RX Family, H8S Family

H8S to RX Migration Guide: DTC

R01AN4392EJ0100
Rev.1.00
Jul 13, 2018

Summary

This application note explains how to migrate from the data transfer controller (DTC) of the H8S Family to the data transfer controller (DTCa) of the RX Family.

Target Devices

- RX Family
- H8S Family

An example of migrating from the H8S Family to the RX Family is presented, with the RX Family represented by the RX231 Group and the H8S Family represented by the H8S/2378 Group. When using this application note with other microcontrollers, appropriate changes should be made to match the specifications of the microcontroller used and thorough evaluation should be performed.

Devices on Which Operation Has Been Confirmed

- RX Family: RX231
- H8S Family: H8S/2378

There are some differences in terminology between the RX Family and H8S Family.

Differences in terminology related to the DTC are listed in the table below.

Table Differences in Terminology between RX Family and H8S Family

Item	RX Family	H8S Family
Name of DTC	DTCa	DTC
Term for chain transfer (below, chain transfer*1)	Chain transfer	Chain transfer
Term for register information	Transfer information	Register information
Term for operation where register information is written back to RAM after data transfer	Write-back	No term specified

Note 1. In the discussion below, the term “chain transfer” is used in reference to the H8S/2378 Group as well.

Contents

1. Points of Difference between DTC

1.1 Points of Difference between Functions

Table 1.1 lists points of difference between the DTC functions.

Table 1.1 Points of Difference between DTC Functions

Item		RX (RX231)	H8S (H8S/2378)
Transfer modes		Normal transfer mode • A single activation results in a single data transfer. • Transfer count: 1 to 65,536 Repeat transfer mode • A single activation results in a single data transfer. • The initial state is restored, and operation continues, after data transfer corresponding to the repeat size. • Transfer count: 1 to 256 Block transfer mode • A single activation results in transfer of a single block of data. • Transfer count: 1 to 65,536	Normal mode • A single activation results in a single data transfer. • Transfer count: 1 to 65,536 Repeat mode • A single activation results in a single data transfer. • The initial state is restored, and operation continues, after data transfer corresponding to the repeat size. • Transfer count: 1 to 256 Block transfer mode • A single activation results in transfer of a single block of data. • Transfer count: 1 to 65,536
Chain transfer		Multiple data transfers are possible on a single activation. Chain transfer operation is selectable between “executed when the counter reaches 0” and “always executed.”	
Transfer source address and transfer destination address		Transfer source address and transfer destination address handling after a data transfer completes is selectable among increment, decrement, and fixed address.	
Transfer space		16 MB in short-address mode (areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh, except reserved areas) 4 GB in full-address mode (area from 0000 0000h to FFFF FFFFh, except reserved areas)	Ability to directly specify 16 MB address space (H'0000 0000 to H'00FF FFFF)
Data transfer units	1 data unit	1 byte, 1 word, or 1 longword	1 byte or 1 word
	1 block	1 to 256 data units	
Activation sources		97 sources (interrupt request)*1	52 sources (software trigger or interrupt request)
Transfer channel		Support for channel transfer corresponding to interrupt source (transferred from ICU by DTC transfer request)	Support for transfers on any number of channels
CPU interrupt request		Ability to generate an interrupt request to the CPU when the DTC is activated by an interrupt	When using software activation, it is possible to generate a software-activated data transfer end interrupt request after the transfer completes. When using interrupt activation, it is possible to generate an interrupt request to the CPU for the interrupt that activates the DTC.

Item	RX (RX231)	H8S (H8S/2378)
Event link function	An event link request is generated after one data transfer (for block transfer operation, after one block transfer).	No such function
Read skip	Ability to specify that reading of transfer information is skipped	No such function
Write-back skip	Write-back skipping is supported if the transfer source address is fixed, or if the transfer destination address is fixed.	No such function
Low power consumption function	Ability to specify transition to module stop state	

Note 1. On the RX231 Group the DTC can be activated by software by means of a software interrupt.

On the RX231 Group the DTC has two selectable address modes: short-address mode and full-address mode. In short-address mode the transfer space consists of the 16 MB in the areas from 0000 0000h to 007F FFFFh and 0080 0000h to FFFF FFFFh, except for reserved areas. In full-address mode the transfer space consists of the 4 GB in the area from 0000 0000h to FFFF FFFFh, except for reserved areas.

The RX231 Group also has a transfer information read skip function and a transfer information write-back skip function. When the transfer information read skip function is enabled, the current DTC vector number is compared with the DTC vector number from the previous activation process, and if they match DTC data transfer is performed without reading the vector address and transfer information. This function can be enabled or disabled by means of the DTCCR.RRS bit.

The transfer information write-back skip function works as follows. When either the transfer source address or transfer destination address is set as a fixed address, write-back (an operation in which transfer information is written back to the RAM) is not performed for a portion of the transfer information. This function is always enabled. Write-back of the contents of the SAR register (DTC transfer source register) is skipped when the transfer source address is set as a fixed address, and write-back of the contents of the DAR register (DTC transfer destination register) is skipped when the transfer destination address is set as a fixed address.

For details of the transfer information read skip and transfer information write-back skip functions, refer to RX230 Group, RX231 Group User's Manual: Hardware.

1.2 Activation Sources

Table 1.2 lists the DTC activation sources on the RX231 Group and H8S/2378 Group.

Table 1.2 DTC Activation Sources on RX231 Group and H8S/2378 Group

RX (RX231)		H8S (H8S/2378)	
Interrupt Request Generation Source	**Activation Source**	**Activation Source Generation Source**	**Activation Source**
ICU	SWINT	Software	Write to DTVECR
	IRQ0	External pin	IRQ0
	IRQ1		IRQ1
	IRQ2		IRQ2
	IRQ3		IRQ3
	IRQ4		IRQ4
	IRQ5		IRQ5
	IRQ6		IRQ6
	IRQ7		IRQ7
	None		IRQ8
			IRQ9
			IRQ10
			IRQ11
			IRQ12
			IRQ13
			IRQ14
			IRQ15
S12AD	S12ADI0	A/D	ADI
	GBADI		None
TPU0	TGI0A	TPU_0	TGI0A
	TGI0B		TGI0B
	TGI0C		TGI0C
	TGI0D		TGI0D
TPU1	TGI1A	TPU_1	TGI1A
	TGI1B		TGI1B
TPU2	TGI2A	TPU_2	TGI2A
	TGI2B		TGI2B
TPU3	TGI3A	TPU_3	TGI3A
	TGI3B		TGI3B
	TGI3C		TGI3C
	TGI3D		TGI3D
TPU4	TGI4A	TPU_4	TGI4A
	TGI4B		TGI4B
TPU5	TGI5A	TPU_5	TGI5A
	TGI5B		TGI5B
TMR0	CMIA0	TMR_0	CMIA0
	CMIB0		CMIB0
TMR1	CMIA1	TMR_1	CMIA1
	CMIB1		CMIB1
TMR2	CMIA2	None	
	CMIB2		
TMR3	CMIA3		
	CMIB3		

RX (RX231)		H8S (H8S/2378)	
Interrupt Request Generation Source	**Activation Source**	**Activation Source Generation Source**	**Activation Source**
DMAC	DMAC0I	DMAC	DMTEND0A
	DMAC1I		DMTEND0B
	DMAC2I		DMTEND1A
	DMAC3I		DMTEND1B
SCI0	RXI0	SCI_0	RXI0
	TXI0		TXI0
SCI1	RXI1	SCI_1	RXI1
	TXI1		TXI1
SCI5	RXI5	SCI_2	RXI2
	TXI5		TXI2
SCI6	RXI6	SCI_3	RXI3
	TXI6		TXI3
SCI8	RXi8	SCI_4	RXI4
	TXI8		TXI4
SCI9	RXI9	None	
	TXI9		
SCI12	RXI12		
	TXI12		
CMT0	CMI0		
CMT1	CMI1		
CMT2	CMI2		
CMT3	CMI3		
USB0	D0FIFO0		
	D1FIFO0		
SDHI	SBFAI		
RSPI0	SPRI0		
	SPTI0		
CAN	COMFRXINT		
CMPB	CMPB0		
	CMPB1		
CTSU	CTSUWR		
	CTSURD		
CMPB1	CMPB2		
	CMPB3		
ELC	ELSR18I		
	ELSR19I		
SSI0	SSIRXI0		
	SSITXI0		
Security	RD		
	WR		
MTU0	TGIA0		
	TGIB0		
	TGIC0		
	TGID0		
MTU1	TGIA1		
	TGIB1		

RX (RX231)		H8S (H8S/2378)	
Interrupt Request Generation Source	**Activation Source**	**Activation Source Generation Source**	**Activation Source**
MTU2	TGIA2	None	
	TGIB2		
MTU3	TGIA3		
	TGIB3		
	TGIC3		
	TGID3		
MTU4	TGIA4		
	TGIB4		
	TGIC4		
	TGID4		
	TCIV4		
MTU5	TGIU5		
	TGIV5		
	TGIW5		
RIIC0	RXI0		
	TXI0		

1.3 Allocation of DTC Vector Configuration and Register Information

Table 1.3 lists the address ranges of the DTC vector table and register information.

Table 1.3 Address Ranges of DTC Vector Table and Register Information

Item	RX (RX231)	H8S (H8S/2378)
DTC vector table	DTCVBR setting value to DTCVBR setting value + 3FFh	H'0400 to H'04FF
Register information	0000 0000h to 0000 FFFFh*1*2	H'FFBC00 to H'FFBFFF*3

Note 1. The address range shown is for products with 64 KB of RAM. The RAM capacity differs among products. For details, refer to RX230 Group, RX231 Group User's Manual: Hardware.

Note 2. Specify the transfer information (n) start address of vector number n as the vector table base address + 4n.

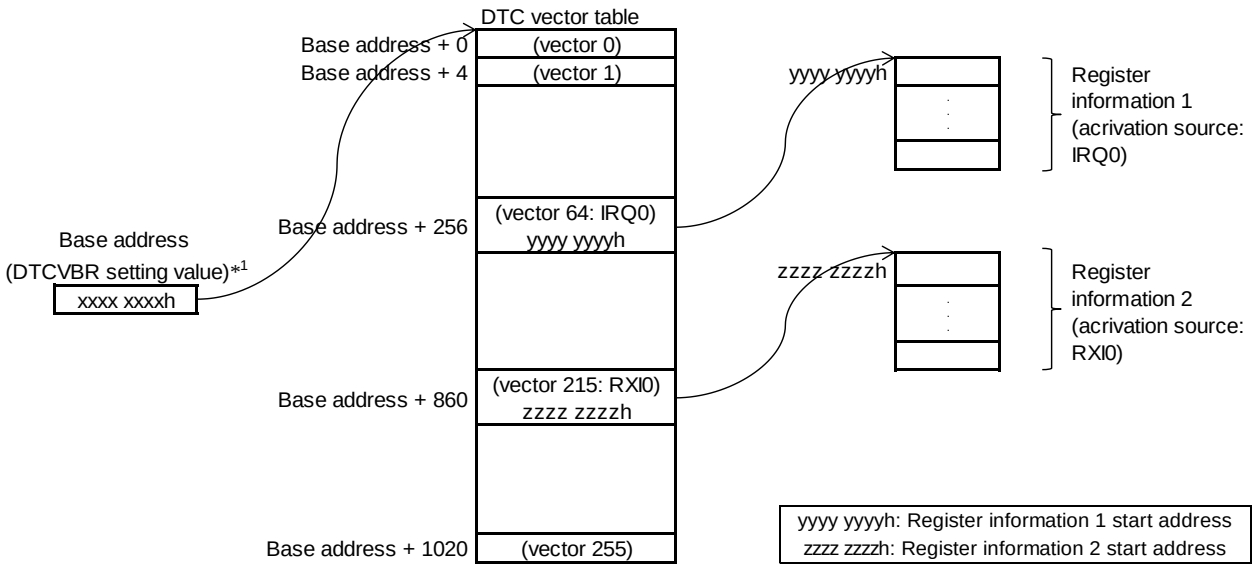
Note 3. Specify as the start address of the register information an address that is a multiple of 4.

The DTC vector address is fixed on the H8S/2378 Group. However, for transfers using a software trigger as the activation source, any address within the range listed in Table 1.3 may be used. The DTC activation vector number can be set in bits DTVEC6 to DTVEC0 in the DTC vector register (DTVECR), after which the DTC vector address becomes H'0400 + vector number × 2. H'0400 to H'04FE is an area in the ROM.

On the RX231 Group, the base address of the DTC vector table can be specified in the DTC vector base register (DTCVBR). The DTC vector address then becomes DTCVBR register setting value + 4n (n = vector number). This allows the DTC vector table to be allocated to any area.

Figure 1.1 shows the DTC vector configuration on the RX231 Group.

Figure 1.2 shows the DTC vector configuration on the H8S/2378 Group.



Note 1. The lower 10 bits of the base address are fixed at 0.

Figure 1.1 DTC Vector Configuration on RX231 Group

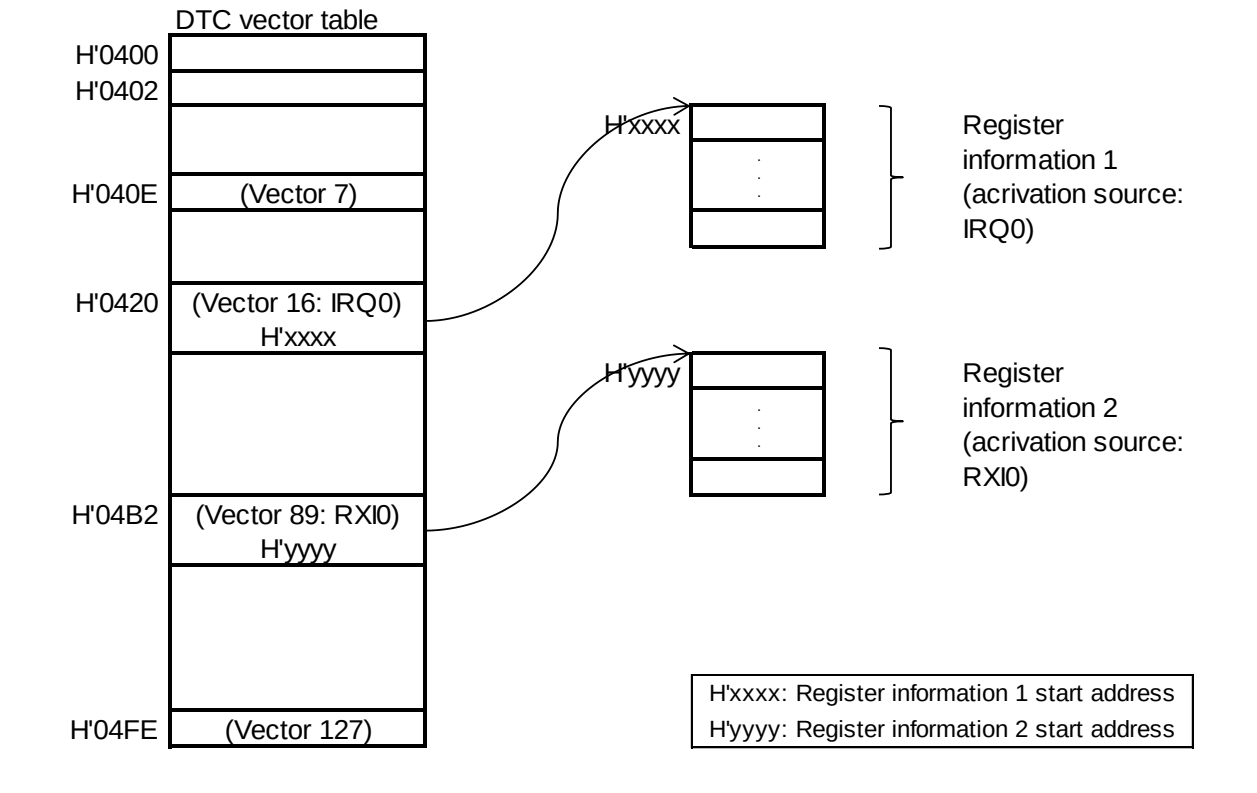


Figure 1.2 DTC Vector Configuration on H8S/2378 Group

On the RX231 Group the method of allocating register information differs according to the address mode.

Figure 1.3 shows the methods of allocating register information on the RX231 Group (short-address mode) and H8S/2378 Group.

Figure 1.4 shows the method of allocating register information on the RX231 Group (full-address mode).

The method of allocating register information on the H8S/2378 Group is the same as that on the RX231 Group in short-address mode. For chain transfer operation, register information should be allocated to contiguous areas as shown in Figure 1.3 and Figure 1.4.

