Renesas RA8 Family

RA8x2 MCU Quick Design Guide

Introduction

This document answers common questions and points out subtleties of the MCU that might be missed unless the hardware manual was extensively reviewed. The document is not intended to be a replacement for the hardware manual; it is intended to supplement the manual by highlighting some key items most engineers will need to start their own design. It discusses some design decisions from an application point of view. The primary focus is on the design guidelines for the Dual-Core series of the RA8P1, RA8D2, RA8M2, RA8T2 MCU families (RA8x2), which are applicable to both the Dual-Core and Single-Core products, although there are some differences in the number of CPU cores and peripherals. Furthermore, it highlights key differences between the Dual-Core and Single-Core series within the family.

The RA8x2 series devices offer a product option that incorporates Octa-SPI Flash manufactured by ISSI within MCU chips. This option is referred to as a SiP (System-in-Package) product. In contrast, products that do not include SiP Flash are designated as Standard products.

Target Device

RA8P1, RA8D2, RA8M2, RA8T2 MCU families.

Please note that some RA8 MCU Groups may not include all the features referred to in this Quick Design Guide. Please refer to the User's Manual for the specific MCU Group for details of which features are included for that device.

MCU Key Features

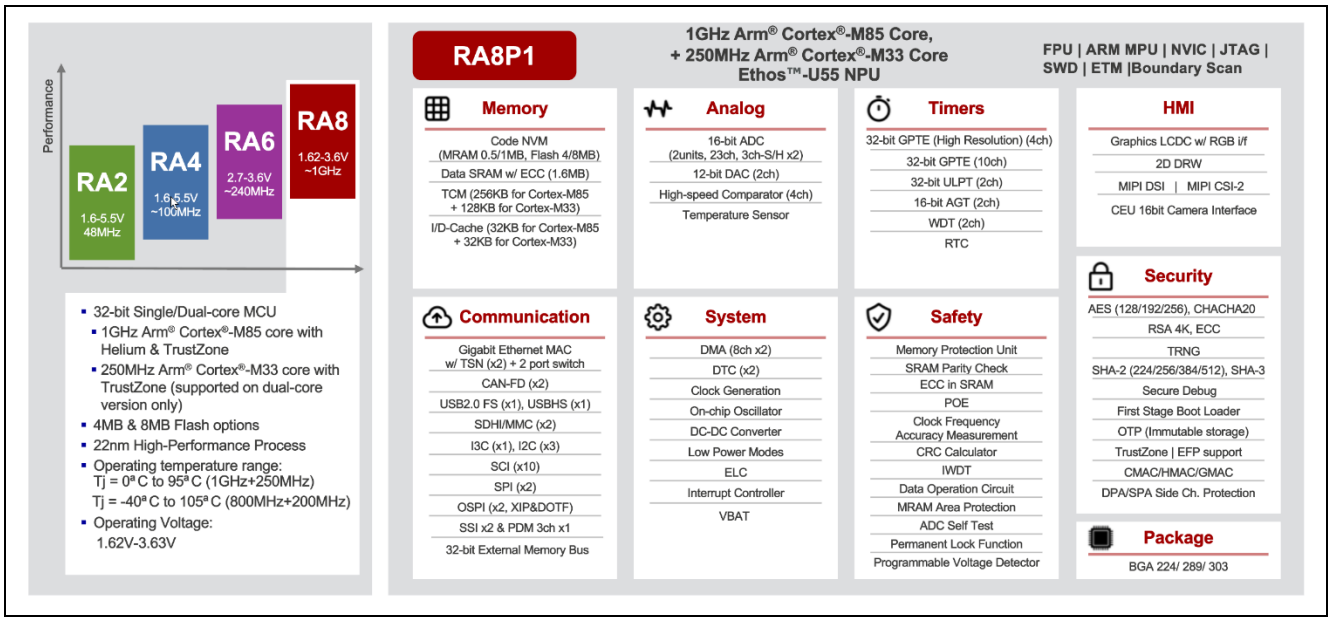


Figure 1. Example of RA8P1 Key Features

Contents

1. Power Supplies

The RA8x2 MCU family has digital power supplies and analog power supplies. The power supplies use the following pins. The system-in-package (SiP) has the dedicated VCC2_n power supply.

Table 1. Digital Power Supplies

Symbol	Function Name	Description
VCC_01 to VCC_10, VCC2_11 to VCC2_15	System Power supply	Power supply pin. Connect it to the system power supply. Connect this pin to the same numbered VSS_01 to VSS_15 by a 100 nF (0.1 µF) capacitor. The capacitor should be placed close to the pin. In the SiP product, connect VCC2_11 to VCC2_15 to the 1.8V system power supply.
VCC2_16 to VCC2_19	SiP (System in Package) power supply	Dedicated power supply pin for the SiP product. Connect it to the 1.8V system power supply. Connect this pin to the same numbered VSS_16 to VSS_19 by a 100 nF (0.1 µF) capacitor. The capacitor should be placed close to the pin.
VCC_DCDC	Switching regulator (DCDC) Power supply	Switching regulator power supply input pin. Connect in either DCDC mode or External VDD mode, as described in Section 1.2 or Section 1.3.
VLO	Switching regulator (DCDC)	Switching regulator I/O pin. Connect in either DCDC mode or External VDD mode, as described in Section 1.2 or Section 1.3.
VCL0 to VCL11	Power supply	Connect in either DCDC mode or External VDD mode, as described in Section 1.2 or Section 1.3.
VBATT	Backup power	Backup power pin. Supplies power to RTC, sub-clock oscillator, backup register, tamper detection, and VBATT_R voltage drop detection in the absence of VCC. When VBATT pin is not used, connect to VCC.
VSS_01 to VSS_15, VSS0 to VSS11	Ground	Ground pin. Connect it to the system power supply (0 V).
VSS_16 to VSS_19, VSS	Ground	Dedicated ground pin for the SiP product. Connect it to the system power supply (0 V).
VSS_DCDC	Switching regulator (DCDC) Ground	Ground pin. Connect it to the system power supply (0 V).
VCC_USB	USB FS power supply	USB Full-speed power supply pin. Connect this pin to the 3.3V system power supply. Connect this pin to VSS_USB via a 100 nF (0.1 µF) capacitor placed close to the VCC_USB pin.
VSS_USB	USB FS ground	USB Full-speed ground pin. Connect this pin to VSS.
VCC_USBHS[1]	USB HS power supply	USB High-speed power supply pin. Connect this pin to the 3.3V system power supply. Connect this pin to VSS1_USBHS and VSS2_USBHS via a 10 nF ceramic capacitor placed close to the VCC_USBHS pin. Also connect this pin to VSS1_USBHS and VSS2_USBHS via a 47 µF electrolytic capacitor.
VSS1_USBHS[1], VSS2_USBHS[1]	USB HS ground	USB High-speed ground pin. Connect this pin to VSS.
VCC18_MIPI[2]	MIPI power supply	MIPI interface power supply pin. Connect to 1.8 V source. Connect this pin to VSS_MIPI via a 100 nF (0.1 µF) capacitor placed close to the VCC18_MIPI pin.
VSS_MIPI[2]	MIPI ground	MIPI interface ground pin. Connect this pin to VSS.

Note: 1. Only for devices with USB High Speed peripheral.
2. Only for devices with MIPI interface

Table 2. Analog Power Supplies

Symbol	Function Name	Description
AVCC0	Analog power supply	Analog voltage supply pin for the respective modules. When the ADC, DAC and High-speed Analog Comparator are not in use, connect to VCC.
AVSS0	Analog ground	Analog ground for the respective modules. Connect this pin to the same voltage as the VSS pin. If connected to VSS, connect at a single point.
VREFH0	16-bit ADC high reference voltage	Analog reference voltage supply pin for the ADC16H (unit 0). Connect this pin to AVCC0 when not using the ADC16H (unit 0).
VREFL0	16-bit ADC low reference voltage	Analog reference ground pin for the ADC16H. Connect this pin to AVSS0 when not using the ADC16H (unit 0).
VREFH	16-bit ADC & DAC analog reference voltage	Analog reference voltage supply pin for the ADC16H (unit 1) and D/A Converter. Connect this pin to AVCC0 when not using the ADC16H (unit 1) and D/A Converter.
VREFL	16-bit ADC & DAC analog reference ground	Analog reference ground pin for the ADC16H and D/A Converter. Connect this pin to AVSS0 when not using the ADC16H (unit 1) and D/A Converter.
IVREFn	ACMPHS reference voltage input	Reference voltage input pins for comparator.
AVCC_USBHS[1]	USB HS analog power supply	USB High-speed analog power supply. Connect this pin to the 3.3V system power supply through an isolation inductor or ferrite. Connect to VSS1_USBHS and VSS2_USBHS through a 10 µF capacitor.
USBHS_RREF[1]	USB HS current reference	USB High-speed reference current source pin. Connect this pin to the VSS1_USBHS and VSS2_USBHS pins through a 2.2 kΩ resistor (±1%)
AVCC_MIPI[2]	MIPI analog power supply	MIPI interface analog power supply. Connect this pin to the 3.3V system power supply. Connect this pin to VSS_MIPI through a 100 nF (0.1 µF) capacitor.

Note: 1. Only for devices with USB High Speed peripheral.
2. Only for devices with MIPI interface

1.1 Dual VCC Power Domains

RA8x2 microcontrollers have two primary power supply voltages, VCC and VCC2. Each of these power supply voltages may be sourced from different external power supplies. Each power supply voltage is used internally for multiple peripherals and I/O blocks. Please refer to the table “Recommended operating conditions” in the “Electrical Characteristics” chapter of the Hardware User’s Manual for details of allowed voltage ranges for each power supply voltage. As an example, RA8x2 devices have the following capabilities: For the standard product, both VCC and VCC2 can operate from 1.62 V to 3.63 V. For the SiP product, VCC can operate from 1.62 V to 3.63 V, and VCC2 can operate from 1.70 V to 2.00 V.

The peripherals associated with VCC include both ESWM and SDRAM. When using SDRAM, VCC and VCC2 should be connected to a nominal 3.3 V supply. When SDRAM is not used and ESWM is used, it is possible to connect VCC and VCC2 to a nominal 2.5V supply . However, there may be circumstances where a lower voltage rail may be desired for some higher speed peripherals. For example, Octal SPI memory devices often have an I/O voltage of 1.8 V for lower power consumption. In this case, VCC2 can be connected to a lower voltage external power supply.

Details of which MCU ports are associated with which power supply voltage can be found in the table “I/O port functions” in the “I/O Ports” chapter of the Hardware User’s Manual. The individual ports associated with each power supply voltage may vary from one RA8x2 device to another. When connecting different voltage supply levels to VCC and VCC2, ensure that all ports associated with each power supply voltage will operate correctly at the relevant voltage level.

When connecting VCC and VCC2 to their respective voltage supplies, follow the guidelines in the “Internal Voltage Regulator” chapter of the Hardware User’s Manual, with one exception. The diagrams in that chapter show VCC and VCC2 both connected to the same external power supply. Instead of connecting both power

supply voltages to the same external power supply, each power supply voltage may be connected to separate external power supplies. Be sure to include the voltage bypass capacitors indicated in the Hardware User's Manual for every VCC and VCC2 pin.

1.2 DCDC Mode

In DCDC mode, VDD is supplied from VCL through the VLO output and an external inductor and capacitor. DCDC mode has the advantages that no external supply is needed for VDD and all power up timing is handled within the microcontroller. However, additional components are needed to support this mode, which can add cost and board space requirements.

In DCDC mode, to minimize power loss and maximize efficiency, Renesas recommends using a 2.2 uH inductor with a DC resistance of 100 mΩ or less. VCC and VCC_DCDC should be shorted together.

