RX Family, M16C Family

Migrating From the M16C Family to the RX Family: External Bus

R01AN2100EJ0100
Rev. 1.00
Sep. 1, 2014

Abstract

This document describes migrating the external bus in the M16C Family to the RX Family.

Products

RX Family

M16C Family

As an example of migrating from the M16C to the RX, the explanation in this document uses the RX210 Group in the RX Family and the M16C/65C Group in the M16C Family. When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Differences in Terminology

Item	RX Family	M16C Family
Operating mode to access the external memory device	● Internal ROM enabled expansion mode ● Internal ROM disabled expansion mode	● Memory expansion mode ● Microprocessor mode
External memory area	External address space CSn area	External area
Peripheral function registers	I/O registers	Special function registers (SFRs)

Contents

1. Differences in the Functions of the External Bus

Table 1.1 lists the Differences in the Functions of the External Bus.

Table 1.1 Differences in the Functions of the External Bus

Item	RX (RX210)	M16C (M16C/65C)
Memory size	16 MB	1 MB/4 MB
Bus type	Separate bus/multiplexed bus	Separate bus/multiplexed bus
Data bus width	8-bit/16-bit	8-bit/16-bit
Data bus width setting	Set using the CSn control register (n = 0 to 3)	Set using the BYTE pin
Address buses	8 to 24 buses selectable	12, 16, or 20 buses selectable
Chip select outputs	4	4
Write access mode	• 1-write strobe mode • Byte strobe mode	Write signal combinations • $\overline{BHE}/\overline{WR}$ • $\overline{WRL}/\overline{WRH}$
Wait cycle	WAIT pin	RDY pin
Software wait	Wait for up to 31 cycles	0 to 8 waits can be inserted (selectable from 1ϕ + 1ϕ, 1ϕ + 2ϕ, 1ϕ + 3ϕ, 2ϕ + 3ϕ, 2ϕ + 4ϕ, 3ϕ + 4ϕ, 4ϕ + 5ϕ)
Recovery cycles	Maximum of 15 cycles can be inserted (select from 8 patterns)	0 to 3 cycles inserted (retain the last accessed address for address output)
Page access	Supported	N/A

From this point forward, the application note describes a separate bus.

1.1 Example of Connecting to an External Bus

Connecting to bus pins on the M16C and RX is the same. However, note that pin names for the bus control pins differ. Figure 1.1 shows Example of Bus Connection When the Bus Width is 16 Bits. Figure 1.2 shows Example of Bus Connection When the Bus Width is 8 Bits.

