RX Family and M32C/R32C Series

Guide for Migration from the M32C/R32C to the RX: Clocks

Abstract

This document describes migration from the clocks in the M32C/R32C Series MCU to the clocks 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 clocks.

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

Item	M32C/R32C Series	RX Family
CPU operating clock	CPU clock	System clock (ICLK)
Peripheral function operating clocks	Peripheral function clocks: fC, fC32, fOCO40M, fOCO-F, fOCO-S, f1	Peripheral module clocks: PCLKA, PCLKB, PCLKD
Pins for the main clock oscillation circuit	XIN, XOUT	EXTAL, XTAL
Modes for reducing power consumption	Wait mode Stop mode	Sleep mode All-module clock stop mode Software standby mode Deep software standby mode
Registers for peripheral functions	Special function registers (SFRs)	I/O registers

Contents

1. General Differences in the Clock Generation Circuit

This chapter describes the general differences in the clock generation circuit between the RX-family MCU and the M32C/R32C Series MCU.

There are differences in the frequencies of the clocks used in the RX Family MCU and the M32C/R32C Series MCU. 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.

In the RX Family, a separate division ratio can be set for the following clocks:

- System clock
- Peripheral module clock
- Flash interface clock (FlashIF clock)
- External bus clock

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

Figure 1.1 shows Illustration of Selecting Various Clocks.

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

Item		M32C (in the case of the M32C/87)	RX (in the case of the RX660)
Maximum operating frequencies	System clock	32 MHz [*1], 24 MHz [*2]	120 MHz
	Peripheral module clock	32 MHz	60 MHz
	External bus clock	32 MHz [*1], 24 MHz [*2]	60 MHz
Frequency	Main clock	0 MHz to 32 MHz [*1], 0 MHz to 24 MHz [*2]	8 MHz to 24 MHz
	Sub-clock	32.768 kHz to 50 kHz	32.768 kHz
	PLL	10 MHz to 32 MHz [*1], 10 MHz to 24 MHz [*2]	120 MHz to 240 MHz
	On-chip oscillator	1 MHz	16/18/20 MHz (HOCO), 240 kHz (LOCO)
	IWDT-dedicated on-chip oscillator	—	120 kHz
WDT cycle period		Approx. 16.384 ms to 2 sec [*1, *3] Approx. 21.845 ms to 2 sec [*2, *4]	Approx. 34.13 μs to 4,096 sec [*5]
Clock after a reset is released		Main clock	LOCO
Oscillation status after a reset	Main clock	Operating	Stopped
	Sub-clock	Stopped	Operating [*6]
	PLL	Stopped	Stopped
	On-chip oscillator	Stopped	Operating/Stopped (HOCO) [*7], Operating (LOCO)
Flash interface clock		CPU clock	FlashIF clock

Notes: 1. Under the condition when VCC1 is in the range from 4.2 to 5.5 V.
2. Under the condition when VCC1 is in the range from 3.0 to 5.5 V.
3. The WDT cycle period becomes shortest when the CPU is operating at 32 MHz with the PLL clock or the main clock. It becomes longest when the CPU is operating at 32.768 kHz with the sub-clock.
4. The WDT cycle period becomes shortest when the CPU is operating at 24 MHz with the PLL clock or the main clock. It becomes longest when the CPU is operating at 32.768 kHz with the sub-clock.
5. The WDT cycle period becomes shortest when the CPU is operating at 120 MHz with the PLL clock. It becomes longest when the CPU is operating at 32.768 kHz with the sub-clock.
6. The sub-clock must be stopped when not in use.

7. 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).

