RX Family and M32C/R32C Series

Guide for Migration from the M32C/R32C to the RX: Serial Communications

Abstract

This document describes migration from the serial interface of the M32C/R32C Series to the serial communication interface (SCI) of 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 serial communications.

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

Item	M32C/R32C Series	RX Family
Abbreviated name of the serial communication interface (SCI)	Serial interface	SCI
Name of the asynchronous serial communications mode	Clock asynchronous serial interface mode (UART mode)	Asynchronous mode
Name of the clock synchronous serial data communications mode	Clock synchronization serial interface mode	Clock synchronous mode
Name of the clock I/O pin for clock synchronous communications (SCK pin)	CLKi pin	SCKi pin
Name of the simple I^2C mode (Simple I^2C mode)	Special mode 1 (I^2C mode)	Simple I^2C mode
Name of the SDA pin	SDAi pin	SSDAi pin
Name of the SCL pin	SCLi pin	SSCLi pin
SCI operating clock (clock source)	Count source	Clock source
Peripheral function operating clock	Peripheral function clocks: f1, f8, f2n, fC32	Peripheral module clocks: PCLKA, PCLKB, PCLKD
Transmit buffer	UiTB register (transmit buffer)	TDR registers: TDRH, TDRL, TDRHL
Transmit shift register	Transmit shift register	TSR register
Receive buffer	UiRB register	RDR registers: RDRH, RDRL, RDRHL
Start condition	Start condition	Start condition
Stop condition	Stop condition	Stop condition
Restart condition	Restart condition	Restart condition
Interrupt after generating the start condition and stop condition	Start condition/stop condition detection interrupt*1	STI interrupt
Transmit interrupt	UARTi transmit interrupt (transmit buffer empty)	TXI interrupt
Transmission complete interrupt	UARTi transmit interrupt (transmission completed)	TEI interrupt
Receive interrupt	UARTi receive interrupt	RXI interrupt
Function to select I/O of peripheral functions for pins	Function select register, input function select register	MPC *2

Notes: 1. An interrupt request occurs when a start condition, restart condition, or stop condition is generated.
2. The MPC is not available in some groups.

Contents

1. General Differences in Serial Communications

1.1 General Differences in Asynchronous Serial Communications

Table 1.1 shows General Differences in Asynchronous Serial Communications Between the M32C/87 and the RX660. Table 1.2 shows General Differences in Asynchronous Serial Communications Between the R32C/118 and the RX660.

Table 1.1 General Differences in Asynchronous Serial Communications Between the M32C/87 and the RX660

Item	M32C (M32C/87)	RX (RX660)
Operation clock source	Selectable from f1, f8, f2n, or fEXT (external clock)	PCLKB
Data length	Selectable from 7 bits, 8 bits or 9bits	Selectable from 7 bits, 8 bits or 9bits
Parity bit	Selectable from even, odd, or no parity	Selectable from even, odd, or no parity
Stop bits	Selectable from 1 bit or 2 bits	Selectable from 1 bit or 2 bits
Data format	Selectable from LSB first or MSB first	Selectable from LSB first or MSB first
Hardware flow control	Available (selectable)	Available (selectable)
Data match detection	Not available	Available
Start bit detection	Falling edge	Low level or falling edge can be selected.
Receive data sampling timing adjustment	Not available	Available
Transmit signal change timing adjustment	Not available	Available
Interrupt sources	Transmit start interrupt Transmit complete interrupt Receive complete interrupt	Transmit data empty (TXI) interrupt Transmit end (TEI) interrupt Receive data full (RXI) interrupt Receive error (TRI) interrupt
Error detection	Overrun error Framing error Parity error	Overrun error Framing error Parity error
Double-speed mode	Not available	Available
Multi-processor function	Not available	Available
Noise cancellation	Not available	On-chip digital noise filter on the RXDn pin input route
Data logic switch	Available	Available
TXD, RXD I/O polarity switch	Available	Not available

Table 1.2 General Differences in Asynchronous Serial Communications Between the R32C/118 and the RX660

Item	R32C (R32C/118)	RX (RX660)
Operation clock source	Selectable from f1, f8, f2n, or fEXT (external clock)	PCLKB
Data length	Selectable from 7 bits, 8 bits or 9bits	Selectable from 7 bits, 8 bits or 9bits
Parity bit	Selectable from even, odd, or no parity	Selectable from even, odd, or no parity
Stop bits	Selectable from 1 bit or 2 bits	Selectable from 1 bit or 2 bits
Data format	Selectable from LSB first or MSB first	Selectable from LSB first or MSB first
Hardware flow control	Available (selectable)	Available (selectable)
Data match detection	Not available	Available
Start bit detection	Falling edge	Low level or falling edge can be selected.
Receive data sampling timing adjustment	Not available	Available
Transmit signal change timing adjustment	Not available	Available
Interrupt sources	Transmit start interrupt Transmit complete interrupt Receive complete interrupt	Transmit data empty (TXI) interrupt Transmit end (TEI) interrupt Receive data full (RXI) interrupt Receive error (TRI) interrupt
Error detection	Overrun error Framing error Parity error	Overrun error Framing error Parity error
Double-speed mode	Not available	Available
Multi-processor function	Not available	Available
Noise cancellation	Not available	On-chip digital noise filter on the RXDn pin input route
Data logic switch	Available	Available
TXD, RXD I/O polarity switch	Available	Not available

1.2 General Differences in Clock Synchronous Serial Data Communications

Table 1.3 shows General Differences in Clock Synchronous Serial Communications Between the M32C/87 and the RX660. Table 1.4 shows General Differences in Clock Synchronous Serial Communications Between the R32C/118 and the RX660.

Table 1.3 General Differences in Clock Synchronous Serial Communications Between the M32C/87 and the RX660

Item	M32C (M32C/87)	RX (RX660)
Operation clock source	Selectable from f1, f8, f2n, or fEXT (external clock)	PCLKB
Data length	8 bits	8 bits
Data format	Selectable from LSB first or MSB first	Selectable from LSB first or MSB first
Hardware flow control	Available (selectable)	Available (selectable)
Interrupt sources	Transmit start interrupt Transmit complete interrupt Receive complete interrupt	Transmit data empty (TXI) interrupt Transmit end (TEI) interrupt Receive data full (RXI) interrupt Receive error (ERI) interrupt
Error detection	Overrun error	Overrun error
Selectable clock polarity	Available (selectable)	Available (selectable)
Data logic switch	Available	Available
TXD, RXD I/O polarity switch	Available	Not available

Table 1.4 General Differences in Clock Synchronous Serial Communications Between the R32C/118 and the RX660

Item	R32C (R32C/118)	RX (RX660)
Operation clock source	Selectable from f1, f8, f2n, or fEXT (external clock)	PCLKB
Data length	8 bits	8 bits
Data format	Selectable from LSB first or MSB first	Selectable from LSB first or MSB first
Hardware flow control	Available (selectable)	Available (selectable)
Interrupt sources	Transmit start interrupt Transmit complete interrupt Receive complete interrupt	Transmit data empty (TXI) interrupt Transmit end (TEI) interrupt Receive data full (RXI) interrupt Receive error (ERI) interrupt
Error detection	Overrun error	Overrun error
Selectable clock polarity	Available (selectable)	Available (selectable)
Data logic switch	Available	Available
TXD, RXD I/O polarity switch	Available	Not available

1.3 Differences in Simple I²C Mode

Table 1.5 shows General Differences in Simple I²C Mode Between the M32C/87 and the RX660. Table 1.6 shows General Differences in Simple I²C Mode Between the R32C/118 and the RX660.

Table 1.5 General Differences in Simple I²C Mode Between the M32C/87 and the RX660

Item	M32C (M32C/87)	RX (RX660)
Operating mode	Master, slave	Master*1
Data length	8 bits	8 bits
Data format	MSB first fixed	MSB first fixed
Interrupt sources	Transmit, NACK interrupt Receive, ACK interrupt Start condition, stop condition detection interrupt	Transmit, NACK (TXI) interrupt Receive, ACK detection (RXI) interrupt Completion of generating a start, restart, or stop condition interrupt (STI interrupt)
Error detection	Overrun error Arbitration loss	Not available
Noise cancellation	Not available	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters. The noise elimination width can be adjusted.
Delay in SDA output	No digital delay or a delay of 2 to 8 cycles can be selected.	The delay length can be selected in the range from 0 to 31 cycles.
Clock phase setting	Whether a clock delay is available can be selected.	Clock delay available

Note: 1. Use the RIIC when performing slave operation with the RX Family.