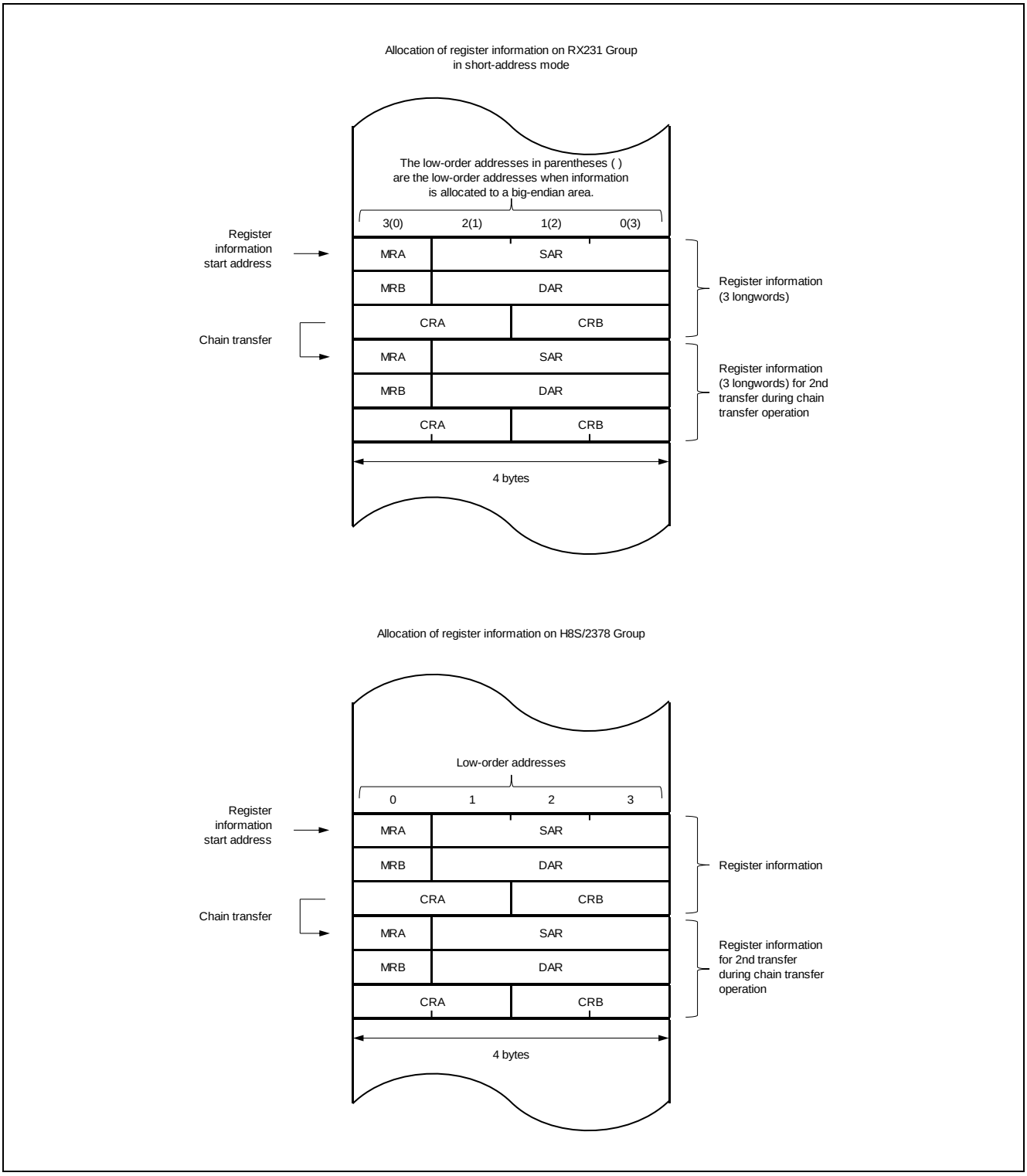


Figure 1.3 Allocation of Register Information on RX231 Group (Short-Address Mode) and H8S/2378 Group

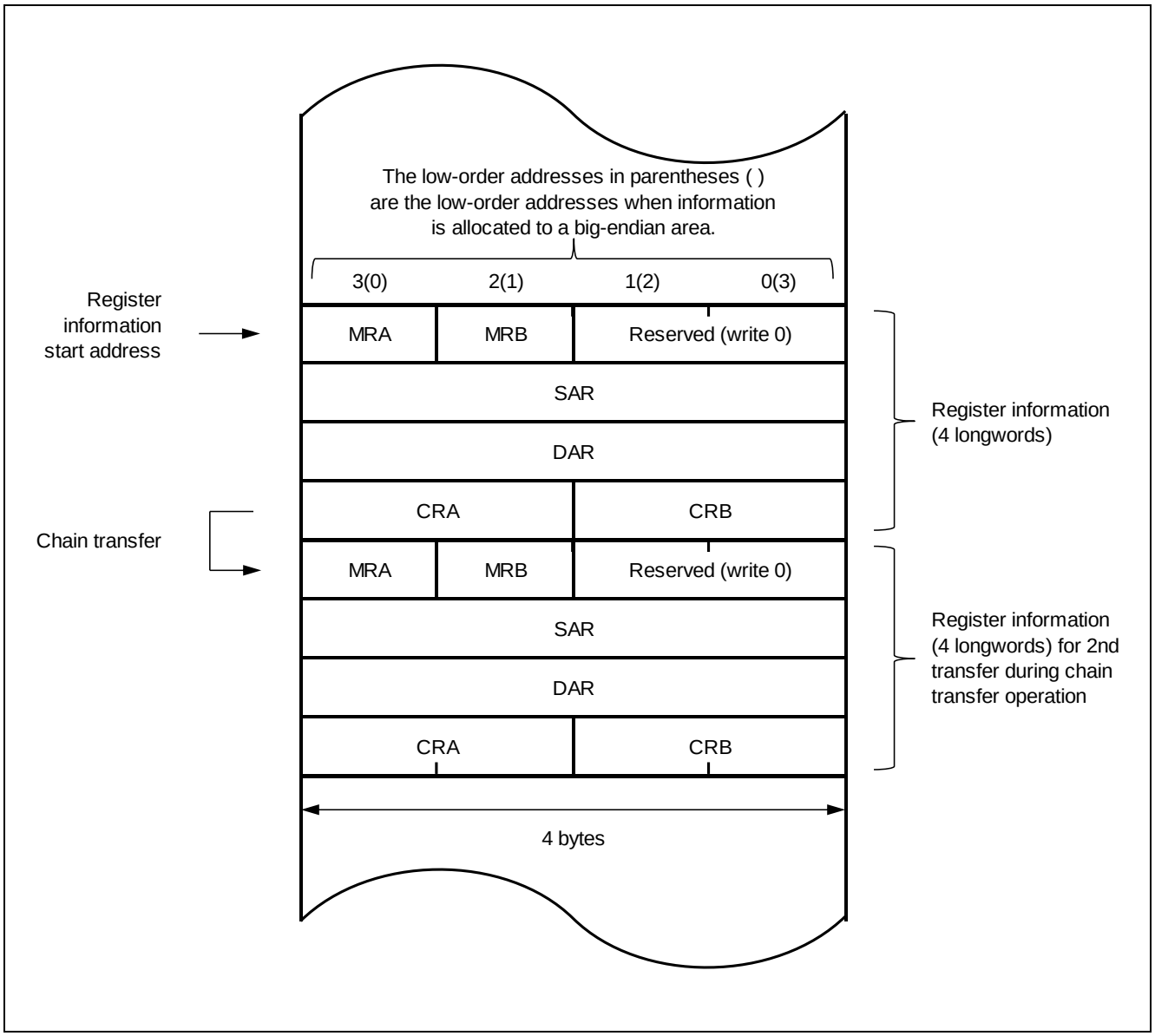


Figure 1.4 Allocation of Register Information on RX231 Group (Full-Address Mode)

On the RX231 Group the register information is affected by the endian format. When allocating register information in the same manner as on the H8S/2378 Group, the area where register information is allocated should be set to big-endian bit format.

In single-chip mode, the endian format can be specified by means of the MDE.MDE[2:0] bits in the option-setting memory. For details, refer to RX230 Group, RX231 Group User's Manual: Hardware.

1.4 Points of Difference between Registers

Table 1.4 lists the DTC registers of the RX231 Group and H8S/2378 Group.

Of the registers listed in Table 1.4, MRA, MRB, SAR, DAR, CRA, and CRB are registers internal to the DTC. They cannot be accessed directly by the CPU. The setting values of these internal registers are allocated to the RAM area as register information. When a transfer request occurs, the DTC reads the register information from the RAM area and makes settings to the internal registers accordingly. After data transfer completes, the DTC writes back the contents of the internal registers to the RAM area as register information.

Table 1.4 List of DTC Registers of RX231 Group and H8S/2378 Group

RX (RX231)	H8S (H8S/2378)
DTC mode register A (MRA)	DTC mode register A (MRA)
DTC mode register B (MRB)	DTC mode register B (MRB)
DTC transfer source register (SAR)	DTC source address register (SAR)
DTC transfer destination register (DAR)	DTC destination address register (DAR)
DTC transfer count register A (CRA)	DTC transfer count register A (CRA)
DTC transfer count register B (CRB)	DTC transfer count register B (CRB)
DTC activation enable register n (DTCERn) (n = interrupt vector number)*1	DTC enable registers A to H (DTCERA to DTCERH)
Software interrupt activation register (SWINTR)*1	None
None	DTC vector register (DTVECR)
DTC vector base register (DTCVBR)	None
DTC control register (DTCCR)	
DTC address mode register (DTCADMOD)	
DTC module start register (DTCST)	
DTC status register (DTCSTS)	

Note 1. Interrupt controller register.

Table 1.5 lists correspondences between register and bit functions. It indicates to which registers and bit fields of the RX231 Group the registers and bit fields of the H8S/2378 Group correspond.

Table 1.5 Correspondences between Register and Bit Functions

RX (RX231)		H8S (H8S/2378)		
Register	**Bit**	**Register**	**Bit**	**Function**
MRA	SM[1:0]	MRA	SM1, SM0	Specifies the operation of the transfer source address after data transfer.
MRB	DM[1:0]		DM1, DM0	Specifies the operation of the transfer destination address after data transfer.
MRA	MD[1:0]		MD1, MD0	Selects the DTC transfer mode.
MRB	DTS		DTS	Settings for repeat area and block area
MRA	SZ[1:0]		Sz	Specifies the transfer data size.
MRB	CHNE	MRB	CHNE	Enables chain transfer operation.
	DISEL		DISEL	Sets the conditions for generating interrupts to the CPU.
	CHNS		CHNS	Sets chain transfer conditions.
SAR		SAR		Transfer source address
DAR		DAR		Transfer destination address
CRA		CRA		Stores data transfer count and acts as transfer counter.
CRB		CRB		Block transfer counter
DTCERn*1	DTCE	DTCERA to DTCERH	DTCE7 to DTCE0	Enables DTC activation.
SWINTR*1	SWINT	DTVECR	SWDTE	Starts DTC transfer by software trigger.
None			DTVEC6 to DTVEC0	Sets DTC software activation vector number.

Note 1. Interrupt controller register.

Table 1.6 lists points of difference between register and bit functions on the RX231 Group and H8S/2378 Group. Register and bit functions implemented only on the RX231 Group are also described.

Table 1.6 Points of Difference between Register and Bit Functions on RX231 Group and H8S/2378 Group

RX (RX231)			H8S (H8S/2378)		
Register	**Bit**	**Function**	**Register**	**Bit**	**Function**
MRA	SM[1:0]	Transfer source address addressing mode bits 0X: SAR address value is fixed. (Write-back to SAR is skipped.) 10: SAR value is incremented after transfer (+1 when SZ[1:0] bits = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b). 11: SAR value is decremented after transfer (−1 when SZ[1:0] bits = 00b, −2 when SZ[1:0] bits = 01b, −4 when SZ[1:0] bits = 10b).	MRA	SM1, SM0	Source address mode 1 and 0 select bits 0X: SAR address value is fixed. 10: SAR value is incremented after transfer (+1 when Sz bit = 0, +2 when Sz bit = 1) 11: SAR value is decremented after transfer (−1 when Sz bit = 0, −2 when Sz bit = 1)
MRB	DM[1:0]	Transfer destination address addressing mode bits 0X: DAR address value is fixed. (Write-back to DAR is skipped.) 10: DAR value is incremented after transfer (+1 when SZ[1:0] bits = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b). 11: DAR value is decremented after transfer (−1 when SZ[1:0] bits = 00b, −2 when SZ[1:0] bits = 01b, −4 when SZ[1:0] bits = 10b).		DM1, DM0	Destination address mode 1 and 0 select bits 0X: DAR address value is fixed. 10: DAR value is incremented after transfer (+1 when Sz bit = 0, +2 when Sz bit = 1) 11: DAR value is decremented after transfer (−1 when Sz bit = 0, −2 when Sz bit = 1)
MRA	MD[1:0]	DTC transfer mode select bits 00: Normal transfer mode 01: Repeat transfer mode 10: Block transfer mode 11: Setting prohibited		MD1, MD0	DTC mode select bits 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited

RX (RX231)			H8S (H8S/2378)		
Register	**Bit**	**Function**	**Register**	**Bit**	**Function**
MRB	DTS	DTC transfer mode select bit 0: Transfer destination is repeat area or block area. 1: Transfer source is repeat area or block area.	MRA	DTS	DTC transfer mode select bit 0: Transfer destination is repeat area or block area. 1: Transfer source is repeat area or block area.
MRA	SZ[1:0]	DTC data transfer size bits 00: Byte (8-bit) transfer 01: Word (16-bit) transfer 10: Longword (32-bit) transfer 11: Setting prohibited		Sz	DTC data transfer size bit 0: Byte-size transfer 1: Word-size transfer
MRB	CHNE	DTC chain transfer enable bit 0: Chain transfer is disabled. 1: Chain transfer is enabled.	MRB	CHNE	DTC chain transfer enable bit 0: Chain transfer is disabled. 1: Chain transfer is enabled.
	DISEL	DTC interrupt select bit 0: An interrupt request to the CPU is generated when the specified data transfer completes. 1: An interrupt request to the CPU is generated each time DTC data transfer is performed.		DISEL	DTC interrupt select bit 0: An interrupt request to the CPU is generated when the specified data transfer completes. 1: An interrupt request to the CPU is generated each time DTC data transfer is performed.
	CHNS	DTC chain transfer select bit 0: Chain transfer is performed continuously. 1: Chain transfer is performed when the transfer counter changes from 1 to 0 or 1 to CRAH.		CHNS	DTC chain transfer select bit 0: Chain transfer is performed continuously. 1: Chain transfer is performed when the transfer counter reaches 0.
SAR		32-bit register used to specify the transfer source start address In full-address mode all 32 bits are valid. In short-address mode the lower 24 bits are valid and the upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.	SAR		24-bit register used to specify the transfer source address For word-size transfer, specify an even source address.
DAR		32-bit register used to specify the transfer destination address In full-address mode all 32 bits are valid. In short-address mode the lower 24 bits are valid and the upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.	DAR		24-bit register used to specify the transfer destination address For word-size transfer, specify an even source address.

RX (RX231)			H8S (H8S/2378)		
Register	**Bit**	**Function**	**Register**	**Bit**	**Function**
CRA		16-bit register used to specify the data transfer count Normal transfer mode Functions as a 16-bit transfer counter (1 to 65,536). The counter value is decremented (−1) at each data transfer, and transfer operation ends when the counter reaches 0000h. Repeat transfer mode CRAH (the upper 8 bits) stores the transfer count, and CRAL (the lower 8 bits) functions as an 8-bit transfer counter (1 to 256). The CRAL value is decremented (−1) at each data transfer, and the CRAH value is transferred to CRAL when the counter reaches 00h. Block transfer mode CRAH (the upper 8 bits) stores the block size, and CRAL (the lower 8 bits) functions as an 8-bit block size counter (1 to 256). The CRAL value is decremented (−1) at each data transfer, and the CRAH value is transferred to CRAL when the counter reaches 00h.	CRA		16-bit register used to specify the data transfer count Normal mode Functions as a 16-bit transfer counter (1 to 65,536). The counter value is decremented (−1) at each data transfer, and transfer operation ends when the counter reaches H'0000. Repeat mode or block transfer mode CRAH (the upper 8 bits) stores the transfer count, and CRAL (the lower 8 bits) functions as an 8-bit transfer counter (1 to 256). The CRAL value is decremented (−1) at each data transfer, and the CRAH value is transferred to CRAL when the counter reaches H'00.
CRB		16-bit register used to specify the block transfer count in block transfer mode Functions as a 16-bit transfer counter (1 to 65,536). The counter value is decremented (−1) when the final data unit of a single block size is transferred, and transfer operation ends when the counter reaches 0000h. This register is not used in normal transfer mode or repeat transfer mode.	CRB		16-bit register used to specify the block transfer count in block transfer mode Functions as a 16-bit transfer counter (1 to 65,536). The counter value is decremented (−1) at each data transfer, and transfer operation ends when the counter reaches H'0000. This register is not used in normal mode or repeat mode.

RX (RX231)			H8S (H8S/2378)		
Register	**Bit**	**Function**	**Register**	**Bit**	**Function**
DTCERn (n = interrupt vector number)	DTCE	DTC activation enable bit When the DTCE bit is set to 1, the corresponding interrupt source is selected as the DTC activation source. [Clearing conditions] • When the specified number of data transfers completes (for chain transfer, when the number of data transfers specified for the last chain transfer complete) • When 0 is written to the DTCE bit	DTCERA to DTCERH	DTCE7 to DTCE0 *1	DTC activation enable bit Setting this bit to 1 selects the corresponding interrupt source as the DTC activation source. [Clearing conditions] • When data transfer ends while the value of the DISEL bit in MRB is 1 • When the specified number of transfers completes (for chain transfer, when the last consecutive transfer is completes) • When 0 is written to DTCE after reading DTCE as 1
SWINTR	SWINT	Software interrupt activation bit When 1 is written to this bit, interrupt request register 027 (IR027) is set to 1. If 1 is written to the SWINT bit when the value of DTC activation enable register 027 (DTCER027) is 0, an interrupt to the CPU is generated. If 1 is written to the SWINT bit when the value of DTC activation enable register 027 (DTCER027) is 1, a DTC activation request is issued. Only 1 can be written to this bit.	DTVECR	SWDTE	DTC software activation enable bit Setting this bit to 1 activates the DTC. [Clearing conditions] • When the value of the DISEL bit is 0 and the specified number of transfers has not ended • When 0 is written to the this bit after a software-activated data transfer end interrupt (SWDTEND) request was sent to the CPU.
None			DTVECR	DTVEC6 to DTVEC0	DTC software activation vector bits 6 to 0 These bits specify the vector number for DTC software activation. The vector address is expressed as H'0400 + vector number × 2. For example, if DTVEC6 to DTVEC0 = H'10, the vector address is H'0420. These bits can be written to when the value of the SWDTE bit is 0.

