RX Family and M32C/R32C Series

Guide for Migration from the M32C/R32C to the RX: DMAC and DTC

Abstract

This document describes migration from the DMAC in the M32C/R32C Series to the DMAC and DTC in the RX Family.

Products

- RX Family
- M32C/80 Series
- R32C/100 Series

This document explains migration from the M32C/R32C Series to the RX Family, using the RX660 Group MCU as an example of the RX Family, the M32C/87 Group MCU as an example of the M32C/80 Series MCU, and the R32C/118 Group MCU as an example of the R32C/100 Series MCU. When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

There are differences in terminology between the RX Family MCU and the M32C/R32C Series MCU.

The table below shows the differences in terminology related to the DMAC and DTC.

Differences in Terminology Between the MCUs of the RX Family and M32C/R32C Series

Item	M32C/R32C Series	RX Family	
	DMAC	DMAC	DTC
Mode for transferring specific data once	Single transfer mode	Normal transfer mode	Normal transfer mode
Mode for transferring specific data multiple times	Repeat transfer	Repeat transfer mode	Repeat transfer mode
Peripheral function registers	Special function registers (SFRs)	I/O registers	

Contents

1. General Differences Between the DMAC/DTC in the RX Family and the DMAC/DMACII in the M32C/R32C Series

Table 1.1 to Table 1.4 show General Differences Between the DMAC in the M32C/87 and the DMAC in the RX660. Table 1.5 shows Difference in the Number of Cycles Between the DMAC/DMACII (M32C/87) and the DMAC/DTC (RX660). Table 1.6 to Table 1.9 show General Differences Between the DMAC in the R32C/118 and the DMAC in the RX660. Table 1.10 shows Difference in the Number of Cycles Between the DMAC/DMACII (R32C/118) and the DMAC/DTC (RX660).

If speed is given priority to transfer data, select the DMAC; if data is to be transferred according to the user-selected request source and transfer method, select the DTC.

- For the DMAC, the transfer address, transfer mode, and other settings are specified in I/O registers provided for each channel. After a request source is generated, data transfer starts according to the I/O register setting values. Therefore, the processing time until transfer starts is shorter than that for the DTC.
- For the DTC, the transfer address, transfer mode, and other information is set in memory (ROM and RAM). After a request source is generated, the information is read from the memory, and then data transfer starts. Therefore, the processing time until transfer starts is longer than that for the DMAC. However, compared to the DMAC, more request sources and transfer methods can be selected, and there is no limit on the number of transfer channels.

Table 1.1 General Differences Between the DMAC in the M32C/87 and the DMAC in the RX660

Item	M32C (M32C/87)	RX (RX660)
	DMAC	DMAC
Number of channels	4 channels	8 channels
Bus request arbitration	Cycle stealing	Fixed priority or round-robin
Transfer space	000000h to FFFFFFh	0000 0000h to 0FFF FFFFh, F000 0000h to FFFF FFFFh
Maximum number of transferred bytes	128 KB (in a 16-bit transfer) / 64 KB (in an 8-bit transfer)	256 MB [*1]
Request source	43 sources • INT pin (falling edge) (4 sources) • INT pin (both edges) (4 sources) • Timers (11 sources) • Communications (10 sources) • A/D conversion (1 source) • Intelligent I/O (12 sources) • Software trigger	112 sources • INT pin (16 sources): Choose Low, falling edge, rising edge, or both edges • Timers (47 sources) • Communications (39 sources) • A/D conversion and comparator (7 sources) • ELC (2 sources) • Software trigger
Channel priority	DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has the highest priority.)	Channel 0 > channel 1 > channel 2 > channel 3 ... > channel 7 (Channel 0 has the highest priority.)
Transfer units	8 or 16 bits	8, 16, or 32 bits
Transfer direction	Non-change → Forward, or Forward → Non-change	4 conditions selectable for the transfer source and transfer destination • Non-change • Increment-by-offset [*2] • Increment • Decrement
Transfer modes	• Single transfer mode • Repeat transfer mode	• Normal transfer mode • Repeat transfer mode • Block transfer mode
Interrupts	An interrupt is generated when the value of the DCTi register changes from "0001h" to "0000h".	• Normal transfer mode: Generated when the specified number of transfers end; Repeat transfer mode: Generated when the specified number of times a transfer is repeated; Block transfer mode: Generated when the specified number of blocks are transferred (transfer end interrupt). • Generated when data of the repeat size is transferred or when the extended repeat area overflows (transfer escape end interrupt).

Notes: 1. The number of transferred bytes becomes the maximum when data is transferred in block transfer mode under the following conditions: 1,024 data pieces × 65,536 blocks × 32 bits.