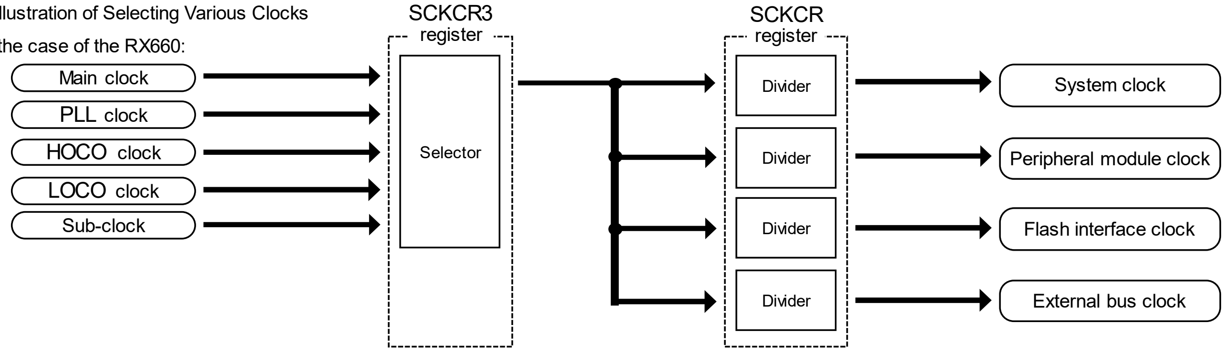
Table 1.2 General Differences Between the R32C/118 and the RX660

Item		R32C (in the case of the R32C/118)	RX (in the case of the RX660)
Maximum operating frequencies	System clock	64 MHz *1, 50 MHz *2	120 MHz
	Peripheral module clock	32 MHz	60 MHz
	External bus clock	32 MHz *1, 25 MHz *2	60 MHz
Frequency	Main clock	4 MHz to 16 MHz	8 MHz to 24 MHz
	Sub-clock	32.768 kHz to 62.5 kHz	32.768 kHz
	PLL	96 MHz to 128 MHz	120 MHz to 240 MHz
	On-chip oscillator	125 kHz	16/18/20 MHz (HOCO) 240 kHz (LOCO)
	IWDT-dedicated on-chip oscillator	—	120 kHz
WDT cycle period		Approx. 16.384 ms to 268.4 sec *1, *3 Approx. 20.972 ms to 268.4 sec *2, *4	Approx. 34.13 µs to 4,096 sec *5
Clock after a reset is released		Main clock	LOCO
Oscillation status after a reset	Main clock	Operating	Stopped
	Sub-clock	Stopped	Operating *6
	PLL	Operating	Stopped
	On-chip oscillator	Stopped	Operating/Stopped (HOCO) *7 Operating (LOCO)
Flash interface clock		CPU clock	FlashIF clock

Notes: 1. This applies to the high-speed version.
2. This applies to the normal-speed version.
3. The WDT cycle period becomes shortest when the external bus is operating at 128 MHz with the PLL clock. It becomes longest when the external bus is operating at 4 MHz with the main clock.
4. The WDT cycle period becomes shortest when the external bus is operating at 100 MHz with the PLL clock. It becomes longest when the external bus is operating at 4 MHz with the main clock.
5. The WDT cycle period becomes shortest when the CPU is operating at 120 MHz with the PLL clock. It becomes longest when the CPU is operating at 125 kHz with the LOCO clock.
6. The sub-clock must be stopped when not in use.
7. 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).

■Illustration of Selecting Various Clocks

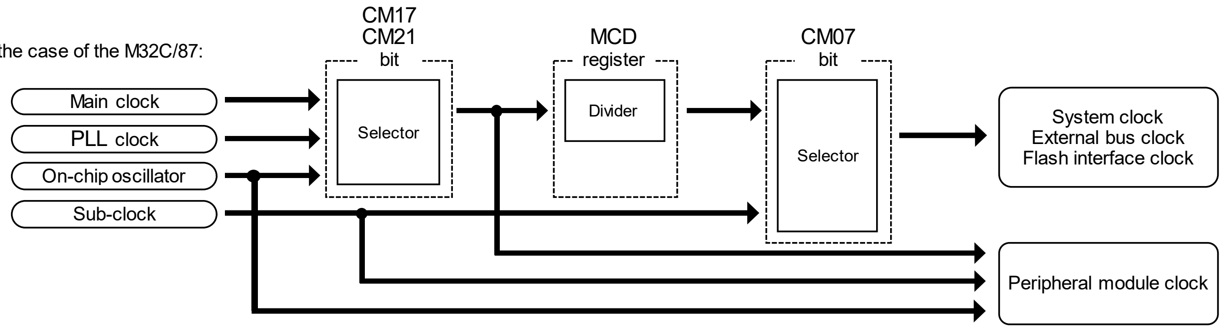
In the case of the RX660:



For the RX660, a separate division ratio can be set for the following clocks:

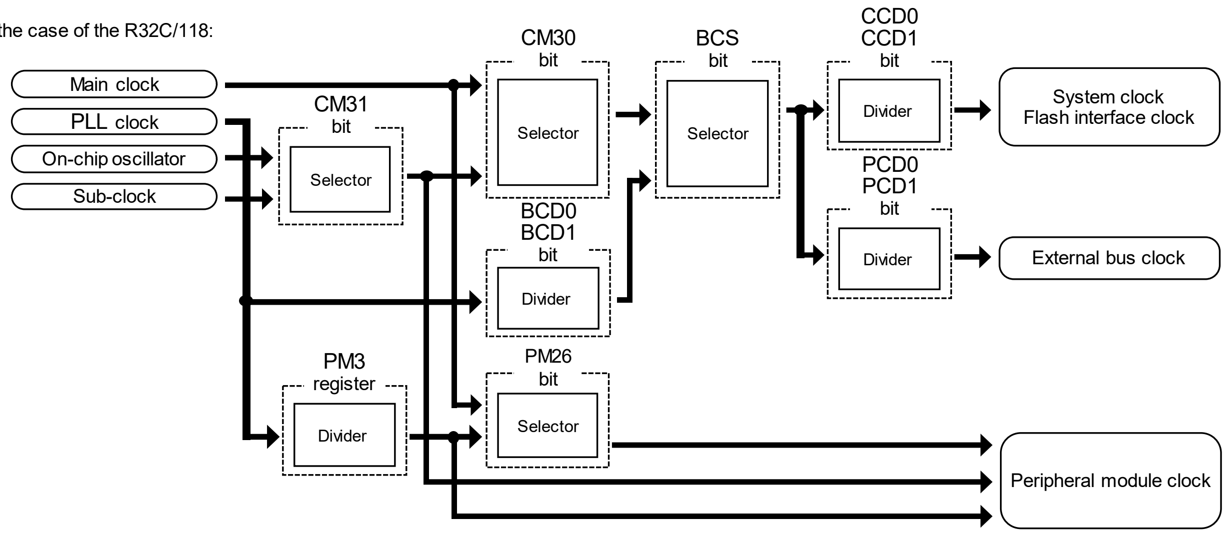
- System clock
- Peripheral module clock
- Flash interface clock
- External bus clock

In the case of the M32C/87:



For the M32C/87, a single clock is used for the system clock, flash interface clock, and external bus clock.
A clock different from the system clock can be used for the peripheral module clock.

In the case of the R32C/118:



For the R32C/118, a single clock is used for the system clock and flash interface clock.
A division ratio can be set separately for the system clock and the external bus clock.
A clock different from the system clock can be used for the peripheral module clock.

Figure 1.1 Illustration of Selecting Various Clocks

2. Functional Differences in Clocks

This chapter describes functional differences in clocks between the RX-family MCU and the M32C/R32C Series MCU.

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 Concept of the Main Clock Oscillation Stabilization Wait Time.

