RX Family, M16C Family

R01AN2148EJ0100
Rev. 1.00
Sep. 16, 2014

Migrating From the M16C Family to the RX Family: A/D and D/A Converters

Abstract

This document describes migrating the A/D and D/A converters in the M16C Family to the RX Family.

Products

RX Family

M16C Family

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

Differences in Terminology

Item	RX Family	M16C Family
Peripheral function registers	I/O registers	Special function registers (SFRs)

Contents

1. Differences in the Functions of the A/D Converter and D/A Converter

Table 1.1 lists the Differences in the A/D Converter.

Table 1.1 Differences in the A/D Converter

Item	RX (RX210) 12-bit A/D converter	M16C (M16C/65C) A/D converter
A/D conversion method	Successive approximation	Successive approximation
Analog input voltage	0 V to AVCC0 (1.62 to 5.5 V)	0 V to AVCC (3.0 to 5.5 V)
Operating clock	Peripheral module clock PCLK and A/D conversion clock ADCLK can be set so that the frequency division ratio should be one of the following. PCLK to ADCLK frequency division ratio = 1:1, 1:2, 1:4, 1:8, 2:1, 4:1	f1, f1 divided by 2, f1 divided by 3, f1 divided by 4, f1 divided by 6, f1 divided by 12, fOCO40M divided by 2, fOCO40M divided by 3, fOCO40M divided by 4, fOCO40M divided by 6, or fOCO40M divided by 12
Resolution	12-bit	10-bit
Operating modes	Single scan mode Continuous scan mode Group scan mode (Double-trigger mode selectable)	One-shot mode Repeat mode Single sweep mode Repeat sweep mode 0 Repeat sweep mode 1
Analog input pins	16 ● 16 analog input pins	26 ● AN0 to AN7 ● AN0_0 to AN0_7 ● AN2_0 to AN2_7 ● ANEX0 and ANEX1
Internally generated analog input	● One for the temperature sensor ● One for the internal reference voltage	N/A
Conditions to start A/D conversion	● Software trigger ● Asynchronous trigger (ADTRG0# pin) *1 ● Synchronous trigger (triggers from the MTU, ELC, and temperature sensor)	● Software trigger ● External trigger *2
Sample and hold function	Available	Available
Low power consumption	Module can be set to the module stop state	N/A
A/D-converted value addition function	Available	N/A
Self-diagnosis function	Self-diagnosis function, analog input disconnection detection assist function	Open-circuit detection assist function
Event link function	A/D conversion can be started when an event signal is input.	N/A
Interrupts	A/D scan end interrupt (S12ADI0), A/D scan end interrupt specially for group B (GBADI)	A/D conversion interrupt
Conversion cycle	1.0 μs (per channel, when ADCLK = 50 MHz)	1.6 μs (per channel, when ϕAD = 25 MHz)
Error	● DNL differential nonlinearity error ±1 LSB (typical) ● INL integral nonlinearity error ±1 LSB (typical), ±3 LSB (maximum) Example when using 3.3 V: Error of ±2.417 mV (maximum) *3	● Integral nonlinearity error AN0 to AN7, AN0_0 to AN0_7, AN2_0 to AN2_7 input: ±3 LSB (maximum) ANEX0, ANEX1 input: ±3 LSB (maximum) Example when using 3.3 V: Error of ±9.668 mV (maximum) *3

Note 1. If an asynchronous trigger is input during A/D conversion, A/D conversion continues.
Note 2. If an external trigger is input during A/D conversion, the current A/D conversion is halted, and A/D conversion starts again.
Note 3. The voltage calculated assumes the conditions listed in the Electrical Characteristics – the actual error may vary.

Table 1.2 Differences in the D/A Converter

Item	RX (RX210)	M16C (M16C/65C)
Resolution	10-bit	8-bit
D/A conversion on channels 0 and 1	Independently controlled or single whole controlled can be selected	N/A
Selectable formats	Data format can be selected as flush with the right end or left end of the data register	N/A
Analog output pins	2: DA0 and DA1	2: DA0 and DA1
Low power consumption	Module can be set to the module stop state	N/A
Event link function	D/A0 conversion can be started when an event signal is input.	N/A

2. Differences in the A/D Converter Functions and Setting Procedures

This chapter explains the differences in using the A/D converter to perform A/D conversion.

