RX Family and M32C/R32C Series

Guide for Migration from the M32C/R32C to the RX: External Bus

Abstract

This document describes migration from the external bus in the M32C/R32C Series MCU to the external bus in the RX Family MCU.

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 external bus.

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

Item	M32C/R32C Series	RX Family
Operating mode to access the external memory device	• Memory expansion mode • Microprocessor mode	• On-chip ROM enabled extended mode • On-chip ROM disabled extended mode
External memory area	External area	External address space CSn area
Peripheral function registers	Special function registers (SFRs)	I/O registers

Contents

1. Functional Differences in the External Bus

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

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

Item	M32C (M32C/87)	RX (RX660)
Memory size	16 MB	8 MB
Bus type	Separate bus/multiplexed bus	Separate bus/multiplexed bus
Data bus width	8-bit/16-bit	8-bit/16-bit
How to set the data bus width	Using the DS register Using the BYTE pin *1	Using the CSn control register (n = 0 to 3)
Address buses	8 to 24 buses selectable	8 to 21 buses selectable
Chip select outputs	4	4
Write access mode	Write signal combinations • BHE/WR • WRL/WRH	1-write strobe mode Byte strobe mode
Wait cycle	RDY pin	WAIT pin
Software wait	A wait of 1 to 8 clock cycles can be inserted in addition to the base clock cycle. *2	Wait for up to 31 cycles
Recovery cycle	1 cycle can be inserted (The last accessed address is retained for address output.)	Maximum of 15 cycles can be inserted (Selectable from 8 patterns)
Page access	—	Supported

Notes: 1. External area 3 only

2. Selectable from the following clock cycle combinations: “1ϕ + 1ϕ”, “1ϕ + 2ϕ”, “1ϕ + 3ϕ”, “1ϕ + 4ϕ”, “1ϕ + 5ϕ”, “1ϕ + 6ϕ”, “2ϕ + 2ϕ”, “2ϕ + 3ϕ”, “2ϕ + 4ϕ”, “2ϕ + 5ϕ”, “3ϕ + 3ϕ”, “3ϕ + 4ϕ”, “3ϕ + 5ϕ”, and “3ϕ + 6ϕ”.

Table 1.2 General Differences Between the R32C/118 and the RX660

Item	R32C (R32C/118)	RX (RX660)
Memory size	64 MB	8 MB
Bus type	Separate bus/multiplexed bus	Separate bus/multiplexed bus
Data bus width	8, 16, or 32 bits *1	8-bit/16-bit
How to set the data bus width	Using the PBC register and the EBCi register (i = 0 to 3)	Using the CSn control register (n = 0 to 3)
Address buses	8 to 24 buses selectable	8 to 21 buses selectable
Chip select outputs	4	4
Write access mode	Write signal combinations • WR0/WR1/WR2/WR3 • WR/BC0/BC1/BC2/BC3	1-write strobe mode Byte strobe mode
Wait cycle	RDY pin	WAIT pin
Software wait	Wait for up to 15 cycles	Wait for up to 31 cycles
Recovery cycle	—	Maximum of 15 cycles can be inserted (Selectable from 8 patterns)
Page access	—	Supported

Note: 1. Supported by only the 144-pin version

Hereafter, the description in this application note assumes that the separate bus interface is used.

1.1 Example of Connecting to an External Bus

The bus pin connections for the M32C/R32C and the RX are the same. Note, however, that the names of bus control pins are different.

Figure 1.1 shows Example of Bus Connection When the Bus Width is 16 Bits (RX/M32C). Figure 1.2 shows Example of Bus Connection When the Bus Width is 16 Bits (RX/R32C). Figure 1.3 shows Example of Bus Connection When the Bus Width is 8 Bits.

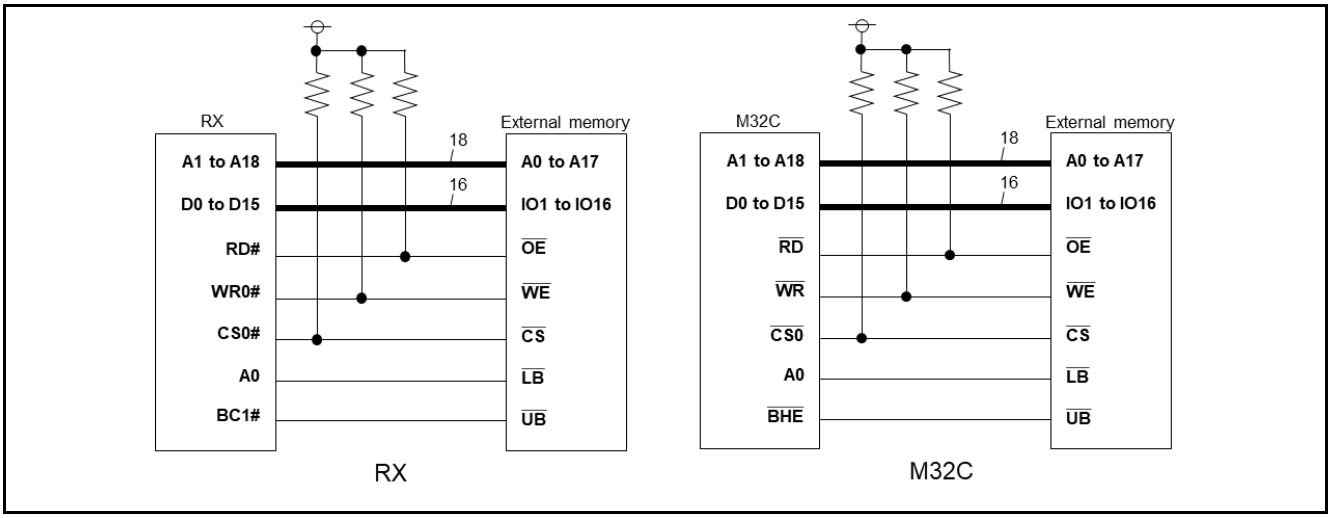


Figure 1.1 Example of Bus Connection When the Bus Width is 16 Bits (RX/M32C)