2.1 Concept of the Main Clock Oscillation Stabilization Wait Time

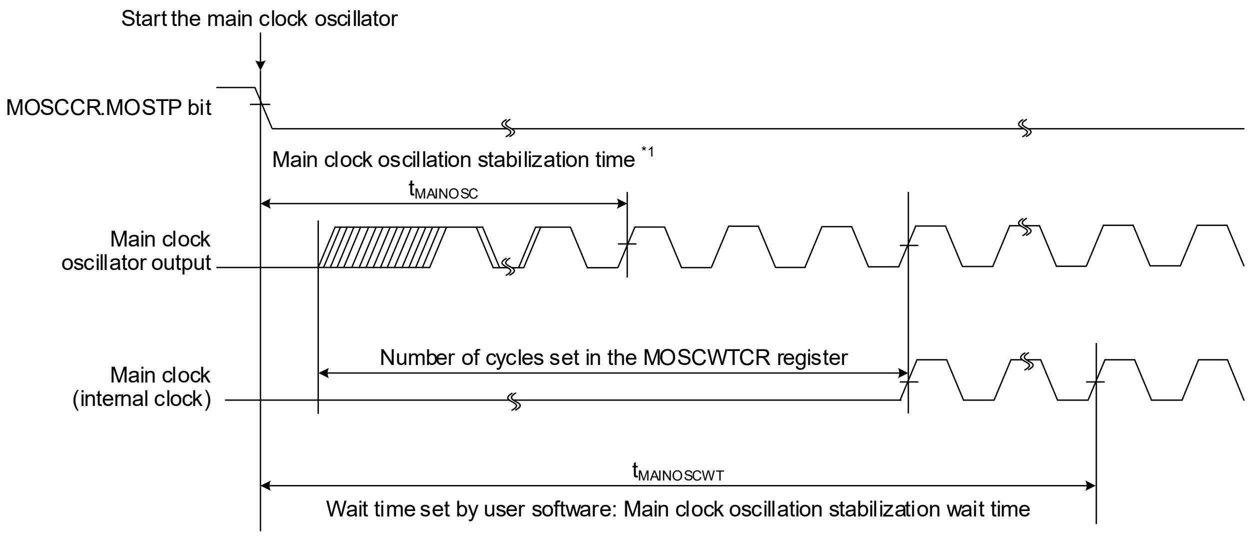
This section describes the concept of the main clock oscillation stabilization wait time in the RX Family.

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 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

3. Differences in Low Power Consumption Modes

The RX Family has several low power consumption modes to reduce power consumption. In order to reduce power consumption, five modes are available on the RX660: Sleep mode, deep sleep mode, software standby mode, all-module clock stop mode, and deep software standby mode. This chapter describes the differences between these five modes and the two modes (wait mode and stop mode) available on the M32C/R32C.

Table 3.1 shows Differences in Low Power Consumption Modes Between the RX Family and the M32C/R32C.

3.1 Sleep Mode

The sleep mode is similar to the wait mode of the M32C/R32C in that the CPU is stopped. The RX Family has a function for automatically switching the clock source when exiting sleep mode.

3.2 Deep Sleep Mode

In this mode, as well as the CPU is stopped as in sleep mode, the clocks for the DMAC, DTC, ROM, and RAM are stopped. Peripheral functions are not stopped.

3.3 Software Standby Mode

The software standby mode is similar to the stop mode of the M32C/R32C in that the CPU, all peripheral functions, and oscillators are stopped.

3.4 All-Module Clock Stop Mode

In this mode, the CPU and all peripheral functions are stopped. Use the module stop function to stop all peripheral functions before entering this mode. In normal operation mode, peripheral functions can be stopped independently. For details, refer to section 9.1.2 Module Stop Function.

3.5 Deep Software Standby Mode

In this mode, power supplies to the CPU, peripheral functions, and oscillators are stopped. Because power supplies are stopped, power consumption can be greatly reduced. In addition, the realtime clock (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 Between the RX Family and the M32C/R32C

Item	M32C/R32C (in the case of the M32C/87 or the R32C/118)		RX (in the case of the RX660)				
Power consumption	More → Less		More → Less				
Mode	Wait mode	Stop mode	Sleep mode	Deep sleep mode	All-module clock stop mode	Software standby mode	Deep software standby mode
CPU	Stopped	Stopped	Stopped	Stopped	Stopped	Stopped	Stopped *1
Main clock Other clocks	Operating *2	Stopped	Operating	Operating	Operating	Stopped	Stopped
Sub-clock	Operating *2	Stopped	Operating	Operating	Operating	Operating	Operating
RAM	Operating	Operating	Operating	Stopped	Stopped	Stopped	Stopped *1
Flash memory	Operating	Stopped	Operating	Stopped	Stopped	Stopped	Stopped
Watchdog timer (WDT)	Operating	Stopped	Stopped	Stopped	Stopped	Stopped	Stopped *1
Independent watchdog timer (IWDT)	N/A	N/A	Operating	Operating	Operating	Operating	Stopped *1
RTC	N/A	N/A	Operating	Operating	Operating	Operating	Operating
8-bit timer	N/A	N/A	Operating	Operating	Operating	Stopped	Stopped *1
Other peripheral functions	Operating	Stopped	Operating	Operating	Stopped	Stopped	Stopped *1
Outline	This mode stops the CPU.	This mode stops the CPU, all peripheral functions, and oscillators.	This mode stops the CPU.	This mode stops the CPU, DMAC, DTC, ROM, and RAM.	This mode stops the CPU and all peripheral functions. (Some peripheral functions are excepted.)	This mode stops the CPU, all peripheral functions, and oscillators. (Only the sub-clock, IWDT, and RTC can operate.)	This mode stops supplying power to all modules. (Only the sub-clock and RTC can operate.)