Table 2.1 Modes Used in A/D Converter Operating Example

No.	Operating Example	RX	M16C	Description
1	A/D convert the output from one sensor (refer to the operation shown in Figure 2.1)	Single scan mode	One-shot mode	Section 2.1
2	A/D convert the output from two sensors with differing responsiveness (refer to the operation shown in Figure 2.2)	Group scan mode	Single sweep mode	Section 2.2

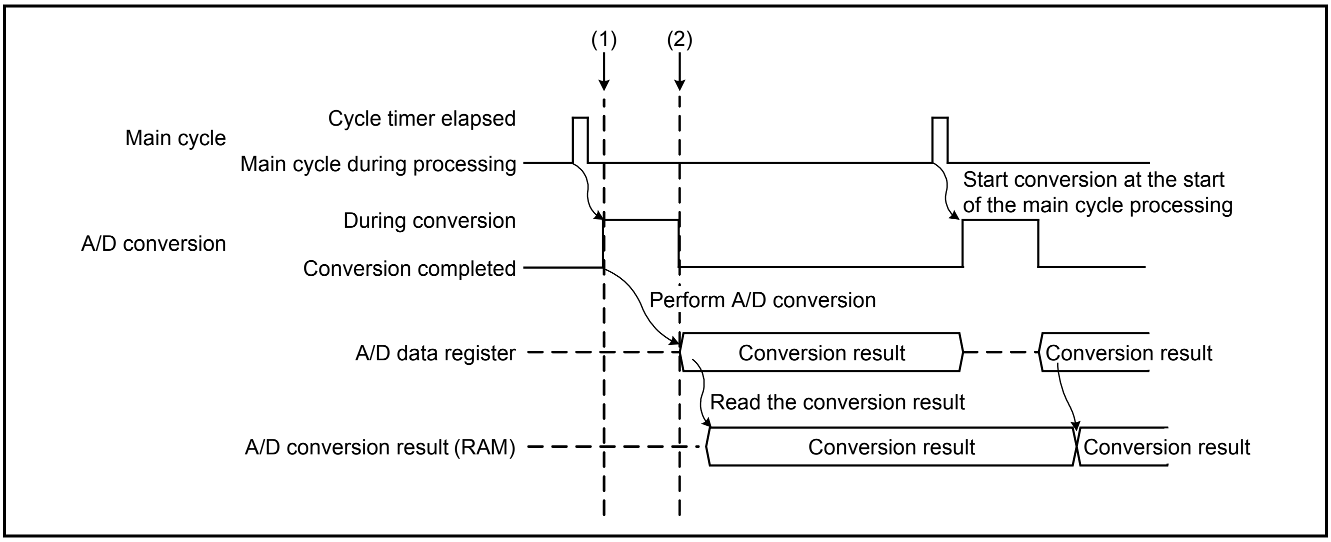

Figure 2.1 Analog Conversion (Example 1)

RX210/M16C

(1) Starting conversion
A/D conversion is started.

(2) Completing A/D conversion
When A/D conversion is completed, the converted value is stored in an A/D register, and an A/D conversion complete interrupt request is generated. The A/D conversion result is read in the interrupt handling.