Table 1.6 General Differences in Simple I²C Mode Between the R32C/118 and the RX660

Item	M32C (M32C/87)	RX (RX660)
Operating mode	Master, slave	Master*1
Data length	8 bits	8 bits
Data format	MSB first fixed	MSB first fixed
Interrupt sources	Transmit, NACK interrupt Receive, ACK interrupt Start condition, stop condition detection interrupt	Transmit, NACK (TXI) interrupt Receive, ACK detection (RXI) interrupt Completion of generating a start, restart, or stop condition interrupt (STI interrupt)
Error detection	Overrun error Arbitration loss	Not available
Noise cancellation	Not available	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters. The noise elimination width can be adjusted.
Delay in SDA output	No digital delay or a delay of 2 to 8 cycles can be selected.	The delay length can be selected in the range from 0 to 31 cycles.
Clock phase setting	Whether a clock delay is available can be selected.	Clock delay available

Note: 1. Use the RIIC when performing slave operation with the RX Family.

2. Peripheral Functions Used

Table 2.1 shows Peripheral Functions and Modes Used When Performing Serial Communications.

Table 2.1 Peripheral Functions and Modes Used When Performing Serial Communications

No.	Operating Example	M32C/R32C		RX	
		Peripheral Function	Mode	Peripheral Function	Mode
1	Asynchronous serial communications (transmit/receive operations)	Serial interface	UART mode	SCI	Asynchronous mode
2	Clock synchronous serial data communications (master transmit and receive operations)		Clock synchronous mode		Clock synchronous mode
3	Clock synchronous serial data communications (master transmit operation)				
4	Clock synchronous serial data communications (slave receive operation)				
5	Master transmit operation in simple I^2C mode		Special mode 1 (I^2C mode)		Simple I^2C mode
6	Master receive operation in simple I^2C mode				

3. Differences in Asynchronous Serial Communications

This section explains the functional differences in asynchronous serial communications between the RX and the M32C/R32C under the example conditions shown in Table 3.1 Conditions for Asynchronous Serial Communications.

Table 3.1 Conditions for Asynchronous Serial Communications

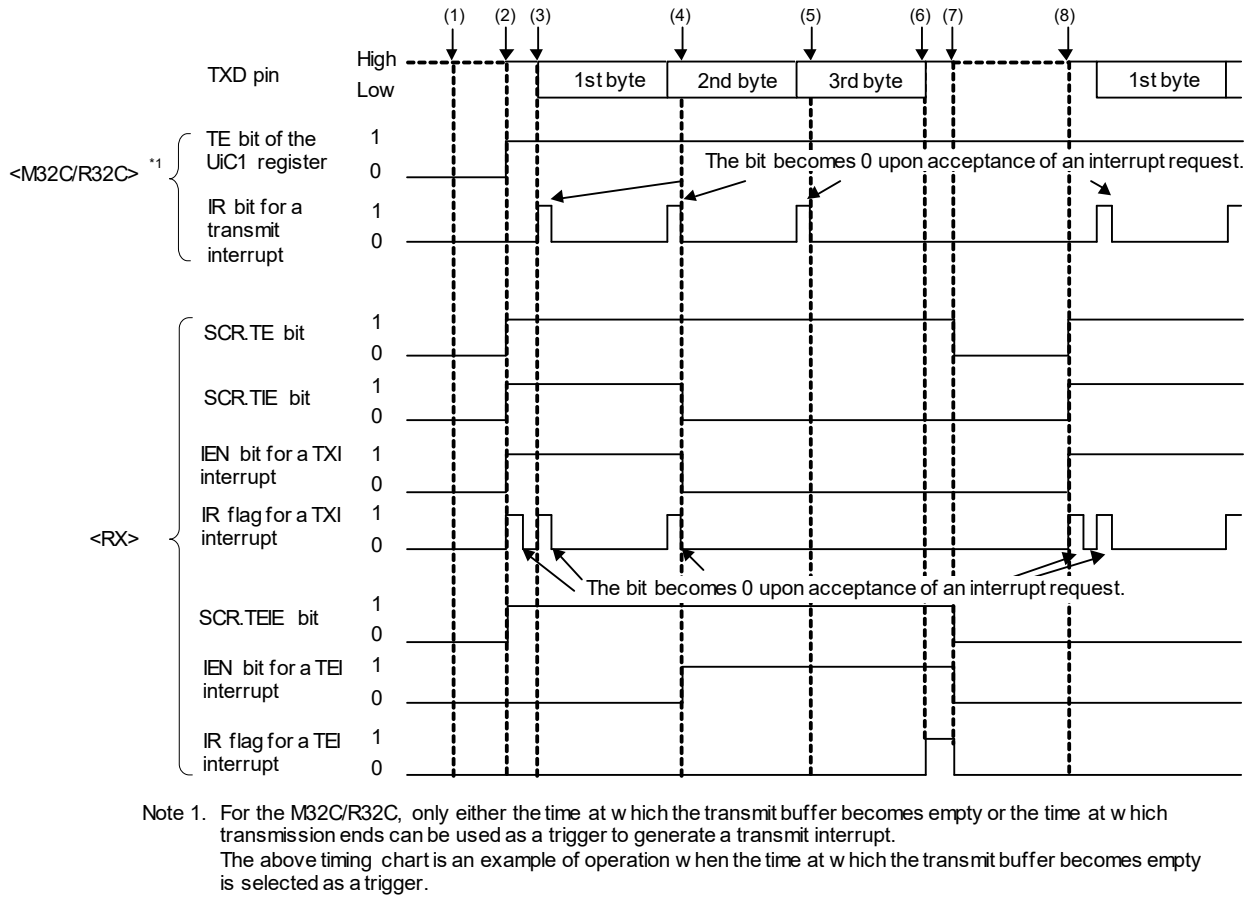
Item	Conditions for Transmission and Reception
Peripheral function operating clock	16 MHz
Transfer rate	9600bps
Data length	8 bits
Stop bits	1 stop bit
Parity	None
Data format	LSB first
Hardware flow control	Not available
Channels used	RX Family: SCI0 M32C/R32C Series: UART0
Pin processing	Pull-up resistors are connected to the TXD and RXD pins.*1

Note: 1. In the RX Family, when the SCR.TE bit is 0 (serial transmission is disabled), the TXD pin is in high-impedance. When a pull-up resistor is not connected, while serial transmission is disabled, switch the pin function to the output state of general I/O ports.

3.1 Transmit/Receive Timing

3.1.1 Differences in Transmitting

Figure 3.1 shows Differences in Timing Between the RX and the M32C/R32C (When Transmitting 3 Bytes). Table 3.2 shows Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (When Transmitting 3 Bytes).



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 RX and the M32C/R32C (When Transmitting 3 Bytes)

Table 3.2 Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (When Transmitting 3 Bytes)

Timing	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
(1) Before transmission starts	The pin status is determined when serial interface mode is selected.	The TXD pin is in high-impedance until the SCR.TE bit is set to 1 (serial transmission is enabled).
(2) When transmission starts	The TE bit is set to 1 (transmission is enabled). The transmit interrupt does not occur even if the TE bit is set to 1. The first byte of data is written in the main processing, etc.	The SCR.TE bit is set to 1, the TIE bit is set to 1 (TXI interrupt request is enabled), the TEIE bit is set to 1 (TEI interrupt request is enabled), and the IEN bit for the TXI interrupt is set to 1 (TXI interrupt is enabled). When the SCR.TE bit is set to 1, the IR flag for the transmit interrupt (TXI interrupt) becomes 1, and the transmit interrupt is generated. The first byte of transmit data is written in the transmit interrupt handling.
(3) When transmit data is transferred to the transmit shift register	The IR flag (IR bit) for the transmit interrupt becomes 1, and the transmit interrupt is generated. The second byte of data is written in the transmit interrupt handling.	
(4) Transmit interrupt when writing the last data	—	The IEN bit for the TEI interrupt is set to 1 (TEI interrupt enabled), the SCR.TIE bit is set to 0 (a TXI interrupt request is disabled), and the IEN bit for the TXI interrupt is set to 0 (TXI interrupt disabled).
(5) Transmit interrupt after writing the last data	Interrupt handling is completed without transmit data being written.	— (No transmit interrupt is generated.)
(6) After outputting the last data	—	The transmit end interrupt is generated.
(7) When transmission is complete		In the transmit end interrupt processing, the SCR.TE bit is set to 0 (serial transmission is disabled), the TEIE bit is set to 0 (a TEI interrupt request is disabled), and the IEN bit for the TEI interrupt is set to 0 (TEI interrupt disabled) to disable transmission. When transmission is disabled, the IR flag for the transmit end interrupt becomes 0, and the TXD pin becomes high-impedance.
(8) When transmission restarts	The next data is written in the main processing, etc.	The same processing as in “(2) When transmission starts” occurs.

