RX Family, M16C Family

Migrating from the M16C Family to the RX Family: Clocks

R01AN1894EJ0100
Rev. 1.00
July 1, 2014

Abstract

This document describes migrating from the clocks in the M16C Family to the clocks in the RX Family.

Products

RX Family, M16C Family

As an example of migrating from the M16C Family to the RX Family, 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.

There are differences in the terminology between the M16C Family and RX Family.
The table below lists the differences in terminology related to clocks.

Differences in Terminology

Item	RX Family	M16C Family
CPU operating clock	System clock (ICLK)	CPU clock
Peripheral function operating clocks	Peripheral module clocks: PCLKA, PCLKB, PCLKC, PCLKD	Peripheral function clocks: f1, fOCO40M, fOCO-F, fOCO-S, fC32
Pins for the main clock oscillation circuit	EXTAL, XTAL	XIN, XOUT
Modes for reducing power consumption	Sleep mode All-module clock stop mode Software standby mode Deep software standby mode	Wait mode Stop mode
Registers for peripheral functions	I/O registers	SFRs

Contents

1. Differences in the Clock Generation Circuit

This chapter describes the differences in the clock generation circuit.

There are differences in the frequencies of the clocks used in the RX Family and M16C Family. Table 1.1 lists the Differences in the Frequencies of Various Clocks.

In the RX Family, settings to divide the following clocks can be done individually.

- System clock
- Peripheral module clock
- Flash interface clock
- External bus clock

In addition, the system clock, peripheral module clock, flash interface clock, and external bus clock are the same clock.

Table 1.1 Differences in the Frequencies of Various Clocks

Item		RX (RX210)	M16C (M16C/65C)
Maximum operating frequencies	System clock	50 MHz	32 MHz
	Peripheral module clock	32 MHz (50 MHz for the A/D only)	32 MHz
	External bus clock	25 MHz	32 MHz *1
Frequencies	Main clock	1 to 20 MHz	2 to 20 MHz
	Sub-clock	32.768 kHz	32.768 to 50 kHz
	PLL clock	50 to 100 MHz	10 to 32 MHz
	High-speed on-chip oscillator (HOCO)	32, 36.864, 40, 50 MHz	40 MHz
	Low-speed on-chip oscillator (LOCO)	125 kHz	125 kHz
	IWDT-dedicated on-chip oscillator	125 kHz	N/A
WDT cycle period		Approx. 128 µs to 4,096 sec *2	Approx. 16.384 ms to 33.6 sec *3
Clock after a reset is released		LOCO	LOCO
Oscillation status after a reset	Main clock	Stopped	Operating
	Sub-clock	Operating *4	Stopped
	HOCO	Operating/Stopped *5	Stopped
	LOCO	Operating	Operating
Flash interface clock		FCLK	CPU clock

Note 1. However, if the frequency goes higher than 25 MHz, the data output hold time becomes 0 ns or less (when VCC = 5 V).

Note 2. The WDT cycle period is shortest when the operating clock of the CPU is 50 MHz of the PLL clock, and longest when the sub-clock is 32.768 kHz.

Note 3. The WDT cycle period is shortest when the operating clock of the CPU is 32 MHz of the PLL clock, and longest when the LOCO clock is 125 kHz.

Note 4. The sub-clock must be stopped when not in use.

Note 5. The state of the HOCO clock after a reset can be set using the HOCO oscillation enable bit in option function select register 1 (OFS1.HOCOEN bit).

Figure 1.1 shows an Example of Selecting Various Clocks.