RX (RX231)			H8S (H8S/2378)		
Register	**Bit**	**Function**	**Register**	**Bit**	**Function**
DTCVBR		32-bit register used to specify the base address for calculating the DTC vector table address. Values written to the upper 4 bits (b31 to b28) are ignored, and the address is extended by the value specified by b27. The lower 10 bits (b9 to b0) are reserved, and their value is fixed at 0. When setting, these bits should be set to 0. The valid ranges are 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h, and settings are made in 1 KB units.	None		
DTCCR	RRS	DTC transfer information read skip enable bit 0: Reading of transfer information is not skipped. 1: Reading of transfer information is skipped when vector numbers match.			
DTCADMOD	SHORT	Short-address mode setting bit 0: Full-address mode 1: Short-address mode			

RX (RX231)			H8S (H8S/2378)		
Register	**Bit**	**Function**	**Register**	**Bit**	**Function**
DTCST	DTCST	DTC module start bit 0: DTC module stop 1: DTC module start Set the DTCST bit to 1 to enable the DTC to accept transfer requests.	None		
DTCSTS	ACT	DTC active flag This flag indicates the state of DTC transfer operation. [Setting condition] • When the DTC is activated by a transfer request. [Clearing condition] • When transfer by the DTC completes in response to a transfer request.			
	VECN[7:0]	DTC-activating vector number monitoring bits While DTC transfer operation is in progress, these bits indicate the vector number corresponding to the activation source for the transfer. When the DTCSTS register is read, the value read from the VECN[7:0] bits is valid if the value of the ACT flag is 1 (DTC transfer in progress). When the DTCSTS register is read, the value read from the VECN[7:0] bits is invalid if the value of the ACT flag is 0 (no current DTC transfer).			

Note 1. Use bit manipulation instructions such as BSET and BCLR to set the DTCE bits. However, multiple activation sources can be set at one time (only at the initial setting) by masking interrupts and writing data after executing a dummy read of the target register.

2. Peripheral Functions Used

Table 2.1 lists the peripheral functions and modes used in the DTC operation examples.

Table 2.1 Peripheral Functions and Modes Used in DTC Operation Examples

	RX (RX231)		H8S (H8S/2378)		Operating Example	
No.	Peripheral Function	Mode	Peripheral Function	Mode	Mode	Reference
1	DTCa	Normal transfer mode	DTC	Normal mode	Normal mode	3.1
2		Repeat transfer mode		Repeat mode	Repeat mode	3.2
3		Block transfer mode		Block transfer mode	Block transfer mode	3.3
4		Chain transfer		Chain transfer	Chain transfer	3.4

3. Points of Difference in Operation

The DTC transfers data based on register information stored in the RAM. When the DTC is activated, it reads the DTC vector corresponding to the vector number. Next, the DTC reads register information from the register information storage address indicated by the DTC vector, transfers data, and then writes the register information back to the RAM after the data transfer.

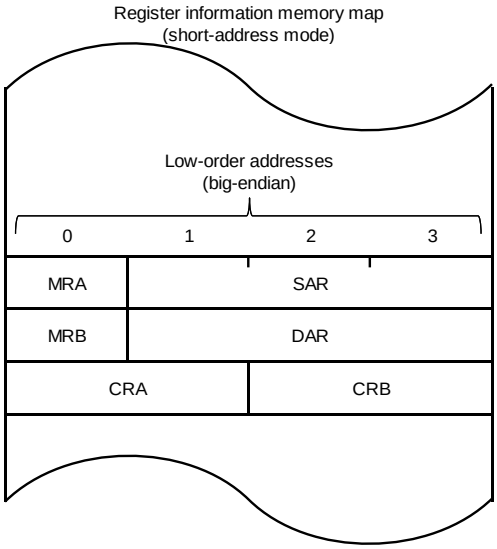
Both the RX231 Group and H8S/2378 Group support normal transfer, repeat transfer, and block transfer modes. In addition, by setting the MRB.CHNE bit to 1 it is possible to perform multiple transfers in response to a single activation (chain transfer). The chain transfer conditions are selected by means of the MRB.CHNS bit.

In order for the DTC to perform data transfers, the register information and DTC vector address must be prepared.

Figure 3.1 and Figure 3.2 show the register information memory maps and structures referred to in the description in this section.

Figure 3.1 shows the register information memory map and structure for the RX231 Group.

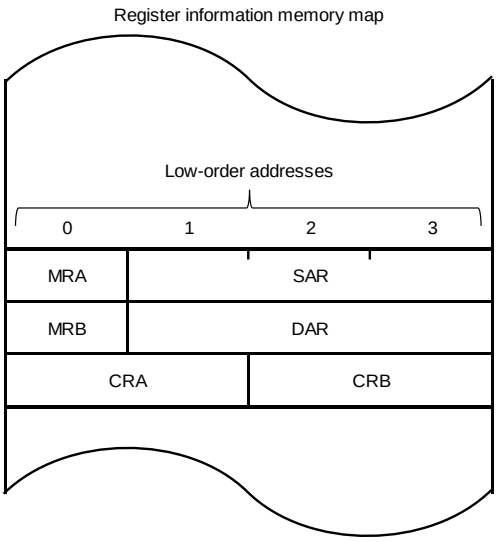
Figure 3.2 shows the register information memory map and structure for the H8S/2378 Group.




```
typedef struct
{
        union {
                unsigned long LONG;
                struct {
                        unsigned long MRA_MD:2;
                        unsigned long MRA_SZ:2;
                        unsigned long MRA_SM:2;
                        unsigned long :2;
                        unsigned long SAR:24;
                } BIT;
        } MRA_SAR;
        union {
                unsigned long LONG;
                struct {
                        unsigned long MRB_CHNE:1;
                        unsigned long MRB_CHNS:1;
                        unsigned long MRB_DISEL:1;
                        unsigned long MRB_DTS:1;
                        unsigned long MRB_DM:2;
                        unsigned long :2;
                        unsigned long DAR:24;
                } BIT;
        } MRB_DAR;
        union {
                unsigned long LONG;
                struct {
                        unsigned long CRA:16;
                        unsigned long CRB:16;
                } WORD;
        } CRA_CRB;
} st_dtc_data;
```

Figure 3.1 RX231 Group Register Information Memory Map and Structure

In Figure 3.1 the address mode is short-address mode, the endian format is big-endian, and within the structure bitfield-type members are arranged starting from the highest-order bit. These settings are made in the development environment.




```
typedef struct
{
        union {
                unsigned long LONG;
                struct {
                        unsigned long MRA_SM:2;
                        unsigned long MRA_DM:2;
                        unsigned long MRA_MD:2;
                        unsigned long MRA_DTS:1;
                        unsigned long MRA_Sz:1;
                        unsigned long SAR:24;
                } BIT;
        } MRA_SAR;
        union {
                unsigned long LONG;
                struct {
                        unsigned long MRB_CHNE:1;
                        unsigned long MRB_DISEL:1;
                        unsigned long MRB_CHNS:1;
                        unsigned long :5;
                        unsigned long DAR:24;
                } BIT;
        } MRB_DAR;
        union {
                unsigned long LONG;
                struct {
                        unsigned long CRA:16;
                        unsigned long CRB:16;
                } WORD;
        } CRA_CRB;
} st_dtc_data;
```

Figure 3.2 H8S/2378 Group Register Information Memory Map and Structure

For the setting procedures for the register information and DTC vector address, refer to the descriptions of the points of difference in the setting procedures for each transfer mode.

3.1 Normal Mode Operation

Table 3.1 lists specifications for normal transfer mode on the RX231 Group and normal mode on the H8S/2378 Group.

Table 3.1 Specifications of RX231 Group Normal Transfer Mode and H8S/2378 Group Normal Mode

Item	RX (RX231) (Normal Transfer Mode)	H8S (H8S/2378) (Normal Mode)
Operation	1 data transfer per activation	
Data size transferrable by single activation	1 byte 1 word 1 longword	1 byte 1 word
Incrementing/decrementing of memory address	Incrementing/decrementing 1, 2, or 4, or fixed address	Incrementing/decrementing 1 or 2, or fixed address
Specifiable transfer count	1 to 65,536	
Interrupt requests to CPU	An interrupt request to the CPU can be generated when the specified number of transfers completes. Alternatively, an interrupt request to the CPU can be generated after every DTC data transfer.	

Table 3.2 lists the normal transfer register functions on the RX231 Group and H8S/2378 Group.

Table 3.2 RX231 Group and H8S/2378 Group Normal Transfer Register Functions

Register	Function	Operation after 1 Data Transfer
SAR	Transfer source address	Increment/decrement/fixed*1
DAR	Transfer destination address	Increment/decrement/fixed*1
CRA	Transfer counter A	CRA – 1
CRB	Transfer counter B	Not updated*2

Note 1. When a fixed address is used on the RX231 Group, write-back is skipped by the transfer information write-back skip function.

Note 2. The CRB register is not used in normal transfer mode. It is used in the block transfer mode only.

Table 3.3 lists the conditions for normal mode operation.

In the description of normal mode operation below, the operation and setting procedure are as listed in Table 3.3, Normal Mode Operation Conditions.

Table 3.3 Normal Mode Operation Conditions

Item	RX (RX231)	H8S (H8S/2378)
Transfer mode	Normal transfer mode	Normal mode
Chain transfer	Not used.	
Address mode	Short-address mode	None
Activation sources	Interrupt controller (ICU) software interrupt (SWINT)	Software
DTC vector base address	0000 FC00h	None
DTC software activation vector address	0000 FC6Ch*1	H'0400
Register information start address	0000 FBF4h	H'FFBC00
Transfer source address	0000 1000h	H'FF4000
Transfer destination address	0000 1010h	H'FF4010
Transfer source address operation	Increment address.	
Transfer destination address operation	Increment address.	
Transfer data size	1 byte	
Transfer count	16	
Generation condition for interrupt requests to CPU	Generation of interrupt request to the CPU at completion of the specified number of data transfers	
Transfer information read skip function	Not used.	No such function.

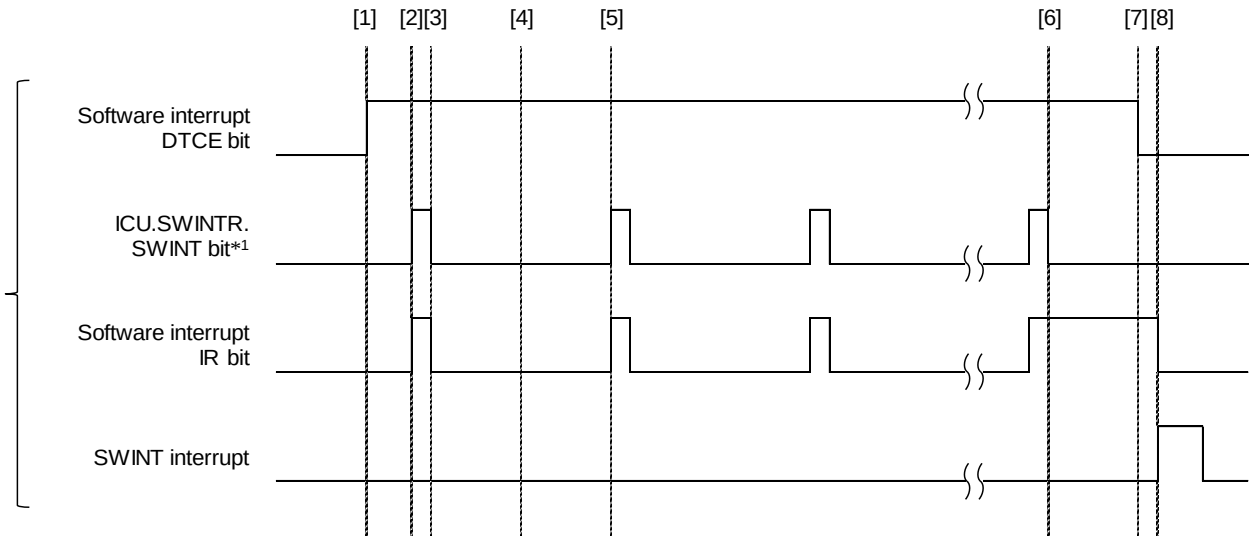
Note 1. On the RX231 Group the DTC software activation vector address is fixed relative to the DTC vector base address.

3.1.1 Operation

Figure 3.3 is a timing chart of normal transfer mode operation on the RX231 Group.

Table 3.4 describes normal transfer mode operation on the RX231 Group.

The numbers in Figure 3.3 correspond to the numbers in Table 3.4.



Note 1. The ICU.SWINTR.SWINT bit (software interrupt activation bit) is used to activate the DTC by software. It is not used when another interrupt is used as the DTC activation source.

Figure 3.3 RX231 Group Normal Transfer Mode Operation Timing

Table 3.4 Description of RX231 Group Normal Transfer Mode Operation

Item		RX (RX231)
[1]	Activation source selection	The software interrupt DTCE bit is set to 1.
[2]	DTC activation	When the ICU.SWINTR.SWINT bit is set to 1, the software interrupt IR bit is set to 1 and a DTC activation request is generated. When the DTC is activated, the DTC vector corresponding to the vector number is read. Next, the transfer information is read from the transfer information storage address indicated by the DTC vector, and data transfer starts.
[3]	Data transfer start at less than specified count	Before transferring data, the DTC determines whether the specified number of transfers has completed. If the transfer count is less than the specified number of transfers (the CRA register value is other than 1), the IR bit is cleared to 0 when data transfer starts.
[4]	End of single transfer	When the first data transfer completes, the CRA register (transfer counter A) is decremented (−1). Also, the transfer information is written back to the RAM. If the transfer count is still less than the specified number of transfers, the value of the DTCE bit remains 1. No interrupt request to the CPU is generated at this point.
[5]	DTC activation	To allow the DTC to be activated continuously by software, set the ICU.SWINTR.SWINT bit to 1 once again.
[6]	Data transfer start at specified count	When the specified number of transfers starts, the IR bit is set to 1.
[7]	End of final transfer of specified count	When the specified number of transfers completes, the DTCE bit is cleared to 0. Also, an software interrupt request to the CPU is generated.
[8]	Clearing of IR bit	When the CPU receives the interrupt request, the IR bit is cleared to 0.

Figure 3.4 is a timing chart of normal mode operation on the H8S/2378 Group.

Table 3.5 describes normal mode operation on the H8S/2378 Group.

The numbers in Figure 3.4 correspond to the numbers in Table 3.5.

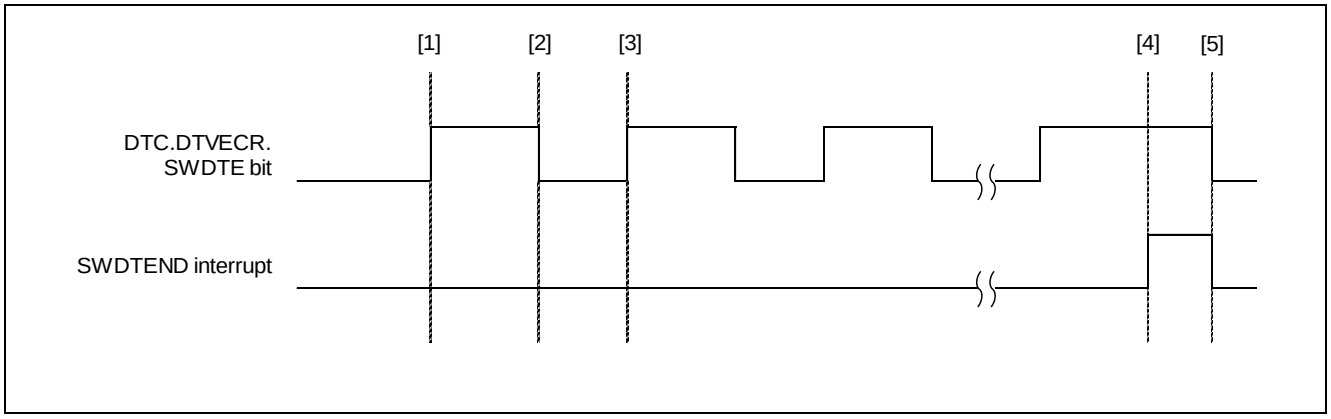


Figure 3.4 H8S/2378 Group Normal Mode Operation Timing

Table 3.5 Description of H8S/2378 Group Normal Mode Operation

Item		H8S (H8S/2378)
[1]	DTC activation	When the DTC.DTVECR.SWDTE bit is set to 1, a DTC software activation request is generated. When the DTC is activated, the DTC vector corresponding to the vector number is read. Next, the register information is read from the register information storage address indicated by the DTC vector, and data transfer starts.
[2]	End of single transfer	When the first data transfer completes, the CRA register (transfer counter A) is decremented (−1). Also, the register information is written back to the RAM. After this, the DTC determines whether the specified number of transfers have completed. If the specified number of transfers have not yet completed (the CRA register value is other than 0), the DTC.DTVECR.SWDTE bit is cleared to 0. No interrupt request to the CPU is generated at this point.
[3]	DTC activation	To allow the DTC to be activated continuously by software, set the DTC.DTVECR.SWDTE bit to 1 once again.
[4]	End of final transfer of specified count	If the specified number of transfers have completed, the value of the DTC.DTVECR.SWDTE bit remains 1. Also, a software-activated data transfer end interrupt request to the CPU is generated.
[5]	Clearing of SWDTE bit	The DTC.DTVECR.SWDTE bit is cleared to 0 by the software-activated data transfer end interrupt handler.

3.1.2 Points of Difference between Setting Procedures

Table 3.6 lists points of difference in the initial setting procedures for normal mode operation. Note that the interrupt controller architectures RX231 Group and H8S/2378 Group differ. For details, refer to section 5, Points of Difference between Interrupts.