3.1.2 Differences in Reception

Figure 3.2 shows Differences in Timing Between the RX and the M32C/R32C (During Reception). Table 3.3 shows Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (During Reception).

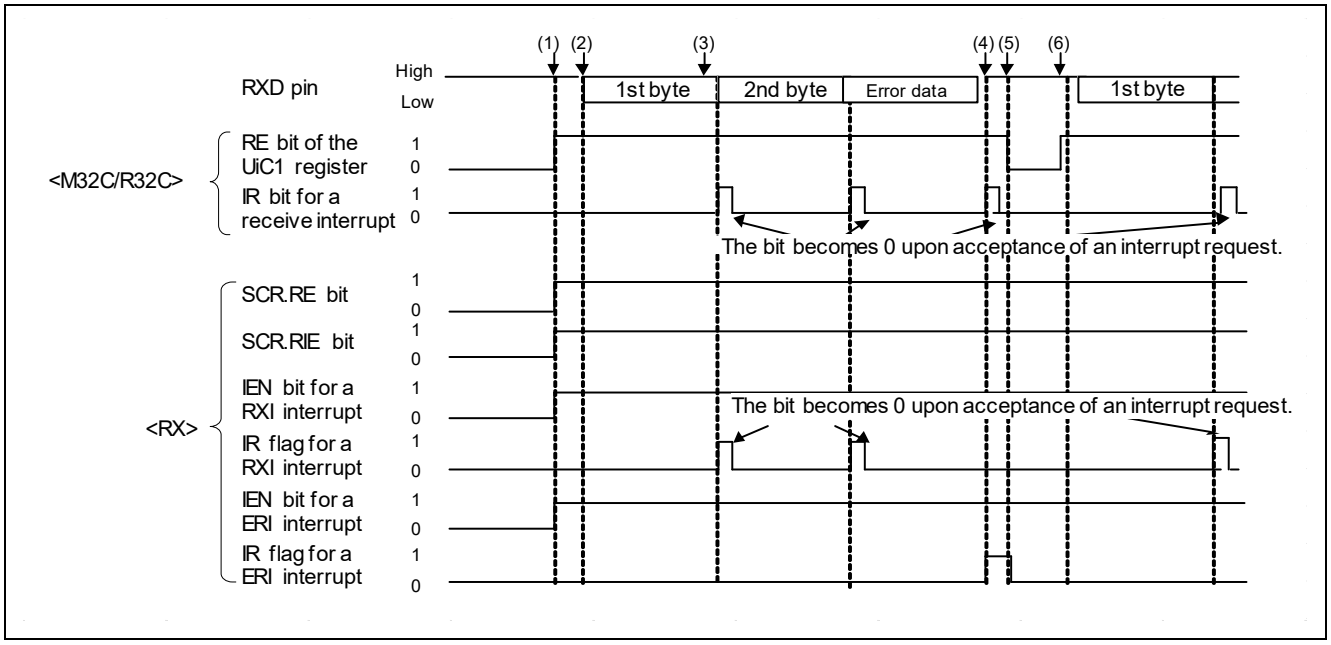


Figure 3.2 Differences in Timing Between the RX and the M32C/R32C (During Reception)

Table 3.3 Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (During Reception)

Timing	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
(1) When reception is enabled	The RE bit is set to 1 (reception is enabled) to enable reception.	The SCR.RE bit is set to 1 (serial reception is enabled), the RIE bit is set to 1 (RXI interrupt request is enabled), the IEN bit for the RXI interrupt is set to 1 (RXI interrupt request enabled), and the IEN bit for the ERI interrupt is set to 1 (ERI interrupt request enabled) to enable reception.
(2) When reception starts	Receive operation starts when the start bit is input to the RXD pin.	
(3) When transmission is complete	When 1 byte of data is received, receive data is transferred to the receive buffer, the IR flag (IR bit) for the receive interrupt (RXI interrupt) becomes 1, and a receive interrupt is generated. The value is read from the receive buffer in the receive interrupt handling.	
(4) When a receive error occurs	A receive interrupt occurs. In the receive interrupt handling, the error flag for the receive buffer register is read to check whether a receive error occurred.	The ERI interrupt is generated. Receive error processing is performed in the ERI interrupt handling.
(5) Clear the receive error flags	The RE bit is set to 0 (reception disabled), and bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled).	After reading the error flags in the SSR register, 0 is written to clear the error flags. After all error flags have been cleared, the IR flag for the ERI interrupt becomes 0, and reception is enabled.
(6) When reception is restarted	When bits SMD2 to SMD0 in the UiMR register are set to 101b (UART mode character length is 8 bits), and the RE bit is set to 1, reception is enabled.	

3.2 Calculating the Bit Rate

The bit rate calculation methods of the RX and the M32C/R32C are different. Table 3.4 shows Differences in Calculating the Bit Rate.

Table 3.4 Differences in Calculating the Bit Rate

Item	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
Calculating the bit rate using the internal clock	Clock source / 16 (*m* + 1) Clock source: f1, f8, or f2n *m*: Value set in the UiBRG register	In the case where BGDM = 0 and ABCS = 0: Clock source / 32 (*N* + 1) *1 In the case where BGDM = 1 and ABCS = 0, or where BGDM = 0 and ABCS = 1: Clock source / 16 (*N* + 1) *1 In the case where BGDM = 1 and ABCS = 1: Clock source / 8 (*N* + 1) *1 In the case where ABCSE = 1: Clock source / 6 (*N* + 1) *1 Clock source: PCLK, PCLK/4, PCLK/16, or PCLK/64 *N*: Value set in the BRR register
Calculating the bit rate using the external clock	fEXT/16 (*m* + 1) fEXT: Input from the CLKi pin *m*: Value set in the UiBRG register	fEXT/16 (when ABCS = 0) fEXT/8 (when ABCS = 0) fEXT: Input from the CLKi pin
Calculating the bit rate when the reference clock is generated by the TMR	—	Only when using SCI5, SCI6, and SCI12: The clock can be input from the TMR. (For details, refer to the user's manual.)

Note: 1. Based on "Relationship between N Setting in the BRR Register and Bit Rate B" in the User's Manual: Hardware (in the case where BGDM = 0 and ABCS = 0):

$B = PCLK / (64 \times 2^{2n-1} \times (N + 1))$
$= PCLK / (32 \times 2^{2n} \times (N + 1))$
$= (PCLK / 2^{2n}) / (32 \times (N + 1))$
$=$ Clock source $/ (32 \times (N + 1))$

4. Differences in Clock Synchronous Serial Data Communications

This section explains the functional differences in clock synchronous serial communications between the RX and the M32C/R32C under the example conditions shown in Table 4.1 Conditions for Clock Synchronous Serial Communications.

Table 4.1 Conditions for Clock Synchronous Serial Communications

Item	Conditions for Transmission and Reception
Peripheral function operating clock	16 MHz
Transfer rate	100 kbps
Data format	LSB first
Hardware flow control	Not available
Channels used	RX Family: SCI0 M32C/R32C Series: UART0

4.1 Differences in Master Transmission and Reception

This section describes the differences in master transmission and reception for clock synchronous operation.

4.1.1 Differences in Timings During Master Transmission and Reception

Figure 4.1 shows Differences in Timing Between the RX and the M32C/R32C (When Transmitting 3 Bytes). Table 4.2 shows Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (When Transmitting and Receiving 3 Bytes).