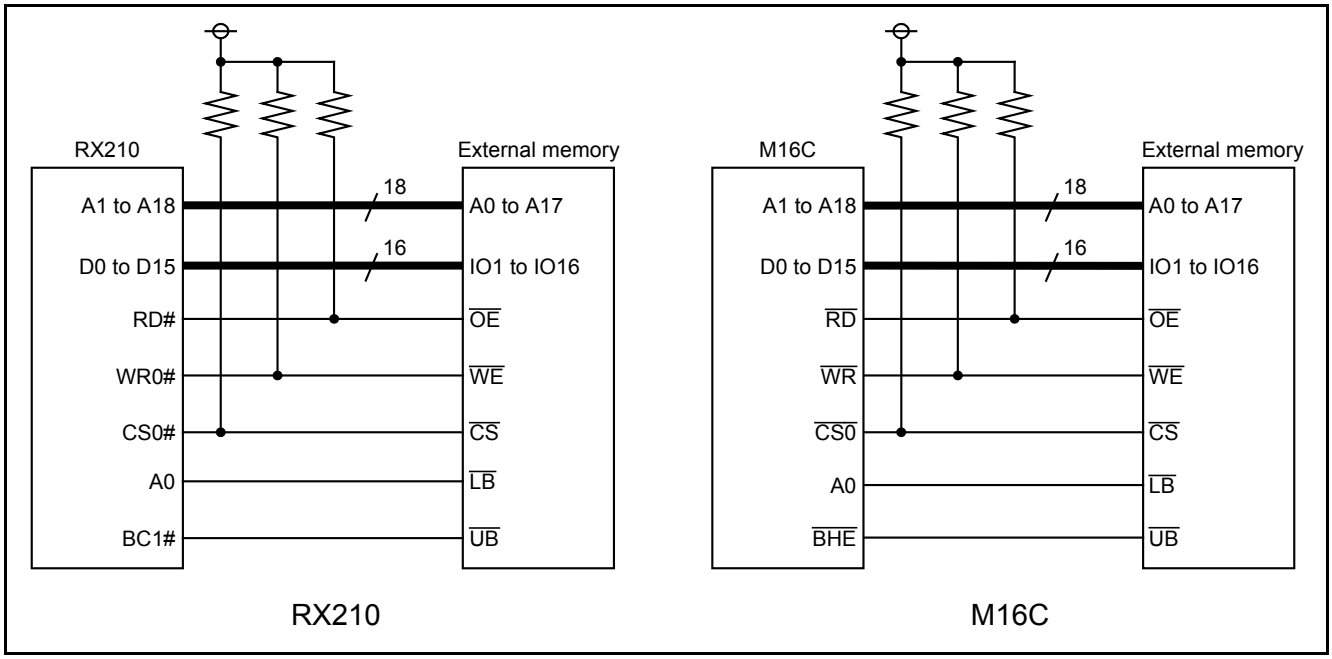


Figure 1.1 Example of Bus Connection When the Bus Width is 16 Bits

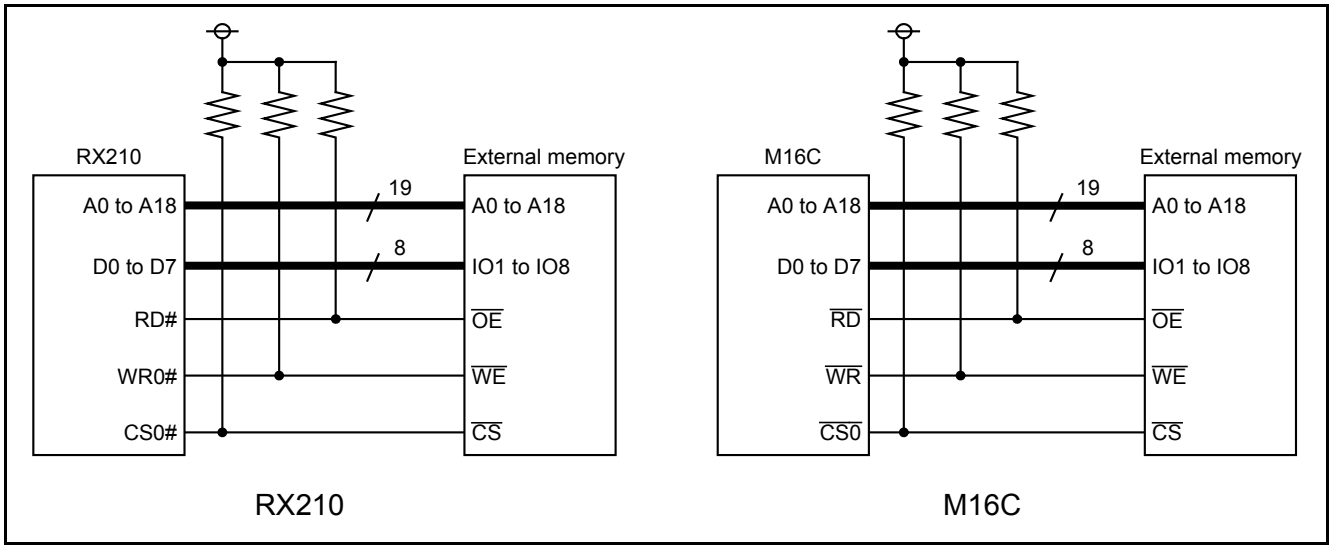


Figure 1.2 Example of Bus Connection When the Bus Width is 8 Bits

1.2 Calculating the Number of Bus Cycles

1.2.1 Calculating the Number of Bus Cycles When Reading

This section shows the differences in the bus timing (when reading) based on the timing shown in Figure 1.3.

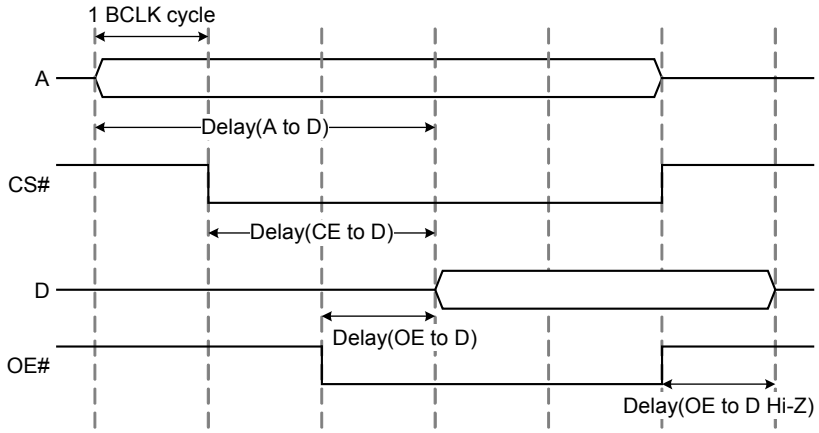
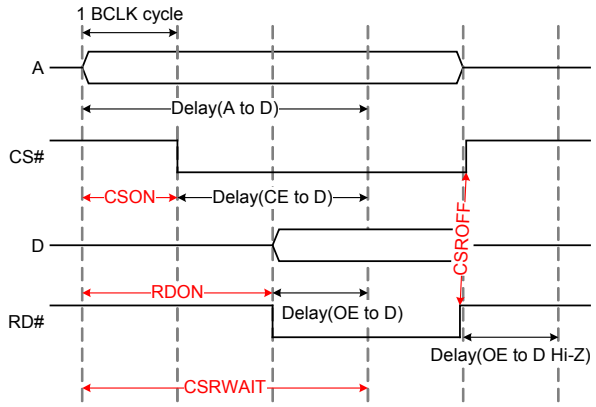
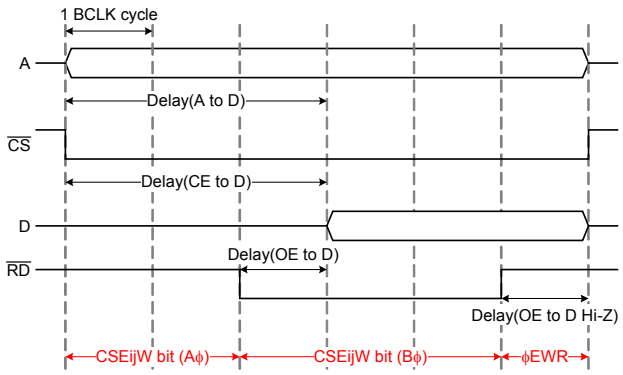