Table 3.6 Points of Difference between Initial Setting Procedures for Normal Mode Operation

Procedure		RX (RX231)	H8S (H8S/2378)
1	Declaration of structure variable for storing register information	#pragma address dtc_data0 = 0x0000FBF4 st_dtc_data dtc_data0;	#pragma section Dtc_reg_info st_dtc_data dtc_data0; #pragma section
2	Declaration of variable for storing DTC vector address*1	#pragma address dtc_vector27 = 0x0000FC6C unsigned long dtc_vector27;	#pragma section Dtc_vect_SW const unsigned int vector_sw = {0xBC00}; #pragma section
3	Setting in DTC vector address of register information start address*1	dtc_vector27 = (unsigned long)&dtc_data0;	
4	Canceling of module stop state*2	SYSTEM.PRCR.WORD = 0xA502; MSTP(DTC) = 0; SYSTEM.PRCR.WORD = 0xA500;	MSTPCR.BIT._DTC = 0;
5	Disabling transfers	DTCE(ICU, SWINT) = 0; DTC.DTCST.BIT.DTCST = 0;	DTC.DTVECR.BIT.SWDTE = 0;
6	Disabling interrupts	IEN(ICU, SWINT) = 0;	set_imask_ccr(1);
7	Clearing RRS bit*3	DTC.DTCCR.BIT.RRS = 0;	No processing
8	Setting DTC address mode	DTC.DTCADMOD.BIT.SHORT = 1;	No processing

9	Setting register information (MRA, MRB, SAR, DAR, and CRA registers)	dtc_data0.MRA_SAR.BIT.MRA_MD = 0;
		dtc_data0.MRA_SAR.BIT.MRA_SM = 2;
		dtc_data0.MRB_DAR.BIT.MRB_DM = 2;
		dtc_data0.MRA_SAR.BIT.MRA_SZ = 0;
		dtc_data0.MRB_DAR.BIT.MRB_CHNE = 0;
		dtc_data0.MRB_DAR.BIT.MRB_DISEL = 0;
		dtc_data0.MRA_SAR.BIT.SAR = 0x00001000;
		dtc_data0.MRB_DAR.BIT.DAR = 0x00001010;
		dtc_data0.CRA_CRB.WORD.CRA = 16;
			dtc_data0.MRA_SAR.BIT.MRA_MD = 0;
			dtc_data0.MRA_SAR.BIT.MRA_SM = 2;
			dtc_data0.MRA_SAR.BIT.MRA_DM = 2;
			dtc_data0.MRA_SAR.BIT.MRA_Sz = 0;
			dtc_data0.MRB_DAR.BIT.MRB_CHNE = 0;
			dtc_data0.MRB_DAR.BIT.MRB_DISEL = 0;
			dtc_data0.MRA_SAR.BIT.SAR = 0xFF4000;
			dtc_data0.MRB_DAR.BIT.DAR = 0xFF4010;
			dtc_data0.CRA_CRB.WORD.CRA = 16;

10	Setting DTC vector base address	DTC.DTCVBR = (void *)0x0000FC00;	No processing
11	Activation source selection	DTCE(ICU, SWINT) = 1;	No processing
12	Setting interrupt control mode	No processing	INTC.INTCR.BIT.INTM = 2;
13	Setting interrupt priority level	IPR(ICU, SWINT) = 0x01;	INTC.IPRE.BIT._DTC = 1;
14	Clearing interrupt request	IR(ICU, SWINT) = 0;	No processing
15	Enabling interrupt requests	IEN(ICU, SWINT) = 1;	No processing
16	Setting processor interrupt priority level	No processing	set_imask_exr(0);

Procedure		RX (RX231)	H8S (H8S/2378)
17	Enabling maskable interrupts	setpsw_i();	No processing

18	Confirming no current DTC transfer	No processing	while(1)
			{
			if(DTC.DTVECR.BIT.SWDTE == 0)
			{

19	DTC activation	ICU.SWINTR.BIT.SWINT = 1; DTC.DTCST.BIT.DTCST = 1;	DTC.DTVECR.BYTE = 0x80;

20	Checking the vector number	No processing	if(DTC.DTVECR.BYTE == 0x80)
			{
			break;
			}
			}
			}

Note 1. The setting steps used when allocating the DTC vector table in the RAM are shown.
Note 2. For information on the module stop function, refer to section 6, Module Stop Function.
Note 3. Clearing the DTCCR.RRS bit to 0 resets the flag for transfer information read skipping. If the DTC is activated after this bit is cleared to 0, no transfer information read skipping takes place. Make this setting after the transfer information is updated.

3.2 Repeat Mode Operation

Table 3.7 lists specifications for repeat transfer mode on the RX231 Group and repeat mode on the H8S/2378 Group.

Table 3.7 Specifications of RX231 Group Repeat Transfer Mode and H8S/2378 Group Repeat Mode

Item	RX (RX231) (Repeat Transfer Mode)	H8S (H8S/2378) (Repeat Mode)
Operation	1 data transfer per activation After transfer of data equal to the repeat size, the initial state is restored and operation continues.	
Data size transferrable by single activation	1 byte 1 word 1 longword	1 byte 1 word
Incrementing/decrementing of memory address	Incrementing/decrementing 1, 2, or 4, or fixed address	Incrementing/decrementing 1 or 2, or fixed address
Specifiable transfer count	1 to 256	
Interrupt requests to CPU*1	Alternatively, an interrupt request to the CPU can be generated after every DTC data transfer.	

Note 1. The transfer counter never reaches 0 during repeat transfer operation, so no interrupt request to the CPU is generated after completion of the specified number of transfers.

Table 3.8 lists the repeat transfer register functions on the RX231 Group and H8S/2378 Group.

Table 3.8 RX231 Group and H8S/2378 Group Repeat Transfer Register Functions

		Operation after 1 Data Transfer		
			Transfer Counter Value = 1	
Register	Function	Transfer Counter Value ≠ 1	Transfer Destination Is Repeat Area	Transfer Source Is Repeat Area
SAR	Transfer source address	Increment/decrement/ fixed*1	Increment/decrement/ fixed*1	SAR register initial value
DAR	Transfer destination address	Increment/decrement/ fixed*1	DAR register initial value	Increment/decrement/ fixed*1
CRAH	Transfer counter value retained	CRAH	CRAH	
CRAL	Transfer counter A	CRAL – 1	CRAH	
CRB	Transfer counter B	Not updated*2		

Note 1. When a fixed address is used on the RX231 Group, write-back is skipped by the transfer information write-back skip function.

Note 2. The CRB register is not used in normal transfer mode. It is used in the block transfer mode only.

Table 3.9 lists the conditions for repeat mode operation.

In the description of repeat mode operation below, the operation and setting procedure are as listed in Table 3.9, Repeat Mode Operation Conditions.

Table 3.9 Repeat Mode Operation Conditions

Item	RX (RX231)	H8S (H8S/2378)
Transfer mode	Repeat transfer mode	Repeat mode
Chin transfer	Not used.	
Address mode	Short-address mode	None
Activation sources	Receive data full interrupt (RXI5) for serial communication interface channel 5 (SCI5)	Receive data full interrupt (RXI1) for serial communication interface channel 1 (SCI_1)
DTC vector base address	0000 FC00h	None
DTC vector address	0000 FF7Ch	H'04BA
Register information start address	0000 FBF4h	H'FFBC00
Transfer source address	0008 A0A5h	H'FFFF85
Transfer destination address	0000 1000h	H'FF4000
Transfer source address operation	Fixed address	
Transfer destination address operation	Increment address.	
Repeat area	Transfer destination	
Transfer data size	1 byte	
Transfer count	16	
Generation condition for interrupt requests to CPU	Generation of interrupt request to CPU for each DTC transfer	
Transfer information read skip function	Not used.	No such function

3.2.1 Operation

Figure 3.5 is a timing chart of repeat transfer mode operation on the RX231 Group.

Table 3.10 describes repeat transfer mode operation on the RX231 Group.

The numbers in Figure 3.5 correspond to the numbers in Table 3.10.

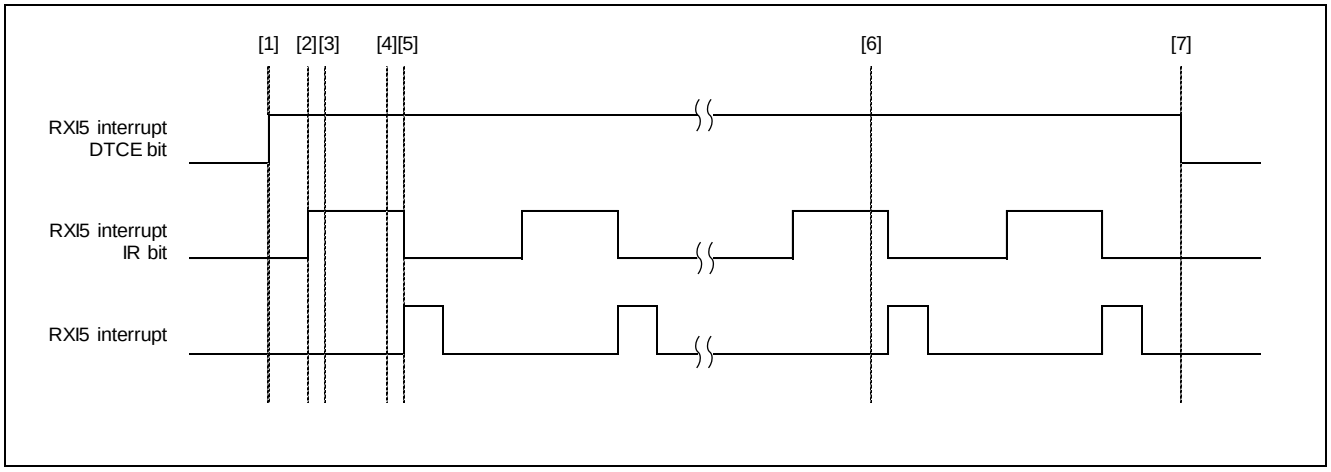


Figure 3.5 RX231 Group Repeat Transfer Mode Operation Timing

Table 3.10 Description of RX231 Group Repeat Transfer Mode Operation

Item		RX (RX231)
[1]	Activation source selection	The RXI5 interrupt DTCE bit is set to 1.
[2]	DTC activation	When data is received on SCI5, the RXI5 interrupt IR bit is set to 1 and a DTC activation request is generated. When the DTC is activated, the DTC vector corresponding to the vector number is read. Next, the transfer information is read from the transfer information storage address indicated by the DTC vector, and data transfer starts.
[3]	Start of data transfer when DISEL = 1	Before transferring data, the DTC determines the value of the MRB.DISEL bit. For data transfer when the DISEL bit is set to 1 (interrupt request to CPU generated for each DTC transfer), the IR bit is not cleared and remains set to 1 when data transfer starts.
[4]	End of single transfer	When the first data transfer completes, the CRAL register (transfer counter A) is decremented (−1). Also, the transfer information is written back to the RAM. During repeat transfer operation, the value of the DTCE bit remains 1. Also, an interrupt request to the CPU is generated.
[5]	Clearing of IR bit	When the interrupt request to the CPU is received, the IR bit is cleared to 0.
[8]	End of final transfer of specified count	After the specified number of transfers have completed, the address register set as the repeat area is restored to its initial state. Also, the value of the CARH register (transfer counter value retained) is transferred to the CRAL register (transfer counter A), and transfer operation repeats.
[9]	End of DTC transfer	To end DTC transfer operation, clear the RXI5 interrupt DTCE bit to 0.

Figure 3.6 is a timing chart of repeat mode operation on the H8S/2378 Group.

Table 3.11 describes repeat mode operation on the H8S/2378 Group.

The numbers in Figure 3.6 correspond to the numbers in Table 3.11.

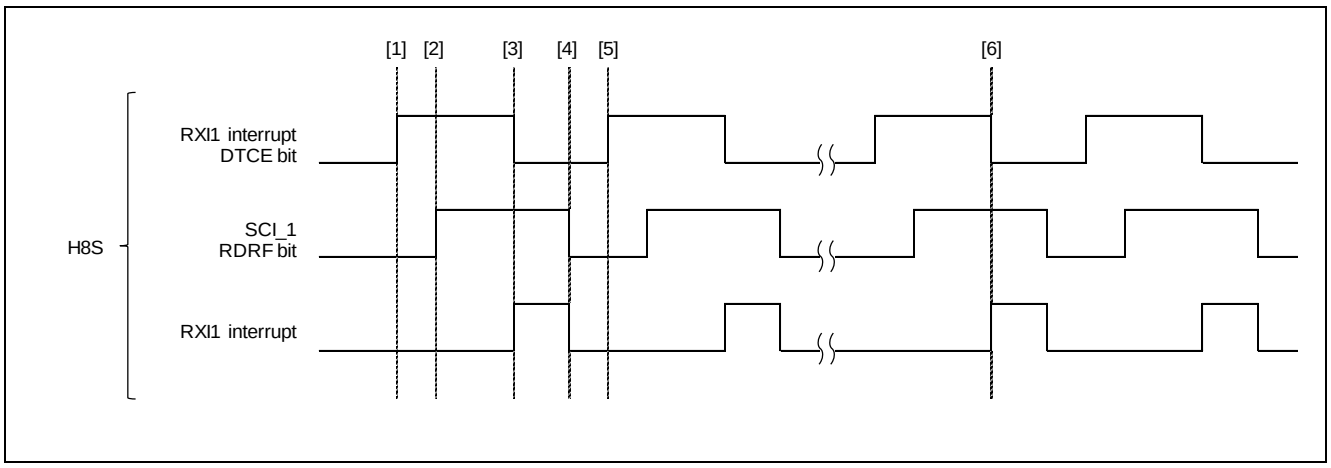


Figure 3.6 H8S/2378 Group Repeat Mode Operation Timing

Table 3.11 Description of H8S/2378 Group Repeat Transfer Mode Operation

Item		H8S (H8S/2378)
[1]	Activation source selection	The RXI1 interrupt DTCE bit is set to 1.
[2]	DTC activation	When data is received on SCI_1, the SCI_1 RDRF bit is set to 1 and a DTC activation request is generated. When the DTC is activated, the DTC vector corresponding to the vector number is read. Next, the register information is read from the register information storage address indicated by the DTC vector, and data transfer starts.
[3]	End of single transfer	When the first data transfer completes, the CRAL register (transfer counter A) is decremented (−1). Also, the register information is written back to the RAM. After this, the DTC determines the value of the MRB.DISEL bit. For data transfer when the DISEL bit is set to 1 (interrupt request to CPU generated for each DTC transfer), the DTCE is cleared to 0 when the transfer completes. Also, the RDRF bit remains set to 1, and an interrupt request to the CPU is generated.
[4]	Clearing of RDRF bit	The RDRF bit is cleared to 0 by the RXI1 interrupt handler.
[5]	DTC activation	To continue activating the DTC by means of the RXI1 interrupt, set the RXI interrupt DTCE bit to 1 once again.
[6]	End of final transfer of specified count	After the specified number of transfers have completed, the address register set as the repeat area is restored to its initial state. Also, the value of the CARH register (transfer counter value retained) is transferred to the CRAL register (transfer counter A), and transfer operation repeats.

3.2.2 Points of Difference between Setting Procedures

Table 3.12 lists points of difference in the initial setting procedures for repeat mode operation. Note that the interrupt controller architectures RX231 Group and H8S/2378 Group differ. For details, refer to section 5, Points of Difference between Interrupts.

Table 3.12 Points of Difference between Initial Setting Procedures for Repeat Mode Operation

Procedure		RX (RX231)	H8S (H8S/2378)
1	Serial communication settings	Settings are made to the serial communication interface (SCI5).	Settings are made to the serial communication interface (SCI_1).
2	Declaration of structure variable for storing register information	#pragma address dtc_data0 = 0x0000FBF4 st_dtc_data dtc_data0;	#pragma section Dtc_reg_info st_dtc_data dtc_data0; #pragma section
3	Declaration of variable for storing DTC vector address*1	#pragma address dtc_vector223 = 0x0000FF7C unsigned long dtc_vector223;	#pragma section Dtc_vect_RXI1 const unsigned int vector_rxi1 = {0xBC00}; #pragma section
4	Setting in DTC vector address of register information start address*1	dtc_vector223 = (unsigned long)&dtc_data0;	
5	Canceling of module stop state*2	SYSTEM.PRCR.WORD = 0xA502; MSTP(DTC) = 0; SYSTEM.PRCR.WORD = 0xA500;	MSTPCR.BIT._DTC = 0;
6	Disabling transfers	DTCE(SCI5, RXI5) = 0; DTC.DTCST.BIT.DTCST = 0;	DTC.DTCEF.BIT.RXI1 = 0;
7	Disabling interrupts	IEN(SCI5, RXI5) = 0; IEN(SCI5, ERI5) = 0;	set_imask_ccr(1);
8	Clearing RRS bit*3	DTC.DTCCR.BIT.RRS = 0;	No processing
9	Setting DTC address mode	DTC.DTCADMOD.BIT.SHORT = 1;	No processing

10	Setting register information (MRA, MRB, SAR, DAR, and CRA registers)	dtc_data0.MRA_SAR.BIT.MRA_MD = 1;
		dtc_data0.MRB_DAR.BIT.MRB_DTS = 0;
		dtc_data0.MRA_SAR.BIT.MRA_SM = 0;
		dtc_data0.MRB_DAR.BIT.MRB_DM = 2;
		dtc_data0.MRA_SAR.BIT.MRA_SZ = 0;
		dtc_data0.MRB_DAR.BIT.MRB_CHNE = 0;
		dtc_data0.MRB_DAR.BIT.MRB_DISEL = 1;
		dtc_data0.MRA_SAR.BIT.SAR = 0x0008A0A5;
		dtc_data0.MRB_DAR.BIT.DAR = 0x00001000;
		dtc_data0.CRA_CRB.WORD.CRA = 0x1010;
			dtc_data0.MRA_SAR.BIT.MRA_MD = 1;
			dtc_data0.MRA_SAR.BIT.MRA_DTS = 0;
			dtc_data0.MRA_SAR.BIT.MRA_SM = 0;
			dtc_data0.MRA_SAR.BIT.MRA_DM = 2;
			dtc_data0.MRA_SAR.BIT.MRA_Sz = 0;
			dtc_data0.MRB_DAR.BIT.MRB_CHNE = 0;
			dtc_data0.MRB_DAR.BIT.MRB_DISEL = 1;
			dtc_data0.MRA_SAR.BIT.SAR = 0xFFFF85;
			dtc_data0.MRB_DAR.BIT.DAR = 0xFF4000;
			dtc_data0.CRA_CRB.WORD.CRA = 0x1010;

11	Setting DTC vector base address	DTC.DTCVBR = (void *)0x0000FC00;	No processing
12	Enabling DTC activation	DTCE(SCI5, RXI5) = 1; DTC.DTCST.BIT.DTCST = 1;	DTC.DTCEF.BIT.RXI1 = 1;
13	Setting interrupt control mode	No processing	INTC.INTCR.BIT.INTM = 2;
14	Setting interrupt priority level	IPR(SCI5, RXI5) = 0x01; IPR(SCI5, ERI5) = 0x01;	INTC.IPRJ.BIT._SCI1 = 1;

Procedure		RX (RX231)	H8S (H8S/2378)
15	Clearing peripheral function interrupt request	SCI5.SSR.BYTE &= 0x87;	SCI1.SSR.BYTE &= 0x87;
16	Clearing interrupt request	IR(SCI5, RXI5) = 0;	No processing
17	Enabling interrupt requests	IEN(SCI5, RXI5) = 1; IEN(SCI5, ERI5) = 1;	No processing
18	Setting processor interrupt priority level	No processing	set_imask_exr(0);
19	Enabling maskable interrupts	setpsw_i();	No processing
20	Enabling serial receive interrupts	SCI5.SCR.BIT.RIE = 1;	SCI1.SCR.BIT.RIE = 1;
21	Enabling serial communication receive operation	SCI5.SCR.BIT.RE = 1;	SCI1.SCR.BIT.RE = 1;

Note 1. The setting steps used when allocating the DTC vector table in the RAM are shown.