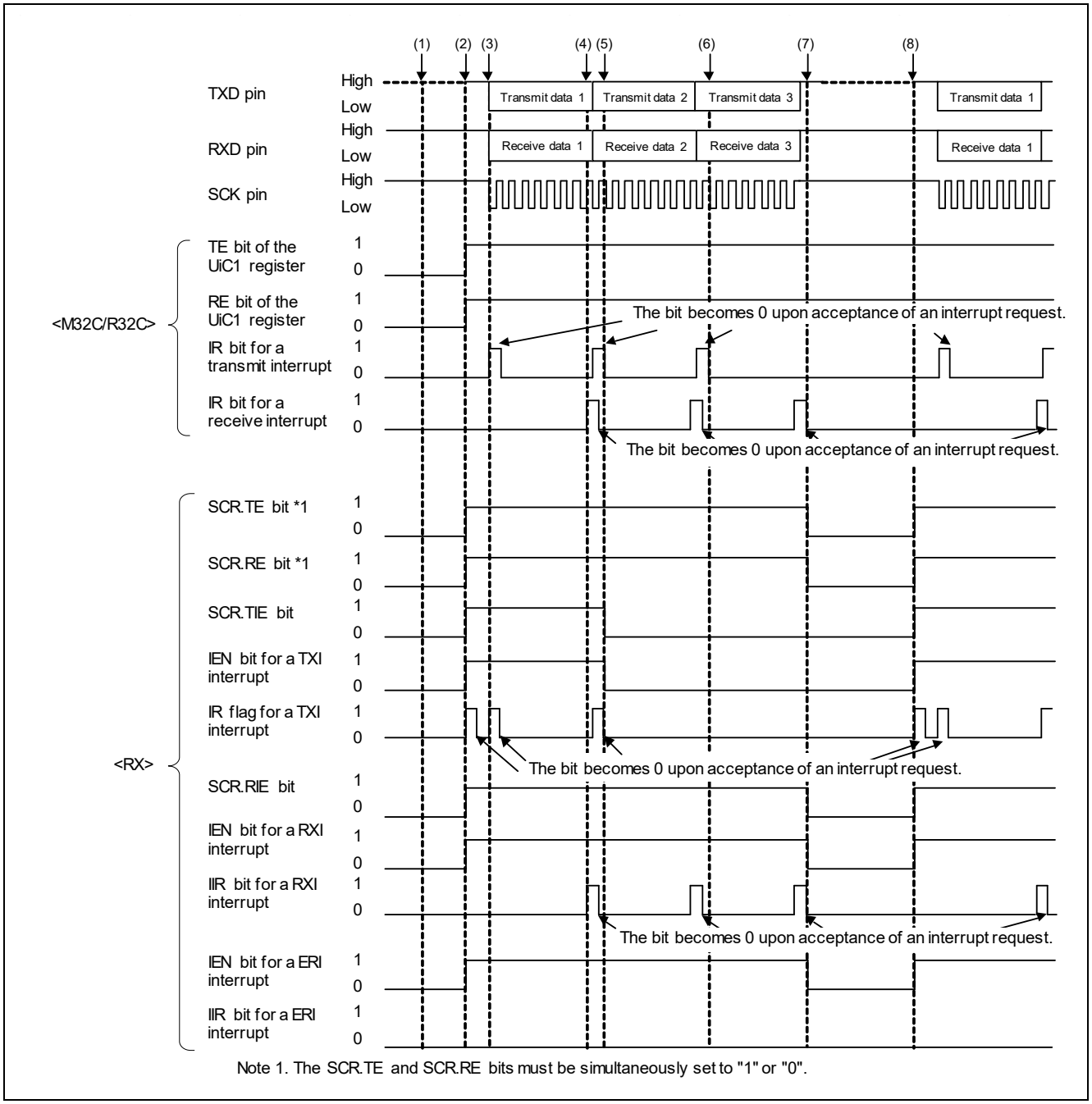


Figure 4.1 Differences in Timings During Transmission and Reception Between the RX and the M32C/R32C (When Transmitting and Receiving 3 Bytes)

Table 4.2 Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (When Transmitting and Receiving 3 Bytes)

Timing	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
(1) Before transmission starts	The pin status is determined when serial I/O mode is selected.	The TXD pin is in high-impedance until the SCR.TE bit is set to 1 (serial transmission is enabled).
(2) When transmission or reception starts	The TE bit is set to 1 (transmission is enabled) and the RE bit is set to 1 (reception is enabled). The transmit interrupt does not occur even if the TE bit is set to 1. The first byte of data is written in the main processing, etc.	To enable transmission and reception, set the SCR.TE bit to 1, and set the SCR.RE bit to 1. To enable interrupts, set the SCR.TIE bit to 1, set the SCR.RIE bit to 1, set the IEN bit for the TXI interrupt to 1, set the IEN bit for the RXI interrupt to 1, and set the IEN bit for the ERI interrupt to 1. When the SCR.TE bit is set to 1, the IR flag for the transmit interrupt (TXI interrupt) becomes 1. Write the first byte of transmit data in the transmit interrupt handling.
(3) When transmit data is transferred to the transmit shift register	The IR flag (IR bit) for the transmit interrupt becomes 1, and the transmit interrupt is generated. The second byte of data is written in the transmit interrupt handling.	
(4) When reception is complete	After receiving 1 byte of data, the data received is stored in the receive buffer, and the IR bit for the receive interrupt (RXI interrupt) becomes 1. The value stored in the receive buffer is read in the receive interrupt handling.	
(5) Transmit interrupt when writing the last data	—	After writing the third byte of transmit data, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled), and set the IEN bit for the TXI interrupt to 0 (TXI interrupt disabled).
(6) Transmit interrupt after writing the last data	Interrupt handling is completed without transmit data being written.	— (No transmit interrupt is generated.)
(7) Receive end interrupt of the last data	The same processing as in “(4) When reception is complete” occurs.	In the receive interrupt handling, after reading the receive data, set the SCR.TE bit to 0 (serial transmission is disabled), and the RE bit to 0 (serial reception is disabled) simultaneously. In addition, set the RIE bit to 0 (RXI interrupt request is disabled), set the IEN bit for the RXI interrupt to 0 (RXI interrupt request is disabled), and set the IEN bit for the ERI interrupt to 0 (ERI interrupt request is disabled). When transmission is disabled, the TXD pin becomes high-impedance.
(8) When transmission restarts	The next data is written in the main processing, etc.	The same processing as in “(2) When transmission or reception starts” occurs.

For differences in timings when a receive error occurs, refer to section 4.3.1 Differences in Timing During Slave Reception.

4.2 Differences in Master Transmission

This section describes the differences in master transmission for clock synchronous operation.

4.2.1 Differences in Timing During Master Transmission

Figure 4.2 shows Differences in Timings During Transmission Between the RX and the M32C/R32C (When Transmitting 3 Bytes). Table 4.3 shows Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (When Transmitting 3 Bytes).