2. Increment-by-offset can be set for DMAC0 only.

Table 1.2 General Differences Between the DMAC in the M32C/87 and the DTC in the RX660

Item	M32C (M32C/87)	RX (RX660)
	DMAC	DTC
Number of channels	4 channels	— *1
Bus request arbitration	Cycle stealing	Fixed priority or round-robin
Transfer space	000000h to FFFFFFh	Short-address mode: 0000 0000h to 007F FFFFh, FF80 0000h to FFFF FFFFh Full-address mode: 0000 0000h to FFFF FFFFh
Maximum number of transferred bytes	128 KB (in a 16-bit transfer) / 64 KB (in an 8-bit transfer)	1,024 bytes *2
Request source	43 sources • INT pin (falling edge) (4 sources) • INT pin (both edges) (4 sources) • Timers (11 sources) • Communications (10 sources) • A/D conversion (1 source) • Intelligent I/O (12 sources) • Software trigger	126 sources • INT pin (18 sources): Choose Low, falling edge, rising edge, or both edges • Timers (55 sources) • Communications (43 sources) • A/D conversion and comparator (7 sources) • ELC (2 sources) • Software trigger
Channel priority	DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has the highest priority.)	— *1
Transfer units	8 or 16 bits	8, 16, or 32 bits
Transfer direction	Non-change → Forward, or Forward → Non-change	3 conditions selectable for the transfer source and transfer destination • Non-change • Increment • Decrement
Transfer modes	• Single transfer mode • Repeat transfer mode	• Normal transfer mode • Repeat transfer mode • Block transfer mode
Interrupts	An interrupt is generated when the value of the DCTi register changes from “0001h” to “0000h”.	• Generated when the DTC starts • Generated after a single data transfer ends • Generated after the specified number of data units are transferred

Notes: 1. There is no concept of channels in the DTC. Transfer corresponding to the interrupt source is possible.

2. The number of transferred bytes becomes the maximum when a transfer is performed 256 times with a data size of 1 longword (32 bits) in repeat transfer mode or when the block size is 256 data pieces × 32 bits in block transfer mode.

Table 1.3 General Differences Between the DMACII in the M32C/87 and the DMAC in the RX660

Item	M32C (M32C/87)	RX (RX660)
	DMACII	DMAC
Transfer space	00000h to 0FFFFh *1	0000 0000h to 0FFF FFFFh, F000 0000h to FFFF FFFFh
Maximum number of transferred bytes	128 KB *2	256 MB *3
Request source	Interrupt requests from all peripheral functions for which the ILVL2 to ILVL0 bits of the interrupt control register are set to “111b” (level 7)	112 sources • INT pin (16 sources): Choose Low, falling edge, rising edge, or both edges • Timers (47 sources) • Communications (39 sources) • A/D conversion and comparator (7 sources) • ELC (2 sources) • Software trigger
Transfer units	8 or 16 bits	8, 16, or 32 bits
Transfer direction	2 conditions selectable for the transfer source and transfer destination • Fixed addresses • Addresses in the forward direction	4 conditions selectable for the transfer source and transfer destination • Non-change • Increment-by-offset *4 • Increment • Decrement
Transfer modes	• Single transfer mode • Burst transfer mode • Multi-transfer mode	• Normal transfer mode • Repeat transfer mode • Block transfer mode
Interrupts	An interrupt is generated when the value of the transfer counter becomes 0.	• Normal transfer mode: Generated when the specified number of transfers end; Repeat transfer mode: Generated when the specified number of times a transfer is repeated; Block transfer mode: Generated when the specified number of blocks are transferred (transfer end interrupt). • Generated when data of the repeat size is transferred or when the extended repeat area overflows (transfer escape end interrupt).

Notes: 1. Note, however, that if the unit of transfer data is 16 bits and the destination address is 0FFFFh, data is transferred to addresses 0FFFFh and 10000h. This also applies when the transfer source address is 0FFFFh.

2. The number of transferred bytes becomes the maximum when 16-bit data is transferred 65,536 times in burst transfer mode.

3. The number of transferred bytes becomes the maximum when data is transferred in block transfer mode under the following conditions: 1,024 data pieces × 65,536 blocks × 32 bits.

4. Increment-by-offset can be set for DMAC0 only.

Table 1.4 General Differences Between the DMACII in the M32C/87 and the DTC in the RX660

Item	**M32C (M32C/87)**	**RX (RX660)**
	DMACII	**DTC**
Transfer space	00000h to 0FFFFh [*1]	Short-address mode: 0000 0000h to 007F FFFFh, FF80 0000h to FFFF FFFFh Full-address mode: 0000 0000h to FFFF FFFFh
Maximum number of transferred bytes	128 KB [*2]	1,024 bytes [*3]
Request source	Interrupt requests from all peripheral functions for which the ILVL2 to ILVL0 bits of the interrupt control register are set to "111b" (level 7)	126 sources • INT pin (18 sources): Choose Low, falling edge, rising edge, or both edges • Timers (55 sources) • Communications (43 sources) • A/D conversion and comparator (7 sources) • ELC (2 sources) • Software trigger
Transfer units	8 or 16 bits	8, 16, or 32 bits
Transfer direction	2 conditions selectable for the transfer source and transfer destination • Fixed addresses • Addresses in the forward direction	3 conditions selectable for the transfer source and transfer destination • Non-change • Increment • Decrement
Transfer modes	• Single transfer mode • Burst transfer mode • Multi-transfer mode	• Normal transfer mode • Repeat transfer mode • Block transfer mode
Interrupts	An interrupt is generated when the value of the transfer counter becomes 0.	• Generated when the DTC starts • Generated after a single data transfer ends • Generated after the specified number of data units are transferred

Notes: 1. Note, however, that if the unit of transfer data is 16 bits and the destination address is 0FFFFh, data is transferred to addresses 0FFFFh and 10000h. This also applies when the transfer source address is 0FFFFh.

2. The number of transferred bytes becomes the maximum when 16-bit data is transferred 65,536 times in burst transfer mode.