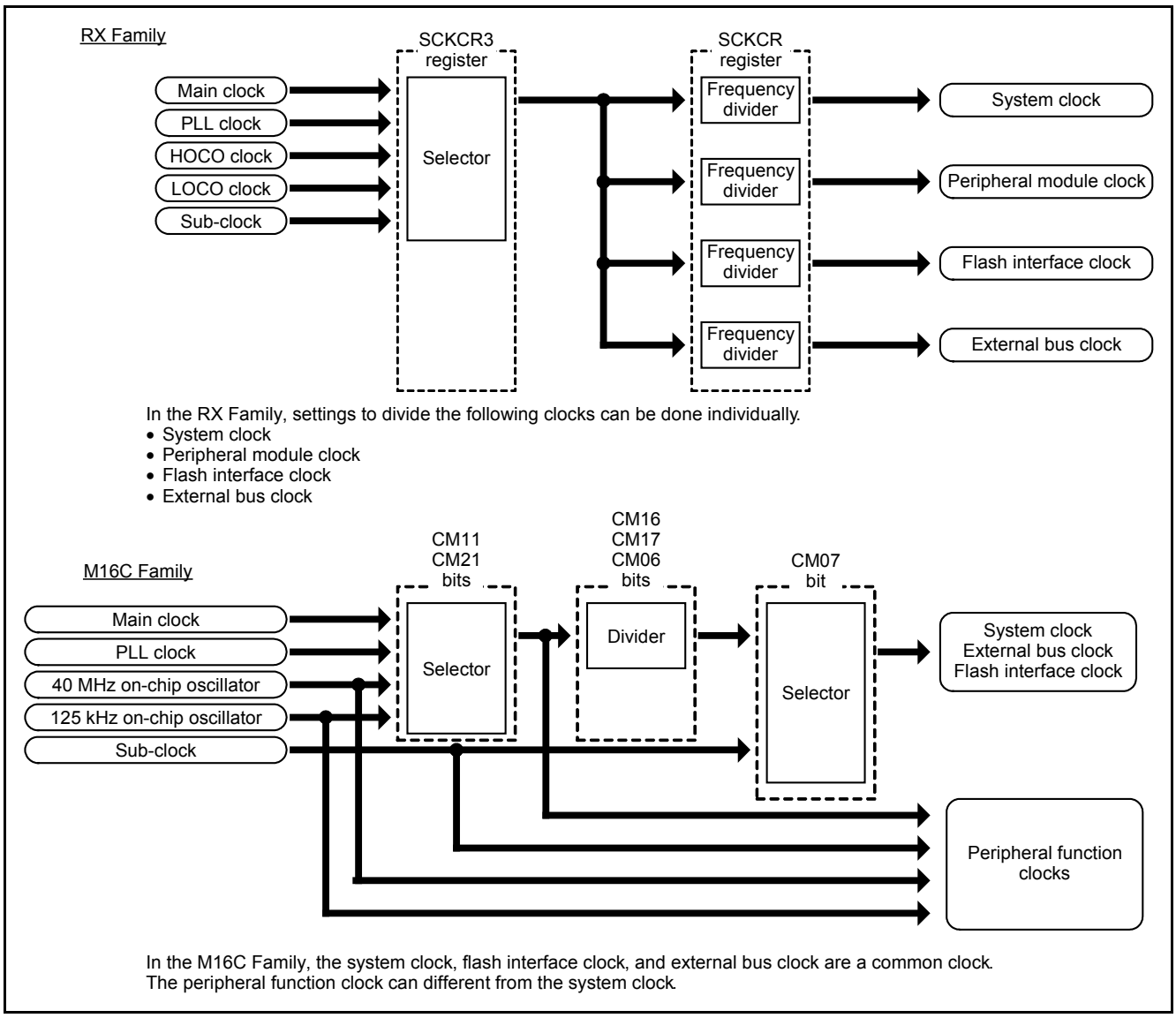


Figure 1.1 Illustration of Selecting Various Clocks

2. Differences in Clock Functions and Configuration

This chapter describes differences between the clock functions and configuration.

In the RX Family has wait control registers for adjusting the time from when clock oscillation starts to when the clock is supplied to the CPU. This will allow a stable clock to be supplied to the CPU, prevent the MCU from operating erroneously. After entering a low power consumption mode, the wait control registers function after exiting the mode.

The concept between the wait control registers and oscillation stabilization wait time is described in section 2.1.

Procedures for setting clocks for the RX Family and M16C Family are described in section 2.2.

Table 2.1 lists Details on Clock Configuration.

Table 2.1 Sections on Clock Configuration

Item	Reference
Procedure for Setting the Main Clock as the System Clock	Section 2.2.1
Procedure for setting the HOCO clock as the system clock	Section 2.2.2
Procedure for setting the sub-clock as the system clock	Section 2.2.3
Procedure for setting the PLL clock as the system clock	Section 2.2.4