To implement DCDC mode:

- VCC_n and VCC2_n pins: Connect each pin to the relevant system power supply. Connect each pin to VSS_n through a 0.1 µF capacitor, with the capacitor close to each pin.
- VCC_DCDC pins: Connect the pins together and connect them to the system power supply. Connect the pins to VSS_DCDC through a single 22 µF and a single 0.1 µF capacitor in parallel. Place the 0.1 µF capacitor close to the pin.
- VCLn pins: Connect each pin to VSSn through a 0.22 µF capacitor. Place the capacitor close to the pin.
- VLO pins: Connect the pins together and connect to an external inductor and capacitor. Place the 2.2 µH inductor and 47 µF capacitor close to the pin. Connect the capacitor to VSS_DCDC.

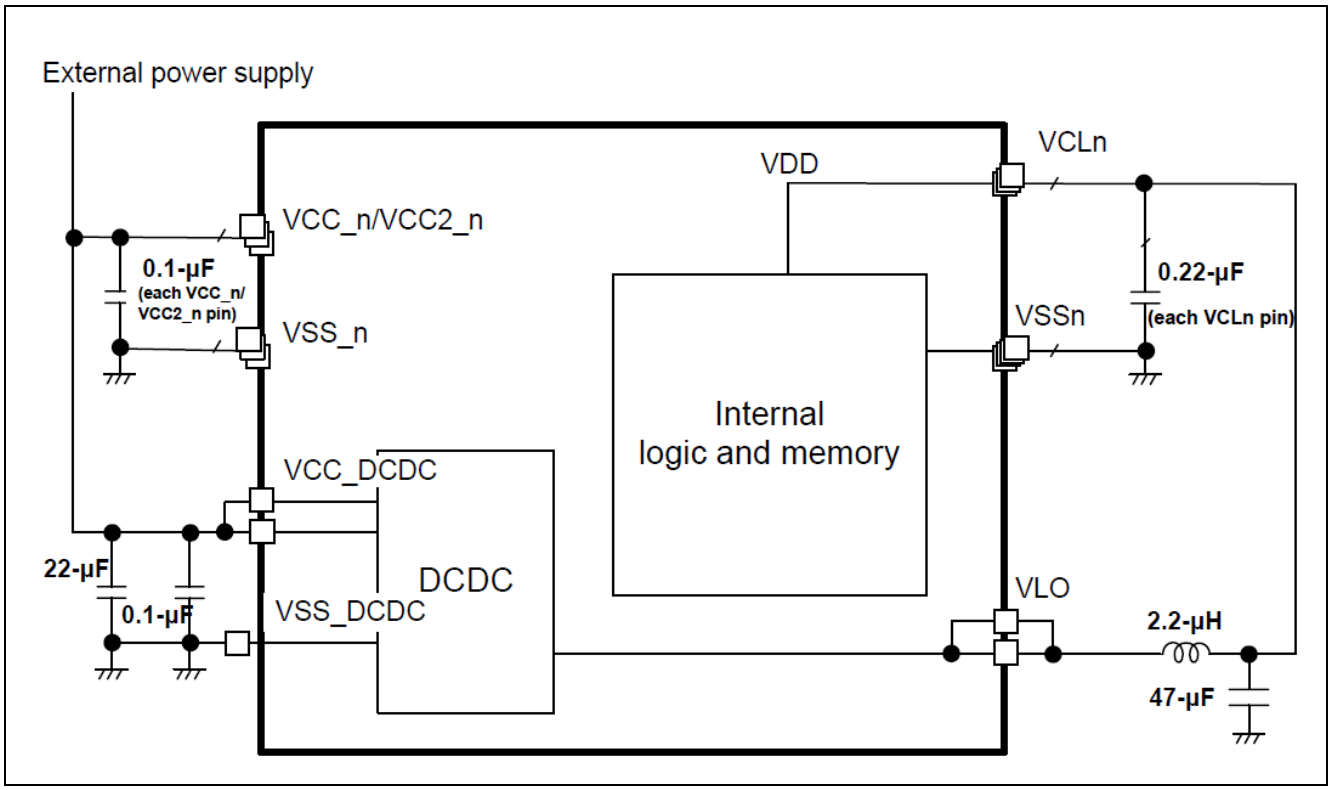


Figure 2. DCDC Mode Configuration

VCC_DCDC and VLO are sensitive to parasitic inductance and parasitic resistance due to routing and component selection. Select devices with low ESR and low parasitic inductance. VCC, VCC2 and VCL are not as sensitive to these parameters, so standard components can be used. Recommended components are shown below.

Table 3. Recommended components for DCDC Mode

Part Type	Value	Manufacturer	Manufacturer's Part Number
Inductor	2.2uH	TDK	SPM5020T-2R2M-LR
Capacitor	47µF	Murata	GRM32ER70J476KE20#
Capacitor	22µF	Murata	GRM31CR70J226KE19#

1.3 External VDD Mode

In External VDD mode, VDD is supplied from VCL through an external power supply. External VDD mode requires fewer components than DCDC mode and provides the flexibility of adjusting VDD independently from VCC/VCC2. However, there are additional timing requirements that must be considered.

In External VDD mode, the voltage of VDD should always be below the voltage of VCC, including the power-on and power-off sequence.

External VDD mode has specific timing requirements. Refer to the Hardware User's Manual for the specific timing requirements for the selected device.

To implement External VDD mode:

- VCC_n and VCC2_n pins: Connect each pin to the system power supply. Connect each pin to VSS_n through a 0.1 µF capacitor, with the capacitor close to each pin.
- VCC_DCDC pins: Connect each pin to the system power supply. Connect each pin to VSS_DCDC through a 0.1 µF capacitor. Place the capacitor close to the pin.
- VCLn pins: Connect each pin to the system power supply. Connect each pin to VSSn through a 0.22 µF capacitor. Place the capacitor close to the pin.
- VLO pins: Keep pin open.

Note: Many operating modes are not available when using External VDD mode. Software Standby mode, Deep Software Standby mode 1, 2, and 3, Voltage Scaling Control, and Battery Backup Function are not supported when using External VDD mode.

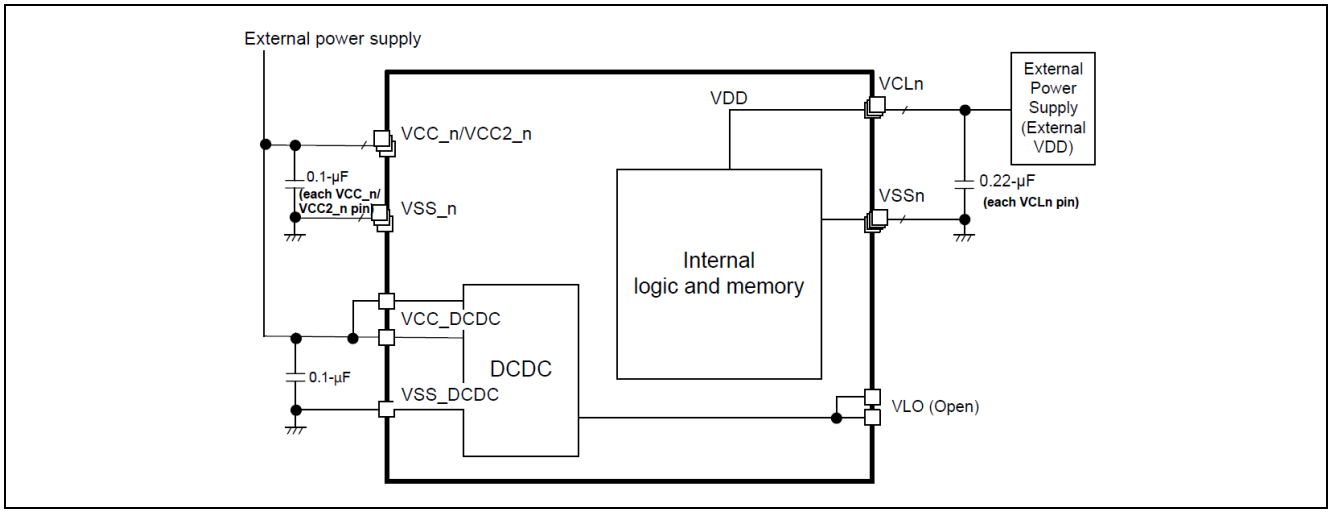


Figure 3. External VDD Mode Configuration

1.4 Voltage Scaling Control

Dynamic Voltage and Frequency Scaling (DVFS) controls low power or high-performance operation. The output voltage of the internal DCDC regulator will automatically adjust based on the selected option in the control register: VSCR for Normal mode and SVSCR for Software Standby mode. Prior to modifying the DVFS configuration, users must make clock adjustments to ensure adherence to the limitations of each DVFS mode. This feature is available only in DCDC mode.

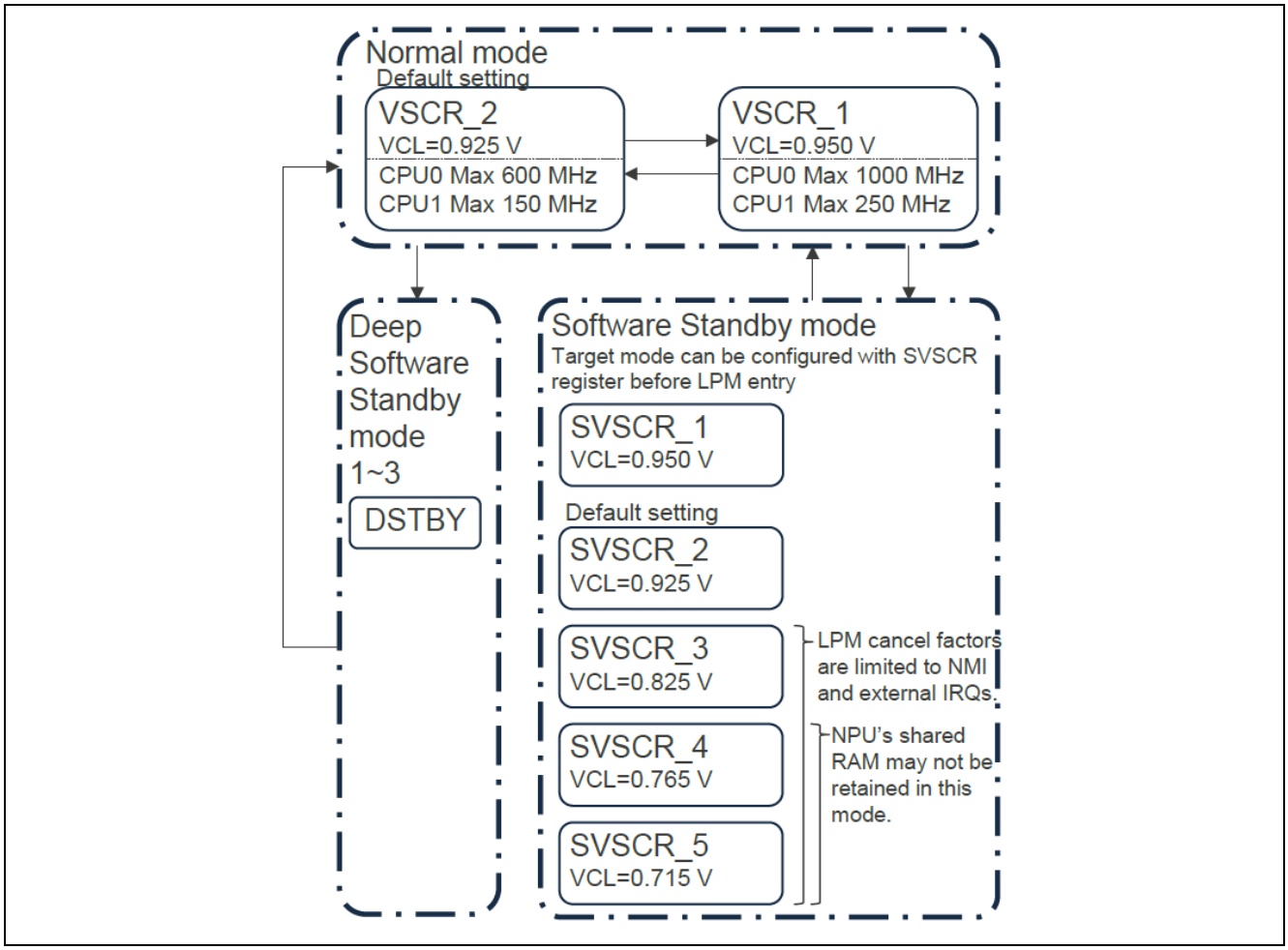


Figure 4. Voltage Scaling (DVFS) Control for Low Power or High Performance Operation

In case of External VDD mode, two voltage ranges are supported.

