

RENESAS M16C-65C Integrated Circuits User Guide

RENESAS M16C-65C Integrated Circuits User Guide

RX Family and M16C Family

Guide for Migration from the M16C to the RX: Clocks

Abstract

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

Products

RX Family

M16C Family

When this document explains migration from the M16C Family to the RX Family, the M16C/65C Group MCU is used as an example of the M16C Family MCU, and the RX231 Group and RX660 Group MCUs are used as examples of the RX Family 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 the terminology between the M16C Family and RX Family.
The table below shows the differences in terminology related to clocks.

Differences in Terminology Between the RX Family and the M16C Family

Item	M16C Family	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 clockoscillation circuit	XIN, XOUT	EXTAL, XTAL
Modes for reducing power consumption	Wait mode Stop mode	Sleep modeAll-module clock stop mode Software standby modeDeep software standby mode
Registers for peripheralfunctions	Special function registers (SFRs)	I/O registers

Contents

- 1 General Differences in the Clock Generation Circuit
- 2 Functional Differences in Clocks
- 3 Differences in Low Power Consumption Modes
- 4 Information Regarding the Function for Lower Operating Power Consumption
- 5 Information Regarding the Clock Frequency Accuracy Measurement Circuit
- 6 Information Regarding the Oscillation Stop Detection Function
- 7 Information on Accessing I/O Registers
- 8 Chapters Associated With the RX User's Manual: Hardware (UMH)
- 9 Appendix
- 10 Documents / Resources
 - 10.1 References
- 11 Related Posts

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 M16C Family MCU.

There are differences in the frequencies of the clocks used in the RX Family and M16C Family. Table 1.1 shows General 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.

Figure 1.1 shows an Illustration of Selecting Various Clocks.

Table 1.1 General Differences in the Frequencies of Various Clocks

Item		M16C (in the case of the M16C/65C)	RX (in the case of the RX231)	RX (in the case of the RX660)
Maximum operating frequencies	System clock	32 MHz	54 MHz	120 MHz
	Peripheral module clock	32 MHz	32 MHz	60 MHz
	External bus clock	32 MHz *1	32 MHz	60 MHz
Frequency	Main clock	2 MHz to 20 MHz	1 MHz to 20 MHz (VCC ≥ 2.4 V) 1 MHz to 8 MHz (VCC < 2.4 V)	8 MHz to 24 MHz
	Sub-clock	32.768 kHz to 50 kHz	32.768 kHz	32.768 kHz
	PLL clock	10 MHz to 32 MHz	24 MHz to 54 MHz	120 MHz to 240 MHz
	High-speed on-chip oscillator (HOCO)	40 MHz	32 MHz/54 MHz	16 MHz/18 MHz/20 MHz
	Low-speed on-chip oscillator (LOCO)	125 kHz	125 kHz	240 kHz
	IWDT-dedicated on-chip oscillator	—	15 kHz	120 kHz
WDT cycle period		Approx. 16.384 ms to 33.6 sec *2	Approx. 75.85 μs to 4,096 sec *3	Approx. 34.13 μs to 4,096 sec *4
Clock after a reset is released		LOCO	LOCO	LOCO
Oscillation status after a reset	Main clock	Operating	Stopped	Stopped
	Sub-clock	Stopped	Operating *5	Operating *5
	High-speed on-chip oscillator (HOCO)	Stopped	Operating/Stopped *6	Operating/Stopped *6
	Low-speed on-chip oscillator (LOCO)	Operating	Operating	Operating
Clock for flash memory		CPU clock	FlashIF clock	FlashIF clock

- Notes:** 1. Note that if the frequency goes higher than 25 MHz, the data output hold time becomes 0 ns or less. (when VCC = 5 V)
- 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.
- The WDT cycle period is shortest when the operating clock of the CPU is 54 MHz of the PLL clock, and longest when the sub-clock is 32.768 kHz.
- The WDT cycle period is shortest when the operating clock of the CPU is 120 MHz of the PLL clock, and longest when the sub-clock is 32.768 kHz.
- The sub-clock must be stopped when not in use.
- 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).