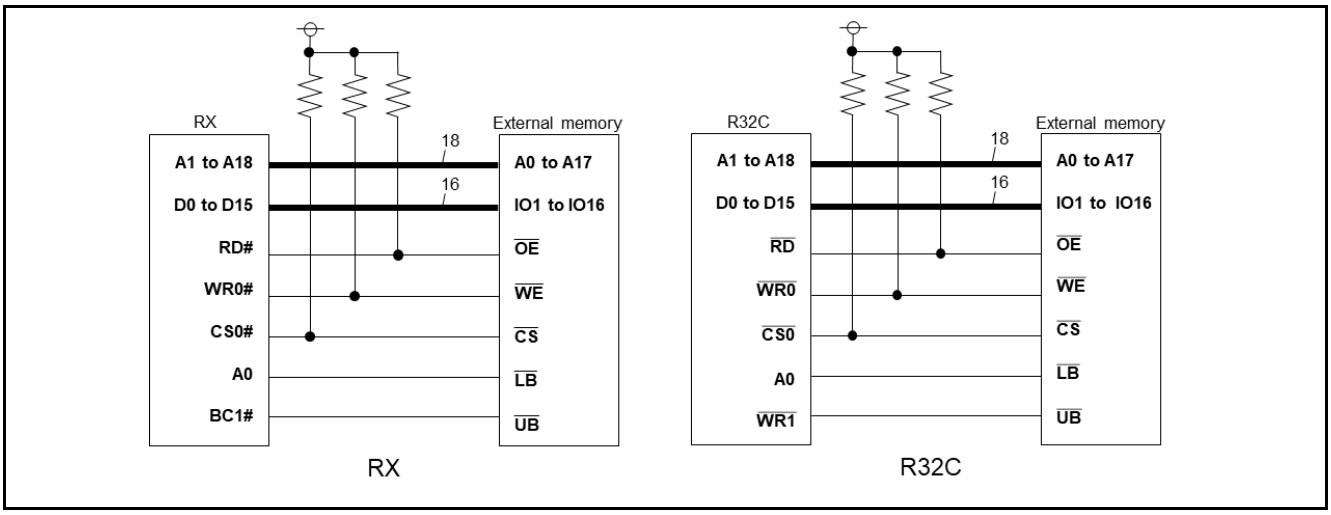


Figure 1.2 Example of Bus Connection When the Bus Width is 16 Bits (RX/R32C)

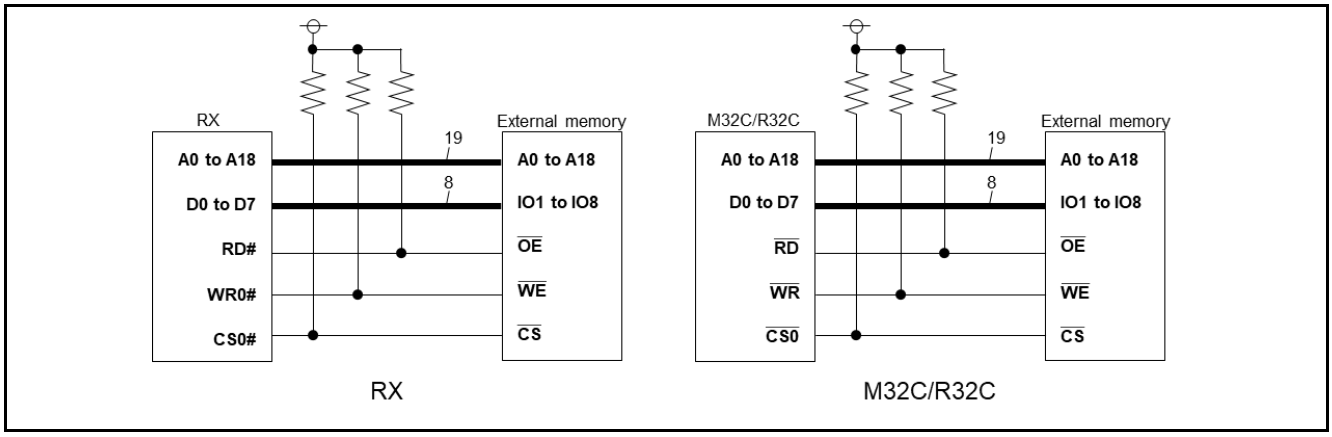


Figure 1.3 Example of Bus Connection When the Bus Width is 8 Bits

1.2 Calculating the Number of Bus Cycles

1.2.1 Calculating the Number of Bus Cycles When Reading

In this section, Table 1.3 Differences in the Bus Timing Settings Between the M32C and the RX (during a Read Operation) and Table 1.4 Differences in the Bus Timing Settings Between the R32C and the RX (during a Read Operation) show the differences in the bus timings to be set based on the timing chart in Figure 1.4 Example of Basic Bus Timings (during a Read Operation).