2.1 Concept of the Main Clock Oscillation Stabilization Wait Time

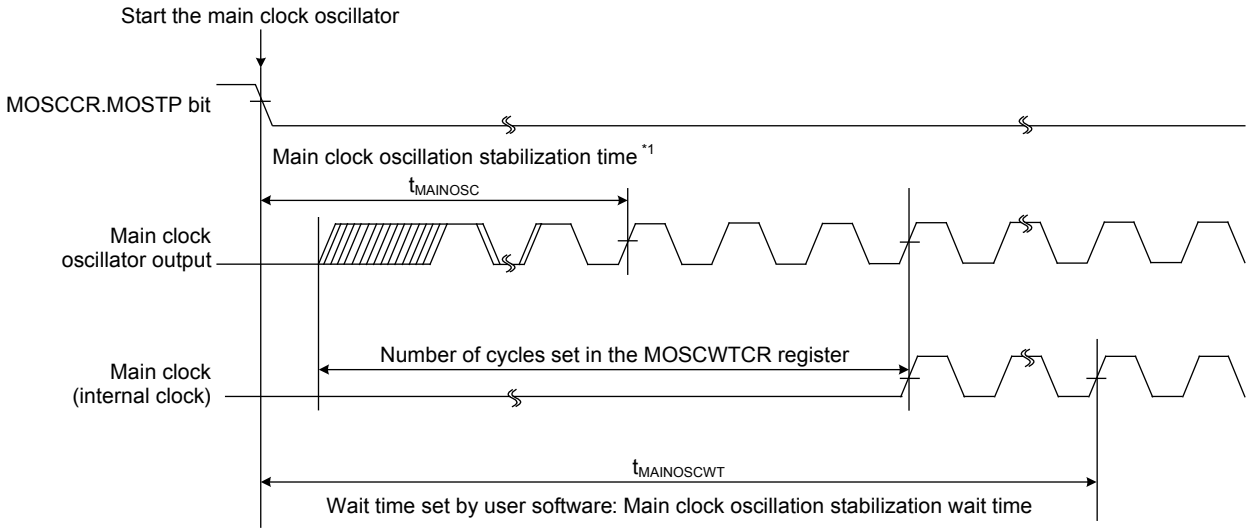
This section describes the concept of the main clock oscillation stabilization wait time in the RX Family.

A "stabilization time value that is greater than the resonator-vendor-recommended value" is set to the wait control register for the main clock (MOSCWTCR register).

The user must use software to wait for the main clock oscillation stabilization wait time. Create a software loop or the like and wait for an adequate amount of time. When using an MCU with oscillation stabilization flag registers, read the corresponding oscillation stabilization flags to determine if oscillation has stabilized.

The recommended main clock oscillation stabilization wait time is "at least twice the clock cycles set in the MOSCWTCR register".

Figure 2.1 shows the Concept of the Main Clock Oscillation Stabilization Wait Time.



Note 1. The oscillation stabilization time for the crystal resonator will differ depending on conditions like the wiring pattern and oscillation constants in the user system. The user should evaluate their system with oscillation stabilization times obtained from the resonator manufacturer.

Figure 2.1 Concept of the Main Clock Oscillation Stabilization Wait Time

2.2 Differences in the Clock Setting Procedures

This section shows the differences in the clock setting procedures after a reset is released.

The sub-clock must be stopped when it is not being used as the system clock or realtime clock (RTC). Refer to the Initial Settings application note of each MCU group for details on the processing to stop the sub-clock and the clock initial settings.

2.2.1 Procedure for Setting the Main Clock as the System Clock

This section shows an example of the procedure for setting the main clock as the system clock when the sub-clock is not being used as the system clock or RTC. Table 2.2 lists the Conditions for Setting the Main Clock as the System Clock, and Table 2.3 lists the Differences in the Procedures for Setting the Main Clock as the System Clock.

Table 2.2 Conditions for Setting the Main Clock as the System Clock

Item		Conditions
System clock		Main clock
Division	System clock	No division
	Peripheral module clock	No division
	Flash interface clock	No division
	External bus clock	No division
Main clock	Oscillator drive capability	Other than 16 to 20 MHz lead type ceramic resonator
	Oscillation source of the oscillator	Resonator
	Wait time	131,072 cycles
Sub-clock		Not used

Table 2.3 Differences in the Procedures for Setting the Main Clock as the System Clock

Step		RX (RX210)	M16C (M16C/65C)
1	Enable writing to registers	SYSTEM.PRCR.WORD = 0xA507;	prc0 = 1;
2	Set the voltage regulator control register	SYSTEM.VRCR = 0x00; *1	N/A
3	Stop the sub-clock	Processing to stop the sub-clock *2	N/A
4	Set the drive capability of the main clock oscillator	SYSTEM.MOFCR.BYTE = 0x30;	cm15 = 1;
5	Set the wait control register for the main clock oscillator	SYSTEM.MOSCWTCR.BYTE = 0x0D; *3	N/A