- Voltage range 1: VCL=0.92~0.99V (clock limit is same as VSCR_1).
- Voltage range 2: VCL=0.87~0.99V (clock limit is same as VSCR_2).

1.5 References

Further information regarding the power supply for the RA8x2 MCU Group can be found in the following documents:

R01UH1064 RA8P1 Group, RA8P1 Group User's Manual: Hardware
R01UH1065 RA8D2 Group, RA8D2 Group User's Manual: Hardware
R01UH1066 RA8M2 Group, RA8M2 Group User's Manual: Hardware
R01UH1067 RA8T2 Group, RA8T2 Group User's Manual: Hardware

The "**Overview**" chapter, lists power pins in each package with recommended bypass capacitors.

The "**Resets**" chapter discusses the Power-on reset and how to differentiate this from other reset sources.

The "**Programmable Voltage Detection**" chapter provides details on using the PVD circuit to monitors the voltage level input to the VCC pin. The detection level can be selected using software programming. The

“**Option-Setting Memory**” chapter additionally describes how to enable Voltage Detection Circuits automatically at startup.

The “**Battery Backup Function**” chapter shows how to provide battery backup to the RTC and sub-clock oscillator.

If you plan to use the on-chip Analog to Digital Converters (ADC) or the Digital to Analog Converter (DAC), see “16-Bit A/D Converter (ADC16H)” and “12-Bit D/A Converter (DAC12)” for chapters in the respective Hardware User’s Manuals for details.

Table 4. RA8x2 MCU Group, User's Manual: Hardware

Chapter Name	Description
Overview	Lists power pins in each package with notes on termination and bypassing. Refer to the Pin functions table.
Electrical Characteristics	Provides symbol parameter information, and values (with units). For e.g.: AVCC_MIPI symbol is the MIPI PHY analog power supply voltage with acceptable maximum rating between –0.3 to +4.0 volts. Typical ratings for each symbol may be identified in subsequent chapters for AC, DC, or function module specific characteristics.
Resets	Discusses the Power-on Reset and how to differentiate this from other reset sources.
Programmable Voltage Detection Circuit	Provides details on the Programmable Voltage Detection Circuit that can be used to monitor the power supply.
Low Power Modes	Using low power modes can allow you to reduce the power consumption of the MCU. See this chapter for details on how operating modes affect power supply requirements for various function modules in the MCU.
Battery Backup Function	Shows how to provide battery backup to the RTC and sub-clock oscillator
16-Bit A/D Converter 12-bit D/A Converter	If you plan to use the on-chip A/D or D/A converters, these chapters give details on how to provide filtered power supplies for these peripherals.
Clock Generation Circuit	Provides detailed descriptions on how to configure and use the available clocks, including PCB design recommendations.

2. Emulator Support

RA MCU devices support debugging using SWD or JTAG communication, and serial programming using SCI or USB FS communication.

The SWD or JTAG emulator interface should be connected to an Arm® standard 10-pin or 20-pin socket. The SWD and JTAG interface can also access the MCU boot firmware, setting up the TrustZone® boundaries. For more information on the TrustZone boundary settings on RA8x2 MCU, please reference section 7.1.2. The SWD and JTAG interface can be used to select the primary CPU (first wakeup core). CPU0 is normally the primary CPU, but CPU1 can be selected as the primary CPU under certain conditions.

To comply with the Arm specification, pull up resistors are required on the JTAG, SWD and SCI signals. Without the correct pull up resistors, the interface may not function correctly. However, RA8x2 MCU devices have internal pull up resistors that are enabled by default for these signals. When the internal pull up resistors are enabled, no external resistors are required on these signals.

Emulator support is useful for product development and prototyping but may not be needed once a design moves to production. If emulator support is no longer needed for a design, make sure to configure the ports according to the “Handling of Unused Pins” section in the I/O Ports chapter of the MCU Hardware User’s Manual. See also section 10.5 in this document. Both cores in Dual-Core products can be debugged using a single SWD/JTAG interface port and one emulator device. Users do not need two emulators for dual-core debugging.

Note that debug capability is not available while the MCU is operating in Boot mode.

2.1 SWD Interface

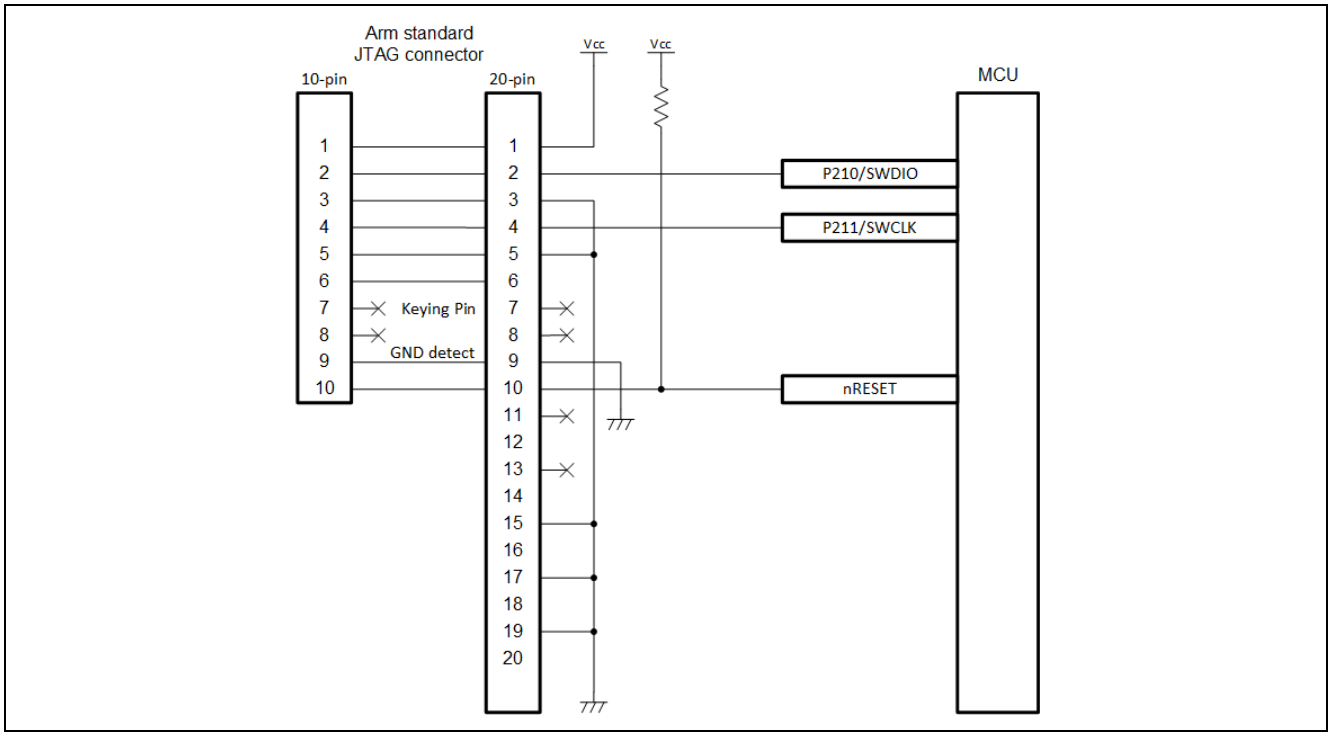


Figure 5. SWD Interface Connections

Note: 1. The output of the reset circuit of the user system must be open collector.

2.2 JTAG Interface

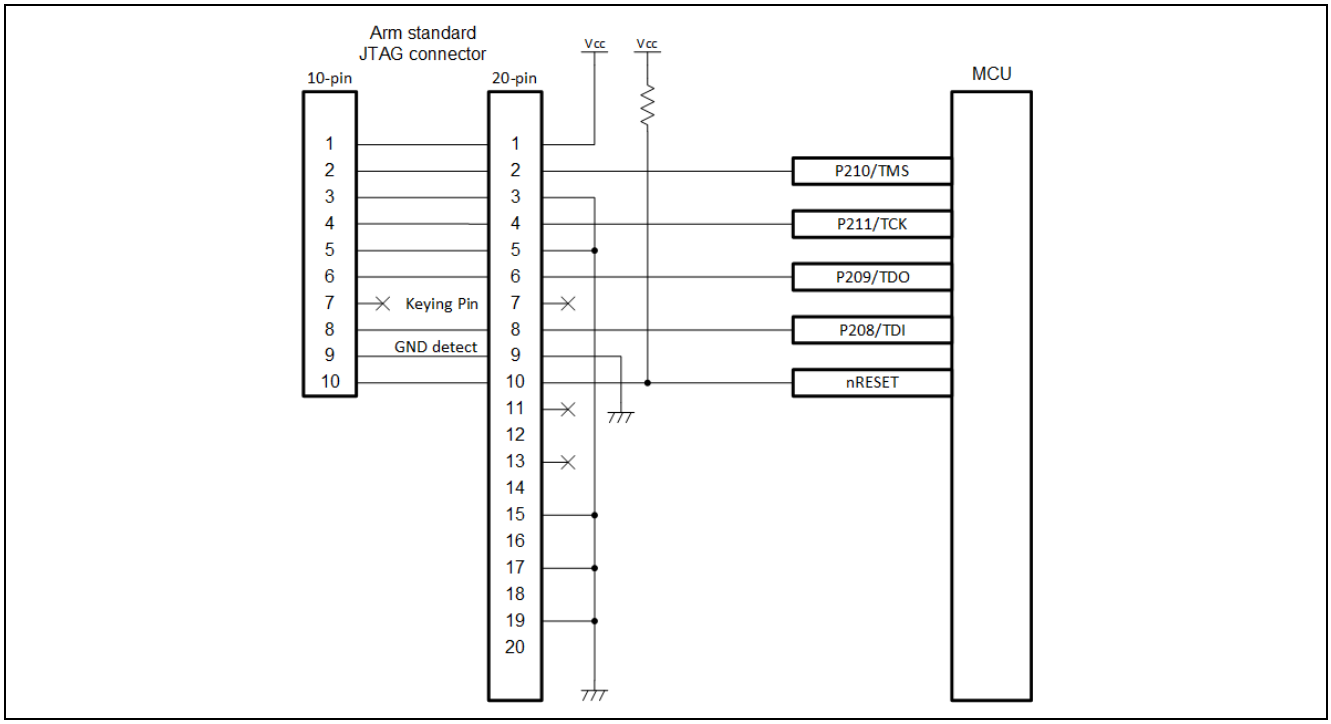


Figure 6. JTAG Interface Connections

Note: 1. The output of the reset circuit of the user system must be open collector.