Figure 1.3 Example of Basic Bus Timing When Reading

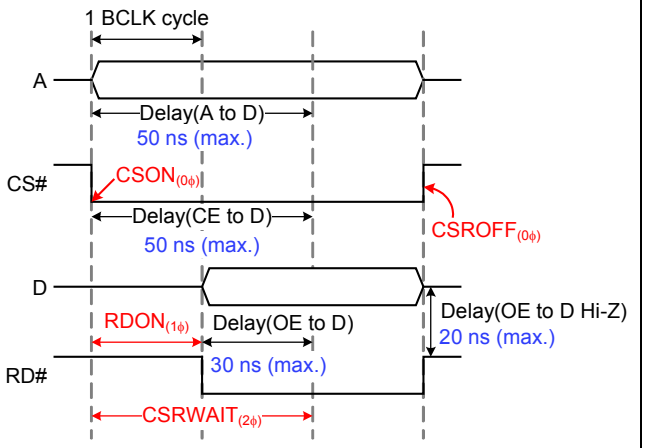
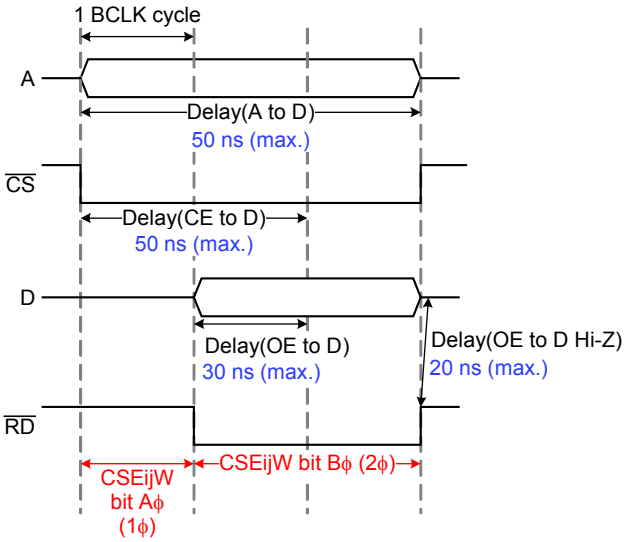
Table 1.2 Differences in the Bus Timing Settings When Reading

<table>
<tr><th colspan="2">RX (RX210)</th><th colspan="2">M16C (M16C/65C)</th></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
<tr><td>CSON</td><td>Sets the number of wait cycles to be inserted before asserting the CSn# signal</td><td colspan="2" rowspan="4">CSEijW bit (Aϕ + Bϕ) (i = 0 to 3, j = 0, 1)
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></tr>
<tr><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>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">N/A</td><td>EWR</td><td>Sets the number of recovery cycles according to the number of needed idle cycles</td></tr>
</table>

When reading the connected external memory has the following attributes, values set to the M16C and RX registers are shown in Table 1.3.

- 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 1.3 Differences in the External Bus Register Setting (When Reading and BCLK = 16 MHz)

<table>
<tr><th colspan="2">RX (RX210)</th><th colspan="2">M16C (M16C/65C)</th></tr>
<tr><td>CSON</td><td>CS0WCR2.BIT.CSON = 0; *1</td><td colspan="2" rowspan="4">CSEijW bit (Aϕ + Bϕ) *2 (i = 0 to 3, j = 0, 1)
CSE = 0x01; /* 2 waits (1ϕ + 2ϕ) */</td></tr>
<tr><td>RDON</td><td>CS0WCR2.BIT.RDON = 1; *1</td></tr>
<tr><td>CSRWAIT</td><td>CS0WCR1.BIT.CSRWAIT = 2; *1</td></tr>
<tr><td>CSROFF</td><td>CS0WCR2.BIT.CSROFF = 0;</td></tr>
<tr><td colspan="2">N/A</td><td>EWR</td><td>EWR = 0x00; / * No recovery cycles */</td></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
</table>

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

Note 2. Select from 1ϕ + 1ϕ, 1ϕ + 2ϕ, 1ϕ + 3ϕ, 2ϕ + 3ϕ, 2ϕ + 4ϕ, 3ϕ + 4ϕ, or 4ϕ + 5ϕ.