6	Operate the main clock oscillator	SYSTEM.MOSCCR.BYTE = 0x00;
		while (0x00 != SYSTEM.MOSCCR.BYTE)
		{
		}
			cm05 = 0;

7	Wait for oscillation stabilization	Wait the main clock oscillation stabilization wait time *4	Wait for main clock oscillation stabilization

8	Set the division	SYSTEM.SCKCR.LONG = 0x00801010;
		while (0x00801010 != SYSTEM.SCKCR.LONG)
		{
		}
			N/A

9	Switch the system clock	SYSTEM.SCKCR3.WORD = 0x0200;
		while (0x0200 != SYSTEM.SCKCR3.WORD)
		{
		}
			cm06 = 1;
			cm11 = 0;
			cm21 = 0;
			cm07 = 0;

10	Set the division	N/A	cm17 = 1;
			cm06 = 0;
			cm1 = cm1 ^(0xC0);
			cm16 = 0;

11	Disable writing to registers	SYSTEM.PRCR.WORD = 0xA500;	prc0 = 0;

Note 1. This step is unnecessary if the register is set once after a reset.

Note 2. Refer to the Initial Settings application note of each MCU group for details.

Note 3. Set a value that is equal to or higher than the main clock oscillation stabilization time recommended by the resonator manufacturer. In the example, if the main clock oscillation stabilization time is 4.2 ms for a 20 MHz resonator, then set a value so the wait time is approximately 6.55 ms.

Note 4. Use software to wait a main clock oscillation stabilization wait time that is "at least twice the clock cycles set in the MOSCWTCR register".

2.2.2 Procedure for Setting the HOCO Clock as the System Clock

This section shows an example of the procedure for setting the HOCO clock as the system clock when the sub-clock is not being used as the system clock or RTC. Table 2.4 lists the Conditions for Setting the HOCO Clock as the System Clock, and Table 2.5 lists the Differences in the Procedures for Setting the HOCO Clock as the System Clock.

Table 2.4 Conditions for Setting the HOCO Clock as the System Clock

Item		Conditions
System clock		HOCO clock
Division	System clock	No division
	Peripheral module clock	Divided by 2
	Flash interface clock	Divided by 2
	External bus clock	Divided by 2
HOCO	Frequency	40 MHz
	Wait time	7,168 cycles
	Oscillation after a reset	HOCO oscillation is disabled after a reset (OFS1.HOCOEN bit is 1)
Sub-clock		Not used

Table 2.5 Differences in the Procedures for Setting the HOCO Clock as the System Clock

Step		RX (RX210)	M16C (M16C/65C)
1	Enable writing to registers	SYSTEM.PRCR.WORD = 0xA507;	prc0 = 1;
2	Set the voltage regulator control register	SYSTEM.VRCR = 0x00; *1	N/A
3	Stop the sub-clock	Processing to stop the sub-clock *2	N/A
4	Select the HOCO frequency	SYSTEM.HOCOCR2.BYTE = 0x02	N/A
5	Set the wait control register for the HOCO	SYSTEM.HOCOWTCR2.BYTE = 0x02 *3	N/A
6	Operate the HOCO	SYSTEM.HOCOCR.BYTE = 0x00;	fra00 = 1;
7	Wait for oscillation stabilization	Wait the HOCO clock oscillation stabilization wait time (tHOCOWT) *4	Wait until 40 MHz on-chip oscillator is stable (tsu(fOCO40M))

8	Set the division	SYSTEM.SCKCR.LONG = 0x10811111;
		while (0x10811111 != SYSTEM.SCKCR.LONG)
		{
		}
			N/A

9	Switch the system clock	SYSTEM.SCKCR3.WORD = 0x0100;
		while (0x0100 != SYSTEM.SCKCR3.WORD)
		{
		}
			cm06 = 1;
			fra01 = 1;
			cm21 = 1;
			cm07 = 0;

10	Set the division	N/A	cm17 = 1;
			cm06 = 0;
			cm1 = cm1 ^(0xC0);

11	Disable writing to registers	SYSTEM.PRCR.WORD = 0xA500;	prc0 = 0;

Note 1. This step is unnecessary if the register is set once after a reset.
Note 2. Refer to the Initial Settings application note of each MCU group for details.
Note 3. If the HOCO clock frequency is 32, 36.864, or 40 MHz, then set the HSTS[3:0] bits to 0010b. If the HOCO clock frequency is 50 MHz, then set the HSTS[3:0] bits to 0011b.
Note 4. Use software to wait a HOCO oscillation stabilization wait time that is at least tHOCOWT (350 µs max.).