Note 2. For information on the module stop function, refer to section 6, Module Stop Function.

Note 3. Clearing the DTCCR.RRS bit to 0 resets the flag for transfer information read skipping. If the DTC is activated after this bit is cleared to 0, no transfer information read skipping takes place. Make this setting after the transfer information is updated.

3.3 Block Transfer Mode Operation

Table 3.13 lists specifications for block transfer mode on the RX231 Group and block transfer mode on the H8S/2378 Group.

Table 3.13 Specifications of RX231 Group Block Transfer Mode and H8S/2378 Group Block Transfer Mode

Item	RX (RX231) (Block Transfer Mode)	H8S (H8S/2378) (Block Transfer Mode)
Operation	1 block of data transfer per activation	
Data size transferrable by single activation	1 to 256 bytes 1 to 256 words 1 to 256 longwords	1 to 256 bytes 1 to 256 words
Incrementing/decrementing of memory address	Incrementing/decrementing 1, 2, or 4, or fixed address	Incrementing/decrementing 1 or 2, or fixed address
Specifiable transfer count	1 to 65,536	
Interrupt requests to CPU	An interrupt request to the CPU can be generated when the specified number of transfers completes. Alternatively, an interrupt request to the CPU can be generated after every DTC data transfer.	

Table 3.14 lists the block transfer register functions on the RX231 Group and H8S/2378 Group.

Table 3.14 RX231 Group and H8S/2378 Group Block Transfer Register Functions

		Operation after 1 Block Transfer	
Register	Function	Transfer Destination Is Block Area	Transfer Source Is Block Area
SAR	Transfer source address	Increment/decrement/fixed*1	SAR register initial value
DAR	Transfer destination address	DAR register initial value	Increment/decrement/fixed*1
CRAH	Block size retention	CRAH	
CRAL	Block size counter	CRAH	
CRB	Block transfer counter	CRB – 1	

Note 1. When a fixed address is used on the RX231 Group, write-back is skipped by the transfer information write-back skip function.

Table 3.15 lists the conditions for block transfer mode operation.

In the description of block transfer mode operation below, the operation and setting procedure are as listed in Table 3.15, Block Transfer Mode Operation Conditions.

Table 3.15 Block Transfer Mode Operation Conditions

Item	RX (RX231)	H8S (H8S/2378)
Transfer mode	Block transfer mode	
Chain transfer	Not used.	
Address mode	Short-address mode	None
Activation sources	12-bit A/D converter (S12ADE) scan end interrupt (S12ADI0)	A/D converter A/D conversion end interrupt (ADI)
DTC vector base address	0000 FC00h	None
DTC vector address	0000 FD98h	H'044C
Register information start address	0000 FBF4h	H'FFBC00
Transfer source address	0008 9020h	H'FFFF90
Transfer destination address	0000 1000h	H'FF4000
Transfer source address operation	Increment address.	
Transfer destination address operation	Increment address.	
Block area	Transfer source	
Transfer data size	3 words	
Transfer count	8	
Generation condition for interrupt requests to CPU	Generation of interrupt request to CPU at completion of specified number of transfers	
Transfer information read skip function	Not used.	No such function

3.3.1 Operation

Figure 3.7 is a timing chart of block transfer mode operation on the RX231 Group.

Table 3.16 describes block transfer mode operation on the RX231 Group.

The numbers in Figure 3.7 correspond to the numbers in Table 3.16.

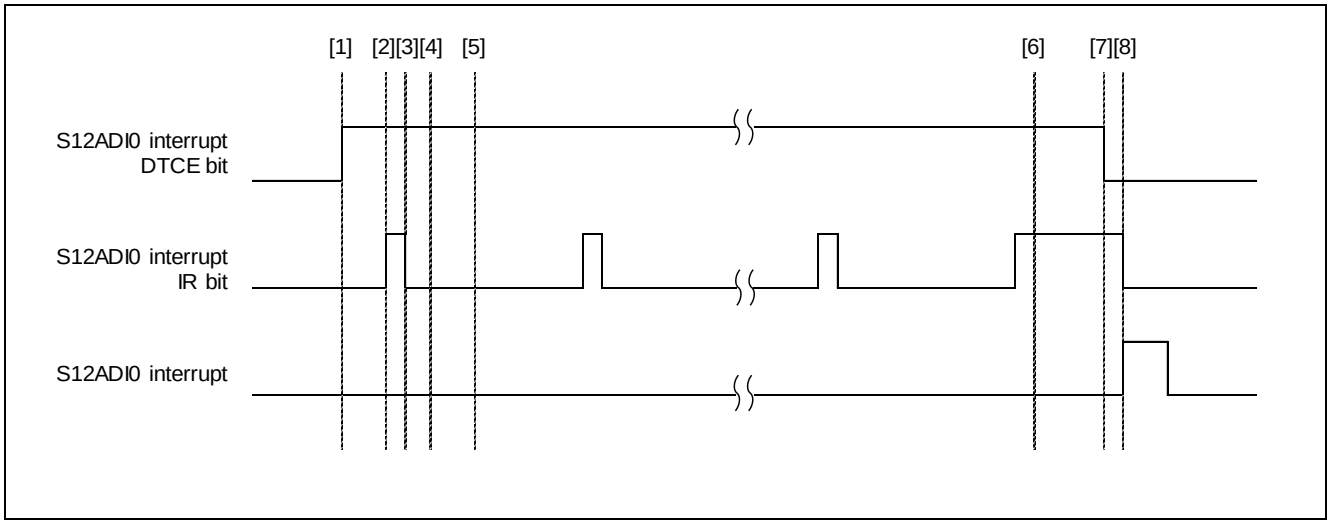


Figure 3.7 RX231 Group Block Transfer Mode Operation Timing

Table 3.16 Description of RX231 Group Block Transfer Mode Operation

Item		RX (RX231)
[1]	Activation source selection	The S12ADI0 interrupt DTCE bit is set to 1.
[2]	DTC activation	When A/D conversion by S12ADE finishes, the S12ADI0 interrupt IR bit is set to 1 and a DTC activation request is generated. When the DTC is activated, the DTC vector corresponding to the vector number is read. Next, the transfer information is read from the transfer information storage address indicated by the DTC vector, and data transfer starts.
[3]	Data transfer start at less than specified count	Before transferring data, the DTC determines whether the specified number of transfers has completed. If the transfer count is less than the specified number of transfers (the CRB register value is other than 1), the IR bit is cleared to 0 when data transfer starts.
[4]	End of transfer of 1 data unit	When transfer of 1 data unit finishes, the value of the CRAL register (block size counter) is decremented (–1).
[5]	End of transfer of 1 block	When transfer of 1 block finishes, the value of the CRB register (block transfer counter) is decremented (–1). Also, the transfer information is written back to the RAM. If the transfer count is still less than the specified number of transfers, the value of the DTCE bit remains 1. No interrupt request to the CPU is generated at this point.
[6]	Data transfer start at specified count	When the specified number of transfers starts, the IR bit is set to 1.
[7]	End of final transfer of specified count	When the specified number of transfers completes, the DTCE bit is cleared to 0. Also, the S12ADI0 interrupt request to the CPU is generated.
[8]	Clearing of IR bit	When the CPU receives the interrupt request, the IR bit is cleared to 0.

Figure 3.8 is a timing chart of block transfer mode operation on the H8S/2378 Group.

Table 3.17 describes block transfer mode operation on the H8S/2378 Group.

The numbers in Figure 3.8 correspond to the numbers in Table 3.17.

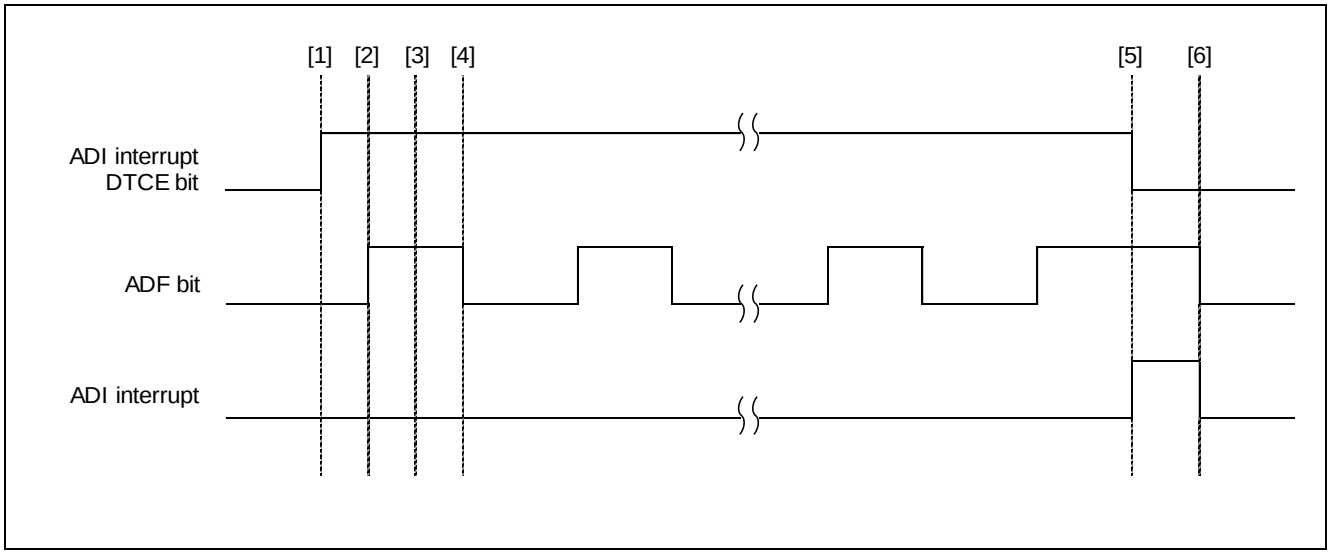


Figure 3.8 H8S/2378 Group Block Transfer Mode Operation Timing

Table 3.17 Description of H8S/2378 Group Block Transfer Mode Operation

Item		H8S (H8S/2378)
[1]	Activation source selection	The ADI interrupt DTCE bit is set to 1.
[2]	DTC activation	When A/D conversion by the A/D converter finishes, the ADF bit is set to 1 and a DTC activation request is generated. When the DTC is activated, the DTC vector corresponding to the vector number is read. Next, the register information is read from the register information storage address indicated by the DTC vector, and data transfer starts.
[3]	End of transfer of 1 data unit	When transfer of 1 data unit finishes, the value of the CRAL register (block size counter) is decremented (–1).
[4]	End of transfer of 1 block	When transfer of 1 block finishes, the value of the CRB register (block transfer counter A) is decremented (–1). Also, the register information is written back to the RAM. After this, the DTC determines whether the specified number of transfers has completed. If the transfer count is less than the specified number of transfers (the CRB register value is other than 0), the value of the DTCE bit remains 1 and the ADF bit is cleared to 0. No interrupt request to the CPU is generated at this point.
[5]	End of final transfer of specified count	When the specified number of transfers complete, the DTCE bit is cleared to 0 and the value of the ADF bit remains 1. Also, the ADI interrupt request to the CPU is generated.
[6]	Clearing of ADF bit	The ADF bit is cleared to 0 by the ADI interrupt handler.

3.3.2 Points of Difference between Setting Procedures

Table 3.18 lists points of difference in the initial setting procedures for block transfer mode operation. Note that the interrupt controller architectures RX231 Group and H8S/2378 Group differ. For details, refer to section 5, Points of Difference between Interrupts.

Table 3.18 Points of Difference between Initial Setting Procedures for Block Transfer Mode Operation

Procedure		RX (RX231)	H8S (H8S/2378)
1	A/D conversion settings	Settings are made to the 12-bit A/D converter (S12ADE).	Settings are made to the A/D converter.
2	Declaration of structure variable for storing register information	#pragma address dtc_data0 = 0x0000FBF4 st_dtc_data dtc_data0;	#pragma section Dtc_reg_info st_dtc_data dtc_data0; #pragma section
3	Declaration of variable for storing DTC vector address*1	#pragma address dtc_vector102 = 0x0000FD98 unsigned long dtc_vector102;	#pragma section Dtc_vect_ADI const unsigned int vector_adi = {0xBC00}; #pragma section
4	Setting in DTC vector address of register information start address*1	dtc_vector102 = (unsigned long)&dtc_data0;	
5	Canceling of module stop state*2	SYSTEM.PRCR.WORD = 0xA502; MSTP(DTC) = 0; SYSTEM.PRCR.WORD = 0xA500;	MSTPCR.BIT._DTC = 0;
6	Disabling transfers	DTCE(S12AD, S12ADI0) = 0; DTC.DTCST.BIT.DTCST = 0;	DTC.DTCEC.BIT.ADI = 0;
7	Disabling interrupts	IEN(S12AD, S12ADI0) = 0;	set_imask_ccr(1);
8	Clearing RRS bit*3	DTC.DTCCR.BIT.RRS = 0;	No processing
9	Setting DTC address mode	DTC.DTCADMOD.BIT.SHORT = 1;	No processing

10	Setting register information (MRA, MRB, SAR, DAR, CRA, and CRB registers)	dtc_data0.MRA_SAR.BIT.MRA_MD = 2;
		dtc_data0.MRB_DAR.BIT.MRB_DTS = 1;
		dtc_data0.MRA_SAR.BIT.MRA_SM = 2;
		dtc_data0.MRB_DAR.BIT.MRB_DM = 2;
		dtc_data0.MRA_SAR.BIT.MRA_SZ = 1;
		dtc_data0.MRB_DAR.BIT.MRB_CHNE = 0;
		dtc_data0.MRB_DAR.BIT.MRB_DISEL = 0;
		dtc_data0.MRA_SAR.BIT.SAR = 0x00089020;
		dtc_data0.MRB_DAR.BIT.DAR = 0x00001000;
		dtc_data0.CRA_CRB.WORD.CRA = 0x0303;
		dtc_data0.CRA_CRB.WORD.CRB = 0x0008;
			dtc_data0.MRA_SAR.BIT.MRA_MD = 2;
			dtc_data0.MRA_SAR.BIT.MRA_DTS = 1;
			dtc_data0.MRA_SAR.BIT.MRA_SM = 2;
			dtc_data0.MRA_SAR.BIT.MRA_DM = 2;
			dtc_data0.MRA_SAR.BIT.MRA_Sz = 1;
			dtc_data0.MRB_DAR.BIT.MRB_CHNE = 0;
			dtc_data0.MRB_DAR.BIT.MRB_DISEL = 0;
			dtc_data0.MRA_SAR.BIT.SAR = 0xFFFF90;
			dtc_data0.MRB_DAR.BIT.DAR = 0xFF4000;
			dtc_data0.CRA_CRB.WORD.CRA = 0x0303;
			dtc_data0.CRA_CRB.WORD.CRB = 0x0008;

11	Setting DTC vector base address	DTC.DTCVBR = (void *)0x0000FC00;	No processing
12	Enabling DTC activation	DTCE(S12AD, S12ADI0)= 1; DTC.DTCST.BIT.DTCST = 1;	DTC.DTCEC.BIT.ADI = 1;
13	Setting interrupt control mode	No processing	INTC.INTCR.BIT.INTM = 2;
14	Setting interrupt priority level	IPR(S12AD, S12ADI0) = 0x01;	INTC.IPRF.BIT._AD = 1;

Procedure		RX (RX231)	H8S (H8S/2378)
15	Clearing peripheral function interrupt request	No processing*4	AD.ADCSR.BIT.ADF = 0;
16	Clearing interrupt request	IR(S12AD, S12ADI0) = 0;	No processing
17	Enabling interrupt requests	IEN(S12AD, S12ADI0) = 1;	No processing
18	Setting processor interrupt priority level	No processing	set_imask_exr(0);
19	Enabling maskable interrupts	setpsw_i();	No processing
20	Enabling A/D conversion end interrupts	S12AD.ADCSR.BIT.ADIE = 1;	AD.ADCSR.BIT.ADIE = 1;
21	A/D conversion start	S12AD.ADCSR.BIT.ADST = 1;	AD.ADCSR.BIT.ADST = 1;

Note 1. The setting steps used when allocating the DTC vector table in the RAM are shown.

Note 2. For information on the module stop function, refer to section 6, Module Stop Function.

Note 3. Clearing the DTCCR.RRS bit to 0 resets the flag for transfer information read skipping. If the DTC is activated after this bit is cleared to 0, no transfer information read skipping takes place. Make this setting after the transfer information is updated.

Note 4. The A/D converter of the RX231 Group does not have a peripheral function interrupt request bit. For details, refer to section 43, 12-Bit A/D Converter (S12ADE), in RX230 Group, RX231 Group User's Manual: Hardware.

3.4 Chain Transfer

Chain transfer allows multiple data transfers to be performed on a single activation.

Figure 3.9 shows chain transfer operation.