2.3 Trace Data Interface

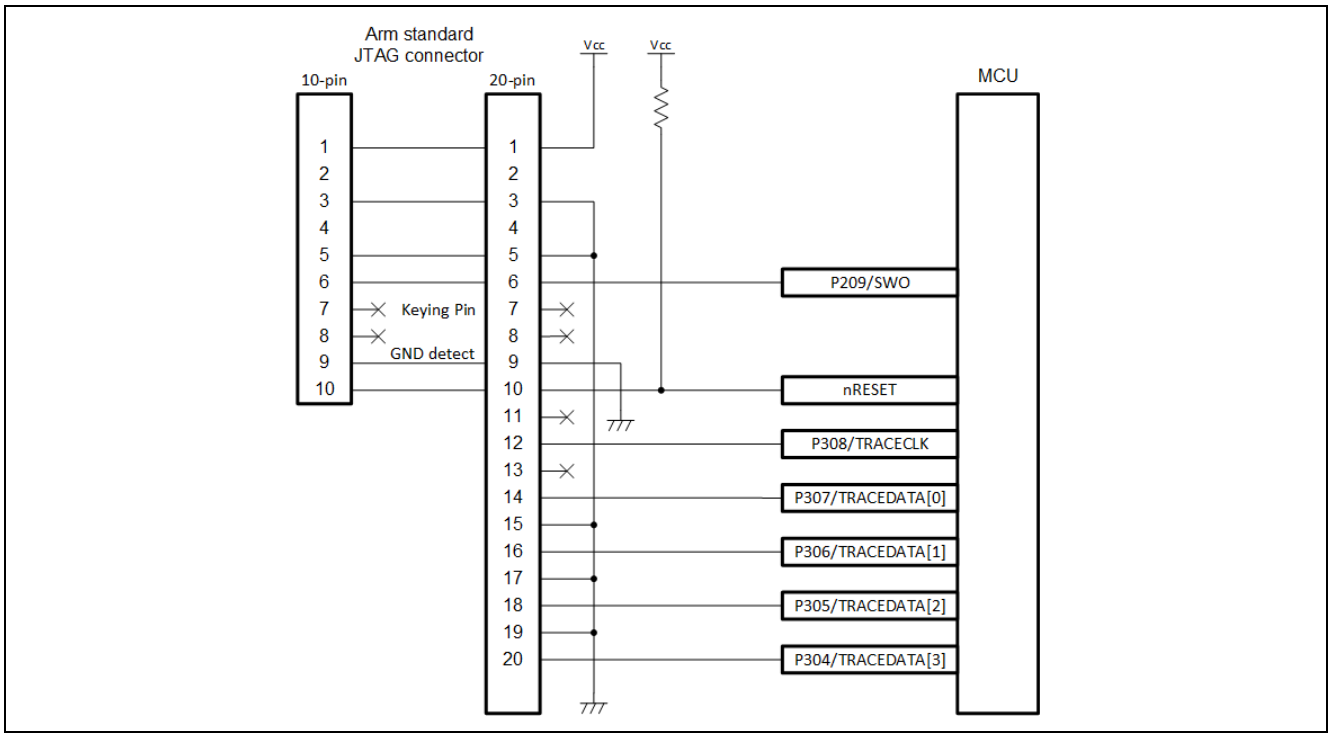


Figure 7. Trace Data Interface

The Trace Port Interface Unit (TPIU) and Serial Wire Output (SWO) provide trace output. The trace output can be sourced from either the Instrumentation Trace Macrocell (ITM) or the Arm® Embedded Trace Macrocell (ETM.). Refer to the “CoreSight ATB Funnel” section in the CPU chapter of the Hardware User’s Manual for more information.

2.4 Using SCI Boot Mode over the Emulator Interface

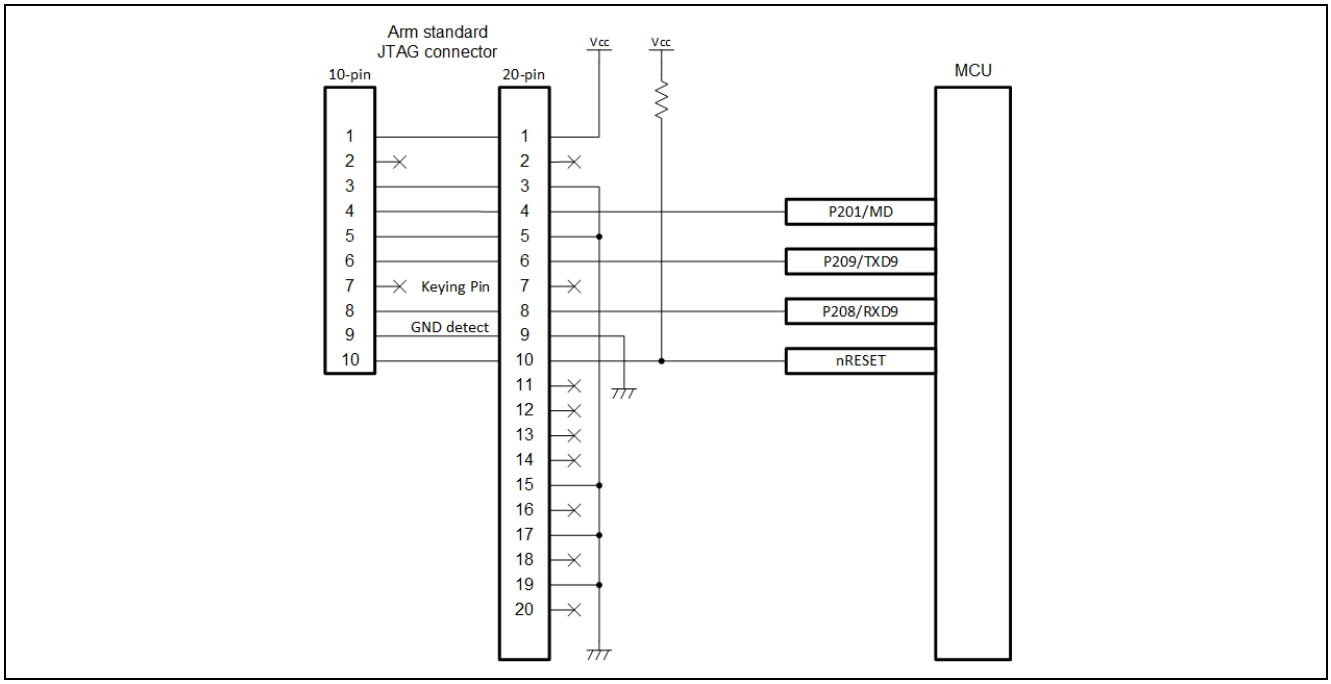


Figure 8. Serial Programming Interface using SCI Connections over Debug Connector

Notes:

1. The output of the reset circuit of the user system must be open collector.
2. To access the MCU boot mode via SCI boot pin over the JTAG connector, connect the P201/MD and the SCI TXD, RXD pin to the JTAG connector per the guideline shown in Figure 8.
3. This configuration can be helpful to simply the debug connector interface. When SCI interface is used on the JTAG connector, Renesas tools provide a convenient way to access the boot mode by controlling the MD pin through the IDE and J-Link driver. In this case, the debugger will put the MCU to boot mode by pulling the MD pin low, access the boot mode and then set the MD pin to high prior to release the MCU from Reset.
4. When SCI boot mode is accessed as a standalone hardware interface, manually pull the MD pin to low prior to accessing the boot mode.

For RA8x2 devices, the TrustZone boundaries and DLM registers may also be accessed using JTAG or SWD. SCI Boot Mode is not required to access these registers.

2.5 Multiple Emulator Interface

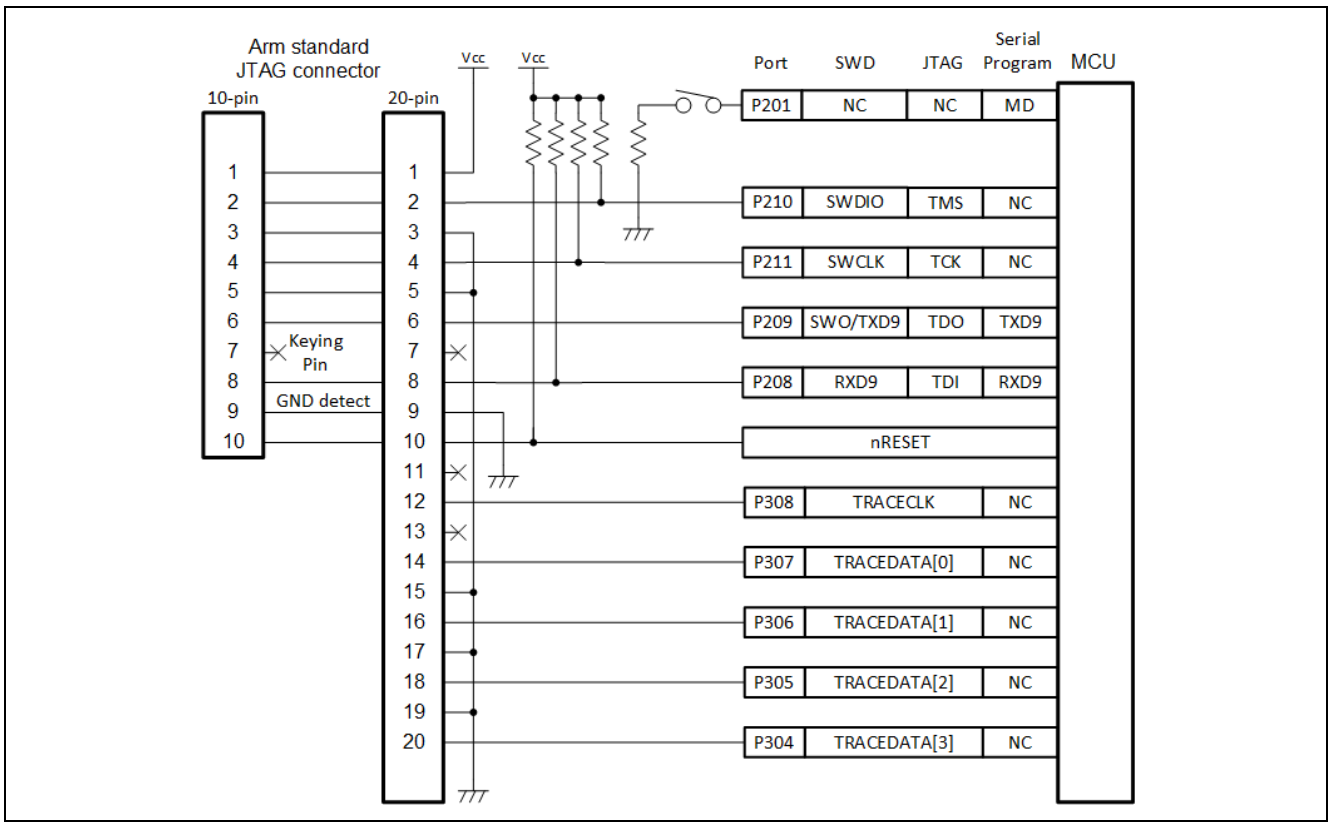


Figure 9. Multiple Emulator Interface Connections

Notes: 1. Reset circuitry on the target must be open-collector. Pull up the nRESET signal. Do not put a capacitor on this signal as it will affect the operation of the power-on reset circuit.

The connection of P201/MD to P211/SWCLK/TCK is dependent on the type of emulator used. Please refer to the User's Manual for the specific emulator for details on the required signal connections.

2.6 Software Setups for Emulator Connections

The procedure to debug RA8x2 MCU devices is documented in the application note Developing with Dual-Core RA8P1 MCU, document No. R01AN7881.

2.6.1 SWD and JTAG Interfaces

Unless Software Debug Control is disabled, SWD and JTAG pins are in default state after reset.

Table 5. SWD/JTAG Pins

Pin	P210	P209	P208	P211
Function	TMS/SWDIO	TDO/SWO	TDI	TCK/SWCLK

2.6.2 Trace Port

A 4-bit Trace Port Interface Unit (TPIU) provides trace output in RA8x2 devices.

Trace ports and clock need to be enabled before they can be used by the debugger script. When using the Trace Port functionality, avoid using the trace pins for other functions.

Table 6. Trace Ports

Pin	P304	P305	P306	P307	P308
Function	TDATA3	TDATA2	TDATA1	TDATA0	TCLK

Trace ports can also be enabled at runtime by using Pin Configurator in Renesas Flexible Software Package (FSP) but some trace data may be lost in this case.