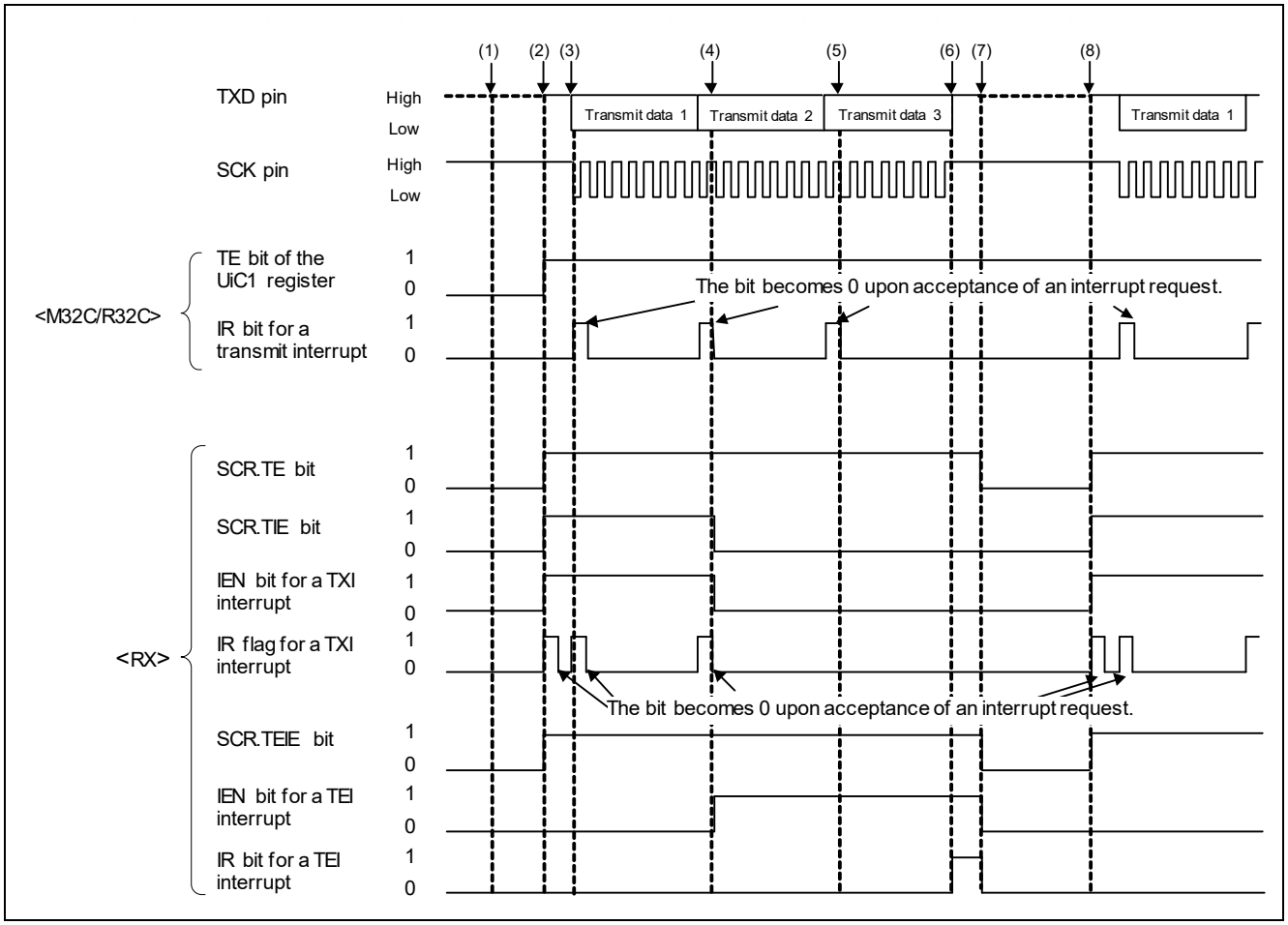


Figure 4.2 Differences in Timings During Transmission Between the RX and the M32C/R32C (When Transmitting 3 Bytes)

Table 4.3 Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (When Transmitting 3 Bytes)

Timing	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
(1) Before transmission starts	The pin status is determined when serial I/O mode is selected.	The TXD pin is in high-impedance until the SCR.TE bit is set to 1 (serial transmission is enabled).
(2) When transmission starts	The TE bit is set to 1 (transmission is enabled). The transmit interrupt does not occur even if the TE bit is set to 1. The first byte of data is written in the main processing, etc.	The SCR.TE bit is set to 1, the TIE bit is set to 1 (TXI interrupt request is enabled), the TEIE bit is set to 1 (TEI interrupt request is enabled), and the IEN bit for the TXI interrupt is set to 1 (TXI interrupt is enabled). When the SCR.TE bit is set to 1, the IR flag for the transmit interrupt (TXI interrupt) becomes 1. Write the first byte of transmit data in the transmit interrupt handling.
(3) When transmit data is transferred to the transmit shift register	The IR flag (IR bit) for the transmit interrupt becomes 1, and the transmit interrupt is generated. The second byte of data is written in the transmit interrupt handling.	
(4) Transmit interrupt when writing the last data	—	Set the IEN bit for the TEI interrupt to 1 (TEI interrupt is enabled), set the SCI.TIE bit to 0 (TXI interrupt request is disabled), and set the IEN bit for the TXI interrupt to 0 (TXI interrupt disabled).
(5) Transmit interrupt after writing the last data	Interrupt handling is completed without transmit data being written.	— (No transmit interrupt is generated.)
(6) When transmission is complete	—	A transmit end interrupt is generated.
(7) Transmission end interrupt handling		In the transmit end interrupt handling, set the SCR.TE bit to 0 (serial transmission is disabled), set the TEIE bit to 0 (a TEI interrupt request is disabled), and set the IEN bit for the transmit end interrupt to 0 (TEI interrupt disabled) to disable transmission. When transmission is disabled, the IR flag for the transmit end interrupt becomes 0, and the TXD pin becomes high-impedance.
(8) When transmission restarts	The next data is written in the main processing, etc.	The same processing as in “(2) When transmission starts” occurs.

4.3 Differences in Slave Reception

This section describes the differences in conditions for enabling reception between the RX and the M32C/R32C. Table 4.4 shows Differences in Conditions for Enabling Reception Between the RX and the M32C/R32C.

Table 4.4 Differences in Conditions for Enabling Reception Between the RX and the M32C/R32C

Setting	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
Master reception	The TE bit and RE bit are set to 1. *1 When continuous receive mode is disabled, if dummy data is written to the transmit buffer, 1 byte of the synchronous clock is output. When continuous receive mode is enabled, if the receive buffer is read, 1 byte of the synchronous clock is output.	In the case where the SCR.TE bit and the SCR.RE bit are set to 1, when dummy data is written to the transmit buffer, reception is enabled and 1 byte of the synchronization clock is output from the SCK pin. In the case where the SCR.TE bit is set to 0 and the SCR.RE bit is set to 1, reception becomes enabled when these settings are made, and either the CTS function is disabled, or the synchronization clock continues to be output from the SCK pin when input to the CTSn# pin is low.
Slave reception	The TE bit and RE bit are set to 1. *1 When continuous receive mode is disabled, if dummy data is written to the transmit buffer, reception is enabled. When continuous receive mode is enabled, if the receive buffer is read, reception is enabled. When reception is enabled, reception starts if the synchronous clock is input to the SCK pin.	When the SCR.RE bit is set to 1, reception is enabled. Reception starts when the synchronization clock is input to the SCK pin. *2

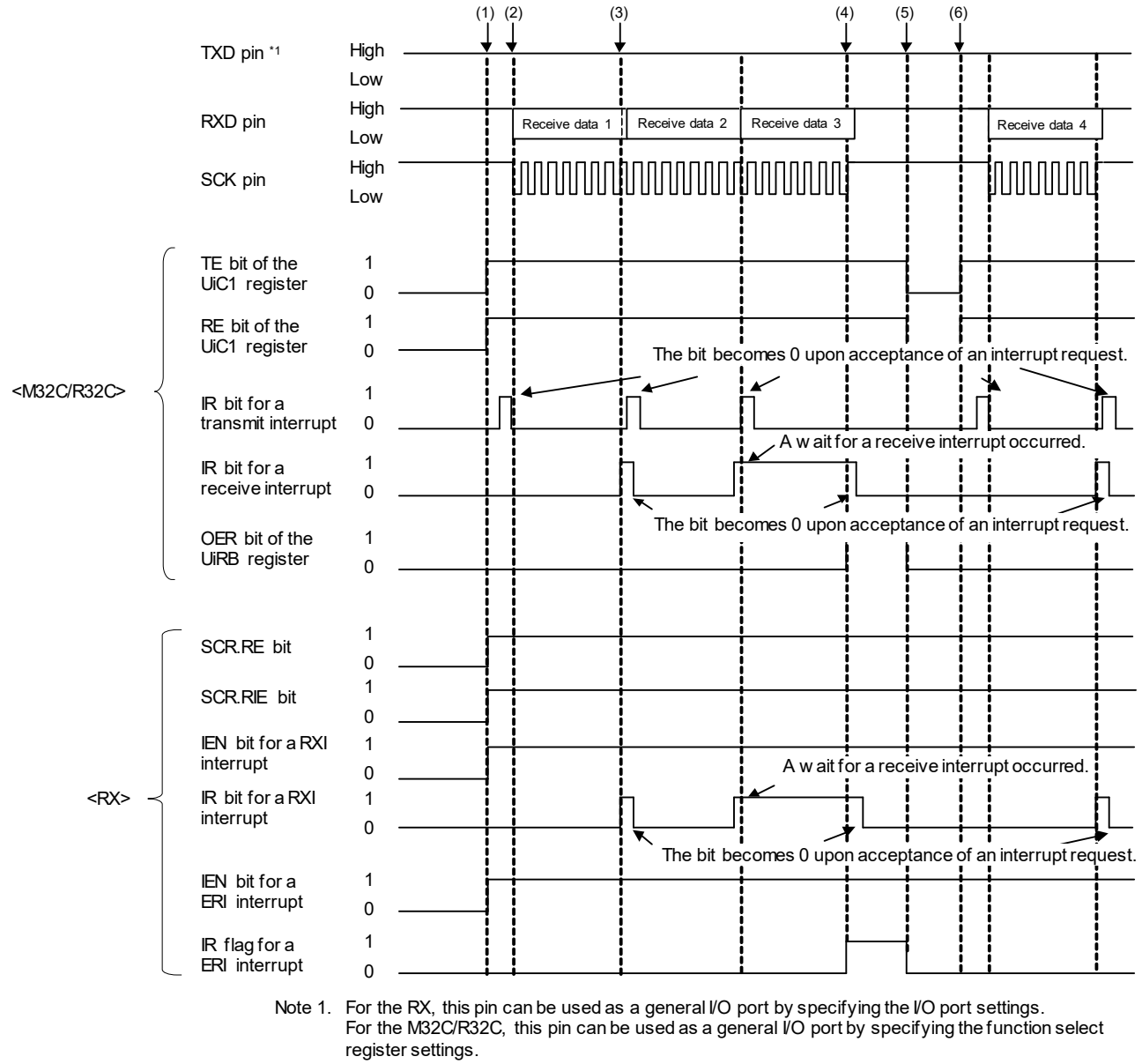
Notes: 1. In the M32C/R32C, the TE bit must be set to 1 even when only receiving data.
2. When the SCR.TE bit is set to 1, write dummy data to the transmit buffer before the synchronization clock is input to the SCK pin.