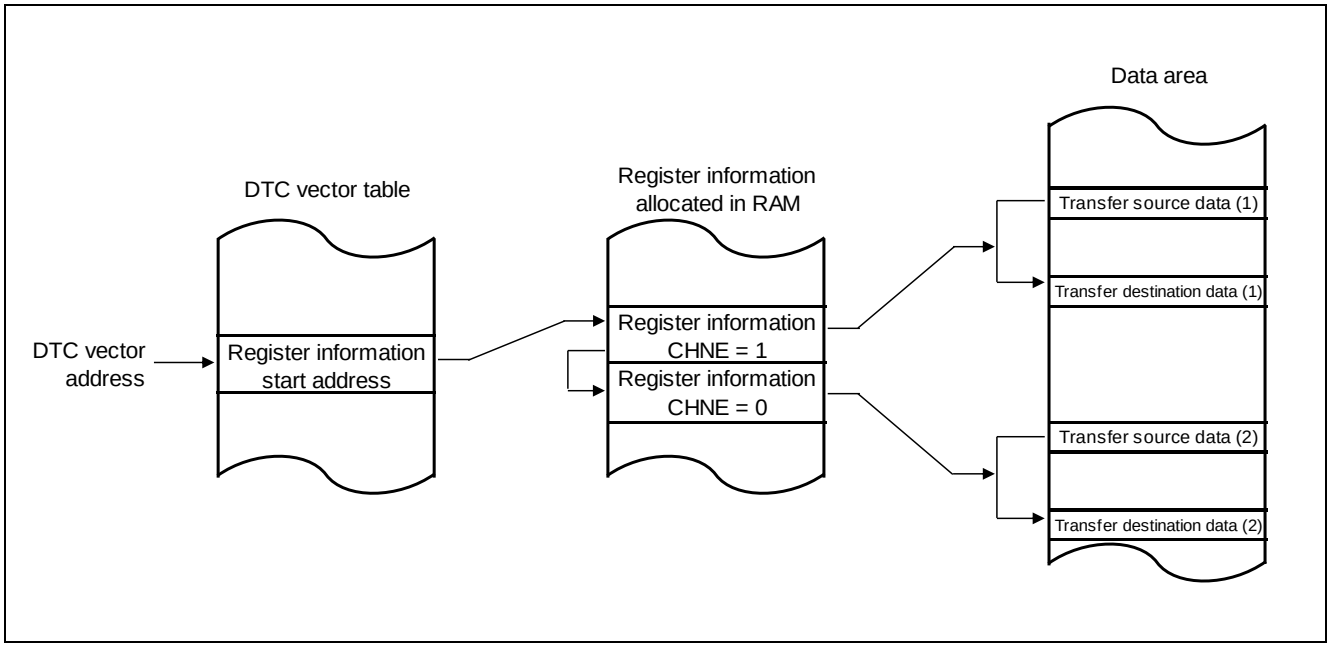


Figure 3.9 Chain Transfer Operation

When the DTC is activated, the register information start address is read from the DTC vector address corresponding to the activation source, and the initial register information is read starting from this start address. If the value of the MRB.CHNE bit is 1, after data transfer completes the next register information is read and another transfer takes place in continuous fashion. This operation continues until the completion of a data transfer corresponding to register information in which the value of the CHNE bit is zero.

The chain transfer conditions can be selected by means of the MRB.CHNS bit.

Table 3.19 lists the functions of the MRB.CHNS bit.

Table 3.19 Functions of MRB.CHNS Bit

Item		RX (RX231)	H8S (H8S/2378)
MRB.CHNS bit	0	Successive chain transfers are performed.	Successive chain transfers are performed.
	1	Chain transfer is performed when transfer counter value changes from 1 to 0 or from 1 to CRAH.	Chain transfer is performed only when transfer counter value = 0.

When the MRB.CHNE bit has been set to 1 and the MRB.CHNS bit cleared to 0, no interrupt request to the CPU is generated when the specified number of transfers completes and the value of the DISEL bit is 1.

Table 3.20 lists the conditions for chain transfer operation.

In the description of chain transfer operation below, the operation and setting procedure are as listed in Table 3.20, Chain Transfer Operation Conditions.

Table 3.20 Chain Transfer Operation Conditions

Item		RX (RX231)	H8S (H8S/2378)
Common	Address mode	Short address mode	None
	DTC vector base address	0000 FC00h	None
	Transfer information read skip function	Not used.	No such function
1st data transfer	Transfer mode	Normal transfer mode	Normal mode
	Chain transfer	Chain transfer is performed when transfer counter value changes from 1 to 0.	Chain transfer is performed only when transfer counter value = 0.
	Activation sources	16-bit timer pulse unit channel 0 (TPU0) compare match interrupt (TGI0A)	16-bit timer pulse unit channel _0 (TPU_0) compare match interrupt (TGI0A)
	DTC vector address	0000 FE38h	H'0450
	Register information start address	0000 FBE8h	H'FFBC00
	Transfer source address	0000 1000h	H'FF4000
	Transfer destination address	0000 1010h	H'FF4010
	Transfer source address operation	Increment address.	
	Transfer destination address operation	Increment address.	
	Transfer data size	1 byte	
	Transfer count	16	
	Generation condition for interrupt requests to CPU	Generation of interrupt request to CPU at completion of specified number of transfers	
2nd data transfer	Transfer mode	Block transfer mode	
	Chain transfer	Not used.	
	Transfer source address	0000 1010h	H'FF4010
	Transfer destination address	0000 1020h	H'FF4020
	Transfer source address operation	Increment address.	
	Transfer destination address operation	Increment address.	
	Block area	Transfer source	
	Transfer data size	16 bytes	
	Transfer count	1	

3.4.1 Operation

Figure 3.10 is a timing chart of chain transfer operation on the RX231 Group.

Table 3.21 describes chain transfer operation on the RX231 Group.

The numbers in Figure 3.10 correspond to the numbers in Table 3.21.

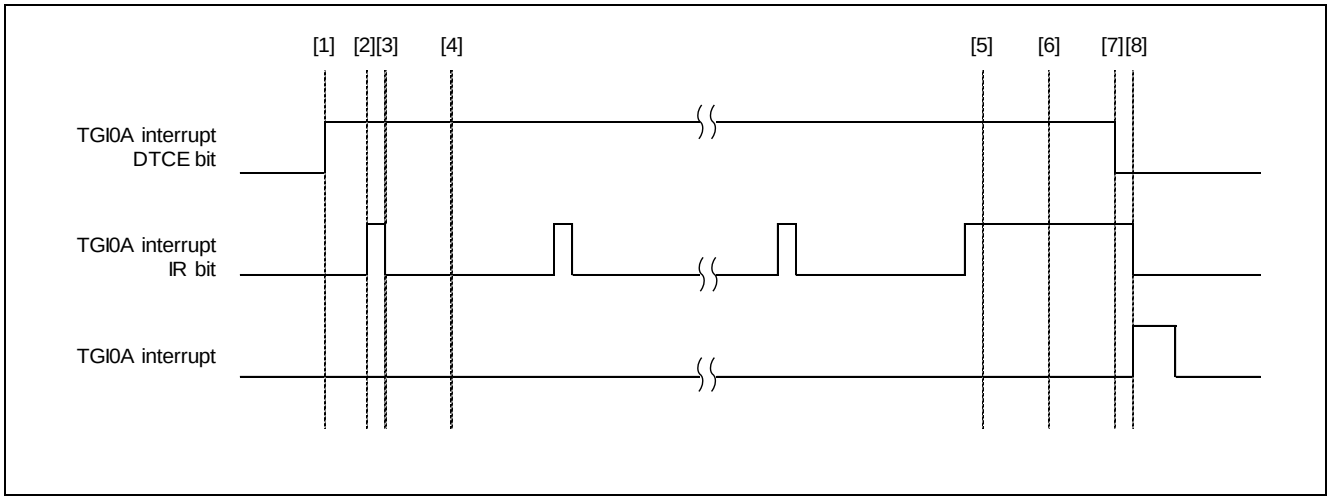


Figure 3.10 RX231 Group Chain Transfer Operation Timing

Table 3.21 Description of RX231 Group Chain Transfer Mode Operation

Item		RX (RX231)
[1]	Activation source selection	The TGI0A interrupt DTCE bit is set to 1.
[2]	DTC activation	When a compare match on TPU0 occurs, the TGI0A interrupt IR bit is set to 1 and a DTC activation request is generated. When the DTC is activated, the DTC vector corresponding to the vector number is read. Next, the transfer information is read from the transfer information storage address indicated by the DTC vector, and the 1st data transfer starts.
[3]	Data transfer start at less than specified count	Before transferring data, the DTC determines whether the specified number of transfers has completed. If the transfer count is less than the specified number of transfers (the CRA register value is other than 1), the IR bit is cleared to 0 when data transfer starts.
[4]	End of single transfer	When the first data transfer completes, the CRA register (transfer counter A) is decremented (−1). Also, the transfer information is written back to the RAM. If the transfer count is less than the specified number of transfers, the value of the DTCE bit remains 1. No interrupt request to the CPU is generated at this point.
[5]	Data transfer start at specified count	If the next transfer is a chain transfer, the value of the IR bit remains 1.
[6]	End of final transfer of specified count	If the next transfer is a chain transfer, the value of the DTCE bit remains 1. No interrupt request to the CPU is generated at this point.
[7]	End of 2nd transfer	When the 2nd transfer completes, the DTCE bit is cleared to 0. Also, the TGI0A interrupt request to the CPU is generated.
[8]	Clearing of IR bit	When the CPU receives the interrupt request, the IR bit is cleared to 0.

Figure 3.11 is a timing chart of chain transfer operation on the H8S/2378 Group.

Table 3.22 describes chain transfer operation on the H8S/2378 Group.

The numbers in Figure 3.11 correspond to the numbers in Table 3.22.

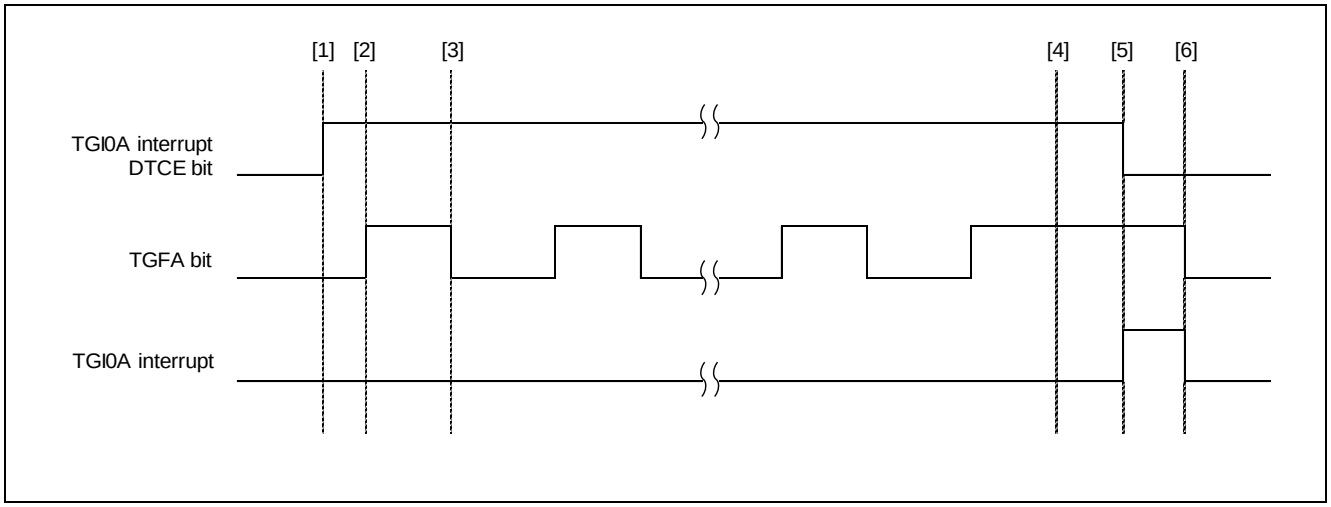


Figure 3.11 H8S/2378 Group Chain Transfer Operation Timing

Table 3.22 Description of H8S/2378 Group Chain Transfer Operation

Item		H8S (H8S/2378)
[1]	Activation source selection	The TGI0A interrupt DTCE bit is set to 1.
[2]	DTC activation	When a compare match on TPU_0 occurs, the TGFA bit is set to 1 and a DTC activation request is generated. When the DTC is activated, the DTC vector corresponding to the vector number is read. Next, the register information is read from the register information storage address indicated by the DTC vector, and the 1st data transfer starts.
[3]	End of single transfer	When the first data transfer completes, the CRA register (transfer counter A) is decremented (−1). Also, the register information is written back to the RAM. After this, the DTC determines whether the specified number of transfers has completed. If the transfer count is less than the specified number of transfers (the CRA register value is other than 0), the value of the DTCE bit remains 1 and the TGFA bit is cleared to 0. No interrupt request to the CPU is generated at this point.
[4]	End of final transfer of specified count	If the next transfer is a chain transfer, the DTCE and TGFA bits remain set to 1. No interrupt request to the CPU is generated at this point.
[5]	End of 2nd transfer	When the 2nd transfer completes, the DTCE bit is cleared to 0. Also, the TGI0A interrupt request to the CPU is generated.
[6]	Clearing of ADF bit	The TGFA bit is cleared to 0 by the TGI0A interrupt handler.

3.4.2 Points of Difference between Setting Procedures

Table 3.23 lists points of difference in the initial setting procedures for chain transfer operation. Note that the interrupt controller architectures RX231 Group and H8S/2378 Group differ. For details, refer to section 5, Points of Difference between Interrupts.

Table 3.23 Points of Difference between Initial Setting Procedures for Chain Transfer Operation

Procedure		RX (RX231)	H8S (H8S/2378)
1	TPU settings	Settings are made to the 16-bit timer pulse unit (TPU0).	Settings are made to the 16-bit timer pulse unit (TPU0).
2	Declaration of structure variable for storing register information	#pragma address dtc_data0 = 0x0000FBE8 st_dtc_data dtc_data0[2];	#pragma section Dtc_reg_info st_dtc_data dtc_data0[2]; #pragma section
3	Declaration of variable for storing DTC vector address*1	#pragma address dtc_vector142 = 0x0000FE38 unsigned long dtc_vector142;	#pragma section Dtc_vect_TGI0A const unsigned int vector_tgi0a = {0xBC00}; #pragma section
4	Setting in DTC vector address of register information start address*1	dtc_vector142 = (unsigned long)dtc_data0;	
5	Canceling of module stop state*2	SYSTEM.PRCR.WORD = 0xA502; MSTP(DTC) = 0; SYSTEM.PRCR.WORD = 0xA500;	MSTPCR.BIT._DTC = 0;
6	Disabling transfers	DTCE(TPU0, TGI0A) = 0; DTC.DTCST.BIT.DTCST = 0;	DTC.DTCEC.BIT.TGI0A = 0;
7	Disabling interrupts	IEN(TPU0, TGI0A) = 0;	set_imask_ccr(1);
8	Clearing RRS bit*3	DTC.DTCCR.BIT.RRS = 0;	No processing
9	Setting DTC address mode	DTC.DTCADMOD.BIT.SHORT = 1;	No processing

10	Setting register information (1st transfer) (MRA, MRB, SAR, DAR, and CRA registers)	dtc_data0[0].MRA_SAR.BIT.MRA_MD = 0;
		dtc_data0[0].MRA_SAR.BIT.MRA_SM = 2;
		dtc_data0[0].MRB_DAR.BIT.MRB_DM = 2;
		dtc_data0[0].MRA_SAR.BIT.MRA_SZ = 0;
		dtc_data0[0].MRB_DAR.BIT.MRB_CHNE = 1;
		dtc_data0[0].MRB_DAR.BIT.MRB_CHNS = 1;
		dtc_data0[0].MRB_DAR.BIT.MRB_DISEL = 0;
		dtc_data0[0].MRA_SAR.BIT.SAR
		= 0x00001000;
		dtc_data0[0].MRB_DAR.BIT.DAR
		= 0x00001010;
		dtc_data0[0].CRA_CRB.WORD.CRA = 16;
			dtc_data0[0].MRA_SAR.BIT.MRA_MD = 0;
			dtc_data0[0].MRA_SAR.BIT.MRA_SM = 2;
			dtc_data0[0].MRA_SAR.BIT.MRA_DM = 2;
			dtc_data0[0].MRA_SAR.BIT.MRA_Sz = 0;
			dtc_data0[0].MRB_DAR.BIT.MRB_CHNE = 1;
			dtc_data0[0].MRB_DAR.BIT.MRB_CHNS = 1;
			dtc_data0[0].MRB_DAR.BIT.MRB_DISEL = 0;
			dtc_data0[0].MRA_SAR.BIT.SAR
			= 0xFF4000;
			dtc_data0[0].MRB_DAR.BIT.DAR
			= 0xFF4010;
			dtc_data0[0].CRA_CRB.WORD.CRA = 16;

Procedure		RX (RX231)	H8S (H8S/2378)

11	Setting register information (2nd transfer) (MRA, MRB, SAR, DAR, CRA, and CRB registers)	dtc_data0[1].MRA_SAR.BIT.MRA_MD = 2;
		dtc_data0[1].MRB_DAR.BIT.MRB_DTS = 1;
		dtc_data0[1].MRA_SAR.BIT.MRA_SM = 2;
		dtc_data0[1].MRB_DAR.BIT.MRB_DM = 2;
		dtc_data0[1].MRA_SAR.BIT.MRA_SZ = 0;
		dtc_data0[1].MRB_DAR.BIT.MRB_CHNE = 0;
		dtc_data0[1].MRB_DAR.BIT.MRB_DISEL = 0;
		dtc_data0[1].MRA_SAR.BIT.SAR = 0x00001010;
		dtc_data0[1].MRB_DAR.BIT.DAR = 0x00001020;
		dtc_data0[1].CRA_CRB.WORD.CRA=0x1010;
		dtc_data0[1].CRA_CRB.WORD.CRB=0x0001;
			dtc_data0[1].MRA_SAR.BIT.MRA_MD = 2;
			dtc_data0[1].MRA_SAR.BIT.MRA_DTS = 1;
			dtc_data0[1].MRA_SAR.BIT.MRA_SM = 2;
			dtc_data0[1].MRA_SAR.BIT.MRA_DM = 2;
			dtc_data0[1].MRA_SAR.BIT.MRA_Sz = 0;
			dtc_data0[1].MRB_DAR.BIT.MRB_CHNE = 0;
			dtc_data0[1].MRB_DAR.BIT.MRB_DISEL = 0;
			dtc_data0[1].MRA_SAR.BIT.SAR = 0xFF4010;
			dtc_data0[1].MRB_DAR.BIT.DAR = 0xFF4020;
			dtc_data0[1].CRA_CRB.WORD.CRA=0x1010;
			dtc_data0[1].CRA_CRB.WORD.CRB=0x0001;

12	Setting DTC vector base address	DTC.DTCVBR = (void *)0x0000FC00;	No processing
13	Enabling DTC activation	DTCE(TPU0, TGI0A) = 1; DTC.DTCST.BIT.DTCST = 1;	DTC.DTCEC.BIT.TGI0A = 1;
14	Setting interrupt control mode	No processing	INTC.INTCR.BIT.INTM = 2;
15	Setting interrupt priority level	IPR(TPU0,TGI0A) = 0x01;	INTC.IPRF.BIT._TPU0 = 1;
16	Clearing peripheral function interrupt request	TPU0.TSR.BIT.TGFA = 0;	TPU0.TSR.BIT.TGFA = 0;
17	Clearing interrupt request	IR(TPU0,TGI0A) = 0;	No processing
18	Enabling interrupt requests	IEN(TPU0,TGI0A) = 1;	No processing
19	Setting processor interrupt priority level	No processing	set_imask_exr(0);
20	Enabling maskable interrupts	setpsw_i();	No processing
21	Enabling TGI0A interrupts	TPU0.TIER.BIT.TGIEA = 1;	TPU0.TIER.BIT.TGIEA = 1;
22	TPU0 operation start	TPU.TSTR.BIT.CST0 = 1;	TPU.TSTR.BIT.CST0 = 1;

Note 1. The setting steps used when allocating the DTC vector table in the RAM are shown.