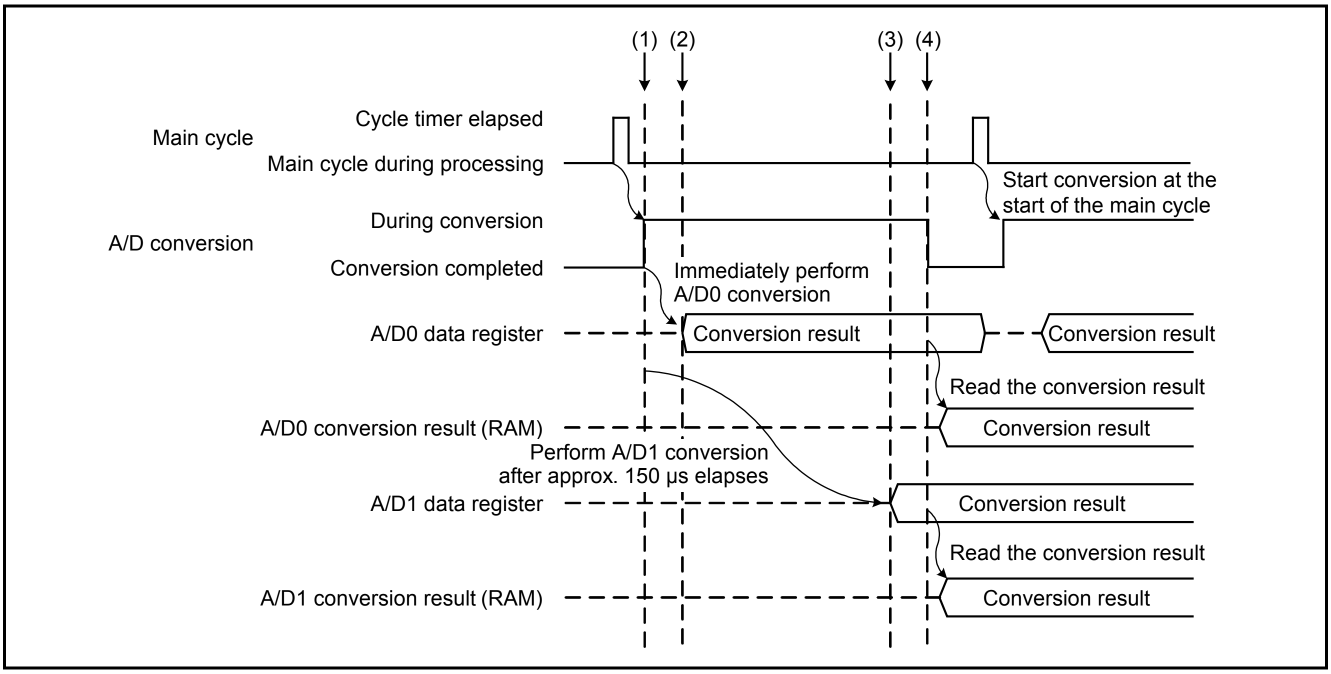

Figure 2.2 Analog Conversion (Example 2)

The above figure shows an example of converting output from a sensor with good response and converting output from a sensor with poor response.

RX210

Output from a sensor with good response is converted on A/D channel 0 using the MTU0.TGRA0 source. Output from a sensor with poor response is converted on A/D channel 1 using the MTU0.TGRB0 source.

(1) Starting conversion
The TGRA0 source is generated, and A/D conversion starts on A/D channel 0.

(2) Completing A/D conversion on A/D channel 0
A/D conversion of A/D channel 0 is completed.

(3) Starting A/D conversion on A/D channel 1
The TGRB0 source is generated approximately 150 µs after the A/D conversion start trigger, and A/D conversion starts on A/D channel 1.

(4) Completing A/D conversion on A/D channel 1
A/D conversion of A/D channel 1 is completed. When A/D conversion of two channels is completed, the conversion results are stored in an A/D register, and the A/D conversion complete interrupt request is generated. The A/D conversion result is read in the interrupt handling.

M16C

In single sweep mode, the sensor with good response uses A/D channel 0 for A/D conversion and the sensor with poor response uses A/D channel 5 for A/D conversion. The approximately 150 µs conversion wait is secured with the AN0 to AN4 pin conversion time.

(1) Starting conversion
A/D conversion is started on A/D channel 0.

(2) Completing A/D conversion on A/D channel 0
A/D conversion is completed on A/D channel 0.

(3) Starting A/D conversion on A/D channel 5
A/D conversion is started on A/D channel 5.

(4) Completing A/D conversion on A/D channel 5
A/D conversion is completed on A/D channel 5. When A/D conversion of two channels is completed, the conversion results are stored in an A/D register, and the A/D conversion complete interrupt request is generated. The A/D conversion result is read in the interrupt handling.

2.1 Differences in the Setting Procedures When A/D Converting One Analog Input

Table 2.2 lists Differences in the Initial Setting Procedure of A/D Converter Output When A/D Converting One Analog Input. Table 2.3 lists Completing A/D Conversion When A/D Converting One Analog Input.

Table 2.2 Differences in the Initial Setting Procedure of A/D Converter Output When A/D Converting One Analog Input

	Procedure	RX (RX210)	M16C (M16C/65C)
1	Exit the module stop state [*1]	SYSTEM.PRCR.WORD = 0xA502; MSTP(S12AD) = 0; SYSTEM.PRCR.WORD = 0xA500;	N/A (no module stop function)
2	Set the A/D input pin	PORT4.PDR.BYTE = 0x00; PORT4.PMR.BYTE = 0x00; MPC.PWPR.BIT.B0WI = 0; MPC.PWPR.BIT.PFSWE = 1; MPC.P40PFS.BIT.ASEL = 1; MPC.PWPR.BIT.PFSWE = 0; MPC.PWPR.BIT.B0WI = 1;	pd10 = 0x00;

3	Set the operating mode, conversion start source, and clock	S12AD.ADCSR.WORD = 0x1000;	adcon2 = 0x00;
			adcon0 = 0x00;
			adcon1 = 0x20;

4	Set the conversion pin	S12AD.ADANSA.WORD = 0x01;	
5	Set the A/D conversion start source	S12AD.ADSTRGR.WORD = 0x0000;	
6	Enable the A/D interrupt [*2]	IR(S12AD,S12ADI0) = 0; IPR(S12AD,S12ADI0) = 0x01; IEN(S12AD,S12ADI0) = 1;	ir_adic = 0; adic = 0x01:
7	Start A/D conversion	S12AD.ADCSR.BIT.ADST = 1;	adst = 1;

Note 1. Refer to section 4.1.3 for details on the module stop function.
Note 2. The method for enabling the interrupt request differs. Refer to section 4.1.1 for details.