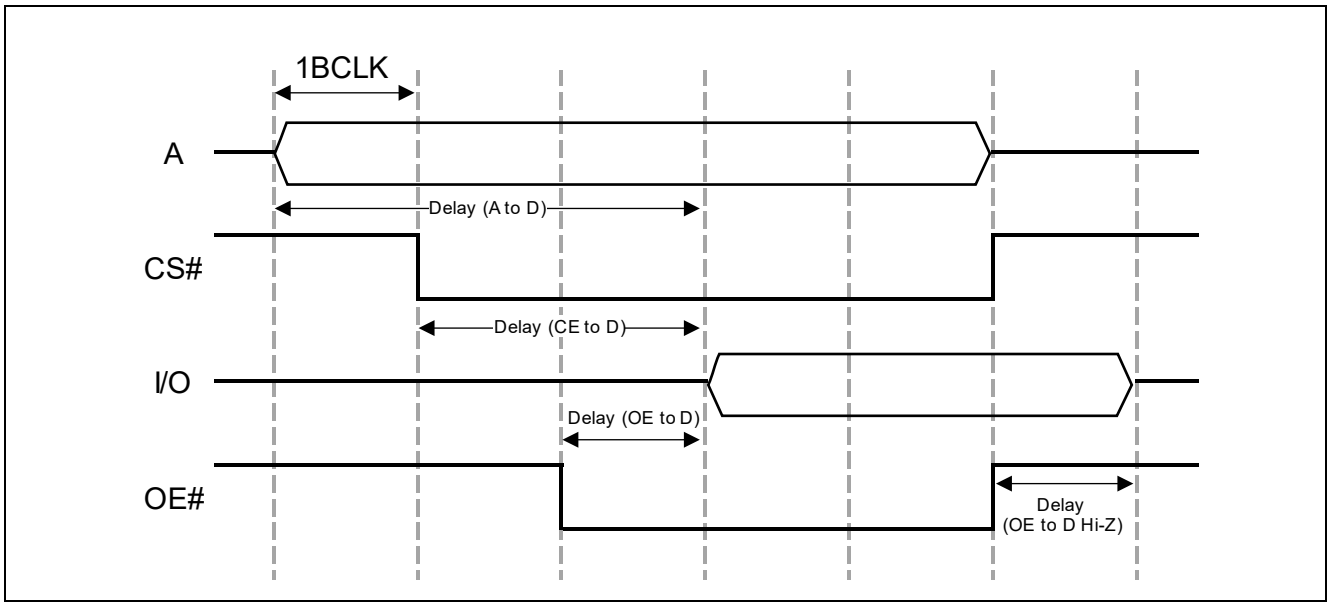


Figure 1.4 Example of Basic Bus Timings (during a Read Operation)

Table 1.3 Differences in the Bus Timing Settings Between the M32C and the RX (during a Read Operation)

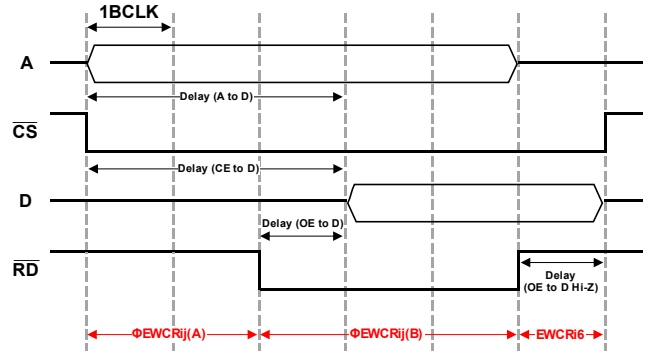
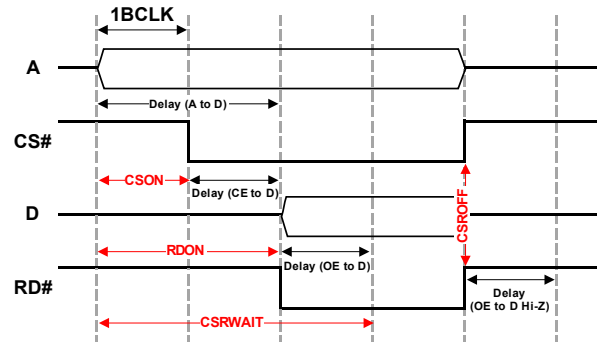
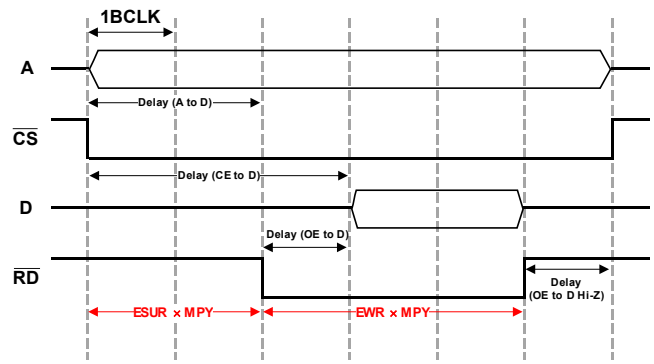
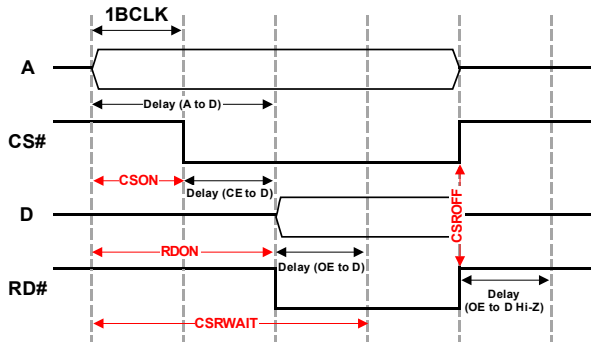
<table>
<tr><th colspan="2">M32C (M32C/87)</th><th colspan="2">RX (RX660)</th></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
<tr><td colspan="2" rowspan="2">EWCRij (Aϕ + Bϕ) (i = 0 to 3, j = 0 to 4)
Aϕ sets the number of cycles from the start of the bus access to the falling edge of the RD signal.
Bϕ sets the number of cycles from falling edge of the RD signal to the rising edge.</td><td>CSON</td><td>Sets the number of wait cycles to be inserted before asserting the CSn# signal</td></tr>
<tr><td>RDON</td><td>Sets the number of wait cycles to be inserted before asserting the RD# signal</td></tr>
<tr><td>EWCRi6</td><td>Sets the number of recovery cycles according to the number of needed idle cycles</td><td>CSRWAIT</td><td>Sets the number of cycles to be inserted in the first access of the normal read cycle</td></tr>
<tr><td colspan="2"></td><td>CSROFF</td><td>Sets the number of cycles from negating the RD# signal during a read access, to negating the CSn# signal</td></tr>
</table>

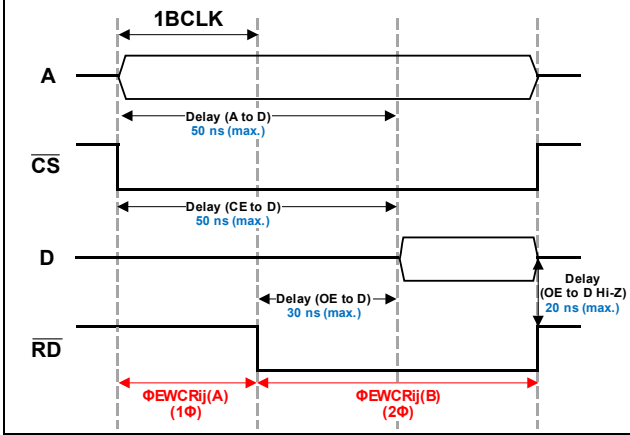
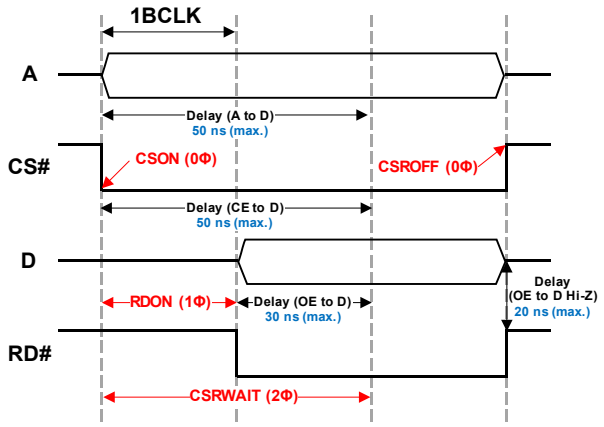
Table 1.4 Differences in the Bus Timing Settings Between the R32C and the RX (during a Read Operation)