Note 2. For information on the module stop function, refer to section 6, Module Stop Function.

Note 3. Clearing the DTCCR.RRS bit to 0 resets the flag for transfer information read skipping. If the DTC is activated after this bit is cleared to 0, no transfer information read skipping takes place. Make this setting after the transfer information is updated.

4. Operation Timing and Number of States

Examples of DTC operation timing during normal transfer and repeat transfer operation are shown below. For detailed information on determining the actual cycle counts, refer to the hardware manual of each MCU group.

Figure 4.1 shows an example of DTC operation timing (short-address mode) on the RX231 Group.

Figure 4.2 shows an example of DTC operation timing on the H8S/2378 Group.

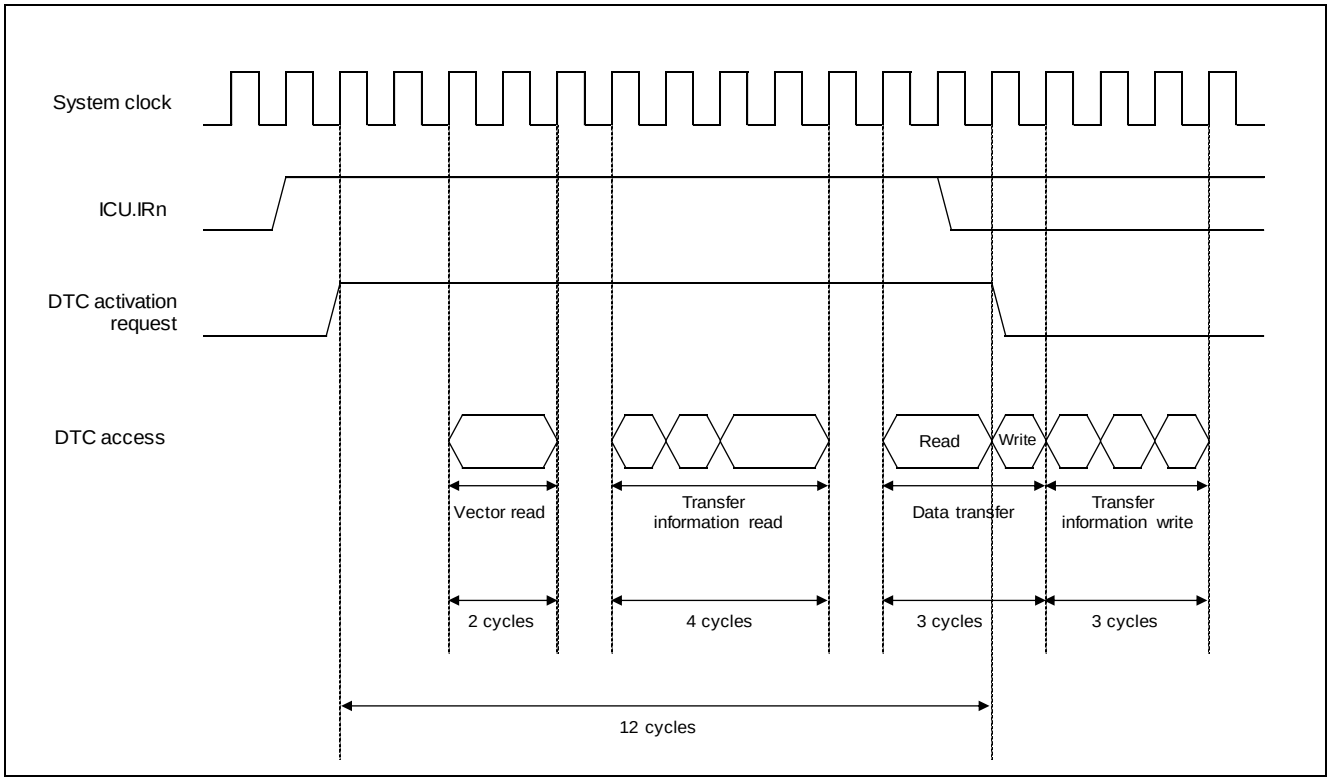


Figure 4.1 RX231 Group DTC Operation Timing Example (Short-Address Mode, Normal Transfer Mode, and Repeat Transfer Mode)

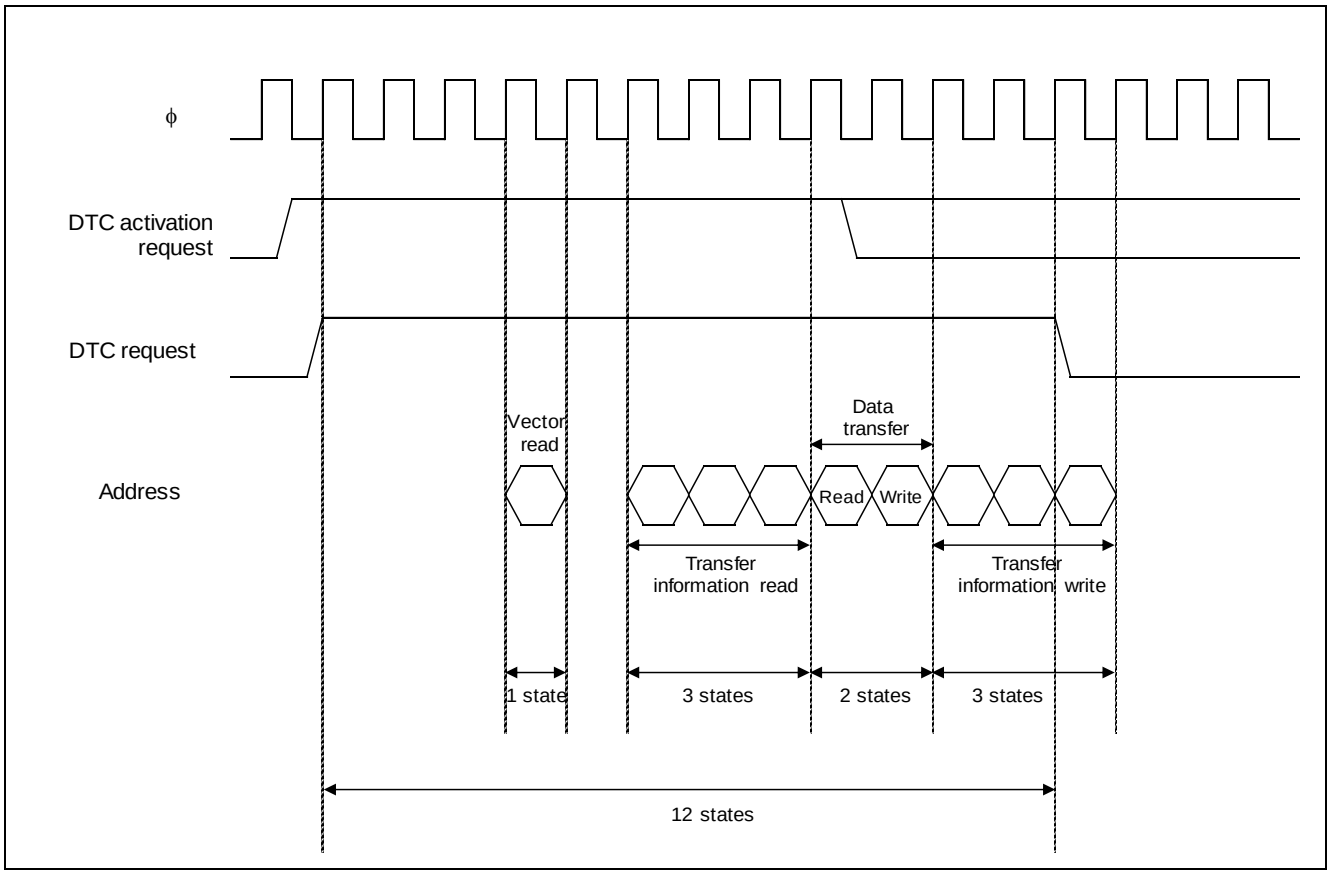


Figure 4.2 H8S/2378 Group DTC Operation Timing Example (Normal Mode and Repeat Mode)

Examples of DTC operation timing during block transfer operation are shown below.

Figure 4.3 shows an example of DTC operation timing (short-address mode) on the RX231 Group.

Figure 4.4 shows an example of DTC operation timing on the H8S/2378 Group.

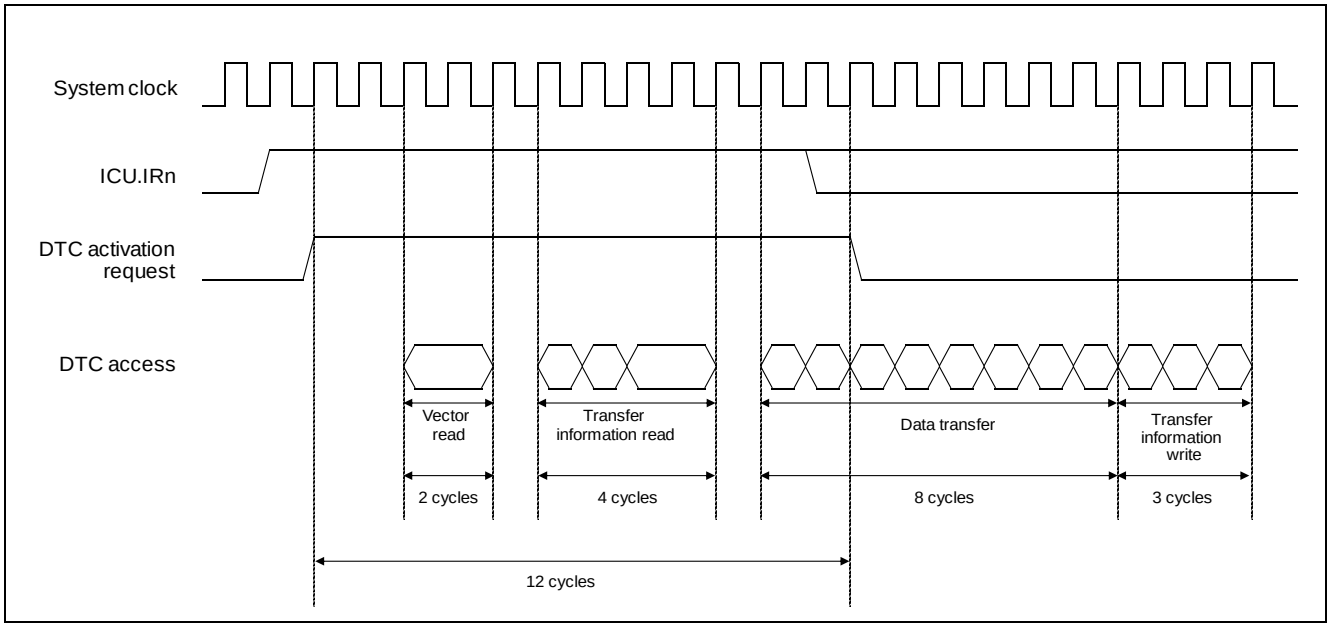


Figure 4.3 RX231 Group DTC Operation Timing Example (Short-Address Mode and Block Transfer Mode)

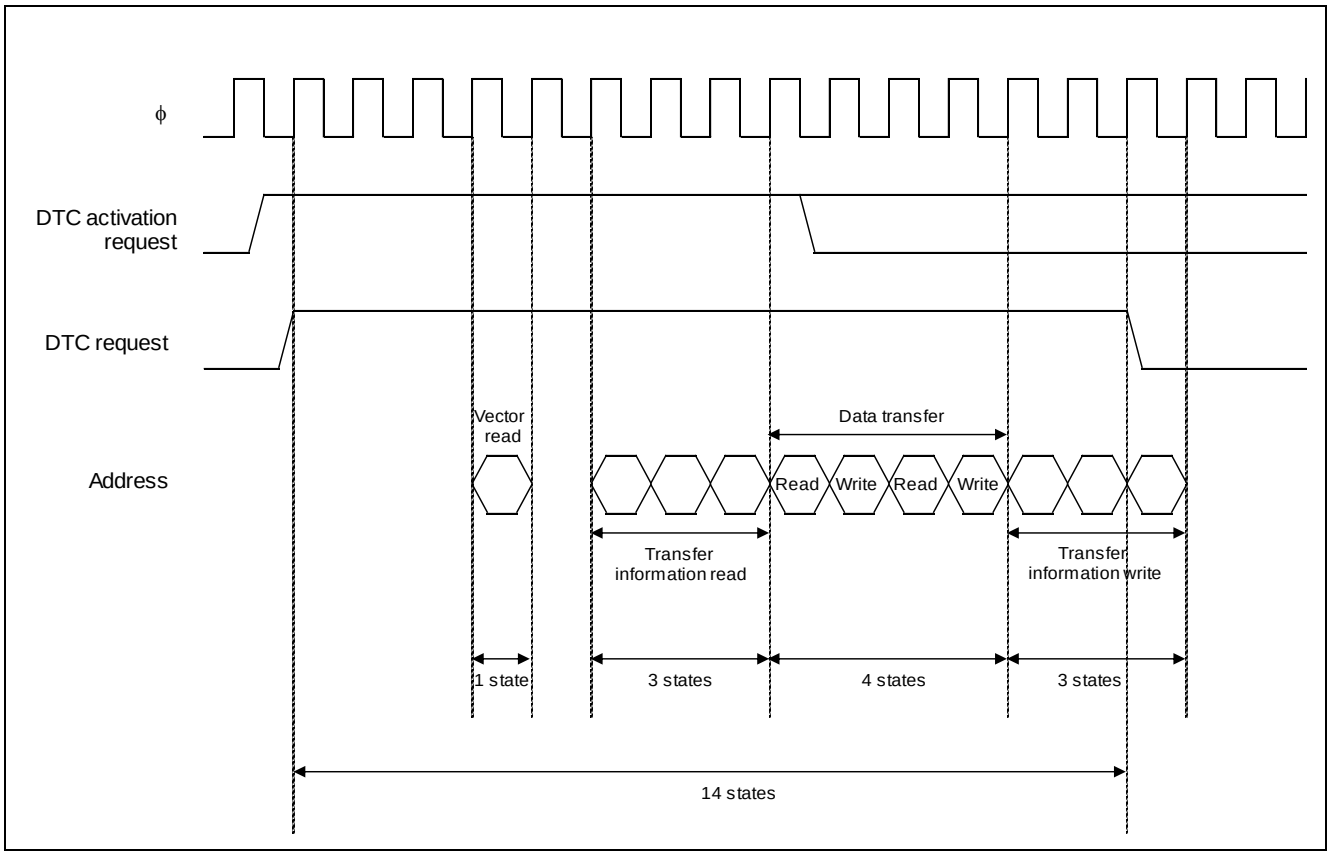


Figure 4.4 H8S/2378 Group DTC Operation Timing Example (Block Transfer Mode)

Examples of DTC operation timing during chain transfer operation are shown below.

Figure 4.5 shows an example of DTC operation timing (short-address mode) on the RX231 Group.

Figure 4.6 shows an example of DTC operation timing on the H8S/2378 Group.