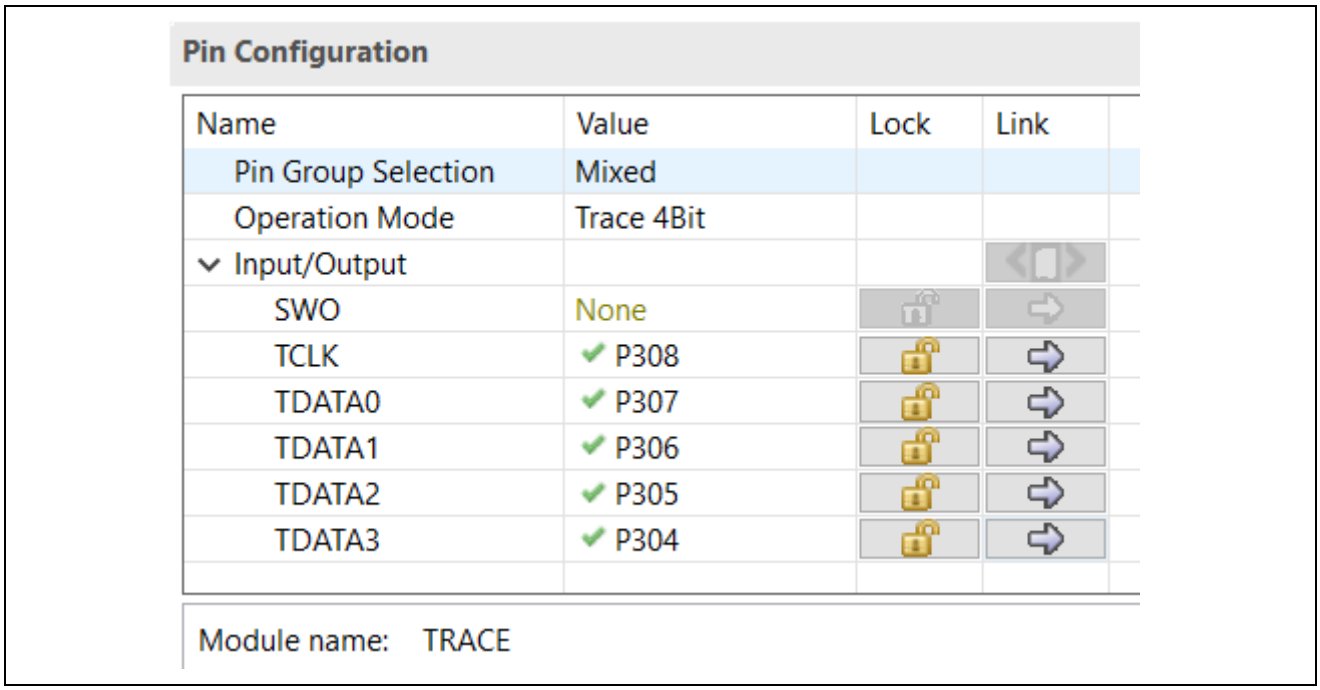


Figure 10. Enabling Trace Ports in Runtime Using FSP Configurator

3. MCU Operating Modes

The RA8x2 MCU can enter one of two modes after reset: Single-chip mode/JTAG Boot Mode or SCI/USB boot mode. The boot mode is selected by the MD pin:

Table 7. Operating Modes Available at Reset

Operating Mode	MD	On-Chip MRAM	External Bus
Single-chip mode/ JTAG Boot Mode	1	Enable	Disable
SCI/USB boot mode	0	Enable	Disable

Operating Mode Transitions as Determined by the Mode-Setting Pin.

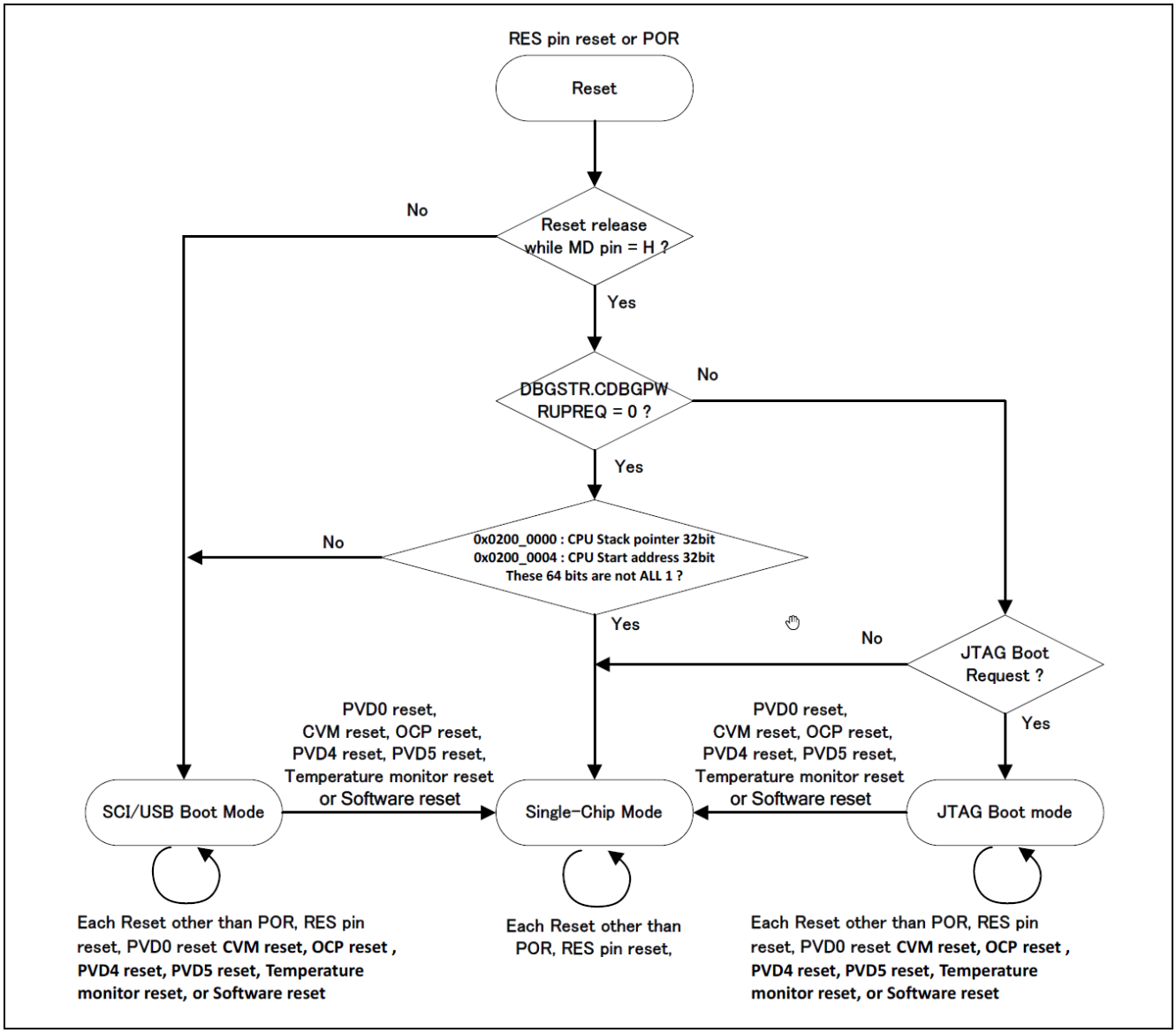


Figure 11. Mode Setting Pin level, Operating Modes and the relationship of mode transition

A typical MCU boot mode circuit includes a jumper and a couple of resistors to allow selections to connect the MD pin to VCC or Ground.

Some emulators support control of the MD pin. For emulators that do support MD pin control, tie the P201/MD pin to the SCK/TCK pin at the emulator connector. Refer to the User's Manual for the selected emulator for further details.

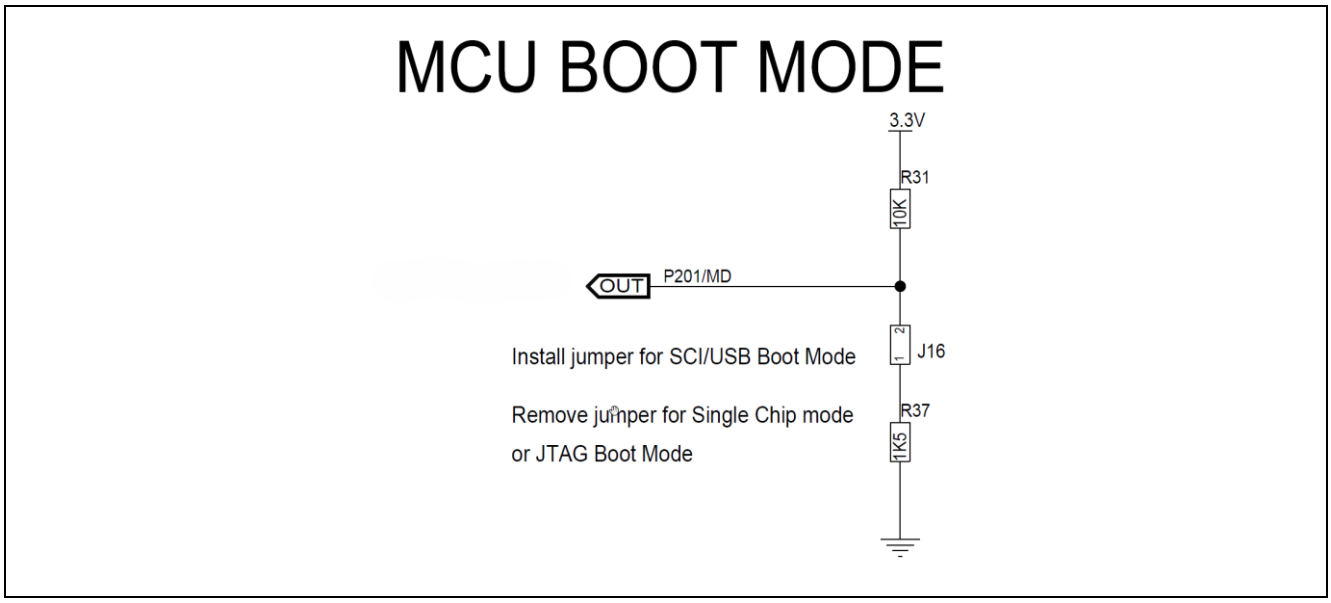


Figure 12. Typical Circuit for MCU Boot Mode Selection

4. Option Setting Memory

The option-setting memory determines the state of the MCU after a reset. It is allocated to the configuration setting area in the extra MRAM memory. The available methods of setting are different for the two areas. Option setting memory may differ in size and layout for RA8x2 devices.

The registers are detailed in the “Option Setting Memory” chapter in the Hardware User’s Manual.

The registers of option-setting memory have a discontinuous address space layout in the extra MRAM memory. Sometimes the registers may be located in a portion of the MRAM memory which is near reserved areas in the extra MRAM. It is possible that some customers may inadvertently store data in the registers of option-setting memory or write into the reserved area in the extra MRAM. This may result in improper functionality. It is advised that the user check to ensure that no unwanted data is written to these locations prior to programming the extra MRAM by reviewing the map file or s-record files generated by the compiler upon creation of the binary. For instance, settings in the MRAM option registers can enable the Independent Watchdog Timer (IWDT) immediately after reset. If data stored in program the extra MRAM inadvertently overlaps the option setting memory register, it is possible to turn on the IWDT on without realizing it. This may cause the debugger to have communications problems with the board. Additionally, Security attribution of extra MRAM option-setting memory can impact which value is applied at runtime, so user must confirm required values are programmed into the option-setting memory.

The image below shows an example from the option setting memory which includes the option function select registers on RA8x2.