1.2.2 Calculating the Number of Bus Cycles When Writing

This section shows the differences in the bus access timing (when writing) based on the timing shown in Figure 1.4.

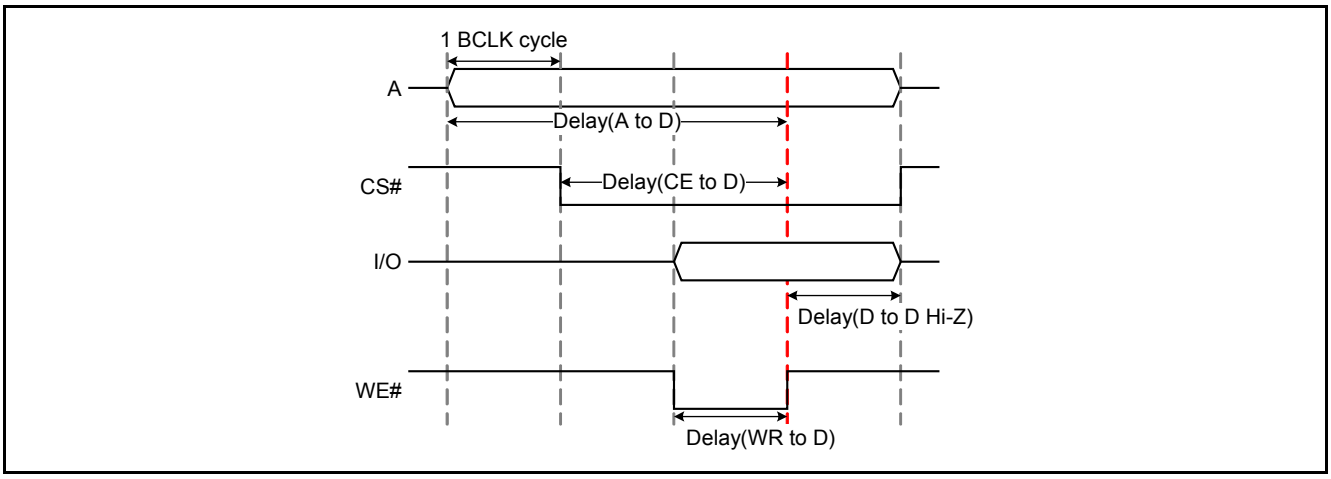
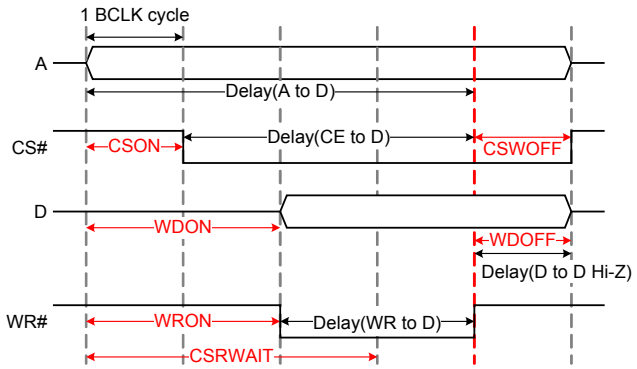
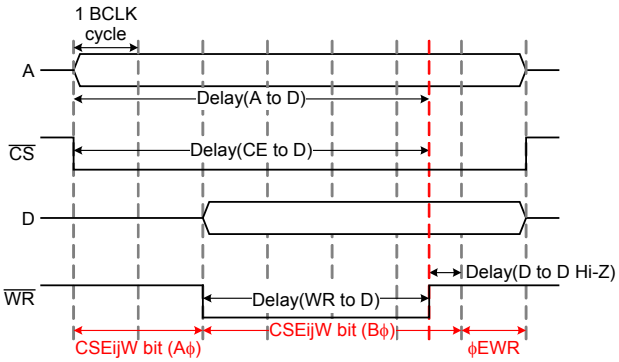


Figure 1.4 Example of Basic Bus Timing When Writing

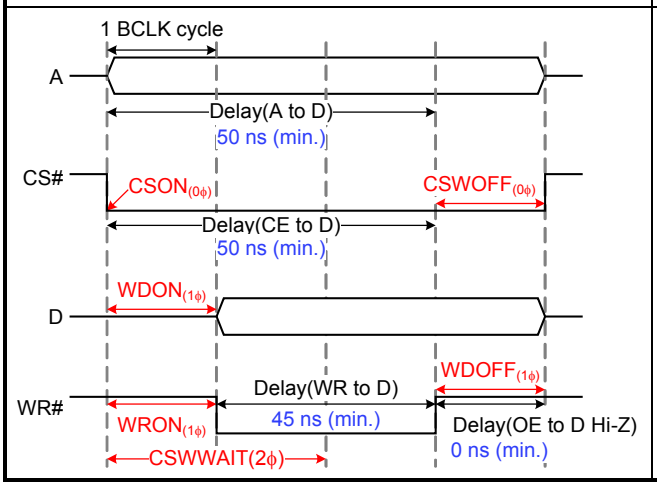
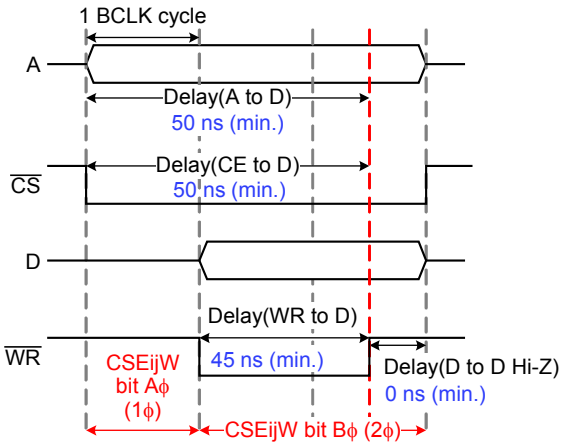
Table 1.4 Differences in the Bus Timing Settings When Writing