This section describes the differences for slave reception when the conditions in the RX Family assume the TE bit is 0 and the RE bit is 1, and when the conditions in the M32C/R32C Series assume the TE bit and RE bit are both 1 and continuous receive mode is disabled.

4.3.1 Differences in Timing During Slave Reception

This section describes an example of the receive interrupt being made to wait because of another interrupt during data reception.

Figure 4.3 shows Differences in Timing Between the RX and the M32C/R32C (During Reception). Table 4.5 shows Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (During Reception).



Note 1. For the RX, this pin can be used as a general I/O port by specifying the I/O port settings.
For the M32C/R32C, this pin can be used as a general I/O port by specifying the function select register settings.

Figure 4.3 Differences in Timing Between the RX and the M32C/R32C (During Reception)

Table 4.5 Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (During Reception)

Timing	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
(1) Setting to enable reception	After setting the TE bit to 1 (transmission is enabled) and setting the RE bit to 1 (reception is enabled), write dummy data to the transmit buffer. When transmit data is transferred to the transfer shift register, the IR bit for the transmit interrupt becomes 1, and the transmit interrupt occurs. Dummy data is rewritten to the transmit buffer in the transmit interrupt handling.	Set the SCR.RE bit to 1 (serial reception is enabled), set the RIE bit to 1 (RXI interrupt request is enabled), set the IEN bit for the RXI interrupt to 1 (RXI interrupt request is enabled), and set the IEN bit for the ERI interrupt to 1 (ERI interrupt request is enabled) to enable reception.
(2) Reception starts	Reception starts when a clock is input to the SCK pin.	
(3) When reception is complete	When 1 byte of data is received, receive data is transferred to the receive buffer, the IR flag (IR bit) for the receive interrupt (RXI interrupt) becomes 1, and a receive interrupt is generated. The value is read from the receive buffer in the receive interrupt handling.	
(4) When a receive error occurs	When an overrun error occurs, the OER bit in the receive buffer (UiRB register) becomes 1.	The ERI interrupt is generated when an overrun error occurs. Receive error processing is performed in the ERI interrupt handling.
(5) Clear the receive error flags	The TE bit is set to 0 (transmission disabled), the RE bit is set to 0 (reception disabled), and bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled).	After reading the error flags in the SSR register, write 0 to clear the error flags. After clearing all the error flags, the IR flag for the ERI interrupt becomes 0, and reception is enabled.
(6) When re-enabling reception	After setting bits SMD2 to SMD0 in the UiMR register to 001b (clock synchronous serial I/O mode), and setting both the TE bit and RE bit to 1, if dummy data is written to the transmit buffer, reception is enabled.	

4.4 Calculating the Bit Rate

The bit rate calculation methods of the RX and the M32C/R32C are different. Table 4.6 shows Differences in Calculating the Bit Rate.

Table 4.6 Differences in Calculating the Bit Rate

Item	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
Calculating the bit rate using the internal clock	Clock source / 2 (*m* + 1) Clock source: f1, f8, or f2n *m*: Value set in the UiBRG register	Clock source / 4 (*N* + 1) *1 Clock source: PCLK, PCLK/4, PCLK/16, or PCLK/64 *N*: Value set in the BRR register
Calculating the bit rate using the external clock	fEXT fEXT: Input from the CLKi pin	fEXT fEXT: Input from the SCKi pin

Note: 1. Based on “Relationship between N Setting in the BRR Register and Bit Rate B” in the User’s Manual: Hardware:

$B = PCLK / (8 \times 2^{2n-1} \times (N + 1))$

$= PCLK / (4 \times 2^{2n} \times (N + 1))$

$= (PCLK / 2^{2n}) / (4 \times (N + 1))$

= Clock source / (4 × (*N* + 1))

5. Differences in Simple I²C Mode

This section describes the differences in simple I²C communication between the RX and the M32C/R32C under the conditions shown in Table 5.1 Conditions for Simple I²C Communication.

Table 5.1 Conditions for Simple I²C Communication

Item	Conditions for Transmission and Reception
Peripheral function operating clock	16 MHz
Transfer rate	100 kbps
Channels used	RX Family: SCI0 M32C/R32C Series: UART0

5.1 Differences in Master Transmission

This section describes the differences in master transmission in simple I²C mode.

5.1.1 Differences in Timing During Master Transmission

Figure 5.1 shows Differences in Timing During Master Transmission Between the RX and the M32C/R32C (When Transmitting 3 Bytes). Table 5.2 shows Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (During Master Transmission).