<table>
<tr><th colspan="2">R32C (R32C/118)</th><th colspan="2">RX (RX660)</th></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
<tr><td>ESUR</td><td>Sets address setup time to be inserted before the RD signal is asserted.
The number of cycles is calculated from the following expression: ESUR × MPY + 0.5</td><td>CSON</td><td>Sets the number of wait cycles to be inserted before asserting the CSn# signal</td></tr>
<tr><td rowspan="2">EWR</td><td rowspan="2">Sets the pulse width of the RD signal.
The number of cycles is calculated from the following expression: EWR × MPY + 0.5</td><td>RDON</td><td>Sets the number of wait cycles to be inserted before asserting the RD# signal</td></tr>
<tr><td>CSRWAIT</td><td>Sets the number of cycles to be inserted in the first access of the normal read cycle</td></tr>
<tr><td>MPY</td><td>Sets a multiple of the number of cycles set in ESUR or EWR.</td><td>CSROFF</td><td>Sets the number of cycles from negating the RD# signal during a read access, to negating the CSn# signal</td></tr>
<tr><td colspan="2"></td><td>—</td><td></td></tr>
</table>

Assume that the read operation for the connected external memory has the following characteristics. Examples of the register settings configured according to these characteristics are shown in Table 1.5 Differences in the External Bus Register Settings Between the M32C and the RX (During a Read Operation) and Table 1.6 Differences in the External Bus Register Settings Between the R32C and the RX (During a Read Operation).

- BCLK = 16 MHz
- Delay (A to D) = 50 ns (max.)
- Delay (CE to D) = 50 ns (max.)
- Delay (OE to D) = 30 ns (max.)
- Delay (OE to D Hi-Z) = 20 ns (max.)

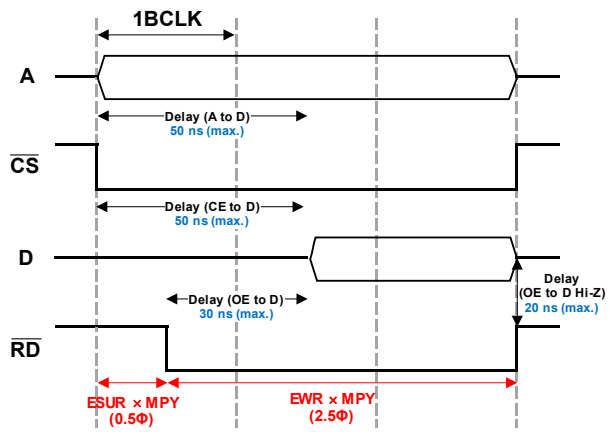
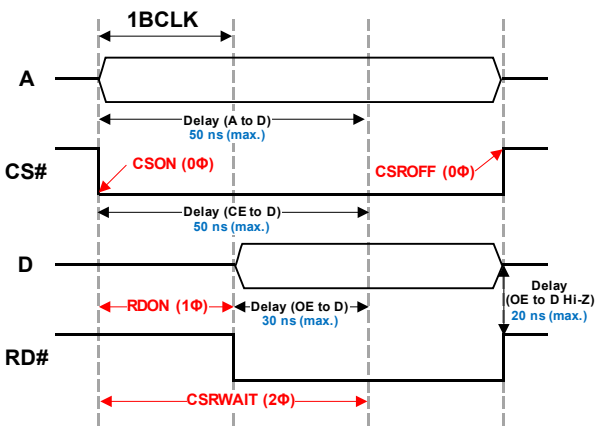
Table 1.5 Differences in the External Bus Register Settings Between the M32C and the RX (During a Read Operation)

<table>
<tr><th colspan="3">M32C (M32C/87)</th><th colspan="2">RX (RX660)</th></tr>
<tr><td colspan="3"> 
</td><td colspan="2"> 
</td></tr>
<tr><td colspan="3" rowspan="2">EWCRij (Aϕ + Bϕ) (i = 0 to 3, j = 0 to 4) *1
EWCR = 0x00010; /* 2 wait cycles (1ϕ + 2ϕ) */</td><td>CSON</td><td>CS0WCR2.BIT.CSON = 0; *2</td></tr>
<tr><td>RDON</td><td>CS0WCR2.BIT.RDON = 1; *2</td></tr>
<tr><td colspan="3" rowspan="2">EWCRi6 EWCRi6 = 0; /* No recovery cycles */</td><td>CSRWAIT</td><td>CS0WCR1.BIT.CSRWAIT = 2; *2</td></tr>
<tr><td>CSROFF</td><td>CS0WCR2.BIT.CSROFF = 0;</td></tr>
</table>

Notes: 1. Set one of the following clock cycle combinations: “1ϕ + 1ϕ”, “1ϕ + 2ϕ”, “1ϕ + 3ϕ”, “1ϕ + 4ϕ”, “1ϕ + 5ϕ”, “1ϕ + 6ϕ”, “2ϕ + 2ϕ”, “2ϕ + 3ϕ”, “2ϕ + 4ϕ”, “2ϕ + 5ϕ”, “3ϕ + 3ϕ”, “3ϕ + 4ϕ”, “3ϕ + 5ϕ”, and “3ϕ + 6ϕ”

2. Satisfy the following condition: CSnWCR2.CSON bit ≤ CSnWCR2.RDON bit ≤ CSnWCR1.CSRWAIT bit.

Table 1.6 Differences in the External Bus Register Settings Between the R32C and the RX (During a Read Operation)