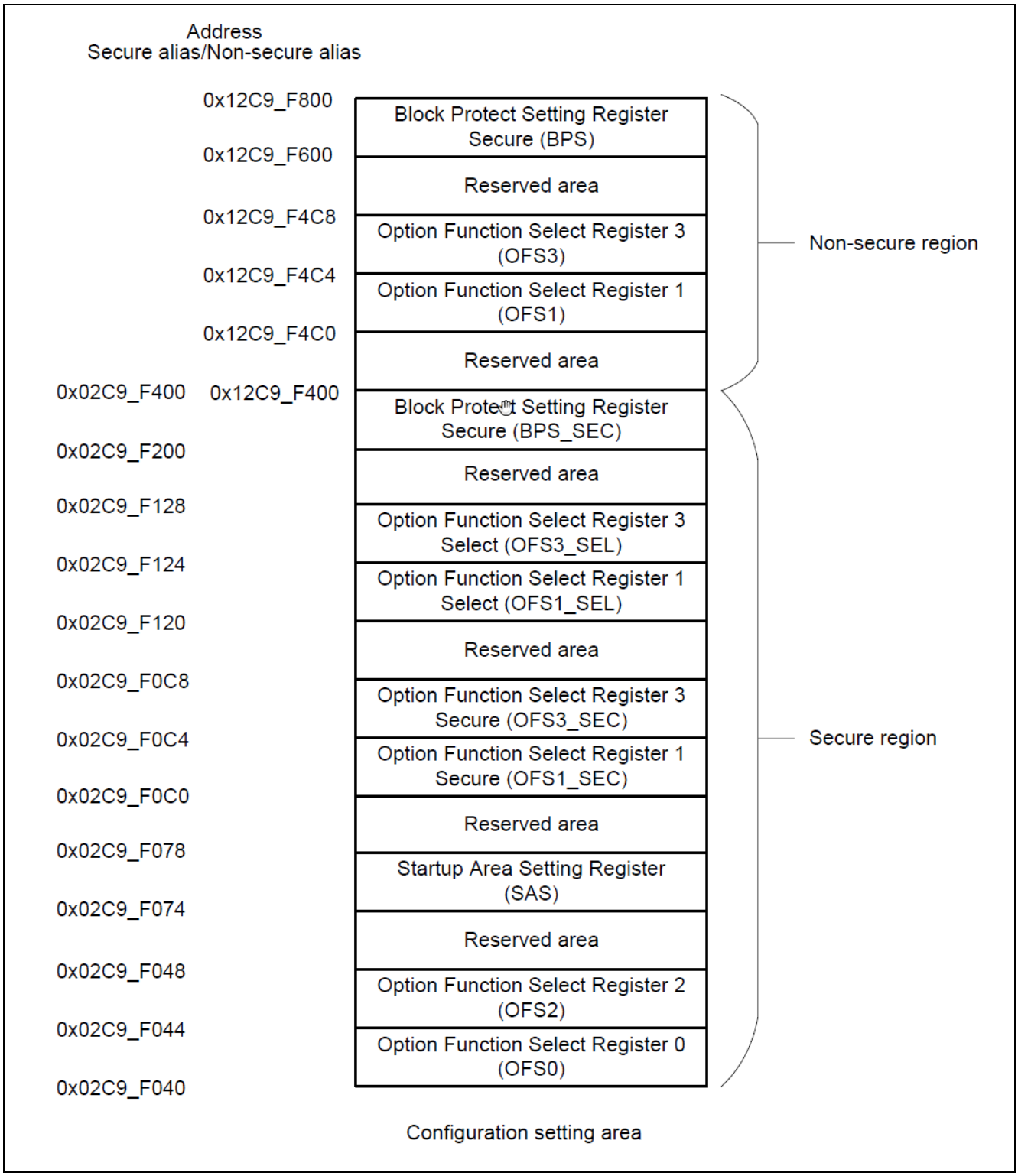


Figure 13. Option Function Select Registers Shown in Memory Map Example

4.1 Option Setting Memory Registers

Following is a summary of the Option Setting Memory registers. Make sure that they are configured properly before first programming of the MCU for startup. The OFSx_SEL register selects the security attribute. Which security attribute is applied, OFS or OFS_SEC, is determined by the setting value of the corresponding bit in the OFSx_SEL register. To check the configuration, review the map file, and output file (using hex or S-Record format) to confirm the values programmed into the Option Setting Memory registers.

MCU Sub-system Control Settings

- OFS0 register
  - Independent Watchdog Timer (IWDT) auto start
  - IWDT timeout, frequency, windowing, interrupt type, and low power mode behavior
  - Watchdog Timer (WDT0) auto start
  - WDT0 timeout, frequency, windowing, interrupt type, and low power mode behavior
- OFS1, OFS1_SEC registers
  - PVD0 startup settings after reset
  - HOCO startup settings after reset
  - Software Debug Control
  - Set ECC function of TCM and CACHE.
- OFS1_SEL register
  - Select settings between OFS1 and OFS1_SEC registers
- OFS2 register
  - DCDC enable at reset.
  - CVM reset enable.
  - NPU Security/Privilege Attribution initial value.
  - External VDD voltage select for SRAM.
- OFS3, OFS3_SEC registers
  - Watchdog Timer (WDT1) auto start
  - WDT1 timeout, frequency, windowing, interrupt type, and low power mode behavior
- OFS3_SEL register
  - Select settings between OFS3 and OFS3_SEC registers

Code MRAM Memory Setting

- SAS register
  - Startup area selection.
- BPS, BPS_SEC registers
  - Invalidates the writing to the code MRAM memory.
- PBPS, PBPS_SEC registers
  - Invalidates writes to bits of BPS and BPS_SEC. Once the bit of this register is set to 0, it is impossible to change the bit to 1.
- POFSPS register
  - Invalidates the programming and erasure to the option function select area

Firmware Verification and Update Control Registers

- FSBLCTRLx registers
  - FSBLCTRL0 configures whether to enable FSBL and selects the FSBLClock speed.
  - FSBLCTRL1 configures the types of validation on the user application which resides in the beginning of MRAM memory and whether to report the boot status: Secure Boot or CRC.
  - FSBLCTRL2 chooses the Port and Pin number used to report the FSBL error status.
  - These registers can be set up using the BSP stack configurations as shown in Figure 14.
- SACCx register
  - Configure the locations in MRAM memory where the code certificates are stored for validating the application when FSBL is operating in Secure Boot.
  - SACC0/1 is selected by the FSBL based on the MCU Start Area Selection and the Bank mode. User can reference the “Secure Boot” section in the Hardware User’s Manual to understand this selection process.

  - The code certificate is signed using the private key of the OEM_BL verification key pair. Its format is Renesas proprietary. The content of the code certificate is described in the table “Detail of code certificate” in the Hardware User’s Manual.
- SAMR register
  - FSBL stores the measurement report to the SRAM address specified by the SAMR register
- REVOKE register
  - This register can be programmed only by MACI command. Sets the configuration for OEM_ROOT_KEY revoke.
- HOEMRTRKn register
  - This register can be programmed only by the MCU boot firmware. It is programmed after a code image is validated. This register contains a processed Hash value.
- ZHUK register
  - This register can be programmed only by MACI command. Enables or disables W-HUK zeroization based on RTCICn tamper event or MREZC.
- ARCLS register
  - Controls Anti Rollback Counter Lock functionality. Note that once this lock functionality is enabled, it can never be reverted. This register can be set using the Renesas Flash Programming (RFP) tool.
- ARCCS register
  - Configures the Anti Rollback Counter operation for the non-secure application. Note that when this configuration is disabled, the “Increment Counter” or “Read Counter” command cannot be issued anymore. This register can be set using the Renesas Flash Programming (RFP) tool.
- ARC_SECn registers
  - Anti-Rollback Counter for Secure Application
- ARC_NSECn registers
  - Anti-Rollback Counter for Non-Secure Application
- ARC_OEMBLn registers
  - Anti-Rollback Counter for OEMBL
- GPOTPn
  - General Purpose One Time Programmable (OTP) register. Secure user can write any value.

Renesas FSP Configurator supports setting of option memory in BSP settings, as shown in the following figure for RA8x2 MCU.

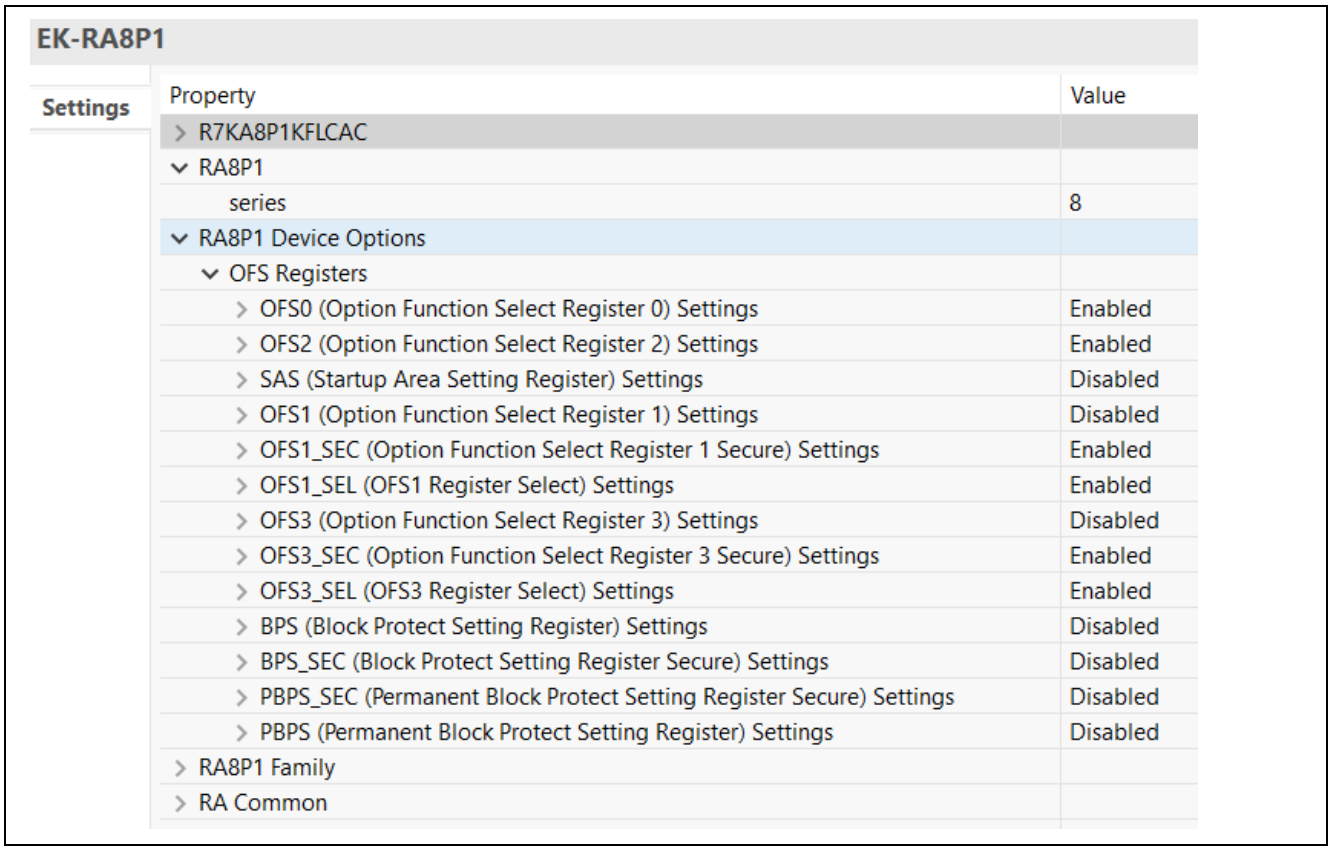


Figure 14. Example of Option Memory Settings in FSP Configuration for RA8x2 MCU

5. Clock Circuits

The RA8x2 MCUs have five primary oscillators. Four of these may be used as the source for the main system clock. In typical systems, requiring higher clock accuracy, the main clock is driven with an external crystal or external clock input. This input is directed to PLLn (n=1,2) where it is eventually multiplied or divided to the PLL clock, then supplied into the CPU clocks (CPUCLKn), system clock (ICLK), Neural Processing Unit clock (NPUCLK), memory clocks (MRICLK, MRPCLK), peripheral module clocks (PCLKn), external bus clock, and trace clock. Additional selectors and frequency dividers are used to generate the clocks for each peripheral. Refer to the Hardware User's Manual "Clock Generation Circuit" chapter for the "Clock generation circuit block diagram".

Each clock has specific tolerances and timing values. Refer to the Hardware User's Manual "AC Characteristics" section in the "Electrical Characteristics" chapter for the Frequency and Clock Timing specifications. Refer to the Hardware User's Manual "Clock Generation Circuit" chapter for the relationship between the various clock frequencies.