3. The number of transferred bytes becomes the maximum when a transfer is performed 256 times with a data size of 1 longword (32 bits) in repeat transfer mode or when the block size is 256 data pieces × 32 bits in block transfer mode.

Table 1.5 Difference in the Number of Cycles Between the DMAC/DMACII (M32C/87) and the DMAC/DTC (RX660)

Item	M32C (M32C/87)		RX (RX660)	
	DMAC	DMACII	DMAC	DTC
Number of cycles [*1]	7 cycles (includes 1 dummy cycle)	32 cycles	6 cycles	18 cycles

Note: 1. This table assumes the following: 8-bit data transfer, transfer source is an I/O register (fixed), the transfer destination is the RAM (increment), the DTC is in full-address mode, and DTC vector table is the ROM, and ICLK = PCLK multiplied by 2.

Table 1.6 General Differences Between the DMAC in the R32C/118 and the DMAC in the RX660

Item	R32C (R32C/118)	RX (RX660)
	DMAC	DMAC
Bus request arbitration	Cycle stealing	Fixed priority or round-robin
Transfer space	0000 0000h to 01FFF FFFh, FE00 0000h to FFFF FFFFh	0000 0000h to 0FFF FFFFh, F000 0000h to FFFF FFFFh
Maximum number of transferred bytes	64 MB (in a 32-bit transfer) / 32 MB (in a 16-bit transfer) / 16 MB (in an 8-bit transfer)	256 MB [*1]
Request source	57 sources • INT pin (falling edge) (7 sources) • INT pin (both edges) (7 sources) • Timers (11 sources) • Communications (18 sources) • A/D conversion (1 source) • Intelligent I/O (12 sources) • Software trigger	112 sources • INT pin (16 sources): Choose Low, falling edge, rising edge, or both edges • Timers (47 sources) • Communications (39 sources) • A/D conversion and comparator (7 sources) • ELC (2 sources) • Software trigger
Channel priority	DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has the highest priority.)	Channel 0 > channel 1 > channel 2 > channel 3 ... > channel 7 (Channel 0 has the highest priority.)
Transfer units	8, 16, or 32 bits	8, 16, or 32 bits
Transfer direction	2 conditions selectable for the transfer source and transfer destination • Non-change • Increment	4 conditions selectable for the transfer source and transfer destination • Non-change • Increment-by-offset [*2] • Increment • Decrement
Transfer modes	• Single transfer mode • Repeat transfer mode	• Normal transfer mode • Repeat transfer mode • Block transfer mode
Interrupts	An interrupt is generated when the value of the DCTi register changes from “00000001h” to “00000000h”.	• Normal transfer mode: Generated when the specified number of transfers end; Repeat transfer mode: Generated when the specified number of times a transfer is repeated; Block transfer mode: Generated when the specified number of blocks are transferred (transfer end interrupt). • Generated when data of the repeat size is transferred or when the extended repeat area overflows (transfer escape end interrupt).

Notes: 1. The number of transferred bytes becomes the maximum when data is transferred in block transfer mode under the following conditions: 1,024 data pieces × 65,536 blocks × 32 bits.

2. Increment-by-offset can be set for DMAC0 only.

Table 1.7 General Differences Between the DMAC in the R32C/118 and the DTC in the RX660

Item	R32C (R32C/118)	RX (RX660)
	DMAC	DTC
Number of channels	4 channels	— [*1]
Bus request arbitration	Cycle stealing	Fixed priority or round-robin
Transfer space	0000 0000h to 01FFFFFFh, FE00 0000h to FFFF FFFFh	Short-address mode: 0000 0000h to 007F FFFFh, FF80 0000h to FFFF FFFFh Full-address mode: 0000 0000h to FFFF FFFFh
Maximum number of transferred bytes	64 MB (in a 32-bit transfer) / 32 MB (in a 16-bit transfer) / 16 MB (in an 8-bit transfer)	1,024 bytes [*2]
Request source	57 sources • INT pin (falling edge) (7 sources) • INT pin (both edges) (7 sources) • Timers (11 sources) • Communications (18 sources) • A/D conversion (1 source) • Intelligent I/O (12 sources) • Software trigger	126 sources • INT pin (18 sources): Choose Low, falling edge, rising edge, or both edges • Timers (55 sources) • Communications (43 sources) • A/D conversion and comparator (7 sources) • ELC (2 sources) • Software trigger
Channel priority	DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has the highest priority.)	— [*1]
Transfer units	8, 16, or 32 bits	8, 16, or 32 bits
Transfer direction	2 conditions selectable for the transfer source and transfer destination • Non-change • Increment	3 conditions selectable for the transfer source and transfer destination • Non-change • Increment • Decrement
Transfer modes	• Single transfer mode • Repeat transfer mode	• Normal transfer mode • Repeat transfer mode • Block transfer mode
Interrupts	An interrupt is generated when the value of the DCTi register changes from “00000001h” to “00000000h”.	• Generated when the DTC starts • Generated after a single data transfer ends • Generated after the specified number of data units are transferred

Notes: 1. There is no concept of channels in the DTC. Transfer corresponding to the interrupt source is possible.

2. The number of transferred bytes becomes the maximum when a transfer is performed 256 times with a data size of 1 longword (32 bits) in repeat transfer mode or when the block size is 256 data pieces × 32 bits in block transfer mode.