2.2.3 Procedure for Setting the Sub-Clock as the System Clock

This section shows an example of the procedure for setting the sub-clock as the system clock when the sub-clock is not being used as the RTC. Table 2.6 lists the Conditions for Setting the Sub-Clock as the System Clock, and Table 2.7 lists the Differences in the Procedures for Setting the Sub-Clock as the System Clock.

Table 2.6 Conditions for Setting the Sub-Clock as the System Clock

Item		Conditions
System clock		Sub-clock
Division	System clock	No division
	Peripheral module clock	No division
	Flash interface clock	No division *1
	External bus clock	No division
Sub-clock	Oscillator drive capability	Drive capability for standard CL (high drive capacity)
	Wait time	2 seconds + 2 cycles

Note 1. The flash cannot be rewritten when the sub-clock is used as the system clock.

Table 2.7 Differences in the Procedures for Setting the Sub-Clock as the System Clock

Step		RX (RX210)	M16C (M16C/65C)
1	Enable writing to registers	SYSTEM.PRCR.WORD = 0xA507;	prc0 = 1;
2	Set the voltage regulator control register	SYSTEM.VRCR = 0x00; *1	N/A
3	Set ports connected to the sub-clock as input ports	N/A	pu21 = 0 pd8_6 = 0 pd8_7 = 0
4	Wait for sub-clock oscillation to stabilize	Wait the sub-clock oscillation stabilization wait time *2	Wait the sub-clock oscillation stabilization time

5	Stop the sub-clock oscillator	SYSTEM.SOSCCR.BYTE = 0x01;
		while (0x01 != SYSTEM.SOSCCR.BYTE)
		{
		}
		RTC.RCR3.BIT.RTCEN = 0:
		for (i = 0; i < 3; i++)
		{
		dummy = RTC.RCR3.BIT.RTCEN;
		}
		while (0 != RTC.RCR3.BIT.RTCEN)
		{
		}
			N/A

6	Wait for the sub-clock to stop	Wait for five cycles of the sub-clock	N/A

7	Set the sub-clock drive capability	RTC.RCR3.BYTE = 0x0C;
		for (i = 0; i < 3; i++)
		{
		dummy = RTC.RCR3.BYTE;
		}
		while (0x0C != RTC.RCR3.BYTE)
		{
		}
			cm03 = 1

8	Set the wait control register for the sub-clock	SYSTEM.SOSCWTCR.BYTE = 0x00 *3	N/A

9	Oscillate the sub-clock	SYSTEM.SOSCCR.BYTE = 0x00;
		while (0x00 != SYSTEM.SOSCCR.BYTE)
		{
		}
			cm04 = 1

10	Wait for the sub-clock to stabilize	Wait the sub-clock oscillation stabilization wait time *2	Wait the sub-clock oscillation stabilization time

11	Set the division	SYSTEM.SCKCR.LONG = 0x00801010;
		while (0x00801010 != SYSTEM.SCKCR.LONG)
		{
		}
			N/A

12	Switch the system clock	SYSTEM.SCKCR3.WORD = 0x0300;
		while (0x0300 != SYSTEM.SCKCR3.WORD)
		{
		}
			cm16 = 1
			cm1 = cm1 ^(0xC0)
			cm06 = 1
			cm07 = 1

13	Disable writing to registers	SYSTEM.PRCR.WORD = 0xA500;	prc0 = 0;

Note 1. This step is unnecessary if the register is set once after a reset.

Note 2. Use software to wait a sub-clock oscillation stabilization wait time that is "at least twice the clock cycles set in the SOSCWTCR register".

Note 3. Set a value that is equal to or higher than the sub-clock oscillation stabilization time recommended by the resonator manufacturer. In the example, if the sub-clock oscillation stabilization time is 1.3 seconds, then set a value so the wait time is 2 seconds + approx. 61 µs.

2.2.4 Procedure for Setting the PLL Clock as the System Clock

This section shows an example of the procedure for setting the PLL clock as the system clock when the sub-clock is not being used as the system clock or RTC. Table 2.8 lists the Conditions for Setting the PLL Clock as the System Clock, and Table 2.9 lists the Differences in the Procedures for Setting the PLL Clock as the System Clock.