<table>
<tr><th colspan="2">R32C (R32C/118)</th><th colspan="2">RX (RX660)</th></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
<tr><td>ESUR</td><td rowspan="2">ESUR = 0x00; /* 0 × MPY(1) + 0.5 (0.5 cycles) */</td><td>CSON</td><td>CS0WCR2.BIT.CSON = 0; *1, *2</td></tr>
<tr><td></td><td></td><td>RDON</td><td>CS0WCR2.BIT.RDON = 1; *1</td></tr>
<tr><td>EWR</td><td rowspan="2">EWR = 0x01; /* 2 × MPY(1) + 0.5 (2.5 cycles) */</td><td>CSRWAIT</td><td>CS0WCR1.BIT.CSRWAIT = 2; *1</td></tr>
<tr><td></td><td></td><td>CSROFF</td><td>CS0WCR2.BIT.CSROFF = 0;</td></tr>
<tr><td>MPY</td><td>MPY = 0x00; /* No multiplication (1x) */</td><td colspan="2">—</td></tr>
</table>

Note: 1. Satisfy the following condition: CSnWCR2.CSON bit ≤ CSnWCR2.RDON bit ≤ CSnWCR1.CSRWAIT bit.

1.2.2 Calculating the Number of Bus Cycles When Writing

In this section, Table 1.7 Differences in the Bus Timing Settings Between the M32C and the RX (During a Write Operation) and Table 1.8 Differences in the Bus Timing Settings Between the R32C and the RX (During a Write Operation) show the differences in the bus timings to be set based on the timing chart in Figure 1.5 Example of Basic Bus Timings (during a Write Operation).

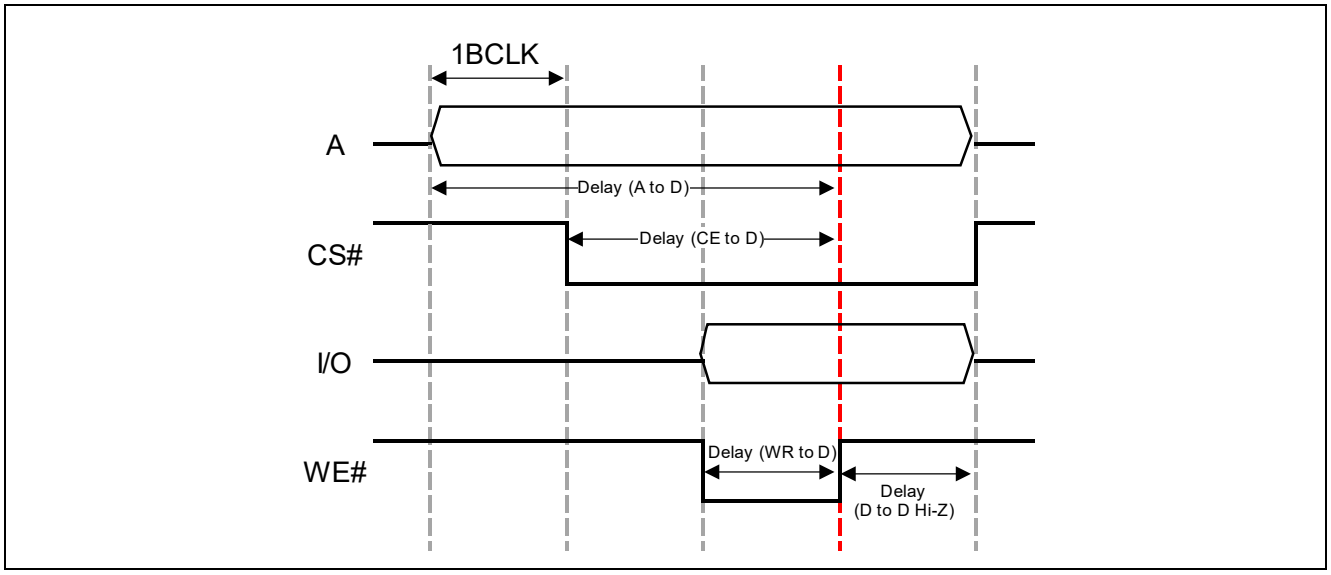


Figure 1.5 Example of Basic Bus Timings (during a Write Operation)

Table 1.7 Differences in the Bus Timing Settings Between the M32C and the RX (During a Write Operation)