Table 1.8 General Differences Between the DMACII in the R32C/118 and the DMAC in the RX660

Item	R32C (R32C/118)	RX (RX660)
	DMACII	DMAC
Transfer space	0000 0000h to 01FF FFFFh, FE00 0000h to FFFF FFFFh	0000 0000h to 0FFF FFFFh, F000 0000h to FFFF FFFFh
Maximum number of transferred bytes	1 MB [*1]	256 MB [*2]
Request source	Interrupt requests from all peripheral functions for which the ILVL2 to ILVL0 bits of the interrupt control register are set to “111b” (level 7)	112 sources • INT pin (16 sources): Choose Low, falling edge, rising edge, or both edges • Timers (47 sources) • Communications (39 sources) • A/D conversion and comparator (7 sources) • ELC (2 sources) • Software trigger
Transfer units	8 or 16 bits	8, 16, or 32 bits
Transfer direction	2 conditions selectable for the transfer source and transfer destination • Fixed • Increment	4 conditions selectable for the transfer source and transfer destination • Non-change • Increment-by-offset [*3] • Increment • Decrement
Transfer modes	• Single transfer mode • Burst transfer mode • Multi-transfer mode	• Normal transfer mode • Repeat transfer mode • Block transfer mode
Interrupts	An interrupt is generated when the value of the transfer counter becomes 0000h.	• Normal transfer mode: Generated when the specified number of transfers end; Repeat transfer mode: Generated when the specified number of times a transfer is repeated; Block transfer mode: Generated when the specified number of blocks are transferred (transfer end interrupt). • Generated when data of the repeat size is transferred or when the extended repeat area overflows (transfer escape end interrupt).

Notes: 1. The number of transferred bytes becomes the maximum when 16-bit data is transferred 65,536 times in burst transfer mode.

2. The number of transferred bytes becomes the maximum when data is transferred in block transfer mode under the following conditions: 1,024 data pieces × 65,536 blocks × 32 bits.

3. Increment-by-offset can be set for DMAC0 only.

Table 1.9 General Differences Between the DMACII in the R32C/118 and the DTC in the RX660

Item	**R32C (R32C/118)**	**RX (RX660)**
	DMACII	**DTC**
Transfer space	0000 0000h to 01FF FFFFh, FE00 0000h to FFFF FFFFh	Short-address mode: 0000 0000h to 007F FFFFh, FF80 0000h to FFFF FFFFh Full-address mode: 0000 0000h to FFFF FFFFh
Maximum number of transferred bytes	1 MB [*1]	1,024 bytes [*2]
Request source	Interrupt requests from all peripheral functions for which the ILVL2 to ILVL0 bits of the interrupt control register are set to “111b” (level 7)	126 sources • INT pin (18 sources): Choose Low, falling edge, rising edge, or both edges • Timers (55 sources) • Communications (43 sources) • A/D conversion and comparator (7 sources) • ELC (2 sources) • Software trigger
Transfer units	8 or 16 bits	8, 16, or 32 bits
Transfer direction	2 conditions selectable for the transfer source and transfer destination • Fixed addresses • Addresses in the forward direction	3 conditions selectable for the transfer source and transfer destination • Non-change • Increment • Decrement
Transfer modes	• Single transfer mode • Burst transfer mode • Multi-transfer mode	• Normal transfer mode • Repeat transfer mode • Block transfer mode
Interrupts	An interrupt is generated when the value of the transfer counter becomes 0.	• Generated when the DTC starts • Generated after a single data transfer ends • Generated after the specified number of data units are transferred

Notes: 1. The number of transferred bytes becomes the maximum when 16-bit data is transferred 65,536 times in burst transfer mode.

2. The number of transferred bytes becomes the maximum when a transfer is performed 256 times with a data size of 1 longword (32 bits) in repeat transfer mode or when the block size is 256 data pieces × 32 bits in block transfer mode.

Table 1.10 Difference in the Number of Cycles Between the DMAC/DMACII (R32C/118) and the DMAC/DTC (RX660)

Item	R32C (R32C/118)		RX (RX660)	
	DMAC	DMACII	DMAC	DTC
Number of cycles [*1]	7 cycles (includes 1 dummy cycle)	32 cycles	6 cycles	18 cycles

Note: 1. This table assumes the following: 8-bit data transfer, transfer source is an I/O register (fixed), the transfer destination is the RAM (increment), the DTC is in full-address mode, and DTC vector table is the ROM, and ICLK = PCLK multiplied by 2.

2. Peripheral Functions Used

Table 2.1 shows Peripheral Functions and Modes Used in the Example for Operating the DMAC and DTC.

Table 2.1 Peripheral Functions and Modes Used in the Example for Operating the DMAC and DTC

No.	Operation	M32C/R32C		RX	
		Peripheral Function	Mode	Peripheral Function	Mode
1	Data transmission	DMAC	Single transfer mode	DMAC	Normal transfer mode
2	Repeated data transmission		Repeat transfer mode		Repeat transfer mode
3	Data transmission		Single transfer mode	DTC	Normal transfer mode
4	Repeated data transmission		Repeat transfer mode		Normal transfer mode

Also, asynchronous serial communications in the serial communications interface are used for the DMAC/DTC request source. Table 2.2 shows Transfer Conditions for the DMAC/DTC (Asynchronous Serial Communications).