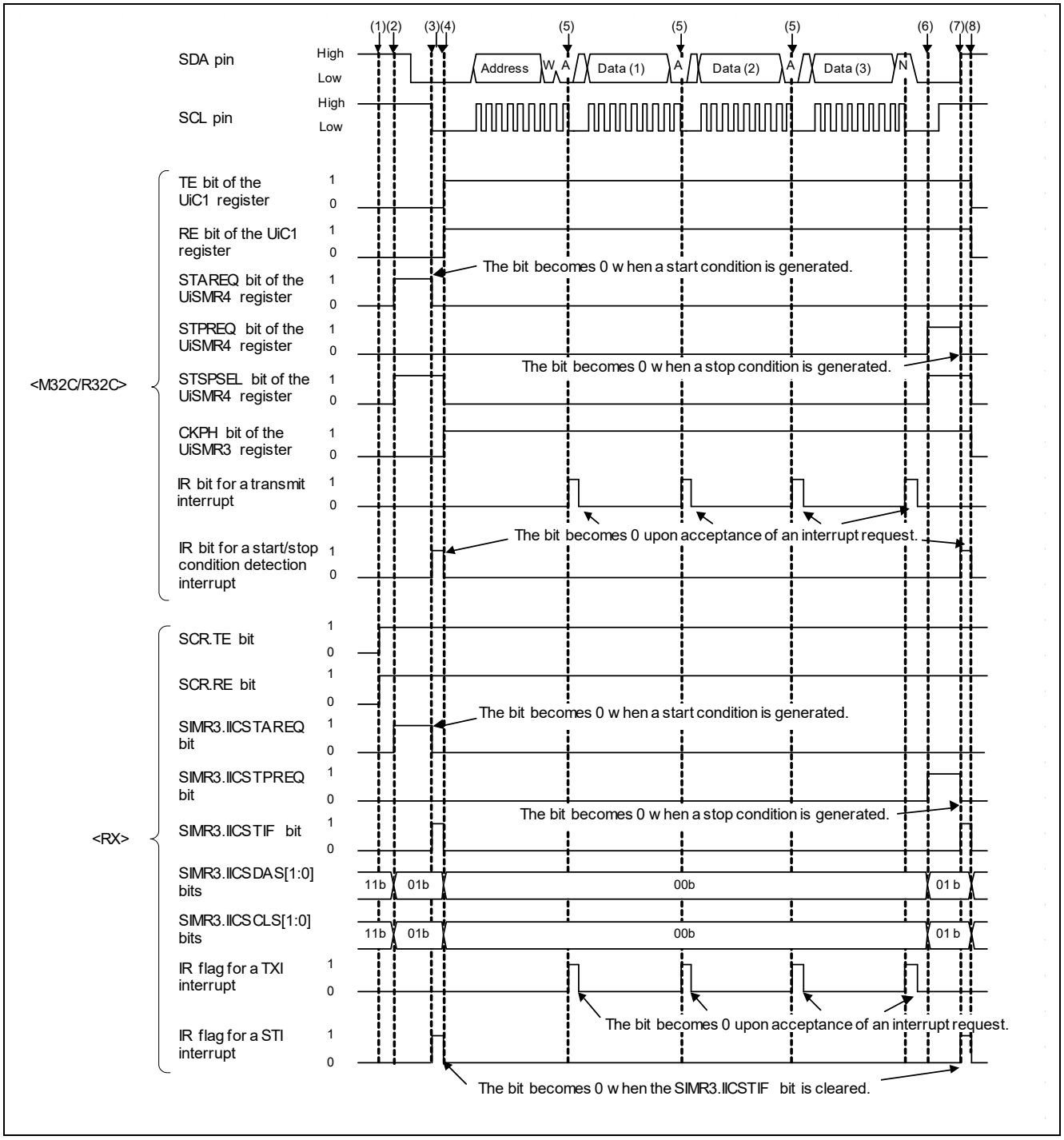


Figure 5.1 Differences in Timing During Master Transmission Between the RX and the M32C/R32C (When Transmitting 3 Bytes)

Table 5.2 Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (During Master Transmission)

Timing	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
(1) Before transmission starts	Bits ILVL2 to ILVL0 in the BCN0IC register are set and the start condition/stop condition interrupt is enabled.	The SCR.TE bit is set to 1 (serial transmission is enabled), the SCR.RE bit is set to 1 (serial reception is enabled), the SCR.TIE bit is set to 1 (TXI interrupt request is enabled), and the SCR.TEIE bit is set to 1 (STI interrupt request is enabled).
(2) Output of a start condition	After the STAREQ bit is set to 1, the STSPSEL bit is set to 1 (select start condition/stop condition generation circuit).	At the same time the SIMR3.IICSTAREQ bit is set to 1 (start condition is generated), the SIMR3.IICSCLS[1:0] bits and the SIMR3.IICSDAS[1:0] bits are set to 01b (generate a start, restart, or stop condition).
(3) Generation of a start condition detection interrupt	The STAREQ bit becomes 0, and the IR bit for the start condition/stop condition detection interrupt becomes 1.	The SIMR3.IICSTAREQ bit is set to 0 (start condition is not generated), the SIMR3.IICSTIF bit is set to 1 (all request generation has been completed), and the IR flag for the STI interrupt becomes 1.
(4) Handling of the start condition detection interrupt	The CKPH bit is set to 1 (with clock delay), the TE bit is set to 1 (transmission enabled), the RE bit is set to 1 (reception enabled), and the STSPSEL bit is set to 0 (select serial I/O circuit). In addition, bits ILVL2 to ILVL0 in the SiTIC register are set and the transmit interrupt is enabled. Then, the first byte of transmit data (slave address and W bit) is written to the transmit buffer.	The SIMR3.IICSTIF bit is set to 0, the SIMR3.IICSCLS[1:0] bits are set to 00b (serial clock output), and the SIMR3.IICSDAS[1:0] bits are set to 00b (serial data output). When the SIMR3.IICSTIF bit is set to 0, the IR flag for the STI interrupt becomes 0. Then, the first byte of transmit data (slave address and R/W bit) is written to the transmit buffer.
(5) When transmission is complete	The IR flag (IR bit) for the transmit interrupt (TXI interrupt) becomes 1, and the transmit interrupt is generated. In the transmit interrupt handling, after ACK or NACK is confirmed, the second byte and successive data is written to the transmit buffer.	
(6) Handling of a transmit interrupt after the last data is output	After the STPREQ bit is set to 1, the STSPSEL bit is set to 1 (select start condition/stop condition generation circuit).	At the same time the SIMR3.IICSTPREQ bit is set to 1 (stop condition is generated), the SIMR3.IICSCLS[1:0] bits and the SIMR3.IICSDAS[1:0] bits are set to 01b.
(7) When transmission is complete	The STPREQ bit becomes 0, and the IR bit for the start condition/stop condition detection interrupt becomes 1.	The SIMR3.IICSTPREQ bit becomes 0 (stop condition is not generated), the SIMR3.IICSTIF bit becomes 1, and the IR flag for the STI interrupt becomes 1.
(8) Handling of a stop condition detection interrupt	The TE bit is set to 0 (transmission disabled), the RE bit is set to 0 (reception disabled), the CKPH bit is set to 0 (no clock delay), the STSPSEL bit is set to 0 (select serial I/O circuit), and bits SMD2 to SMD0 in the UiMR register are set to 000b.	The SIMR3.IICSTIF bit is set to 0, and bits SIMR3.IICSCLS[1:0] and SIMR3.IICSDAS[1:0] are set to 11b (high-impedance). When the SIMR3.IICSTIF bit is set to 0, the IR flag for the STI interrupt becomes 0.

5.2 Differences in Master Reception

This section describes the differences in master reception in simple I²C mode.

5.2.1 Differences in Timings During Master Reception

Figure 5.2 shows Differences in Timing During Master Reception Between the RX and the M32C/R32C (When Receiving 3 Bytes). Table 5.3 shows Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (During Master Reception).