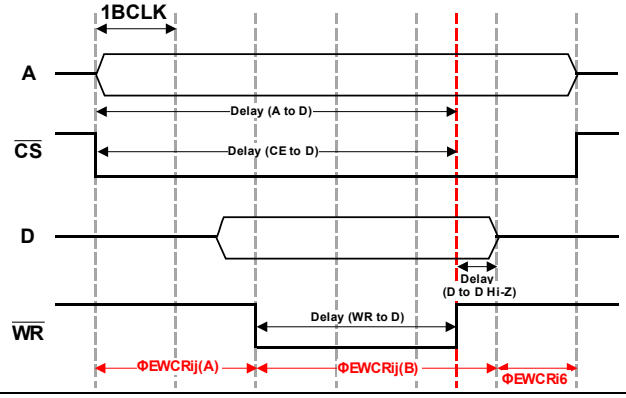
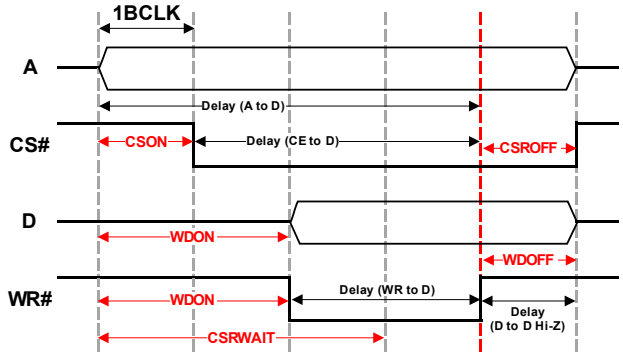
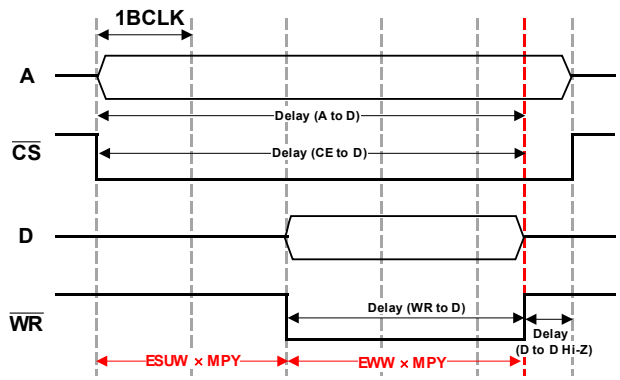
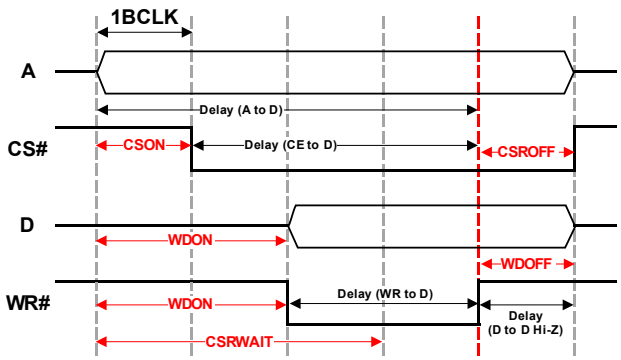
<table>
<tr><th colspan="2">M32C (M32C/87)</th><th colspan="2">RX (RX660)</th></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
<tr><td colspan="2">EWCRij (Aϕ + Bϕ) (i = 0 to 3, j = 0 to 4)
Aϕ sets the number of cycles from the start of the bus access to the falling edge of the WR signal.
Bϕ sets the number of cycles from falling edge of the WR signal to the rising edge.</td><td>CSON</td><td>Sets the number of wait cycles to be inserted before asserting the CSn# signal</td></tr>
<tr><td colspan="2"></td><td>WDON</td><td>Sets the number of wait cycles to be inserted before outputting write data</td></tr>
<tr><td colspan="2"></td><td>WRON</td><td>Sets the number of wait cycles to be inserted before asserting the WRn# signal</td></tr>
<tr><td colspan="2"></td><td>WDOFF</td><td>Sets the number of cycles from negating the WRn# signal during a write access, to write data output completion</td></tr>
<tr><td colspan="2"></td><td>CSWWAIT</td><td>Sets the number of cycles to be inserted in the first access of the normal write cycle</td></tr>
<tr><td>EWCRi6</td><td>Sets the number of recovery cycles according to the number of needed idle cycles</td><td>CSWOFF</td><td>Sets the number of cycles from negating the WRn# signal during a write access, to negating the CSn# signal</td></tr>
</table>

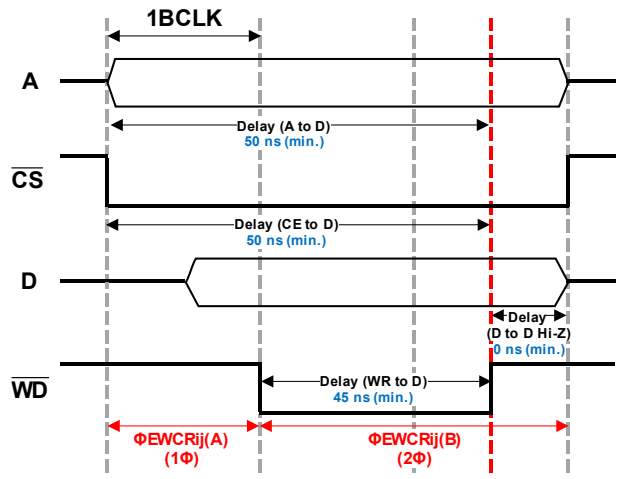
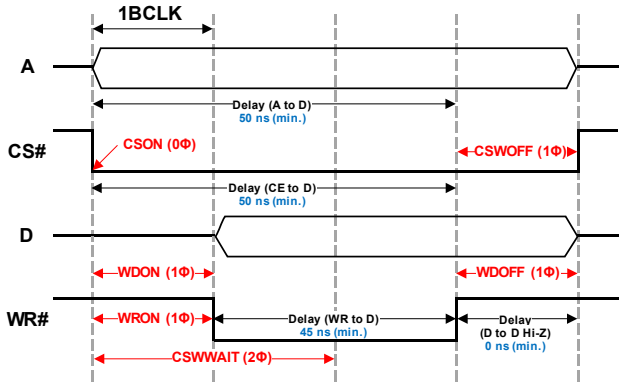
Table 1.8 Differences in the Bus Timing Settings Between the R32C and the RX (During a Write Operation)

<table>
<tr><th>R32C (R32C/118)</th><th colspan="2">RX (RX660)</th></tr>
<tr><td> 
</td><td colspan="2"> 
</td></tr>
<tr><td>ESUW</td><td>Sets address setup time to be inserted before the WR signal is asserted.
The number of cycles is calculated from the following expression: ESUW × MPY + 0.5</td><td>CSON</td><td>Sets the number of wait cycles to be inserted before asserting the CSn# signal</td></tr>
<tr><td rowspan="2">EWW</td><td rowspan="2">Sets the pulse width of the WR signal.
The number of cycles is calculated from the following expression: EWW × MPY + 0.5</td><td>WDON</td><td>Sets the number of wait cycles to be inserted before outputting write data</td></tr>
<tr><td>WRON</td><td>Sets the number of wait cycles to be inserted before asserting the WRn# signal</td></tr>
<tr><td rowspan="3">MPY</td><td rowspan="3">Sets a multiple of the number of cycles set in ESUW or EWW.</td><td>WDOFF</td><td>Sets the number of cycles from negating the WRn# signal during a write access, to write data output completion</td></tr>
<tr><td>CSWWAIT</td><td>Sets the number of cycles to be inserted in the first access of the normal write cycle</td></tr>
<tr><td>CSWOFF</td><td>Sets the number of cycles from negating the WRn# signal during a write access, to negating the CSn# signal</td></tr>
<tr><td>—</td><td colspan="2">—</td></tr>
</table>

Assume that the write operation for the connected external memory has the following characteristics. Examples of the register settings configured according to these characteristics are shown in Table 1.9 Differences in the External Bus Register Settings Between the M32C and the RX (During a Write Operation) and Table 1.10 Differences in the External Bus Register Settings Between the R32C and the RX (During a Write Operation).

- BCLK = 16 MHz
- Delay (A to D) = 50 ns (min.)
- Delay (CE to D) = 50 ns (min.)
- Delay (WR to D) = 45 ns (min.)
- Delay (D to D Hi-Z) = 0 ns (min.)