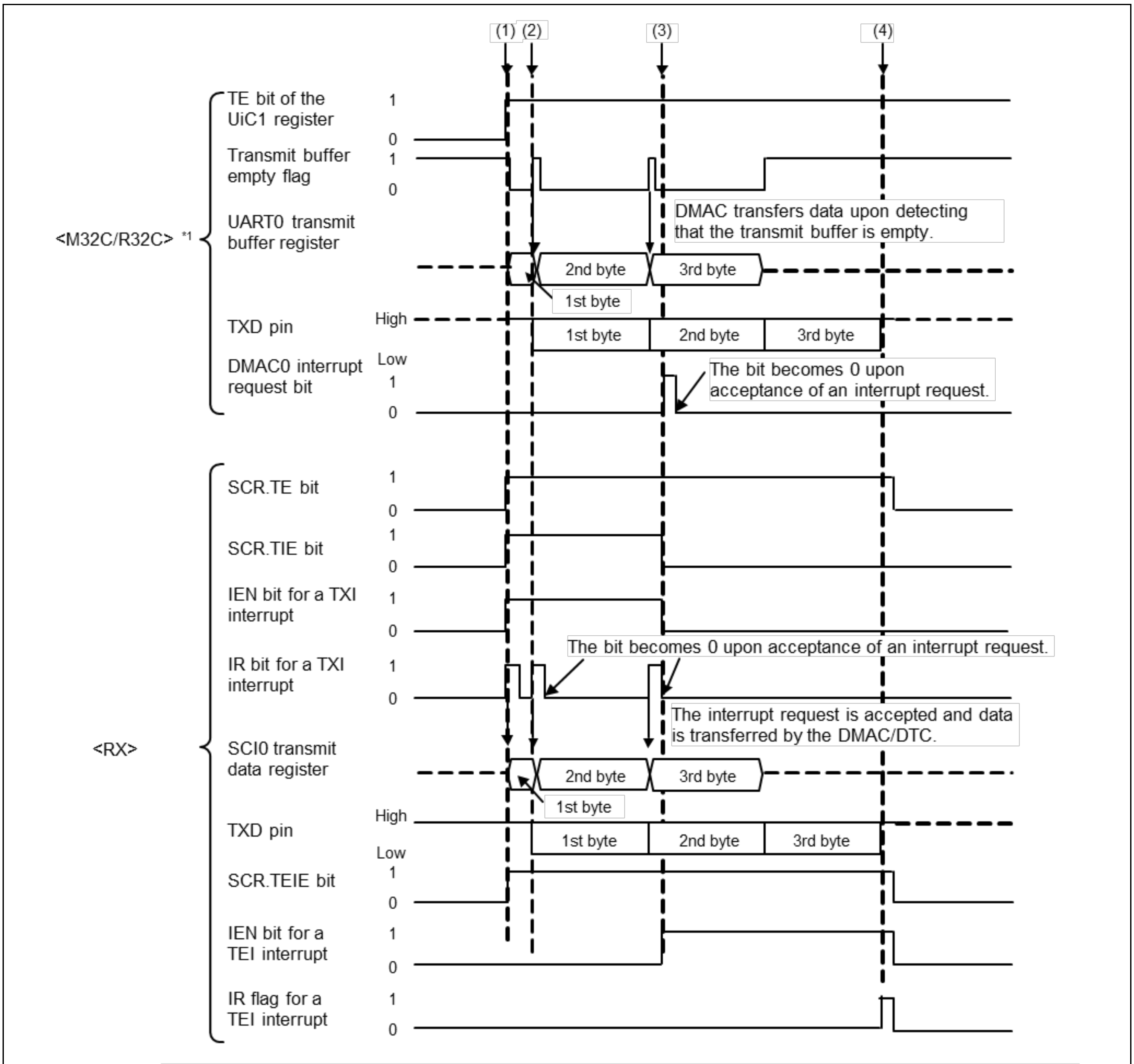
Table 2.2 Transfer Conditions for the DMAC/DTC (Asynchronous Serial Communications)

Item	Conditions for Transmission and Reception
Request source	Serial communications interface (SCI) Transmit data empty of the asynchronous serial communications
Channel used for asynchronous serial communications	RX Family: SCI0 M32C/R32C Series: UART0

3. Transfer Timing

Figure 3.1 shows Differences in Timing Between the M32C/R32C and the RX. Table 3.1 shows Comparison of Operation and Processing at Various Timings Between the M32C/R32C and the RX.

Both Figure 3.1 and Table 3.1 assume 3 bytes of serial data are transferred using the DMAC or DTC.


(1)
(2)
(3)
(4)
<M32C/R32C> *1
TE bit of the UiC1 register
Transmit buffer empty flag
UART0 transmit buffer register
TXD pin
DMAC0 interrupt request bit
1
0
High
Low
2nd byte
3rd byte
1st byte
DMAC transfers data upon detecting that the transmit buffer is empty.
The bit becomes 0 upon acceptance of an interrupt request.
<RX>
SCR.TE bit
SCR.TIE bit
IEN bit for a TXI interrupt
IR bit for a TXI interrupt
SCI0 transmit data register
TXD pin
SCR.TEIE bit
IEN bit for a TEI interrupt
IR flag for a TEI interrupt
The bit becomes 0 upon acceptance of an interrupt request.
The interrupt request is accepted and data is transferred by the DMAC/DTC.

Note 1. For the M32C/R32C, only either the time at which the transmit buffer becomes empty or the time at which transmission ends can be used as a trigger to generate a transmit interrupt.
The above timing chart is an example of operation when the time at which the transmit buffer becomes empty is selected as a trigger.