RX (RX210)		M16C (M16C/65C)	
			
CSON	Sets the number of wait cycles to be inserted before asserting the CSn# signal	CSEijW bit (Aϕ + Bϕ) (i = 0 to 3, j = 0, 1) Aϕ sets the number of cycles from the start of the bus access to the falling edge of the $\overline{WR}$ signal. Bϕ sets the number of cycles from falling edge of the $\overline{WR}$ signal to the rising edge.	
WDON	Sets the number of wait cycles to be inserted before outputting write data		
WRON	Sets the number of wait cycles to be inserted before asserting the WRn# signal		
WDOFF	Sets the number of cycles from negating the WRn# signal during a write access, to write data output completion		
CSWWAIT	Sets the number of cycles to be inserted in the first access of the normal read cycle		
CSWOFF	Sets the number of cycles from negating the WRn# signal during a write access, to negating the CSn# signal		
N/A		EWR	Sets the number of recovery cycles according to the number of needed idle cycles

When writing to the connected external memory has the following attributes, values set to the M16C and RX registers are shown in Table 1.5.

- 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 1.5 Differences in the External Bus Register Setting (When Writing and BCLK = 16 MHz)

<table>
<tr><th colspan="2">RX (RX210)</th><th colspan="2">M16C (M16C/65C)</th></tr>
<tr><td>CSON</td><td>CS0WCR2.BIT.CSON = 0; *1</td><td colspan="2" rowspan="6">CSEijW bit (Aϕ + Bϕ) *3 (i = 0 to 3, j = 0, 1)
CSE = 0x01; /* 2 waits (1ϕ + 2ϕ) */</td></tr>
<tr><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>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 colspan="2">N/A</td><td>EWR</td><td>EWR = 0x00; / * No recovery cycles */</td></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
</table>

Note 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.

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

Note 3. Select from 1ϕ + 1ϕ, 1ϕ + 2ϕ, 1ϕ + 3ϕ, 2ϕ + 3ϕ, 2ϕ + 4ϕ, 3ϕ + 4ϕ, or 4ϕ + 5ϕ.

1.2.3 Recovery Cycles

The differences in the recovery cycle specifications are that in the M16C 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

In the M16C Family, 1 to 3 recovery cycles can be inserted. When a recovery cycle is inserted, output for the data bus, address bus, and CS is extended.

2. Differences in Operating Modes

Table 2.1 lists the Differences in the Functions of Operating Modes Used by the External Memory and Internal Memory, and Table 2.2 lists the Differences in the Functions of Operating Modes Used Only by the External Memory.

Table 2.1 Differences in the Functions of Operating Modes That Use the External Memory and Internal Memory

Item	RX (RX210) On-chip ROM Enabled Extended Mode	M16C (M16C/65C) Memory Expansion Mode
Access area	I/O registers, on-chip RAM, on-chip ROM, external areas	SFRs, internal RAM, internal ROM, external areas
External memory area	0500 0000h to 07FF FFFFh (CS1, CS2, CS3)	Addresses 04000h to CFFFFh

Table 2.2 Differences in the Functions of Operating Modes That Use Only the External Memory

Item	RX (RX210) On-Chip ROM Disabled Extended Mode	M16C (M16C/65C) Microprocessor Mode
Access area	I/O registers, on-chip RAM, on-chip ROM (only enabled on start-up), external areas	SFRs, internal RAM, external areas
External memory area	0500 0000h to 07FF FFFFh (CS1, CS2, CS3) FF00 0000h to FFFF FFFFh (CS0)	Addresses 04000h to CFFFFh

3. Peripheral Functions Used

Table 3.1 lists the Operating Modes Used With the External Bus.

Table 3.1 Operating Modes Used With the External Bus

No.	RX Mode	M16C Mode	Operating Example	Description
1	On-chip ROM enabled extended mode	Memory expansion mode	Read the data for the external memory from program in the on-chip ROM.	Section 3.2
2	On-chip ROM disabled extended mode	Microprocessor mode	Read the data for the external memory from program in the external ROM.	Section 3.3

3.1 Device Operating Modes

The RX Family has three operating modes – single-chip mode, on-chip ROM enabled extended mode, and on-chip ROM disabled extended mode. Set the user program to enter each mode.