Table 1.9 Differences in the External Bus Register Settings Between the M32C and the RX (During a Write Operation)

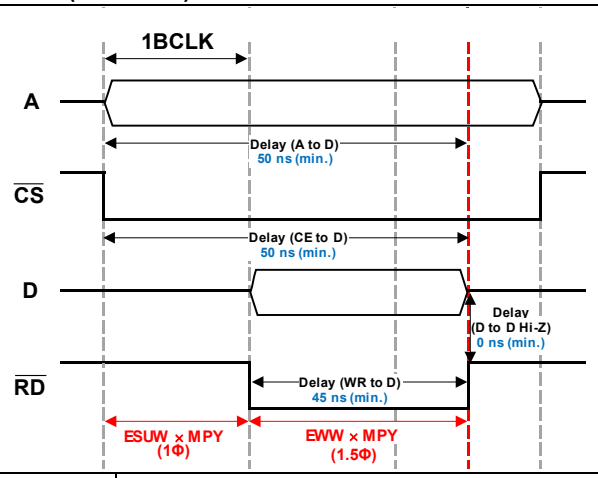
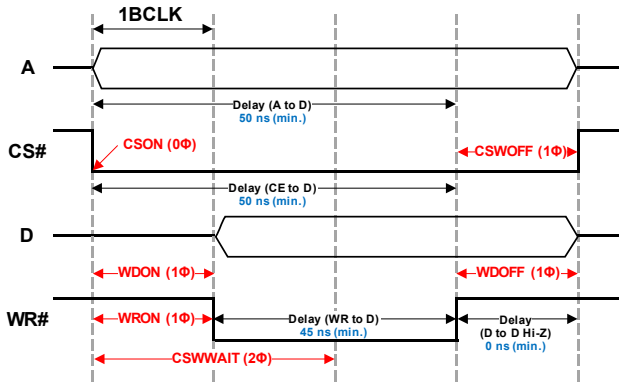
<table>
<tr><th colspan="2">M32C (M32C/87)</th><th colspan="2">RX (RX660)</th></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
<tr><td colspan="2" rowspan="5">EWCRij (Aϕ + Bϕ) (i = 0 to 3, j = 0 to 4) *1
EWCR = 0x00010; /* 2 wait cycles (1ϕ + 2ϕ) */</td><td>CSON</td><td>CS0WCR2.BIT.CSON = 0; *2, *1</td></tr>
<tr><td>WDON</td><td>CS0WCR2.BIT.WDON = 1; *2, *1</td></tr>
<tr><td>WRON</td><td>CS0WCR2.BIT.WRON = 1; *2</td></tr>
<tr><td>WDOFF</td><td>CS0WCR2.BIT.WDOFF = 1; *3</td></tr>
<tr><td>CSWWAIT</td><td>CS0WCR1.BIT.CSWWAIT = 2; *2</td></tr>
<tr><td>EWCRi6</td><td>EWCRi6 = 0; /* No recovery cycles */</td><td>CSWOFF</td><td>CS0WCR2.BIT.CSWOFF = 1; *3</td></tr>
</table>

Notes: 1. Set one of the following clock cycle combinations: “1ϕ + 1ϕ”, “1ϕ + 2ϕ”, “1ϕ + 3ϕ”, “1ϕ + 4ϕ”, “1ϕ + 5ϕ”, “1ϕ + 6ϕ”, “2ϕ + 2ϕ”, “2ϕ + 3ϕ”, “2ϕ + 4ϕ”, “2ϕ + 5ϕ”, “3ϕ + 3ϕ”, “3ϕ + 4ϕ”, “3ϕ + 5ϕ”, and “3ϕ + 6ϕ”

2. Satisfy the following conditions: 1 ≤ CSnWCR2.WDON bit ≤ CSnWCR2.WRON bit ≤ CSnWCR1.CSWWAIT bit, and CSnWCR2.CSON bit ≤ CSnWCR2.WRON bit ≤ CSnWCR1.CSWWAIT bit.

3. Satisfy the following condition: CSnWCR2.WDOFF bit ≤ CSnWCR2.CSWOFF bit.

Table 1.10 Differences in the External Bus Register Settings Between the R32C and the RX (During a Write Operation)

<table>
<tr><th colspan="2">R32C (R32C/118)</th><th colspan="2">RX (RX660)</th></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
<tr><td>ESUW</td><td rowspan="3">ESUW = 0x00; /* 0 × MPY(1) + 1 (1 cycles) */</td><td>CSON</td><td>CS0WCR2.BIT.CSON = 0; *1, *1</td></tr>
<tr><td rowspan="3">EWW</td><td>WDON</td><td>CS0WCR2.BIT.WDON = 1; *1</td></tr>
<tr><td>WRON</td><td>CS0WCR2.BIT.WRON = 1; *1</td></tr>
<tr><td rowspan="3">EWW = 0x01; /* 2 × MPY(1) - 0.5 (1.5 cycles) */</td><td>WDOFF</td><td>CS0WCR2.BIT.WDOFF = 1; *2</td></tr>
<tr><td>CSWWAIT</td><td>CS0WCR1.BIT.CSWWAIT = 2; *1</td></tr>
<tr><td>CSWOFF</td><td>CS0WCR2.BIT.CSWOFF = 1; *2</td></tr>
<tr><td>MPY</td><td>MPY = 0x00; /* No multiplication (1x) */</td><td colspan="2">—</td></tr>
</table>

Notes: 1. Satisfy the following conditions: 1 ≤ CSnWCR2.WDON bit ≤ CSnWCR2.WRON bit ≤ CSnWCR1.CSWWAIT bit, and CSnWCR2.CSON bit ≤ CSnWCR2.WRON bit ≤ CSnWCR1.CSWWAIT bit.

2. Satisfy the following condition: CSnWCR2.WDOFF bit ≤ CSnWCR2.CSWOFF bit.

1.2.3 Recovery Cycles

The differences in the recovery cycle specifications are that in the M32C Family, recovery cycles are inserted before CS is negated, while in the RX Family, recovery cycles are inserted after CS is negated.