Figure 3.1 Differences in Timing Between the M32C/R32C and the RX

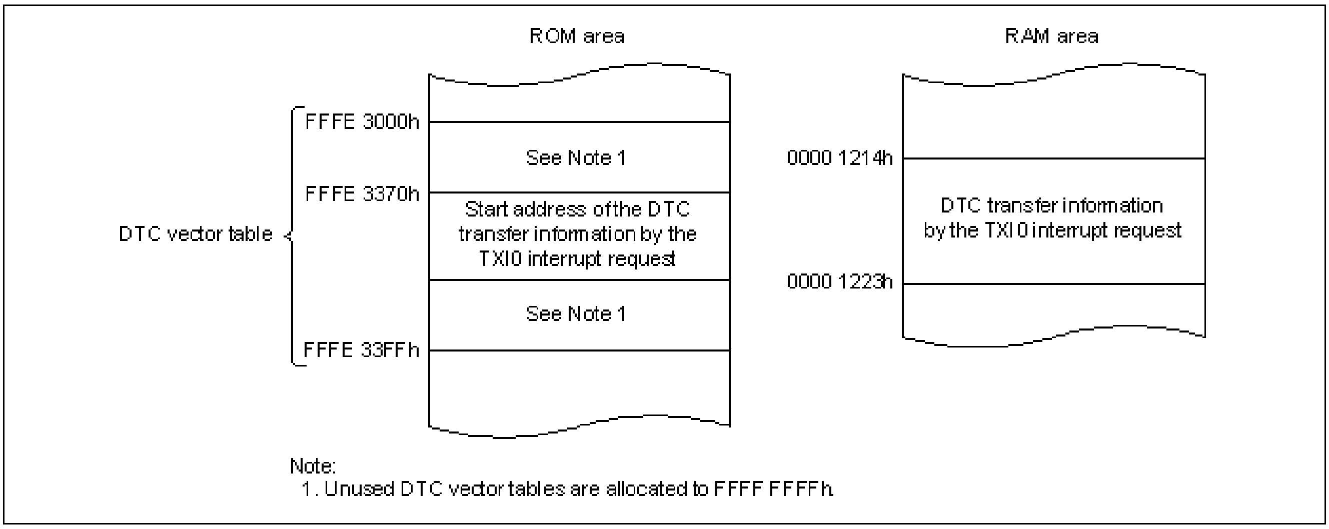
Table 3.1 Comparison of Operation and Processing at Various Timings Between the M32C/R32C and the RX

Timing	M32C/R32C (M32C/78, R32C/118)	RX (RX660)
(1) Transmission starts	Transmission is enabled. However, even if transmission is enabled, a transmit interrupt is not generated, so an instruction like the MOV instruction of the CPU must be used in the main processing in order to write the first byte of data to the transmit buffer.	When transmission is enabled, the transmit interrupt (TXI interrupt) is generated. The interrupt request triggers the DMAC/DTC to transfer the first byte of data to the transmit buffer.
(2) Transmit data is transferred to the transmit shift register	When a transmit interrupt request is generated, the DMAC transfers the second byte of data to the transmit buffer.	When a transmit interrupt request is generated, the DMAC/DTC transfers the second byte of data to the transmit buffer.
(3) DMA transfer interrupt (DMAC) or TXI interrupt (DTC) generated when the last data is written	DMAC interrupt request is generated.	When the last data is written to the transmit buffer, the DMA transfer interrupt (DMAC) or TXI interrupt (DTC) is generated. This interrupt enables the transmit end interrupt (TEI interrupt) and disables the transmit interrupt.
(4) After the last data is output	—	The transmit end interrupt is generated. In the interrupt handling, the transmit end interrupt and transmission are disabled. When transmission is disabled, the TXD pin becomes high-impedance.