Notes: 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.
2. The peripheral function clock stop function can be used to stop some clocks.

4. Information Regarding the Function for Lower Operating Power Consumption

Some RX MCUs are equipped with the function for lower operating power consumption. This function reduces 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.

The names of the low power consumption modes available on the RX Family MCU may resemble high-speed mode, medium-speed mode, and low-speed mode available as normal operating modes of the M32C/R32C. However, the modes available on the M32C/R32C simply specify the differences in the operating clock of the CPU.

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 RX660 MCUs are 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 measurement ends or 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) between the RX and the M32C/R32C.

Table 6.1 shows 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		
	M32C (in the case of the M32C/87)	R32C (in the case of the R32C/118)	RX (in the case of the RX660)
Main clock	On-chip oscillator	— (This function must not be used.)	On-chip oscillator (LOCO)
Sub-clock		No change (remains as sub-clock)	No change (remains as sub-clock)
On-chip oscillator		No change (remains as on-chip oscillator)	No change (remains as on-chip oscillator)
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.

On an RX Family MCU, while data is being written to I/O registers, the CPU can execute the subsequent instructions without waiting for the write operation to finish. In addition, when accessing I/O registers, the operating clock for peripheral functions is used. Therefore, in cases such as when the peripheral function clock for the I/O registers to be accessed is slower than the CPU clock, before the settings programmed on I/O registers are applied, the subsequent instructions may be executed.

There may be situations where the changes to I/O registers must be applied before executing subsequent instructions. These situations include when interrupt requests should be disabled by clearing the interrupt request enable bit (ICU.IERn.IENj bit) before executing subsequent instructions, and when the preprocessing to enter the power-down state occurs before executing a WAIT instruction. In such a situation, make sure that the CPU waits for the write operation to finish and then executes the subsequent instructions.

Table 7.1 shows 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

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

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

When migrating from the M32C/R32C Series MCU to the RX Family MCU, refer to the following sections of the user's manual for hardware:

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

9. Appendix

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

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

9.1.1 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 9.1 shows Comparison of Conditions for Interrupt Generation Between the M32C/R32C and the RX.

Table 9.1 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.

9.1.2 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.

9.1.3 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 9.2 shows Comparison of I/O Settings for Peripheral Function Pins Between the M32C and the RX.
Table 9.3 shows Comparison of I/O Settings for Peripheral Function Pins Between the R32C and the RX.

Table 9.2 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 9.3 Comparison of I/O Settings for Peripheral Function Pins Between the R32C and the RX

<table>
<tr><th>Function</th><th>R32C (in the case of the R32C/118)</th><th>RX (in the case of the RX660)</th></tr>
<tr><td>Select the pin function</td><td rowspan="2">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.</td><td>With the PFS register, I/O ports for peripheral functions can be assigned by selecting from multiple pins.</td></tr>
<tr><td>Switch between general I/O port and peripheral function</td><td>With the PMR register, the corresponding pin function can be selected as a general I/O port or a peripheral function.</td></tr>
</table>

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.

9.2 I/O Register Macros

Macro definitions listed in Table 9.4 can be found in the RX I/O register definitions (iodefine.h).

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

Table 9.4 shows Macro Usage Examples.

Table 9.4 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.

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

Table 9.5 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.

10. 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/.