RX Family

- 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

M16C Family

In the M16C Family, the system clock, flash interface clock, and external bus clock are a common clock. The peripheral function clock can be different from the system clock.

Functional Differences in Clocks

This chapter describes functional differences in clocks between the RX Family MCU and the M16C Family 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

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 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.

Differences in Low Power Consumption Modes

The RX Family has several low power consumption modes to reduce power consumption. There are five modes available on RX Family MCUs as follows: Sleep mode and software standby mode on the RX231 and RX660 MCUs; deep sleep mode on the RX231 MCU; all-module clock stop mode and deep software standby mode on the RX660 MCU. This chapter describes the differences between the five modes available on the RX Family MCUs and the two modes (wait mode and stop mode) available on the M16C Family MCUs.

Table 3.1 shows Differences in Low Power Consumption Modes Between the RX and the M16C

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.

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.

Software Standby Mode

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

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. Refer to section 9.1.3 Module Stop Function for details.

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 and the M16C

Item	M16C (in the case of the M16C/65C)		RX (in the case of the RX231/RX660)				
	More	Less	More				Less
Power consumption							
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	Stopped	Operating	Operating	Operating	Stopped	Stopped
Sub-clock	Operating	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	Operating	Stopped	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.)

Note: 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.

Information Regarding the Function for Lower Operating Power Consumption

The RX231 MCU is 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 above shown modes available on the RX Family MCU resemble high-speed mode, medium-speed mode, and low-speed mode available on the M16C Family MCU. However, the modes available on the M16C Family MCU simply specify the differences in the operating clock of the CPU.

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 RX231 and 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.

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 M16C.

Table 6.1 shows Differences in the Oscillation Stop Detection Function

Clocks When Oscillation is Stopped	Clocks After an Oscillation Stop is Detected	
	M16C (in the case of the M16C/65C)	RX (in the case of the RX231/RX660)
Main clock	LOCO	LOCO
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: However, the frequency becomes the self-oscillation frequency.

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

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	

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

Appendix

Points on Migration From the M16C Family to the RX Family

This section explains points on migration from the M16C Family to the RX Family

Interrupts

For the RX Family, when an interrupt request is received while all of the following conditions are met, the interrupt occurs.

- The I flag (PSW.I bit) is 1.
- Registers IER and IPR in the ICU are set to enable interrupts.
- The interrupt request is enabled by the interrupt request enable bits for the peripheral function

Table 9.1 shows Comparison of Conditions for Interrupt Generation Between the RX and the M16C.

Table 9.1 Comparison of Conditions for Interrupt Generation Between the RX and the M1

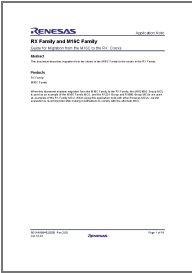
Item	M16C	RX
I flag	When the I flag is set to 1 (enabled), the maskable interrupt request can be accepted.	
Interrupt request flag	When there is an interrupt request from a peripheral function, the interrupt request flag becomes 1 (interrupt requested).	
Interrupt priority level	Selected by setting bits ILVL2 to ILVL0.	Selected by setting the IPR[3:0] bits.
Interrupt request enable	—	Specified by setting the IER register.
Interrupt enable for peripheral functions	—	Interrupt enable or disable can be specified in each peripheral function.

For more information, refer to sections Interrupt Controller (ICU), CPU, and sections for other peripheral functions used in the UMH.



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.

	<p>RENESAS M16C-65C Integrated Circuits [pdf] User Guide</p> <p>M16C-65C Integrated Circuits, M16C-65C, Integrated Circuits, Circuits</p>
---	---

References

-  [Renesas Electronics Corporation](#)
-  [Distributor Network](#) | [Renesas Semiconductors](#) | [Renesas](#)
- [User Manual](#)

[Manuals+](#), [Privacy Policy](#)

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.