Table 3.2 Entering Modes in the RX Family

Mode	Program Setting
Single-chip mode	N/A
On-chip ROM enabled extended mode	Set the SYSCR0.ROME bit to 0, and the SYSCR0.EXBE bit to 1
On-chip ROM disabled extended mode	Set the SYSCR0.ROME bit to 1, and the SYSCR0.EXBE bit to 1 *2

Note 1. Set the bus width in the CSn control register (CSnCR register).
Note 2. Set this register in an area other than the external memory and ROM.

The RX Family's on-chip ROM disabled extended mode differs from the M16C Family's processor mode, and requires a program to transition to the mode on the on-chip ROM.

The M16C Family has three processor modes – single-chip mode, memory expansion mode, and microprocessor mode. The pin and program must be set to enter each processor mode.

Table 3.3 Entering Modes in the M16C Family *1

Mode	Mode Pin (CNVSS)	Program Setting
Single-chip mode	Low	N/A (when the MCU starts-up, bits PM01 and PM00 become 00b)
Memory expansion mode	Low	Set bits PM01 and PM00 to 01b *2
Microprocessor mode	High	N/A *3 (when the MCU starts-up, bits PM01 and PM00 become 11b)

Note 1. The bus width is set by the BYTE pin. Configure settings so the bus width is 8 bits when the BYTE pin is high, and 16 bits when the BYTE pin is low.
Note 2. The bus must be set. Configure settings in accordance with the external memory.
Note 3. The bus does not need to be set. Configure the bus when using a setting other than 1ϕ + 1ϕ for reading the external memory.

3.2 Differences in the Setting Procedure When Using the Program in the On-chip ROM to Read Data From the External Memory

Table 3.4 lists the Differences in Executing the Program in On-chip ROM and Reading Data From the External Memory (16-bit Bus Width).

Table 3.4 Differences in Executing the Program in On-chip ROM and Reading Data From the External Memory (16-bit Bus Width)

Step		RX Family (RX210)	M16C Family (M16C/65C)
1	Set pins	Input a high signal to the MD pin	Input a low signal to the CNVSS pin Input a low signal to the BYTE pin
2	Exit the reset state	Change the signal to the RESET pin from low to high	Change the signal to the RESET pin from low to high
3	Set clocks	Set the frequency of the operating clock.	
4	Set write access mode, external waits, and page access	BSC.CS1MOD.WORD = 0x0000;	N/A (no processing)
5	Set the bus timing (write)	BSC.CS1WCR1.BIT.CSWWAIT = 7; BSC.CS1WCR2.BIT.CSWOFF = 0; BSC.CS1WCR2.BIT.WDON = 0; BSC.CS1WCR2.BIT.WRON = 0; BSC.CS1WCR2.BIT.WDOFF = 0;	CSE = 0x00; EWC = 0x00;
6	Set the bus timing (read)	BSC.CS1WCR1.BIT.CSRWAIT = 7; BSC.CS1WCR2.BIT.CSROFF = 7; BSC.CS1WCR2.BIT.CSON = 0; BSC.CS1WCR2.BIT.RDON = 0;	
7	Set the number of recovery cycles	BSC.CS1REC.BIT.RRCV = 0; BSC.CS1REC.BIT.WRCV = 0; BSC.CSRECEN.WORD = 0x0000;	EWR = 0x00;
8	Set the bus pins	MPC.PFCSE.BIT.CS1E = 1; MPC.PFAOE0.BYTE = 0xFF; MPC.PFAOE1.BYTE = 0x0F; MPC.PFBCR0.BYTE = 0x11;	CSR = 0x01;
9	Select the bus width, endian, and bus mode, and start operation	BSC.CS1CR.WORD = 0x0001;	N/A (no processing)

10	Switch the operating mode	SYSTEM.PRCR.WORD = 0xA502;
		SYSTEM.SYSCR0.WORD = 0x5A03;
		SYSTEM.PRCR.WORD = 0xA500;
			prcr = 0x02;
			pm0 = 0x01;
			prcr = 0x00;

Table 3.5 lists the Differences in Executing the Program in On-chip ROM and Reading Data From the External Memory (8-bit Bus Width).

Table 3.5 Differences in Executing the Program in On-chip ROM and Reading Data From the External Memory (8-bit Bus Width)

Step		RX Family (RX210)	M16C Family (M16C/65C)
1	Set pins	Input a high signal to the MD pin	Input a low signal to the CNVSS pin Input a high signal to the BYTE pin
2	Exit the reset state	Change the signal to the RESET pin from low to high	Change the signal to the RESET pin from low to high
3	Set clocks	Set the frequency of the operating clock.	
4	Set write access mode, external waits, and page access	BSC.CS1MOD.WORD = 0x0000;	N/A (no processing)
5	Set the bus timing (write)	BSC.CS1WCR1.BIT.CSWWAIT = 7; BSC.CS1WCR2.BIT.CSWOFF = 0; BSC.CS1WCR2.BIT.WDON = 0; BSC.CS1WCR2.BIT.WRON = 0; BSC.CS1WCR2.BIT.WDOFF = 0;	CSE = 0x00; EWC = 0x00;
6	Set the bus timing (read)	BSC.CS1WCR1.BIT.CSRWAIT = 7; BSC.CS1WCR2.BIT.CSROFF = 7; BSC.CS1WCR2.BIT.CSON = 0; BSC.CS1WCR2.BIT.RDON = 0;	
7	Set the number of recovery cycles	BSC.CS1REC.BIT.RRCV = 0; BSC.CS1REC.BIT.WRCV = 0; BSC.CSRECEN.WORD = 0x0000;	EWR = 0x00;
8	Set the bus pins	MPC.PFCSE.BIT.CS1E = 1; MPC.PFAOE0.BYTE = 0xFF; MPC.PFAOE1.BYTE = 0x0F; MPC.PFBCR0.BYTE = 0x11;	CSR = 0x01;
9	Select the bus width, endian, and bus mode, and start operation	BSC.CS1CR.WORD = 0x0021;	N/A (no processing)