4. Settings to Use the DTC

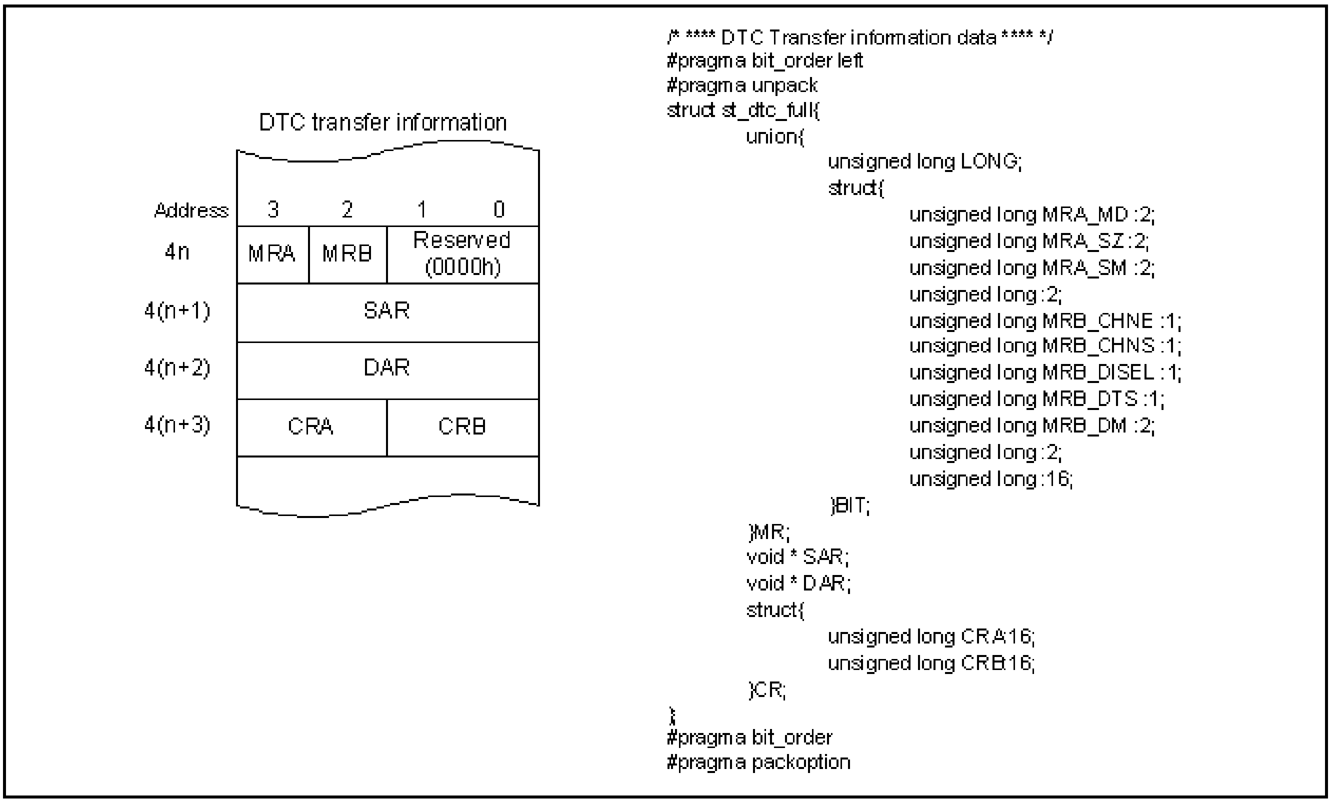
When using the DTC, the DTC vector table and DTC transfer information must be prepared. Figure 4.1 shows Memory Map When the DTC Vector Table is Allocated to the ROM Area. The start address of the DTC transfer information is stored in the DTC vector table, and the DTC vector table is allocated to the ROM or RAM. The base address of the vector table is allocated so the lower 12 bits are 0. The start address of the DTC transfer information with vector number *n* for an interrupt to be used as an activation trigger is stored in the +4*n*-th address from the base address.

The DTC transfer information is allocated to the RAM. Figure 4.2 shows Memory Map and Structure of the DTC Transfer Information.


ROM area
RAM area
DTC vector table
FFFE 3000h
See Note 1
FFFE 3370h
Start address of the DTC transfer information by the TXI0 interrupt request
See Note 1
FFFE 33FFh
0000 1214h
DTC transfer information by the TXI0 interrupt request
0000 1223h

Note:
1. Unused DTC vector tables are allocated to FFFF FFFFh.

Figure 4.1 Memory Map When the DTC Vector Table is Allocated to the ROM Area



```
/* **** DTC Transfer information data **** */
#pragma bit_order left
#pragma unpack
struct st_dtc_full{
          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 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 :16;
                    }BIT;
          }MR;
          void * SAR;
          void * DAR;
          struct{
                    unsigned long CRA:16;
                    unsigned long CRB:16;
          }CR;
}
#pragma bit_order
#pragma packoption
```

Figure 4.2 Memory Map and Structure of the DTC Transfer Information

5. Appendix

5.1 Points on Migration from the M32C/R32C to the RX

This section explains points on migration from the M32C/R32C to the RX.

5.1.1 Interrupts

For the RX Family, when an interrupt request is received while all of the following conditions are met, the interrupt occurs.

- The I flag (PSW.I bit) is 1.
- Registers IER and IPR in the ICU are set to enable interrupts.
- The interrupt request is enabled by the interrupt request enable bits for peripheral functions.

Table 5.1 shows Comparison of Conditions for Interrupt Generation Between the M32C/R32C and the RX.

Table 5.1 Comparison of Conditions for Interrupt Generation Between the M32C/R32C and the RX

Item	M32C/R32C	RX
I flag	When the I flag is set to 1 (enabled), the maskable interrupt request can be accepted.	
Interrupt request flag	When an interrupt request is generated by a peripheral function, the interrupt request flag becomes 1 (interrupt requested).	
Interrupt priority level	Selected by setting bits ILVL2 to ILVL0.	Selected by setting the IPR[3:0] bits.
Interrupt request enable	—	Specified by setting the IER register.
Interrupt enable for peripheral functions	—	Interrupts can be enabled or disabled in each peripheral function.

For more information, refer to sections Interrupt Controller (ICU), CPU, and sections for other peripheral functions used in the user's manual for hardware.

5.1.2 Module Stop Function

The RX Family has the ability to stop each peripheral module individually.

By transitioning unused peripheral modules to the module stop state, power consumption can be reduced.

After a reset is released, all modules (with a few exceptions) are in the module stop state.

Registers for modules in the module stop state cannot be written to or read.

For more information, refer to the Low Power Consumption section in the user's manual for hardware.

5.1.3 I/O Ports

In the RX Family, the MPC must be configured in order to assign I/O signals from peripheral functions to pins.