Table 2.8 Conditions for Setting the PLL Clock as the System Clock

Item		Conditions
System clock		PLL clock
Division	System clock	Divided by 2
	Peripheral module clock	Divided by 4
	Flash interface clock	Divided by 4
	External bus clock	Divided by 4
Main clock	Oscillator drive capability	Other than 16 to 20 MHz lead type ceramic resonator
	Oscillation source of the oscillator	Resonator
	Wait time	131,073 cycles
PLL clock	PLL input division ratio/multiplication factor	Divided by 2 and multiplied by 8
	Wait time	32,768 cycles
Sub-clock		Not used

Table 2.9 Differences in the Procedures for Setting the PLL Clock as the System Clock

Step		RX (RX210)	M16C (M16C/65C)
1	Enable writing to registers	SYSTEM.PRCR.WORD = 0xA507;	prcr = 0x01
2	Set the voltage regulator control register	SYSTEM.VRCR = 0x00; *1	N/A
3	Stop the sub-clock	Processing to stop the sub-clock *2	N/A
4	Set the drive capability for the main clock oscillator	SYSTEM.MOFCR.BYTE = 0x30;	cm15 = 1;
5	Set the wait control register for the main clock oscillator	SYSTEM.MOSCWTCR.BYTE = 0x0D; *3	N/A

6	Operate the main clock oscillator	SYSTEM.MOSCCR.BYTE = 0x00;
		while (0x00 != SYSTEM.MOSCCR.BYTE)
		{
		}
			cm05 = 0;

7	Wait for oscillation to stabilize	Wait the main clock oscillation stabilization wait time *4	Wait for the main clock oscillation to stabilize
8	Set the PLL input frequency division ratio and the frequency multiplication factor	SYSTEM.PLLCR.WORD = 0x0701; *5	plc0 = 0x14; *5
9	Set the wait control register for the PLL clock oscillator	SYSTEM.PLLWTCR.BYTE = 0x09; *6	N/A
10	Operate the PLL	SYSTEM.PLLCR2.BYTE = 0x00	plc07 = 1;
11	Wait for oscillation to stabilize	Wait the PLL clock oscillation stabilization wait time *7	Wait (tsu(PLL) until the PLL clock stabilizes

12	Set the division	SYSTEM.SCKCR.LONG = 0x21821212;
		while (0x21821212 != SYSTEM.SCKCR.LONG)
		{
		}
			N/A

13	Switch the system clock	SYSTEM.SCKCR3.WORD = 0x0400;
		while (0x0400 != SYSTEM.SCKCR3.WORD)
		{
		}
			cm06 = 1;
			cm11 = 1;
			cm21 = 0;
			cm07 = 0

14	Set the division	N/A	cm17 = 1;
			cm06 = 0;
			cm1 = cm1 ^(0xC0);
			cm16 = 0;

15	Disable writing to registers	SYSTEM.PRCR.WORD = 0xA500;	prc0 = 0;

Note 1. This step is unnecessary if the register is set once after a reset.
Note 2. Refer to the Initial Settings application note of each MCU group for details.
Note 3. Set a value that is equal to or higher than the main clock oscillation stabilization time recommended by the resonator manufacturer. In the example, if the main clock oscillation stabilization time is 4.2 ms for a 20 MHz resonator, then set a value so the wait time is approximately 6.55 ms.
Note 4. Use software to wait a main clock oscillation stabilization wait time that is "at least twice the clock cycles set in the MOSCWTCR register".
Note 5. In the RX Family, set the PLL input frequency division ratio select bits so the frequency after division is 4 to 12.5 MHz, and set the frequency multiplication factor select bits so the frequency after multiplication is 50 to 100 MHz. In the M16C Family, set the reference frequency counter set bits so the frequency after division is 2 to 5 MHz, and set the PLL multiplying factor select bits to the frequency after multiplication is within f(PLL).
Note 6. A value of at least tPLL1 (500 µs max.) must be set. In the example, if the PLL clock is 80 MHz, set a value to wait approx. 819.2 µs.
Note 7. Use software to wait a PLL clock oscillation stabilization wait time that is at least tPLLWT1 (1.5 ms min.).

3. Differences in Low Power Consumption Modes

The RX Family has several low power consumption modes to reduce power consumption. This chapter describes the differences in the RX210 modes (sleep mode, software standby mode, all-module clock stop mode, and deep software standby mode) and the M16C Family modes (wait mode and stop mode).