10	Switch the operating mode	SYSTEM.PRCR.WORD = 0xA503;
		SYSTEM.SYSCR0.WORD = 0x5A03;
		SYSTEM.PRCR.WORD = 0xA500;
			prcr = 0x02;
			pm0 = 0x01;
			prcr = 0x00;

3.3 Differences in the Setting Procedure When Using the Program in the External Memory to Read Data From the External Memory

Table 3.6 lists the Differences in Executing the Program in the External Memory (16-bit Bus Width) Without Using the On-chip ROM.

Table 3.6 Differences in Executing the Program in the External Memory (16-bit Bus Width) Without Using the On-chip ROM

Step		RX Family (RX210)	M16C Family (M16C/65C)
1	Set pins	Input a high signal to the MD pin	Input a high signal to the CNVSS pin Input a low signal to the BYTE pin
2	Exit the reset state	Change the signal to the RESET pin from low to high	Change the signal to the RESET pin from low to high
3	Set write access mode, external waits, and page access	BSC.CS1MOD.WORD = 0x0000;	N/A (no processing)
4	Set the bus timing (write)	BSC.CS0WCR1.BIT.CSWWAIT = 7; BSC.CS0WCR2.BIT.CSWOFF = 0; BSC.CS0WCR2.BIT.WDON = 0; BSC.CS0WCR2.BIT.WRON = 0; BSC.CS0WCR2.BIT.WDOFF = 0;	
5	Set the bus timing (read)	BSC.CS0WCR1.BIT.CSRWAIT = 7; BSC.CS0WCR2.BIT.CSROFF = 7; BSC.CS0WCR2.BIT.CSON = 0; BSC.CS0WCR2.BIT.RDON = 0;	
6	Set the number of recovery cycles	BSC.CS0REC.BIT.RRCV = 0; BSC.CS0REC.BIT.WRCV = 0; BSC.CSRECEN.WORD = 0x0000;	
7	Set the bus pins	MPC.PFCSE.BIT.CS0E = 1; MPC.PFAOE0.BYTE = 0xFF; MPC.PFAOE1.BYTE = 0x0F; MPC.PFBCR0.BYTE = 0x11;	
8	Select the bus width, endian, and bus mode, and start operation	BSC.CS0CR.WORD = 0x0001;	
9	Execute the program on the RAM	Transfer the program to be executed in steps 10 and 11 to the RAM. After transferring the program to the RAM, set the program counter in step 9.	
10	Switch the operating mode	SYSTEM.PRCR.WORD = 0xA502; SYSTEM.SYSCR0.WORD = 0x5A02; SYSTEM.PRCR.WORD = 0xA500;	
11	Jump to the program in the external memory that is to be executed first	Jump to the predetermined external memory address	Read the reset vector (FFFFCh to FFFFEh) from the external memory, and jump to the specified address
—	Set the bus to be used when accessing the external memory	Access the CS0 area using the specified number of cycles	Access the CS0 area with 1 wait (1ϕ + 1ϕ)
—	Set clocks	Set the frequency of the operating clock.	

Table 3.7 lists the Differences in Executing the Program in the External Memory (8-bit Bus Width) Without Using the On-chip ROM.

Table 3.7 Differences in Executing the Program in the External Memory (8-bit Bus Width) Without Using the On-chip ROM