Table 8. RA8x2 Oscillators

Oscillator	Input Source	Frequency	Primary Uses[1]
Main clock (MOSC)	External crystal/resonator (EXTAL, XTAL)	8 MHz to 48 MHz	CPU0 clock, CPU1 clock, PLL1 input, PLL2 input, NPU clock, MRAM clock, system clock, external bus clock, peripheral clocks, trace clock, USB clocks
	External clock (EXTAL)	Up to 48 MHz	
Sub-clock (SOSC)	External crystal/resonator (XCIN, XCOUT)	32.768 kHz	Real-time clock, system clock in low power modes, CLKOUT, AGT clock, CAC clock, ULPT clock
	External clock (EXCIN)		
High-speed on-chip (HOCO)	On-chip oscillator	16/18/20/32/48 MHz	CPU0 clock, CPU1 clock, PLL1 input, PLL2 input, NPU clock, MRAM clock, system clock, external bus clock, peripheral clocks, trace clock
Middle-speed on-chip (MOCO)	On-chip oscillator	8 MHz	System clock at startup, CLKOUT, PDMIF clock, CAC clock, SysTick timer clock
Low-speed on-chip (LOCO)	On-chip oscillator	32.768 kHz	Main system clock in low power modes (Software Standby and Deep Software Standby Mode 1), and during main oscillator stop detection, AGT clock, CAC clock, Real-time clock, ULPT clock, IWDT clock

Note: 1. Additional uses may be available. See the MCU Hardware User's Manual for details.

5.1 Reset Conditions

After reset, RA8x2 MCUs begin running with the middle-speed on-chip oscillator (MOCO) as the main clock source. At reset, the main oscillator and PLL1 and PLL2 are off by default. The HOCO and IWDT operation may be on or off depending on the settings in the Option Setting Memory (see section 2).

5.2 Clock Frequency Requirements

The bits specifying the frequency of various clocks are specified in the respective sub-sections of the Internal Clock section in the Clock Generation Circuits chapter of the MCU hardware user's Manual. Verifying the values in the registers specified in the sub-sections can help confirm if the clock generation circuit has been set up correctly to generate the desired clock frequencies.

The "Overview" section of the "Clock Generation Circuit" chapter in the Hardware User's Manual details the specifications for all the clock sources, including the available frequency ranges. The table "Clock generation circuit specifications for the internal clocks" in that section provides the details. For the MCU to operate correctly, users must adhere to the notes for that table. Additional details can be found in the "AC Characteristics" section of the "Electrical Characteristics" chapter in the MCU Hardware User's Manual.

Table 9 provides an overview of the maximum possible clock frequencies for each clock and minimum clock frequencies where applicable. The maximum frequency for some clocks may be limited by the device configuration. See the MCU Hardware User's Manual for more details.

Table 9. Frequency Range for RA8P1 MCU Internal Clocks

Clock Frequency [MHz]	CPUCLK0 [1]	CPUCLK1 [1]	NPUCLK [1]	ICLK [1]	MRICLK	MRPCLK	
Maximum	1000	250	500	250	250	125	
Minimum	—	—	—	—	—	—	

1. Maximum CPUCLK0, CPUCLK1, NPUCLK, ICLK, MRICLK, and MRPCLK frequency depends on the package type and operating junction temperature.
PLLCCR.PLSRCSEL bit must be set to 0 when CPUCLK0 is set to over 960 MHz.
While open process of the API of security engine (R_RSIP_Open) is running, the CPUCLK0 frequency and the ICLK frequency must be the same.

Clock Frequency [MHz]	PCLKA [2]	PCLKB [2]	PCLKC [3]	PCLKD [2]	PCLKE [2]	BCLK	TRCLK
Maximum	125	62.5	125	250	250	125	125
Minimum	—	—	—	—	—	—	—

2. Maximum PCLKA, PCLKB, PCLKD, PCLKE, BCLK frequency depends on the package type and operating junction temperature.
3. PCLKC is not used.

Clock Frequency [MHz]	EBCLK	SDCLK	SCI CLK	SPICLK	OCTACLK	CANFD CLK [4]	ADCCLK
Maximum	60	133	120	333.33	333.33	80	120
Minimum	—	—	—	—	—	—	

4. If the CAN-FD is used, clock frequency ratio is constrained to be PCLKA:PCLKE = 1:2.

Clock Frequency [MHz]	GPTCLK	LCDCLK	LCDCLK pins [6]	USBCLK	USB60 CLK	I3CCLK
Maximum	300	240	54/60	48	60	200
Minimum		—	—	—	—	48

Clock Frequency [MHz]	BCLKA	ETHPHY CLK	CLKOUT	MIPIMCLK			
Maximum	133	25/50	60	48			
Minimum	—			—			

5.2.1 Requirements for USB Communications

The USB 2.0 Full-Speed Module (USBFS) and USB 2.0 High-Speed Module (USBHS) available on some members of the RA family require a 48 MHz USB clock signal (USBCLK). A 60 MHz clock must be supplied when using the USBHS module in classic-only mode (CL-only mode).

When not operating in CL-Only mode, the main oscillator (MOSC) is used as the source for the operating clock for USB-PHY clock (USBMCLK). When the USB-PHY clock is generated from an external source, the main clock oscillator frequency is restricted to either 12 MHz, 20 MHz, 24 MHz or 48 MHz. This is due to two factors. (1) There are limited multiplication and division ratios available in the PLL circuit in the USBHS

module. (2) A 48 MHz clock input is required by the USB Function modules. USBMCLK does not need to be supplied when operating in CL-Only mode.

When operating in CL-Only mode, there are two internal operating clocks:

- The USB clock USBCLK is a 48 MHz clock that must be supplied when using the USBFS module, or when using the USBHS in Classic-Only (CL-Only) mode. USBCLK is not required for USBHS when USBHS is not operating in CL-Only mode.
- The USB clock USB60CLK is a 60 MHz clock that must be supplied when using the USBHS module in CL-Only mode. USB60CLK is not required for USBHS when USBHS is not operating in CL-Only mode.

5.2.2 Requirements for Ethernet Controller

When the Ethernet Switch Module (ESWM) is used, the following conditions must be met:

ICLK x 1.5 ≥ ESWCLK

ESWCLK > PCLKA

ICLK > PCLKA

5.2.3 Requirements for SDRAM Controller

The SDCLK is sourced from the external bus BCLK or BCLKA. Do not set SDCLK to a frequency higher than the system clock (ICLK).

5.2.4 Requirements for MIPI D-PHY

The clock for the MIPI PHY (MIPIMCLK) is sourced directly from the Main clock oscillator. PCLKA is used for the reference clock of the counter circuit which controls the internal timing of the D-PHY module. PCLKA must be 40 MHz or more to use the MIPI function. The operating power control mode of the system must also be set to high-speed mode.

5.3 Lowering Clock Generation Circuit (CGC) Power Consumption

To aid in saving power, set the dividers for any unused clocks (for example, BCLK) to the highest possible value whenever possible. Also, if not using a clock then make sure that it has been stopped by setting the appropriate register(s). The registers for controlling each clock source are shown in the following table.

Table 10. Clock Source Configuration Registers

Oscillator	Register	Description
Main clock	MOSCCR	Starts/stops main clock oscillator
Sub-clock	SOSCCR	Starts/stops sub-clock oscillator
High-speed on-chip (HOCO)	HOCOCR	Starts/stops HOCO
Middle-speed on-chip (MOCO)	MOCOCR	Starts/stops MOCO
Low-speed on-chip (LOCO)	LOCOCR	Starts/stops LOCO

5.4 Writing the System Clock Control Registers

Care should be taken when writing to the individual bit fields in the System Clock Division Control Register (SCKDIVCR), System Clock Division Control Register 2 (SCKDIVCR2), and System Clock Source Control Register (SCKSCR). Details can be found in the section “Clock Setting” section of the Hardware User’s Manual.

Make sure to follow the procedures outlined in the “Clock Setting" section of the Hardware User’s Manual. The example procedures ensure that changing the clock frequency does not interfere with normal processing.

When changing the value of SCKSCR to a different clock source, follow the procedure in the “Clock Setting” section of the Hardware User’s Manual. The example procedure ensures that the clock oscillation output is stable before operation continues.

The recommended method to measure the wait time is to do so in software by counting instruction cycles. Be sure to consider the worst-case use conditions to ensure that the required wait time elapses.

5.5 Clock Setup Example

Renesas FSP provides a simple, visual clock configuration tool for RA8x2 MCUs shown as follows. The tool will highlight if setting conditions violate specifications for the device.

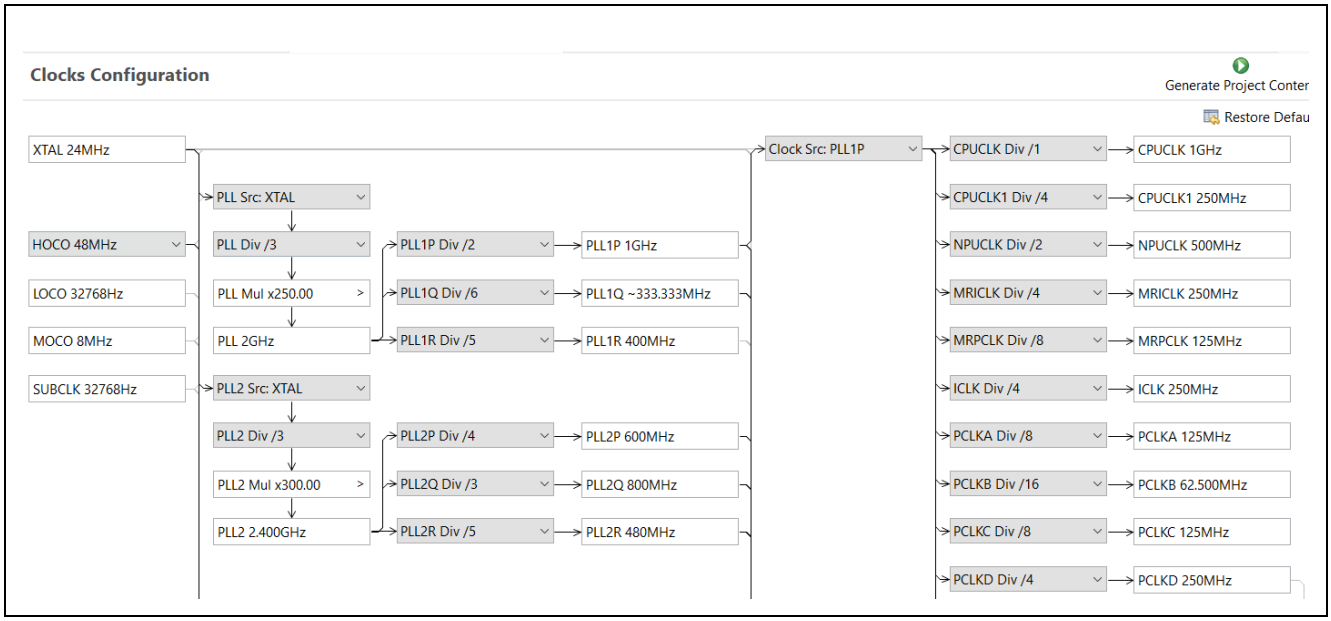


Figure 15. Clock Settings Using Renesas FSP Configurator Example

5.6 HOCO Accuracy

The internal high-speed on-chip oscillator (HOCO) runs at 16 MHz, 18 MHz, 20 MHz, 32 MHz, or 48 MHz with an accuracy of ±1.8 % or better. The accuracy of the HOCO may be improved by enabling the Frequency Locked Loop (FLL) function, which results in a clock accuracy of ±0.25% or better. Refer to the Electrical Specifications in the hardware manual for details.

The HOCO may be used as an input to the PLL circuits. When the HOCO is used this way, no external oscillator is required. This may be an advantage when space constraints or other limitations require a reduced component count in a PCB design. However, there are performance tradeoffs and limitations due to the clock accuracy which should be evaluated for your application.

5.7 Board Design

Refer to the “Usage Notes” section of the Clock Generation Circuit (CGC) chapter in the Hardware User’s Manual for more information on using the CGC and for board design recommendations.