Table 3.1 lists the Differences in Low Power Consumption Modes.

3.1 Sleep Mode

The RX Family sleep mode is similar to the M16C Family wait mode in that the CPU is stopped. The RX Family has a function for automatically switching the clock source when exiting sleep mode.

3.2 Software Standby Mode

The RX Family software standby mode is similar to the M16C Family stop mode in that the CPU, peripheral functions, and oscillators are stopped.

3.3 All-Module Clock Stop Mode

In this mode, the CPU and peripheral functions are stopped. Use the module stop function to stop peripheral functions before entering this mode. In normal operation mode, peripheral functions can be stopped independently. Refer to section 7.1.3 for details.

3.4 Deep Software Standby Mode

In this mode, the power supplied to the CPU, peripheral functions, and oscillators is stopped. By stopping the power supply, power consumption can be greatly reduced. In addition, the RTC can be operated in this mode. The MCU must be reset to exit this mode.

Table 3.1 Differences in Low Power Consumption Modes

Item	RX (RX210)				M16C (M16C/65C)	
Power consumption	More → Less				More → Less	
Mode	Sleep mode	All-module clock stop mode	Software standby mode	Deep software standby mode	Wait mode	Stop mode
CPU	Stopped	Stopped	Stopped	Stopped *1	Stopped	Stopped
Main clock Other clocks	Operating	Operating	Stopped	Stopped	Operating	Stopped
Sub-clock	Operating	Operating	Operating	Operating	Operating	Stopped
RAM	Operating	Stopped	Stopped	Stopped *1	Operating	Operating
Flash memory	Operating	Stopped	Stopped	Stopped	Operating	Stopped
Watchdog timer (WDT)	Stopped	Stopped	Stopped	Stopped *1	Operating	Stopped
Independent watchdog timer (IWDT)	Operating	Operating	Operating	Stopped *1	N/A	N/A
RTC	Operating	Operating	Operating	Operating	Operating	Stopped
8-bit timer	Operating	Operating	Stopped	Stopped *1	N/A	N/A
Other peripheral functions	Operating	Stopped	Stopped	Stopped *1	Operating	Stopped
Outline	This mode stops the CPU.	This mode stops the CPU and peripheral functions (with some exceptions).	This mode stops the CPU, peripheral functions, and oscillators (only the sub-clock, IWDT, and RTC can be operated).	This mode stops supplying power to all functions (only the sub-clock and RTC can be operated).	This mode stops the CPU.	This mode stops the CPU, peripheral functions, and oscillators.

Note 1. In order to stop supplying power, register values for the CPU and internal peripheral functions (excluding the RTC alarm, RTC period, SCL-DS, and SDA-DS) become undefined, and data in the RAM becomes undefined.

4. Information Regarding the Function for Lower Operating Power Consumption

The RX Family is equipped with a function for lower operating power consumption to reduce power consumption while the MCU is operating.

The function for lower operating power consumption has a high-speed operating mode, middle-speed operating modes, and low-speed operating modes. The slower the mode, the more power consumption can be reduced. As the power supply voltage, clocks, and frequencies differ for each mode, select an appropriate mode based on the conditions of use. When slowing down and speeding up clocks, the procedure for changing the operating power control mode differs.

Slowing the clock to reduce CPU power consumption

(1) Set the clock source and switch the division ratio.

(2) Change the operating power control mode.

Speeding up the clock to quicken CPU operation

(1) Change the operating power control mode.

(2) Set the clock source and switch the division ratio.

Although the M16C Family uses similar terminology for its normal operating mode (high-speed mode, medium-speed mode, and low-speed mode), these modes specify differences in the operating clock of the CPU, and are therefore different from the RX Family.

5. Information Regarding the Clock Frequency Accuracy Measurement Circuit

The RX Family is equipped with functions for monitoring the clock frequencies and detecting abnormal frequencies. The RX210 is equipped with a clock frequency accuracy measurement circuit (CAC).

The CAC monitors the clock frequency based on a reference signal input to the MCU externally or another clock source, and generates interrupts when the frequency is outside the set range.

For example, when monitoring the sub-clock frequency by the on-chip oscillator, if an abnormal frequency is detected and the sub-clock stops, an interrupt can be generated.