Table 2.3 Completing A/D Conversion When A/D Converting One Analog Input

	Procedure	RX (RX210)	M16C (M16C/65C)
1	Read the A/D converted result	ad_result = S12AD.ADDR0;	ad_result = ad0;

2.2 Differences in the Setting Procedures When A/D Converting Two Analog Inputs

Table 2.4 lists Differences in the Initial Setting Procedure of A/D Converter Output When A/D Converting Two Analog Inputs. Table 2.5 lists Completing A/D Conversion When A/D Converting Two Analog Inputs.

Table 2.4 Differences in the Initial Setting Procedure of A/D Converter Output When A/D Converting Two Analog Inputs

	Procedure	RX (RX210)	M16C (M16C/65C)
1	Exit the module stop state [*1]	SYSTEM.PRCR.WORD = 0xA502; MSTP(S12AD) = 0; SYSTEM.PRCR.WORD = 0xA500;	N/A (no module stop function)
2	Set the A/D input pins	PORT4.PDR.BYTE = 0x00; PORT4.PMR.BYTE = 0x00; MPC.PWPR.BIT.B0WI = 0; MPC.PWPR.BIT.PFSWE = 1; MPC.P40PFS.BIT.ASEL = 1; MPC.P41PFS.BIT.ASEL = 1; MPC.PWPR.BIT.PFSWE = 0; MPC.PWPR.BIT.B0WI = 1;	pd10 = 0x00;
3	Set the operating mode and clock	S12AD.ADCSR.WORD = 0x3240;	

4	Set the A/D conversion start condition	S12AD.ADSTRGR.WORD = 0x0201;	adcon2 = 0x10;
			adcon0 = 0x10;
			adcon1 = 0x22; [*3]

5	Set the conversion pins	S12AD.ADANSA.WORD = 0x0001; S12AD.ADANSB.WORD = 0x0002;	
6	Enable the A/D interrupt [*2]	IR(S12AD,S12ADI0) = 0; IPR(S12AD,S12ADI0) = 0x01; IEN(S12AD,S12ADI0) = 1;	ir_adic = 0; adic = 0x01:
7	Set the timer	Setting for multifunction pulse unit 2 (MTU2a)	N/A (no processing)
8	Start A/D conversion	N/A (no processing)	adst = 1;

Note 1. Refer to section 4.1.3 for details on the module stop function.
Note 2. The method for enabling the interrupt request differs. Refer to section 4.1.1 for details.
Note 3. AN0 and AN5 are used as the A/D conversion pins.

Table 2.5 Completing A/D Conversion When A/D Converting Two Analog Inputs

	Procedure	RX (RX210)	M16C (M16C/65C)
1	Read the A/D converted result (group A scan complete interrupt)	ad_result[0] = S12AD.ADDR0;	ad_result[0] = ad0 ad_result[1] = ad5
2	Read the A/D converted result (group B scan complete interrupt)	ad_result[1] = S12AD.ADDR1;	N/A (function not available)

3. Differences in the D/A Converter Functions and Setting Procedures

This chapter explains the differences in using the D/A converter to perform D/A conversion.

This chapter explains the differences of how the RX210 Group and M16C/65 Group use the D/A converter to perform the operation listed in Table 3.1.

Table 3.1 Example of D/A Converter Operation

No.	Operating Example	Description
1	Periodic D/A output from two channels	Section 3.1

3.1 Differences in the Setting Procedures When D/A Converting Two Analog Inputs

Table 3.2 lists Differences in the Initial Setting Procedure of D/A Converter Output. Table 3.3 lists Differences in the Output Voltage Switch Settings for D/A Converter Output.