Before performing pin I/O control in the RX Family, perform the following two operations:

- In the MPC.PFS register, select the peripheral functions that are assigned to the appropriate pins.
- In the PMR register for I/O ports, select the function for the pin to be used as a general I/O port or I/O port for a peripheral function.

The M32C/R32C provides a function select register that allows the user to select whether to use the pin as an I/O port or for the output port for a specific peripheral function.

Before performing pin I/O control in the M32C, perform the following two operations:

- Function select registers B to E: Use these registers to select the peripheral function that can be assigned to the target pin.
- Function select register A: Use this register to select whether the target pin is to be used as a general I/O port or for the selected peripheral function.

Before performing pin I/O control in the R32C, perform the following operation:

- Function select register: Use this register to select the peripheral function that can be assigned and to select whether the target pin is to be used as a general I/O port or for the selected peripheral function.

Table 5.2 shows Comparison of I/O Settings for Peripheral Function Pins Between the M32C and the RX.
Table 5.3 shows Comparison of I/O Settings for Peripheral Function Pins Between the R32C and the RX.

Table 5.2 Comparison of I/O Settings for Peripheral Function Pins Between the M32C and the RX

Function	M32C (in the case of the M32C/87)	RX (in the case of the RX660)
Select the pin function	With the function select registers B to E, I/O ports for peripheral functions can be assigned by selecting from multiple pins.	With the PFS register, I/O ports for peripheral functions can be assigned by selecting from multiple pins.
Switch between general I/O port and peripheral function	With the function select register A, the corresponding pin function can be selected as a general I/O port or a peripheral function.	With the PMR register, the corresponding pin function can be selected as a general I/O port or a peripheral function.

Table 5.3 Comparison of I/O Settings for Peripheral Function Pins Between the R32C and the RX

Function	R32C (in the case of the R32C/118)	RX (in the case of the RX660)
Select the pin function	With the function select register, the corresponding pin function can be selected as a general I/O port or a peripheral function. Output ports for peripheral functions can be assigned by selecting from multiple pins.	With the PFS register, I/O ports for peripheral functions can be assigned by selecting from multiple pins.
Switch between general I/O port and peripheral function		With the PMR register, the corresponding pin function can be selected as a general I/O port or a peripheral function.

For details on the RX, refer to the chapters on the multi-function pin controller (MPC) and I/O ports in the user’s manual for hardware.

For details on the M32C, refer to the chapter on programmable I/O ports in the user’s manual for hardware.

For details on the R32C, refer to the chapter on I/O ports in the user’s manual for hardware.

5.2 I/O Register Macros

Macro definitions listed in Table 5.4 can be found in the RX I/O register definitions (iodefine.h).

The readability of programs can be achieved with these macro definitions.

Table 5.4 shows Macro Usage Examples.

Table 5.4 Macro Usage Examples

Macro	**Usage Example**
IR("module name", "bit name")	**IR(MTU0,TGIA0)** = 0 ; The IR bit corresponding to MTU0.TGIA0 is cleared to 0 (no interrupt request is generated).
DTCE("module name", "bit name")	**DTCE (MTU0, TGIA0)** = 1 ; The DTCE bit corresponding to MTU0.TGIA0 is set to 1 (DTC activation is enabled).
IEN("module name", "bit name")	**IEN(MTU0, TGIA0)** = 1 ; The IEN bit corresponding to MTU0.TGIA0 is set to 1 (interrupt enabled).
IPR("module name", "bit name")	**IPR(MTU0, TGIA0)** = 0x02 ; The IPR bit corresponding to MTU0.TGIA0 is set to 2 (interrupt priority level 2).
MSTP("module name")	**MSTP(MTU)** = 0 ; The MTU0 Module Stop bit is set to 0 (module stop state is canceled).
VECT("module name", "bit name")	#pragma interrupt (Excep_MTU0_TGIA0 (vect=**VECT(MTU0, TGIA0)**) The interrupt function is declared for the corresponding MTU0.TGIA0 register.

5.3 Intrinsic Functions

The RX Family has intrinsic functions for setting control registers and special instructions. When using intrinsic functions, include machine.h.

Table 5.5 shows Examples of Differences in the Settings of Control Registers and Descriptions of Special Instructions Between the M32C/R32C and the RX.

Table 5.5 Examples of Differences in the Settings of Control Registers and Descriptions of Special Instructions Between the M32C/R32C and the RX

Item	Description	
	M32C/R32C	RX
Set the I flag to 1	asm("fset i");	setpsw_i (); [*1]
Set the I flag to 0	asm("fclr i");	clrpsw_i (); [*1]
Expanded into the WAIT instruction	asm("wait");	wait(); [*1]
Expanded into the NOP instruction	asm("nop");	nop(); [*1]

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

6. Reference Documents

User’s Manual: Hardware

RX660 Group User’s Manual: Hardware (R01UH0937EJ)
M32C/87 Group (M32C/87, M32C/87A, M32C/87B) Hardware Manual (REJ09B0180)
R32C/118 Group User’s Manual: Hardware (R01UH0212EJ)
If you are using a product that does not belong to the above groups, refer to the applicable user's manual for hardware.
The latest versions can be downloaded from the Renesas Electronics website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

User’s Manual: Development Tools

RX Family CC-RX Compiler User’s Manual (R20UT3248EJ)
C Compiler Package for the M32C Series (M3T-NC308WA)
C Compiler Package for the R32C Series
The latest versions can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jan. 10, 24	—	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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.