6. Information Regarding the Oscillation Stop Detection Function

This chapter describes the differences in the clock oscillation stop function. There are differences in some functions, such as the clocks after oscillation stop is detected. Table 6.1 lists the Differences in the Oscillation Stop Detection Function.

Table 6.1 Differences in the Oscillation Stop Detection Function

Clocks When Oscillation is Stopped	Clocks After an Oscillation Stop is Detected	
	RX (RX210)	M16C (M16C/65C)
Main clock	LOCO	Low-speed on-chip oscillator
Sub-clock	No change (remains as sub-clock)	
LOCO clock	No change (remains as LOCO)	
HOCO clock	No change (remains as HOCO)	
PLL clock	No change (remains as PLL clock *1)	No change (remains as PLL clock *1)

Note 1. However, the frequency becomes the self-oscillation frequency.

7. Information on Accessing I/O Registers

This chapter describes accessing the I/O registers in the RX Family.

When writing to I/O registers in the RX Family, the CPU does not wait for a write access to be complete before executing subsequent instructions. In addition, when accessing I/O registers, the operating clock for peripheral functions is used. For this reason, since clocks for peripheral functions of the accessed I/O registers are sometimes slower than the CPU clock, subsequent instructions may be executed before the value can be written to the I/O register.

If subsequent instructions must be executed after the change to an I/O register setting is reflected, after clearing an interrupt request enable bit (ICU.IERn.IENj bit), when executing subsequent instructions while the interrupt request is disabled, or when a WAIT instruction is executed after the preprocessing to enter a low power consumption mode, wait for the write access to be complete before executing subsequent instructions.

Table 7.1 lists the Instructions That Wait for the I/O Register Write Value to be Reflected.

Table 7.1 Instructions That Wait for the I/O Register Write Value to be Reflected

Step		Instruction Example
1	Write to I/O registers	MOV.L #SFR_ADDR, R1 MOV.B #SFR_DATA, [R1] CMP [R1].UB, R1
2	Values written to I/O registers are read to general-purpose registers	
3	Use the values read to perform calculations	
4	Execute subsequent instructions	

8. Chapters Associated With the RX User's Manual: Hardware (UMH)

When migrating from the M16C Family to the RX Family, refer to the following chapters in the UMH.

- I/O registers
- Clock generation circuit
- Low power consumption
- Register write protection function
- RTC

9. Appendix

9.1 Points on Migrating from the M16C Family to the RX Family

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

9.1.1 Interrupts

In the RX Family, if an interrupt request is received when all of the following conditions are met, the interrupt can be accepted.

- 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 by the interrupt request enable bits for the peripheral function.

Table 9.1 lists the Comparison of Conditions for Generating an Interrupt.

Table 9.1 Comparison of Conditions for Generating an Interrupt

Item	RX	M16C
I flag	When the I flag is set to 1 (enabled), the maskable interrupts are 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 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	Interrupt enable or disable can be specified 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 UMH.

9.1.2 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 controlling the input and output 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 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.

Table 9.2 lists Comparison of I/O Settings for Peripheral Function Pins.

Table 9.2 Comparison of I/O Settings for Peripheral Function Pins

Function	RX (RX210)	M16C (M16C/65C)
Select the pin function	A pin to which a peripheral I/O is assigned can be selected from multiple pins. The PFS register is used to select a peripheral function I/O that is assigned to a pin used.	These are not available in the M16C Group. *1 When a mode of a peripheral function is selected, an appropriate pin is assigned as an I/O pin for the peripheral function.
Switch between general I/O port and peripheral function	With the PMR register, the pin function can be selected as a general I/O port or a peripheral function.	

Note 1. The M32C Group and R32C Group have registers to assign a peripheral function to a pin.

For more information, refer to the Multi-Function Pin Controller (MPC) and I/O port sections in the UMH.

9.1.3 Module Stop Function

The RX Family has the ability to stop peripheral modules individually. By placing unused peripheral modules in 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 read or write accessed.

For more information, refer to the Low Power Consumption section in the UMH.

9.2 I/O Register Macros

Macro definitions listed in Table 9.3 can be found in the RX I/O register definitions (iodefine.h).

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

Table 9.3 lists examples of Macros.

Table 9.3 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).
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 register.

9.3 Intrinsic Functions

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

Table 9.4 lists examples of Descriptions of Special Instructions and Control Register Settings.

Table 9.4 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.

10. 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 UMH 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: Clocks

Rev.	Date	Description	
		Page	Summary
1.00	July 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