Step		RX Family (RX210)	M16C Family (M16C/65C)
1	Set pins	Input a high signal to the MD pin	Input a high signal to the CNVSS pin Input a high signal to the BYTE pin
2	Exit the reset state	Change the signal to the RESET pin from low to high	Change the signal to the RESET pin from low to high
3	Set write access mode, external waits, and page access	BSC.CS1MOD.WORD = 0x0000;	N/A (no processing)
4	Set the bus timing (write)	BSC.CS0WCR1.BIT.CSWWAIT = 7; BSC.CS0WCR2.BIT.CSWOFF = 0; BSC.CS0WCR2.BIT.WDON = 0; BSC.CS0WCR2.BIT.WRON = 0; BSC.CS0WCR2.BIT.WDOFF = 0;	
5	Set the bus timing (read)	BSC.CS0WCR1.BIT.CSRWAIT = 7; BSC.CS0WCR2.BIT.CSROFF = 7; BSC.CS0WCR2.BIT.CSON = 0; BSC.CS0WCR2.BIT.RDON = 0;	
6	Set the number of recovery cycles	BSC.CS0REC.BIT.RRCV = 0; BSC.CS0REC.BIT.WRCV = 0; BSC.CSRECEN.WORD = 0x0000;	
7	Set the bus pins	MPC.PFCSE.BIT.CS0E = 1; MPC.PFAOE0.BYTE = 0xFF; MPC.PFAOE1.BYTE = 0x0F; MPC.PFBCR0.BYTE = 0x11;	
8	Select the bus width, endian, and bus mode, and start operation	BSC.CS0CR.WORD = 0x0021;	
9	Execute the program on the RAM	Transfer the program to be executed in steps 10 and 11 to the RAM. Then, set the program counter to the address for step 10 of the transferred program.	
10	Switch the operating mode	SYSTEM.PRCR.WORD = 0xA502; SYSTEM.SYSCR0.WORD = 0x5A02; SYSTEM.PRCR.WORD = 0xA500;	
11	Jump to the program in the external memory that is to be executed first	Jump to the predetermined external memory address	Read the reset vector (FFFFCh to FFFFEh) from the external memory, and jump to the specified address
—	Set the bus to be used when accessing the external memory	Access the CS0 area using the specified number of cycles	Access the CS0 area with 1 wait (1ϕ + 1ϕ)
—	Set clocks	Set the frequency of the operating clock.	

4. Appendix

4.1 Points on Migrating From the M16C Family to the RX Family

This chapter explains points on migrating from the M16C Family to the RX Family.

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 interrupt enabled.
- The interrupt request is enabled in the interrupt request enable bit for peripheral functions.

Table 4.1 lists a Comparison of Conditions for Interrupt Generation.

Table 4.1 Comparison of Conditions for Interrupt Generation

Item	RX210	M16C/65C
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 the IPR[3:0] bits.	Selected by setting bits ILVL2 to ILVL0.
Interrupt request enable	Specified by setting the IER register.	N/A
Interrupt enable for peripheral functions	Interrupts can be enabled or disabled in each peripheral function.	N/A

For more information, refer to sections Interrupt Controller (ICU), CPU, and sections for other peripheral functions used in the User's Manual: Hardware.

4.1.2 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 controlling the I/O pins in the RX Family, the following two items must be set.

- In the MPC.PFS register, select the peripheral functions that are assigned to the pins.
- In the PORTn.PMR register, select the pin function from a general I/O port or peripheral function.

Table 4.2 lists a Comparison of I/O Settings for Peripheral Function Pins.

Table 4.2 Comparison of I/O Settings for Peripheral Function Pins

Function	RX210	M16C/65C
Select the pin function	With the PFS register, I/O ports for peripheral functions can be assigned by selecting from multiple pins.	These are not available in the M16C Family. *1 When a mode is set for a peripheral function, appropriate pins are assigned as I/O pins for the peripheral function.
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.	

Note 1. Register for similar functions are available in the M32C Group and R32C Group.

For more information, refer to the Multi-Function Pin Controller (MPC) and I/O port sections in the User's Manual: Hardware.

4.1.3 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.

4.2 I/O Register Macros

Macro definitions listed in Table 4.3 can be found in the RX I/O register definitions (iodefine.h).
The readability of programs can be achieved with these macro definitions.
Table 4.3 lists examples of macros.

Table 4.3 Using Macros

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 request enabled).
IPR("module name", "bit name")	**IPR(MTU0, TGIA0)** = 0x02 ; The IPR[3:0] bits corresponding to MTU0.TGIA0 are set to 0010b (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.

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 4.4 lists examples of Descriptions of Special Instructions and Control Register Settings.

Table 4.4 Differences in the Descriptions of Special Instructions and Control Register Settings

Item	Description	
	RX	M16C
Set the I flag to 1	setpsw_i (); *1	asm("fset i");
Set the I flag to 0	clrpsw_i (); *1	asm("fclr i");
Expanded into the WAIT instruction	wait(); *1	asm("wait");
Expanded into the NOP instruction	nop(); *1	asm("nop");

Note 1. "machine.h" must be included.

5. Reference Documents

User's Manual: Hardware

RX210 Group User's Manual: Hardware Rev.1.50 (R01UH0037EJ)
M16C/65C Group User's Manual: Hardware Rev.1.10 (R01UH0093)
Refer to the corresponding User's Manual: Hardware when using products other than the RX210 Group and M16C/65C Group.
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 C/C++ Compiler Package V.1.01 User's Manual Rev.1.00 (R20UT0570EJ)
M16C Series, R8C Family C Compiler Package V5.45
C Compiler User's Manual Rev.3.00
The latest versions can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website

http://www.renesas.com

Inquiries

http://www.renesas.com/contact/

REVISION HISTORY		RX Family, M16C Family Application Note Migrating From the M16C Family to the RX Family: External Bus	
Rev.	Date	Description	
		Page	Summary
1.00	Sep. 1, 2014	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

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

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

2. Processing at Power-on

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

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

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

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

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

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

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

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

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

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

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

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0