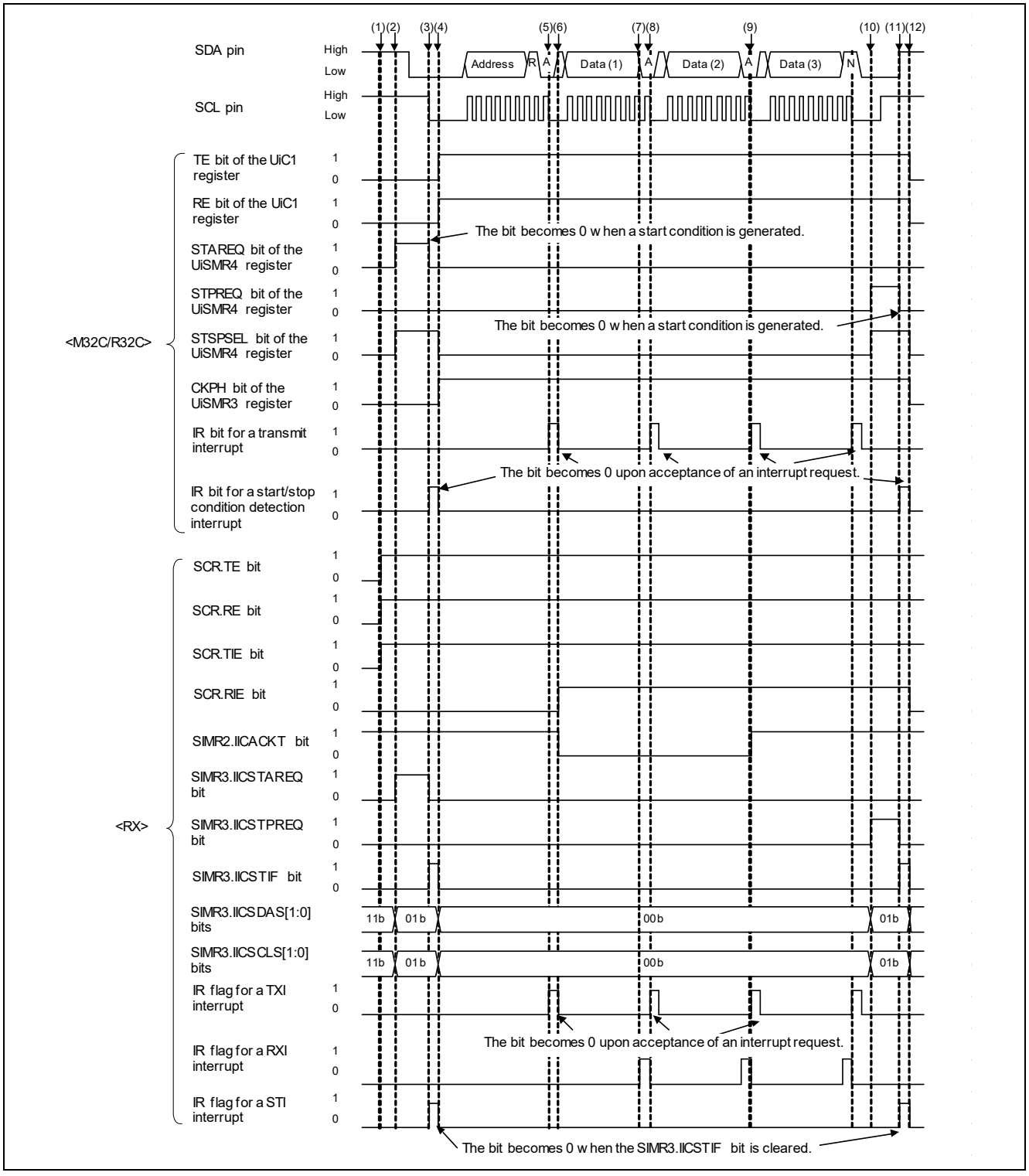


Figure 5.2 Differences in Timing During Master Reception Between the RX and the M32C/R32C (When Receiving 3 Bytes)

Table 5.3 Differences in Operation and Processing at Various Timings Between the RX and the M32C/R32C (During Master Reception)

Timing		M32C/R32C (M32C/87, R32C/118)	RX (RX660)
(1)	When reception is enabled	Bits ILVL2 to ILVL0 in the BCN0IC register are set and the start condition/stop condition interrupt is enabled.	The SCR.TE bit is set to 1 (serial transmission is enabled), the SCR.RE bit is set to 1 (serial reception is enabled), the SCR.TIE bit is set to 1 (TXI interrupt request is enabled), and the SCR.TEIE bit is set to 1 (STI interrupt request is enabled).
(2)	When reception starts	After the STAREQ bit is set to 1, the STSPSEL bit is set to 1 (select start condition/stop condition generation circuit).	At the same time the SIMR3.IICSTAREQ bit is set to 1 (start condition is generated), the SIMR3.IICSCLS[1:0] bits and the SIMR3.IICSDAS[1:0] bits are set to 01b (generate a start, restart, or stop condition).
(3)	When transmission is complete	The STAREQ bit becomes 0, and the IR bit for the start condition/stop condition detection interrupt becomes 1.	The SIMR3.IICSTAREQ bit is set to 0 (start condition is not generated), the SIMR3.IICSTIF bit is set to 1 (all request generation has been completed), and the IR flag for the STI interrupt becomes 1.
(4)	When a receive error occurs	The CKPH bit is set to 1 (with clock delay), the TE bit is set to 1 (transmission enabled), the RE bit is set to 1 (reception enabled), and the STSPSEL bit is set to 0 (select serial I/O circuit). Then, the first byte of transmit data (slave address and R bit) is written to the transmit buffer.	The SIMR3.IICSTIF bit is set to 0, the SIMR3.IICSCLS[1:0] bits are set to 00b (serial clock output), and the SIMR3.IICSDAS[1:0] bits are set to 00b (serial data output). When the SIMR3.IICSTIF bit is set to 0, the IR flag for the STI interrupt becomes 0. Then, the first byte of transmit data (slave address and R/W bit) is written to the transmit buffer.
(5)	Clear the receive error flags	The IR flag (IR bit) for the transmit interrupt (TXI interrupt) becomes 1, and the transmit interrupt is generated.	
(6)	When reception is restarted	0x00FF is written as dummy data to the transmit buffer.	The SCR.RIE bit is set to 1 (RXI interrupt request is enabled), the SIMR2.IICACKT bit is set to 0 (ACK transmission), and dummy data (0xFF) is written to the transmit buffer.
(7)	When data is received	—	The IR flag for the RXI interrupt becomes 1. Data received is read in the RXI interrupt handling.
(8)	ACK transmission complete	The IR bit for the transmit interrupt becomes 1. After the transmit interrupt is used to read the receive data, dummy data (0x00FF) is written to the transmit buffer.	The IR flag for the TXI interrupt becomes 1. Dummy data (0xFF) is written to the transmit buffer in the TXI interrupt handling.
(9)	Handling of a transmit complete interrupt before the last data is written	After reading the data received, 0x01FF is written to the transmit buffer as dummy data.	The SIMR2.IICACKT bit is set to 1 (NACK transmission), and dummy data (0xFF) is written to the transmit buffer.

Timing	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
(10) Handling of a transmit interrupt after the last data is output	After the STPREQ bit is set to 1, the STSPSEL bit is set to 1 (select start condition/stop condition generation circuit).	At the same time the SIMR3.IICSTPREQ bit is set to 1 (stop condition is generated), the SIMR3.IICSCLS[1:0] bits and the SIMR3.IICSDAS[1:0] bits are set to 01b.
(11) Stop condition detection interrupt generation	The STPREQ bit becomes 0, and the IR bit for the start condition/stop condition detection interrupt becomes 1.	The SIMR3.IICSTPREQ bit becomes 0 (stop condition is not generated), the SIMR3.IICSTIF bit becomes 1, and the IR flag for the STI interrupt becomes 1.
(12) Handling of a stop condition detection interrupt	The TE bit is set to 0 (transmission disabled), the RE bit is set to 0 (reception disabled), the CKPH bit is set to 0 (no clock delay), the STSPSEL bit is set to 0 (select serial I/O circuit), and bits SMD2 to SMD0 in the UiMR register are set to 000b.	The SCR.RIE bit is set to 0 (RXI interrupt request is disabled), the SIMR3.IICSTIF bit is set to 0, and bits SIMR3.IICSCLS[1:0] and SIMR3.IICSDAS[1:0] are set to 11b (high-impedance). When the SIMR3.IICSTIF bit is set to 0, the IR flag for the STI interrupt becomes 0.

5.3 Calculating the Bit Rate

The bit rate calculation methods of the RX and the M32C/R32C are different. Table 5.4 shows Differences in Calculating the Bit Rate.

Table 5.4 Differences in Calculating the Bit Rate

Item	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
Calculating the bit rate using the internal clock	Clock source / 2 (*m* + 1) Clock source: f1, f8, or f2n *m*: Value set in the UiBRG register	Clock source / 32 (*N* + 1) *1 Clock source: PCLK, PCLK/4, PCLK/16, or PCLK/64 *N*: Value set in the BRR register

Note: 1. Based on “Relationship between N Setting in the BRR Register and Bit Rate B” in the User’s Manual: Hardware:

$B = PCLK / (64 \times 2^{2n-1} \times (N + 1))$

$= PCLK / (32 \times 2^{2n} \times (N + 1))$

$= (PCLK / 2^{2n}) / (32 \times (N + 1))$

$=$ Clock source $/ (32 \times (N + 1))$

6. Appendix

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

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

6.1.1 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 the peripheral function.

Table 6.1 shows Comparison of Conditions for Interrupt Generation Between the M32C/R32C and the RX.

Table 6.1 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 there is an interrupt request from 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	—	Interrupt enable or disable can be specified 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: Hardware.

6.1.2 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: Hardware.

6.1.3 I/O Ports

In the RX Family, the MPC must be configured in order to assign I/O signals of 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 6.2 shows Comparison of I/O Settings for Peripheral Function Pins Between the M32C and the RX.
Table 6.3 shows Comparison of I/O Settings for Peripheral Function Pins Between the R32C and the RX.

Table 6.2 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 6.3 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.

6.2 I/O Register Macros

Macro definitions listed in Table 6.4 can be found in the RX I/O register definitions (iodefine.h).

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

Table 6.4 shows Macro Usage Examples.

Table 6.4 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.

6.3 Intrinsic Functions

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

Table 6.5 shows Examples of Differences in the Settings of Control Registers and Descriptions of Special Instructions Between the M32C/R32C and the RX.

Table 6.5 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.

7. 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, 2024	—	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/.