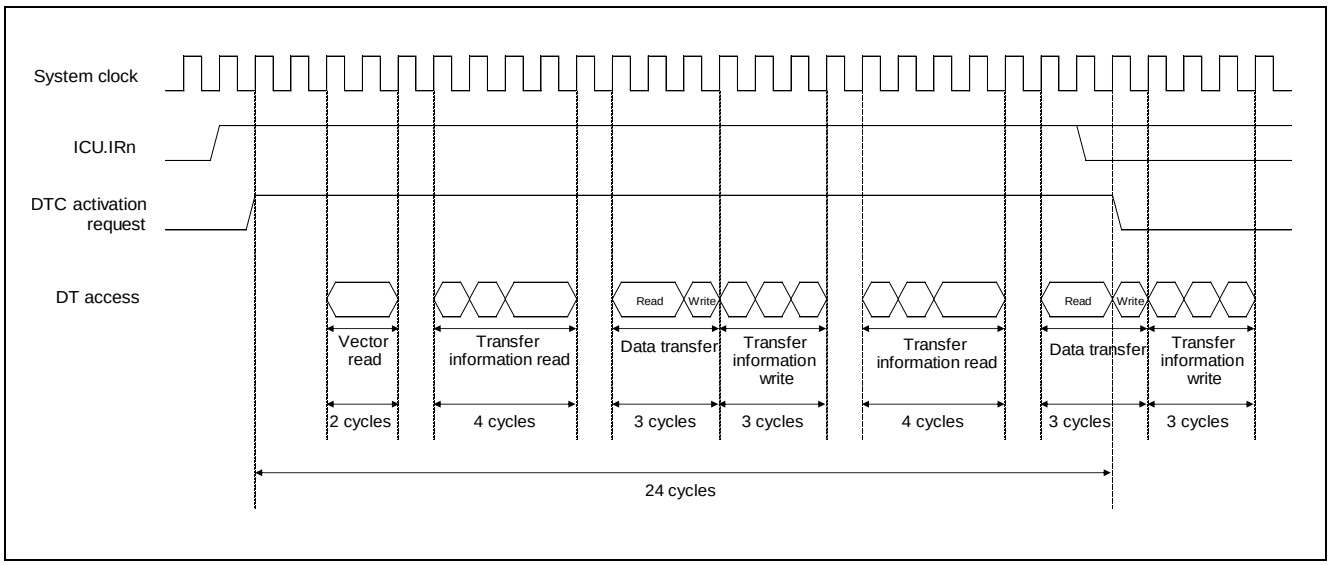


Figure 4.5 RX231 Group DTC Operation Timing Example (Chain Transfer Mode)

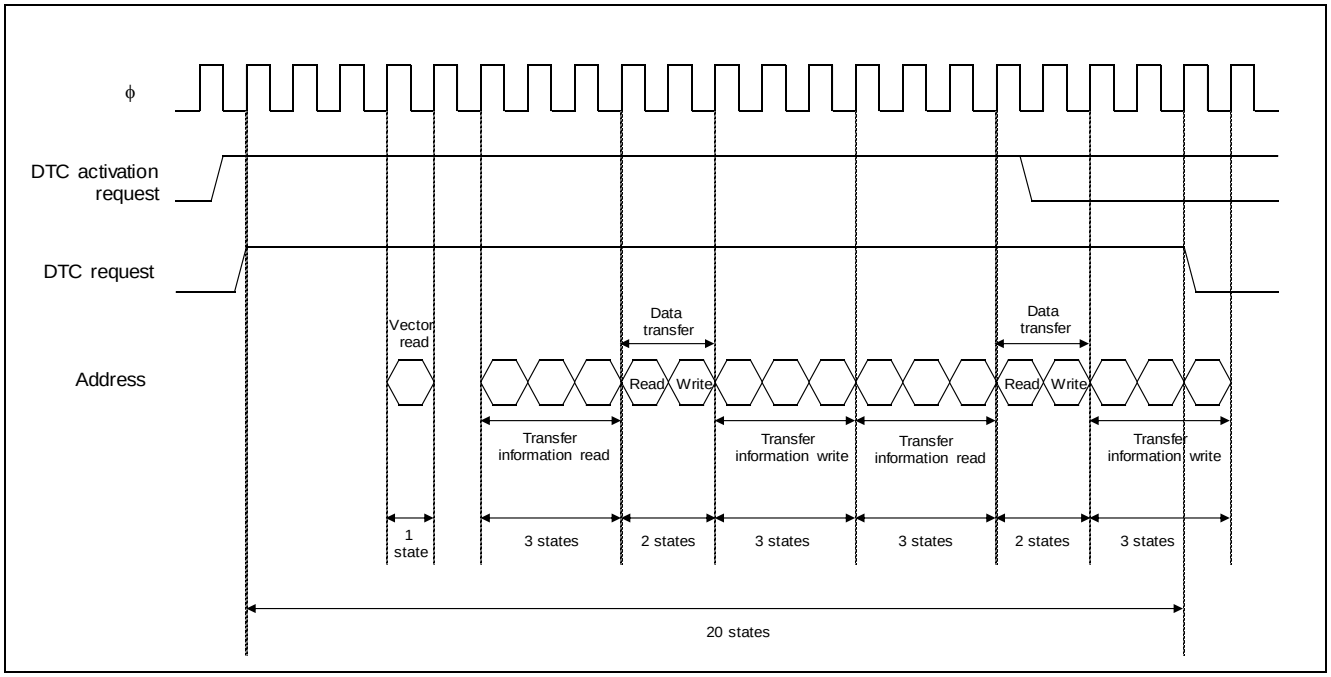


Figure 4.6 H8S/2378 Group DTC Operation Timing Example (Chain Transfer Mode)

Figure 4.7 shows an example of DTC operation timing (full-address mode, normal transfer mode, and repeat transfer mode) on the RX231 Group.

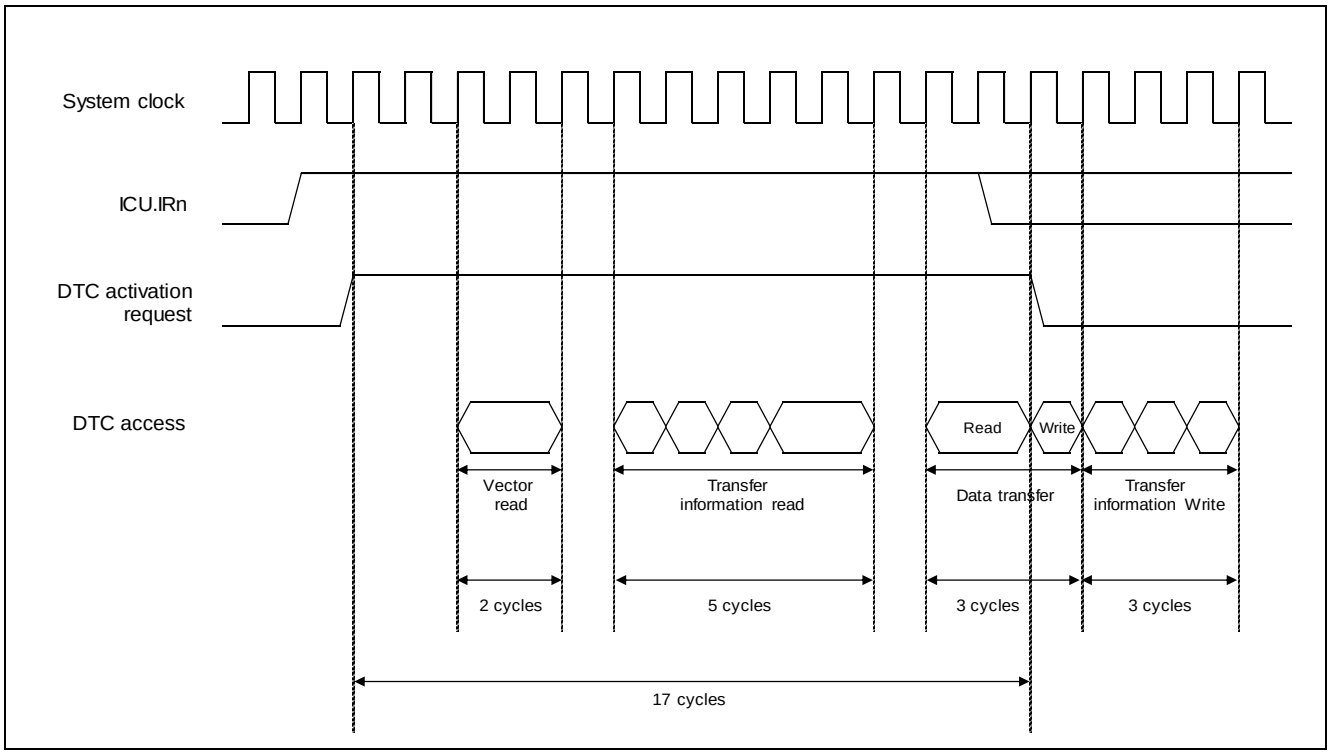


Figure 4.7 RX231 Group DTC Operation Timing Example (Full-Address Mode, Normal Transfer Mode, and Repeat Transfer Mode)

5. Points of Difference between Interrupts

Table 5.1 lists the registers and bits used for DTC activation.

Table 5.1 Registers and Bits Used for DTC Activation

RX (RX231)		H8S (H8S/2378)	
Register.Bit	**Function**	**Register.Bit**	**Function**
DTCERn,DTCE (n = interrupt vector number)	Enabling DTC activation	DTCERA to H.DTCE	Enabling DTC activation
SWINTR.SWINT	Software interrupt activation	DTVECR.SWDTE	Enabling DTC activation by software

On the RX231 Group the DTCERn registers and the SWINTR register are incorporated into the interrupt controller, and on the H8S/2378 Group registers DTCERA to DTCERH and the DTVECR register are incorporated into the DTC.

Table 5.2 lists DTC interrupt sources.

Table 5.2 DTC Interrupt Sources

RX (RX231)		H8S (H8S/2378)	
Activation Source	**Interrupt Source**	**Activation Source**	**Interrupt Source**
Interrupt	Generation of request to the CPU for the interrupt that activated the DTC*1	Software	Generation of software-activated data transfer end interrupt (SWDTE)
		Interrupt	Generation of request to the CPU for the interrupt that activated the DTC

Note 1. On the RX231 Group the DTC can be activated by software by means of a software interrupt.

Interrupts can be accepted on the RX231 Group when the following conditions are met:

- The I flag (PSW.I bit) is set to 1.
- The interrupt is enabled in the IER and IPR registers of the ICU.
- Interrupt requests are enabled by the corresponding peripheral function interrupt request enable bit.

Table 5.3 is a comparative listing of the interrupt generation conditions on the RX231 Group and H8S/2378 Group.

Table 5.3 Comparative Listing of Interrupt Generation Conditions on RX231 Group and H8S/2378 Group

Item	RX (RX231 Group)	H8S (H8S/2378)
Interrupt enable bit (I bit)	Setting the I bit in the PSW register to 1 (enabled) enables acceptance of maskable interrupts.	In interrupt control mode 0, setting the I bit to 0 (enabled) in the CCR register enables acceptance of maskable interrupts. In interrupt control mode 2 the I bit in the CCR register is not used.
Processor interrupt priority level	Only interrupt requests with a higher priority level than that indicated by the IPL[3:0] bits in the PSW register are accepted.	In interrupt control mode 2 only interrupt requests with a higher priority level than that indicated by bits I2 to I0 in the EXR register are accepted. In interrupt control mode 0 the bits I2 to I0 in the EXR register is not used.
Interrupt priority level	Set in the IPR register.	In interrupt control mode 0 the default settings are used. In interrupt control mode 2 the IPR register settings are used.
Interrupt request flag	The interrupt controller manages all interrupt status flags for peripheral functions, external pins, NMI interrupts, etc.	The interrupt controller manages interrupt status flags for external interrupts, and interrupt status flags for internal interrupt sources are managed within each on-chip peripheral function.
Interrupt request enable	Set in the IER register for maskable interrupts and in the NMIER register for non-maskable interrupts.	IRQ interrupts are enabled by settings in the IER register.
Peripheral function interrupt enable	Interrupts can be enabled or disabled by each peripheral function.	

Table 5.4 lists points of difference in the enabling and priority levels of processor interrupts.

On the RX231 Group the processor interrupt priority level is 0 (lowest level) by default when the PSW.I bit is set to 1 (interrupt enabled), so maskable interrupts are enabled.

On H8S/2378 Group, in interrupt control mode 0, processor interrupt priority levels are not used when the CCR.I bit is cleared to 0 (interrupt enabled), so maskable interrupts are enabled.

On H8S/2378 Group, in interrupt control mode 2, the processor interrupt priority level is 7 (highest level) by default, so maskable interrupts are enabled by setting bits I2 to I0 in EXR.

Table 5.4 Points of Difference in Enabling and Priority Levels of Processor Interrupts

		H8S (H8S/2378)	
Item	**RX (RX231)**	**Interrupt Control Mode 0**	**Interrupt Control Mode 2**
Interrupt enable default value	PSW.I bit: 0 (interrupt mask)	CCR.I bit: 1 (interrupt mask)	Not used
Processor interrupt priority level default value	PSW.IPL[3:0] bits: 0000b (lowest level)	Not used	EXR bits I2 to I0: 111b (highest level)
Operation after a reset	Maskable interrupts are not accepted.		

Table 5.5 lists some of the embedded functions used for enabling interrupts.

Table 5.5 Embedded Functions Used for Enabling Interrupts (Partial Listing)

	Description		
		H8S (H8S/2378)	
Item	**RX (RX231)**	**Interrupt Control Mode 0**	**Interrupt Control Mode 2**
Processor interrupt enable setting	setpsw_i(); *1	set_imask_ccr(0); *1	Not used
Processor interrupt priority level setting (setting = 0)	set_ipl(0); *1	Not used	set_imask_exr(0); *1

Note 1. The file machine.h must be included.

For details, refer to the sections describing the interrupt controller (ICU), CPU, and peripheral functions used in User's Manual: Hardware.

6. Module Stop Function

On the H8S/2378 Group and RX231 Group it is possible to halt the functioning of individual peripheral modules.

Power consumption can be reduced by transitioning unused peripheral modules to the module stop state. Modules not listed in Table 6.1 are in the module stop state after a reset.

Table 6.1 Modules that Operate under Initial Settings on RX231 Group and H8S/2378 Group

RX (RX231)	H8S (H8S/2378)
DMAC, DTC, RAM	EXDMAC, DMAC, DTC

When a module is in the module stop state, its registers cannot be read or written to.

Before using any module not listed in Table 6.1, it is necessary to cancel the module stop state and then make initial settings.

For details, refer to the section describing the low power consumption functions in User's Manual: Hardware.

7. Register Write Protection Function

On the RX231 Group it is possible to protect important registers from being overwritten if program runaway occurs. The protect register (PRCR) is used to specify the registers that are protected by this function.

Register protection can be enabled for the clock generation circuit–related registers, flash memory–related registers, operating mode–related registers, low power consumption function–related registers, low-power timer–related registers, LVD–related registers, and software reset register.

For details, refer to the section on the register write protection function in User's Manual: Hardware.

8. Key Points when Migrating from H8S to RX

Some points to keep in mind when migrating from the H8S/2378 Group to the RX231 Group are described below.

8.1 I/O Ports

On the RX231 Group it is necessary to make settings to the MPC to assign pins to peripheral function I/O signals.

To apply I/O control to a pin on the RX231 Group, make the following two settings:

- PFS register of MPC: Select the peripheral function to be assigned to the pin.
- PMR register of I/O port: Select whether to assign the pin to a general I/O port or a peripheral function.

Table 8.1 provides a comparative listing of I/O settings for peripheral function pins on the RX231 Group and H8S/2378 Group.

Table 8.1 Comparison of I/O settings for Peripheral Function Pins on RX231 Group and H8S/2378 Group

Function	RX (RX231)	H8S (H8S/2378)
Pin function selection	I/O pins for peripheral functions can be assigned from a selection of multiple pins by making settings in the PFS register.	Pins can be switched between general I/O port and peripheral function settings and pin functions selected through combinations of the MCU operating mode, the setting of the SYSCR.EXPE bit, the PFCR registers, the DDR registers, and the settings of the various peripheral functions.
General I/O port/peripheral function switching	Settings in the PMR register can be used to select whether specific pins are used as I/O ports or as peripheral functions.	

For details, refer to the sections describing the multi-function pin controller (MPC) and I/O ports in User's Manual: Hardware.

8.2 I/O Register Macros

The macro definitions listed below are contained in the I/O register definition file (iodefine.h) of the RX231 Group.

Using macro definitions can make program code easier to read.

Table 8.2 lists macro usage examples.

Table 8.2 Macro Usage Examples

Macro	Usage Example
IR(“module name”,”bit name”)	IR(MTU0,TGIA0) = 0; Clears the IR bit corresponding to TGIA0 of MTU0 to 0 (clear interrupt request).
DTCE(“module name”,”bit name”)	DTCE(MTU0,TGIA0) = 1; Sets the DTCE bit corresponding to TGIA0 of MTU0 to 1 (enable DTC start).
IEN(“module name”,”bit name”)	IEN(MTU0,TGIA0) = 1; Sets the IEN bit corresponding to TGIA0 of MTU0 to 1 (enable interrupt).
IPR(“module name”,”bit name”)	IPR(MTU0,TGIA0) = 0x02; Sets the IPR bits corresponding to TGIA0 of MTU0 to 2 (interrupt priority level 2).
MSTP(“module name”)	MSTP(MTU) = 0; Clears the module stop setting bit of MTU0 to 0 (cancel module stop state).
VECT(“module name”,”bit name”)	#pragma interrupt(Excep_MTU0_TGIA0(vect=VECT(MTU0,TGIA0))) Declares the interrupt function corresponding to TGIA0 of MTU0.

8.3 Embedded Functions

On the RX231 Group interrupt functions are provided to implement control register settings or special instructions. To use these embedded functions, include the file machine.h.

Table 8.3 lists (examples of) points of difference between control register settings and special instructions on the RX231 Group and H8S/2378 Group.

Table 8.3 Points of Difference between Control Register Settings and Special Instructions on RX231 Group and H8S/2378 Group (Example)

	Format	
Item	**RX (RX231)**	**H8S (H8S/2378)**
Set I flag to 1.	setpsw_i(); ∗1	set_imask_ccr(1); ∗1∗2
Clear I flag to 0.	clrpsw_i(); ∗1	set_imask_ccr(0); ∗1∗2
Expand to WAIT instruction.	wait(); ∗1	None
Expand to NOP instruction.	nop(); ∗1	nop(); ∗1

Note 1. It is necessary to include the file machine.h.

Note 2. I = 1 means enable interrupts on the RX231 Group, and I = 1 means mask interrupts on the H8S/2378 Group.

9. Reference Documents

User's Manual: Hardware

H8S/2378 Group, H8S/2378R Group Hardware Manual Rev.7.00 (REJ09B0109-0700)
RX230 Group and RX231 Group User's Manual: Hardware Rev.1.10 (R01UH0496EJ0110)
(The latest versions can be downloaded from the Renesas Electronics website.)

Application Notes

RX Family, M16C Family Migrating From the M16C Family to the RX Family: DMAC and DTC Application Note (R01AN2099EJ0100)
RX631 Group SH7044 to RX631 Microcontroller Migration Guide Application Note (R01AN2207EJ0100)
(The latest versions can be downloaded from the Renesas Electronics website.)

Technical Update/Technical News

(The latest versions can be downloaded from the Renesas Electronics website.)

User's Manual: Development Environment

CC-RX Compiler User's Manual Rev.1.05 (R20UT3248EJ0105)
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor
Compiler Package Ver.6.01 User's Manual (REJ10B0161-0100)
(The latest versions can be downloaded from the Renesas Electronics website.)

Website and Support

Renesas Electronics Website
http://www.renesas.com/
Inquiries
http://www.renesas.com/contact/

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jul. 13, 2018	—	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0