Table 3.2 Differences in the Initial Setting Procedure of D/A Converter Output

	Procedure	RX (RX210)	M16C (M16C/65C)
1	Exit the module stop state *1	SYSTEM.PRCR.WORD = 0xA502; MSTP(DA) = 0; SYSTEM.PRCR.WORD = 0xA500;	N/A (no module stop function)

2	Set the I/O port functions *2	PORT0.PDR.BIT.B3 = 0;
		PORT0.PDR.BIT.B5 = 0;
		MPC.PWPR.BIT.B0WI = 0;
		MPC.PWPR.BIT.PFSWE = 1;
		MPC.P03PFS.BYTE = 0x80;
		MPC.P05PFS.BYTE = 0x80;
		MPC.PWPR.BYTE = 0x80;
		PORT0.PMR.BIT.B3 = 0;
		PORT0.PMR.BIT.B5 = 0;
			prcr = 0x08;
			pd9 = 0x00;
			prcr = 0x00;

3	Select single D/A conversion for channels 0 and 1	DA.DACR.BYTE = 0x3F;	N/A (no processing)
4	Set the register for the digital value	DA.DADR0 = cycle_tbl0[0]; DA.DADR1 = cycle_tbl1[0];	da0 = cycle_tbl0[0]; da1 = cycle_tbl1[0];
5	Enable D/A converter output	DA.DACR.BYTE = 0xFF;	dacon = 0x03;

Note 1. Refer to section 4.1.3 for details on the module stop function.

Note 2. In the RX Family, peripheral function pin settings are performed in the MPC. Refer to section 4.1.2 for details.

Table 3.3 Differences in the Output Voltage Switch Settings for D/A Converter Output

	Procedure	RX (RX210)	M16C (M16C/65C)
1	Set the register for the digital value	DA.DADR0 = cycle_tbl0[cycle_cnt0]; DA.DADR1 = cycle_tbl1[cycle_cnt1];	da0 = cycle_tbl0[cycle_cnt0]; da1 = cycle_tbl1[cycle_cnt1];

4. Appendix

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

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

4.1.1 Interrupts

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

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

Table 4.1 lists a Comparison of Conditions for Interrupt Generation.

Table 4.1 Comparison of Conditions for Interrupt Generation

Item	RX210	M16C/65C
I flag	When the I flag is set to 1 (enabled), the interrupt request can be accepted.	
Interrupt request flag	When an interrupt request is generated by a peripheral function, the interrupt request flag becomes 1 (interrupt requested).	
Interrupt priority level	Selected by setting the IPR[3:0] bits.	Selected by setting bits ILVL2 to ILVL0.
Interrupt request enable	Specified by setting the IER register.	N/A
Interrupt enable for peripheral functions	Interrupts can be enabled or disabled in each peripheral function.	N/A

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

4.1.2 I/O Ports

In the RX Family, the MPC must be configured in order to assign I/O signals from peripheral functions to pins. Before controlling the 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 pins.
- In the PORTn.PMR register, the pin function from a general I/O port or peripheral function.

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

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

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

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

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

4.1.3 Module Stop Function

The RX Family has the ability to stop each peripheral module individually. By transitioning unused peripheral modules to the module stop state, power consumption can be reduced. After a reset is released, all modules (with a few exceptions) are in the module stop state. Registers for modules in the module stop state cannot be written to or read.

For more information, refer to the Low Power Consumption section in the User's Manual: Hardware.

4.2 I/O Register Macros

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

Table 4.3 Using Macros

Macro	Usage Example
IR("module name", "bit name")	**IR(MTU0,TGIA0)** = 0 ; The IR bit corresponding to MTU0.TGIA0 is cleared to 0 (no interrupt request is generated).
DTCE("module name", "bit name")	**DTCE (MTU0, TGIA0)** = 1 ; The DTCE bit corresponding to MTU0.TGIA0 is set to 1 (DTC activation is enabled).
IEN("module name", "bit name")	**IEN(MTU0, TGIA0)** = 1 ; The IEN bit corresponding to MTU0.TGIA0 is set to 1 (interrupt request enabled).
IPR("module name", "bit name")	**IPR(MTU0, TGIA0)** = 0x02 ; The IPR[3:0] bits corresponding to MTU0.TGIA0 are set to 0010b (interrupt priority level 2).
MSTP("module name")	**MSTP(MTU)** = 0 ; The MTU0 Module Stop bit is set to 0 (module stop state is canceled).
VECT("module name", "bit name")	#pragma interrupt (Excep_MTU0_TGIA0 (vect=**VECT(MTU0, TGIA0)**) The interrupt function is declared for the corresponding MTU0.TGIA0.

4.3 Intrinsic Functions

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

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

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

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

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

5. Reference Documents

User’s Manual: Hardware

RX210 Group User's Manual: Hardware Rev.1.50 (R01UH0037EJ)
M16C/65C Group User's Manual: Hardware Rev.1.10 (R01UH0093)
Refer to the corresponding 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: A/D and D/A Converters	
Rev.	**Date**	**Description**	
		Page	**Summary**
1.00	Sep. 16, 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

Colophon 4.0