In the RX Family, 1 to 15 recovery cycles can be inserted. The condition for inserting a recovery cycle can be selected from the following:

- After a read access, the same external bus area is read accessed.
- After a read access, the same external bus area is write accessed.
- After a read access, a different external bus area is read accessed.
- After a read access, a different external bus area is write accessed.
- After a write access, the same external bus area is read accessed.
- After a write access, the same external bus area is write accessed.
- After a write access, a different external bus area is read accessed.
- After a write access, a different external bus area is write accessed

For the M32C, one recovery cycle can be inserted. When a recovery cycle is inserted, output for the data bus, address bus, and CS is extended.

For the R32C, no recovery cycle can be inserted.

2. Operating Modes Available

Table 2.1 shows Operating Modes Available for the External Bus.

Table 2.1 Operating Modes Available for the External Bus

No.	Operating Example	M32C/R32C (M32C/87, R32C/118)	RX (RX660)
		Mode	Mode
1	A program on the on-chip ROM reads data from external memory.	Memory expansion mode	On-chip ROM enabled extended mode
2	A program on external memory reads data from the external memory.	Microprocessor mode	On-chip ROM disabled extended mode

2.1 Device Operating Modes

This section describes the operating mode settings on each device. Table 2.2 shows Mode Entry Settings in the RX. Table 2.3 shows Mode Entry Settings in the M32C. Table 2.4 shows Mode Entry Settings in the R32C.

The RX has the following modes: single-chip mode, on-chip ROM enabled extended mode, and on-chip ROM disabled extended mode. The mode is selected by specifying the program settings.

The M32C has the following modes: single-chip mode, memory expansion mode, microprocessor mode, and boot mode. The mode is selected by specifying the processor mode selection pin and program settings.

The R32C Family has three processor modes: single-chip mode, memory expansion mode, and microprocessor mode. The mode is selected by specifying the processor mode selection pin and program settings.

Table 2.2 Mode Entry Settings in the RX

Mode	Program Settings
Single-chip mode	N/A
On-chip ROM enabled extended mode	Set the SYSCR0.ROME bit to 0b, and the SYSCR0.EXBE bit to 1b
On-chip ROM disabled extended mode	Set the SYSCR0.ROME bit to 1b, and the SYSCR0.EXBE bit to 1b *1

Note: 1. Specify these register settings in an area other than the external memory and ROM.

Unlike the processor mode of the M32C/R32C, the on-chip ROM disabled extended mode requires a program to transition to the mode on the on-chip ROM.

Table 2.3 Mode Entry Settings in the M32C

Mode	Mode Pin		Program Settings
	CNVSS	$\overline{\text{EPM}}$	
Single-chip mode	Low	—	N/A (Activated when bits PM01 and PM00 are set to 00b.)
Memory expansion mode	High	—	Bits PM01 and PM00 are set to 01b.
Microprocessor mode	High	High	N/A (Activated when bits PM01 and PM00 are set to 11b.)
Boot mode	High	Low	N/A *1

Note: 1. A special setup procedure must be performed.

Table 2.4 Mode Entry Settings in the R32C

Mode	Mode Pin (CNVSS)	Program Settings
Single-chip mode	Low	N/A (Activated when bits PM01 and PM00 are set to 00b.)
Memory expansion mode	High	Bits PM01 and PM00 are set to 01b.
Microprocessor mode	High	N/A (Activated when bits PM01 and PM00 are set to 11b.)

3. Differences in Operating Modes

Table 3.1 shows Functional Differences in the Operating Modes That Use the External Memory and On-Chip Memory. Table 3.2 shows Functional Differences in the Operating Modes That Use Only the External Memory.

Table 3.1 Functional Differences in the Operating Modes That Use the External Memory and On-Chip Memory

Item	M32C (M32C/87)	R32C (R32C/118)	RX (RX660)
	Memory expansion mode	**Memory expansion mode**	**On-chip ROM enabled extended mode**
Access area	SFRs, internal RAM, internal ROM, external areas	SFRs, internal RAM, internal ROM, external areas	I/O registers, on-chip RAM, on-chip ROM, external areas
External memory area	Addresses 010000h to F00000h (CS0/1/2/3) *1	0008 0000h to FFDF FFFFh (CS0/1/2/3) *2	05E0 0000h to 05FF FFFFh (CS3) 06E0 0000h to 06FF FFFFh (CS2) 07E0 0000h to 07FF FFFFh (CS1)

Notes: 1. There is an area that cannot be used in external area mode. External area mode can be set by using bits PM10 and PM11.
2. Addresses in the range from 0200 0000h to FDFF FFFFh cannot be used.

Table 3.2 Functional Differences in the Operating Modes That Use Only the External Memory

Item	M32C (M32C/87)	R32C (R32C/118)	RX (RX660)
	Microprocessor mode	**Microprocessor mode**	**On-chip ROM disabled extended mode**
Access area	SFRs, internal RAM, external areas	SFRs, internal RAM, external areas	I/O registers, on-chip RAM, on-chip ROM *1, external areas
External memory area	Addresses 010000h to FFFFFFh (CS0/1/2/3) *2	0008 0000h to FFFF FFFFh (CS0/1/2/3) *3	05E0 0000h to 05FF FFFFh (CS3) 06E0 0000h to 06FF FFFFh (CS2) 07E0 0000h to 07FF FFFFh (CS1) FFE0 0000h to FFFF FFFFh (CS0)

Notes: 1. Enabled at startup only
2. There is an area that cannot be used in external area mode. External area mode can be set by using bits PM10 and PM11.
3. Addresses in the range from 0200 0000h to FDFF FFFFh cannot be used.

4. Appendix

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

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

4.1.1 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 4.1 shows Comparison of Conditions for Interrupt Generation Between the M32C/R32C and the RX.

Table 4.1 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.

4.1.2 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.

4.1.3 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 4.2 shows Comparison of I/O Settings for Peripheral Function Pins Between the M32C and the RX.
Table 4.3 shows Comparison of I/O Settings for Peripheral Function Pins Between the R32C and the RX.

Table 4.2 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 4.3 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.

4.2 I/O Register Macros

Macro definitions listed in Table 4.4 can be found in the RX I/O register definitions (iodefine.h).

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

Table 4.4 shows Macro Usage Examples.

Table 4.4 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.

4.3 Intrinsic Functions

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

Table 4.5 shows Examples of Differences in the Settings of Control Registers and Descriptions of Special Instructions Between the M32C/R32C and the RX.

Table 4.5 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.

5. 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/.