In general, place the crystal resonator and its load capacitors as close to the MCU clock pins (XTAL/EXTAL, XCIN/XCOUT) as possible. Avoid routing any other signals between the crystal resonator and the MCU. Minimize the number of connecting vias used on each trace.

5.8 External Crystal Resonator selection

An external crystal resonator may be used as the main clock source. The external crystal resonator is connected across the EXTAL and XTAL pins of the MCU. The frequency of the external crystal resonator must be in the frequency range of the main clock oscillator.

Selection of a crystal resonator will be largely dependent on each unique board design. Due to the large selection of crystal resonators available that may be suitable for use with RA8x2 MCU devices, carefully evaluate the electrical characteristics of the selected crystal resonator to determine the specific implementation requirements.

The following diagram shows a typical example of a crystal resonator connection.

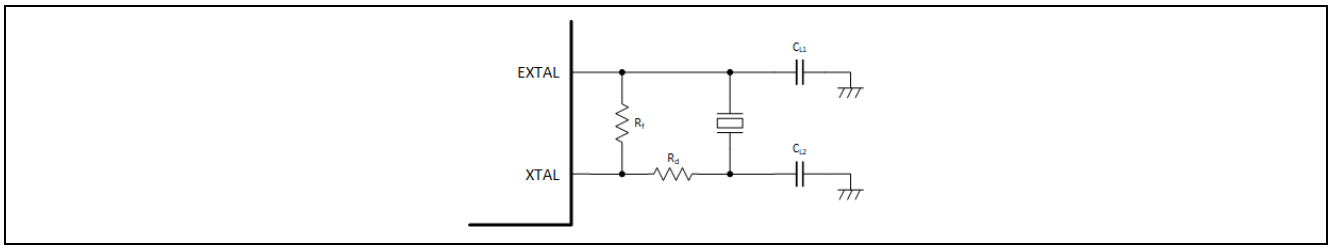


Figure 16. Example of Crystal Resonator Connection

Careful evaluation must be used when selecting the crystal resonator and the associated capacitors. The external feedback resistor (R_f) and damping resistor (R_d) may be added if recommended by the crystal resonator manufacturer.

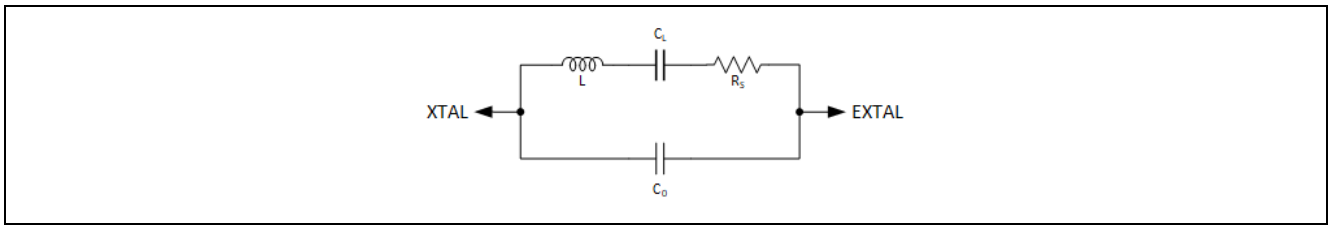


Figure 17. Equivalent Circuit of the Crystal Resonator

Selection of the capacitor values for CL1 and CL2 will affect the accuracy of the internal clock. To understand the impact of the values for CL1 and CL2, the circuit should be simulated using the equivalent circuit of the crystal resonator in the figure above. For more accurate results, also take into account the stray capacitance associated with the routing between the crystal resonator components.

For more information on designing clock circuits for RA devices, please see the Application Note “Renesas RX and RA Families Design Guide for Main Clock Circuits and Sub-Clock Circuits” available at www.renesas.com.

6. Reset Requirements and the Reset Circuit

There are twenty-one types of resets.

Table 11. Example of RA8P1 Resets

	Reset Name	Source
System reset	Pin reset	RES# pin is driven low
	Power-on reset	VCC rises (voltage detection: VPOR)
	Voltage monitor 0 reset	VCC falls (voltage detection Vdet0)
	Voltage monitor 1 reset	VCC rises/falls (voltage detection Vdet1)
	Voltage monitor 2 reset	VCC rises/falls (voltage detection Vdet2)
	Voltage monitor 4 reset	VCC rises/falls (voltage detection Vdet4)
	Voltage monitor 5 reset	VCC rises/falls (voltage detection Vdet5)
	Core Voltage monitor reset	VDD falls (voltage detection Vdet_VDDL) VDD rises (voltage detection Vdet_VDDH)
	Temperature monitor reset	Detect temperature outside a certain range
	Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh occurs
	CPU0 Watchdog timer reset	The watchdog timer underflows, or a refresh error for CPU0 occurs
	CPU1 Watchdog timer reset	The watchdog timer underflows, or a refresh error for CPU1 occurs
	CPU0 Lockup reset	This reset is generated when CPU0 encounters lockup
	CPU1 Lockup reset	This reset is generated when CPU1 encounters lockup
	Bus Error reset	A bus error occurred (MSAU, MMPU, Illegal address, TrustZone® Filter, Slave Bus, or Bufferable write errors)
	Common Memory Error reset	RAM error (ECC error of SRAM)
	Local Memory 0 error reset	RAM error (ECC error of CPU0 Cache or TCM)
	Local Memory 1 error reset	RAM error (ECC error of CPU1 Cache or TCM)
	Deep software standby reset	Deep software standby mode is canceled by an interrupt
	Software reset	Register setting (use the software reset bit AIRCR.SYSRESETREQ)
VBATT_POR reset		VBATT_R voltage drop detection below $V_{PORBATT}$

These resets are described briefly in the following sections. For further details, refer to the “Operation” sub-section of the “Resets” chapter in the MCU Hardware User’s Manual.

6.1 Pin Reset

When the RES# pin is driven low, all processing is aborted and the MCU enters a reset state. To reset the MCU while it is running, RES# should be held low for the specified reset pulse width. Refer to the “Reset Timing” section of the “Electrical Characteristics” chapter of the Hardware User’s Manual for more detailed timing requirements. Also refer to section 2 of this document, “

Emulator Support" for details on reset circuitry in relation to debug support. Additional details may also be found in the User's Manual for each emulator. For example, details for using the Renesas E2 Emulator can be found in "E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RA Devices)" (Document # R20UT4686).
There is no need to use an external capacitor on the RES# line because the POR circuit holds it low internally for a good reset and a minimum reset pulse is required to initiate this process.

6.2 Power-On Reset

There are two conditions that will generate a power-on reset (POR):

1. If the RES# pin is in a high-level state when power is supplied.
2. If the RES# pin is in a high-level state when VCC is dropped below V_{POR}.

After VCC has exceeded the power on reset voltage (V_{POR}) and the power-on reset time (t_{POR}) has elapsed, the chip is released from the power-on reset state. The power-on reset time is a period that allows for stabilization of the external power supply and the MCU. Refer to the "POR and PVD Characteristics" section of the "Electrical Characteristics" chapter of the Hardware User's Manual for voltage level and timing details.

Because the POR circuit relies on having RES# high concurrently with VCC, don't place a capacitor on the reset pin. This will slow the rise time of RES# in relation to VCC, preventing the POR circuit from properly recognizing the power-on condition.

If the RES# pin is high when the power supply (VCC) falls to or below V_{POR} , a power-on reset is generated. The chip is released from the power-on state after VCC has risen above V_{POR} and the t_{POR} has elapsed.

After a power on reset, the PORF bit in RSTSR0 is set to 1. Following a pin reset, PORF is cleared to 0.

6.3 Voltage-Monitor Resets

The RA8x2 group includes circuitry that allows the MCU to protect against unsafe operation during brownouts. On-board comparators check the supply voltage against multiple reference voltages, for example, V_{det0}, V_{det1}, V_{det2}, and so forth. As the supply dips below each reference voltage an interrupt or a reset can be generated. The detection voltage V_{det0} is selectable from 8 different levels. V_{det1}, V_{det2}, V_{det4}, and V_{det5} are each selectable from 16 different levels.

When VCC subsequently rises above V_{det0}, V_{det1}, V_{det2}, V_{det4}, or V_{det5}, release from the voltage-monitor reset proceeds after a stabilization time has elapsed.

Low Voltage Detection using V_{det0} after a reset may be enabled or disabled by setting the OFS1.PVDAS register bit.

Low Voltage Detection using V_{det1} or V_{det2} is disabled after a power on reset. Voltage monitoring can be enabled, and the detection voltage set by using the PVDmCMPCR register. For more details, see the chapter "Programmable Voltage Detection (PVD)" in the Hardware User's Manual.

Low Voltage Detection using V_{det4} or V_{det5} is disabled after a power on reset. Voltage monitor can be enabled, and the detection voltage set by using the PVDnCMPCR register. For more details, see the chapter "Programmable Voltage Detection (PVD)" in the Hardware User's Manual.

After a PVD Reset, the PVDnRF (n = 0, 1, 2, 4, 5) bit in RSTSR0 is set to 1.

6.4 Core Voltage Monitor Reset

This is an internal reset generated by the Core Voltage Monitor circuit.

If the Core Voltage Detection Circuit disable bit (CVMRDIS) in Option Function Select Register 0 (OFS2) is 0 (Core Voltage monitor reset is enabled after a reset) and VDD falls below Vdet_VDDL or rises above Vdet_VDDH, the RSTSR3.CVMRF flag is set to 1 and the Core Voltage detection circuit generates a Core Voltage monitor reset.

6.5 Temperature Monitor Reset

Temperature monitor reset is an internal reset generated by temperature sensor (TSN). TSN detects ambient temperature in the surrounding environment of the MCU. If it is outside from the specified range, a reset is issued. For more details, see the section "Temperature Monitor Reset" in the chapter "Reset" in the Hardware User's Manual.

6.6 Independent Watchdog Timer Reset

This is an internal reset generated by the Independent Watchdog Timer (IWDT).

When the IWDT underflows, an independent watchdog timer reset is optionally generated (NMI can be generated instead) and the underflow bit UNDFF in the IWDT Status register IWDTSR is set to a 1. The reset signal is output for one count cycle, then the IWDT reset is released.

6.7 CPU Watchdog Timer Resets

These are internal resets generated by the CPU Watchdog Timers (WDT0 for CPU0 and WDT1 for CPU1).

When the CPU WDT overflows, a watchdog timer reset is optionally generated (NMI can be generated instead), and the underflow flag bit in WDT Status Register WDTSR is set to a 1. The reset signal is output for one count cycle, then the WDT reset is released.

6.8 Deep Software Standby Reset

This is an internal reset generated when Deep Software Standby mode is canceled by an associated interrupt.

The Deep Software Standby reset is canceled after tDSBY (return time after Deep Software Standby mode cancellation) elapses. At the same time, Deep Software Standby mode is also canceled. When tDSBYWT (wait time after Deep Software Standby mode cancellation) elapses after Deep Software Standby mode has

been canceled, the internal reset is canceled, and the CPU starts the reset exception handling. For details of the deep software standby mode refer to the “Low Power Modes” chapter in the Hardware User’s Manual.

After a Deep Software Standby Reset, an interrupt can be generated, and the corresponding flag in the DPSIFRn register is set to 1.

6.9 Software Reset

This is an internal reset generated by writing a 1 to the SYSRESETREQ bit in the AIRCR register. When using software reset, make sure that the watchdogs are serviced first before issuing the software reset command.

When a software reset is generated, the SWRF bit in RSTSR1 is set to a 1. After a short delay (tRESW2) the internal software reset is canceled, and the CPU starts the reset exception handling.

6.10 Bus Error Reset

Bus error reset is an internal reset generated by buses.

It is an integrated reset generated by buses, and it consists of the following: Bus error reset (MSAU error, MMPU error, Illegal address error, Slave TrustZone Filter error, Slave Bus error, Bufferable write error).

Output of Bus error reset can be selected by MSAOAD.OAD or MMPUOAD.OAD or BUSOAD.ILERROAD or BUSOAD.SLERROAD or BUSOAD.BWERROAD.

6.11 Common Memory Error Reset

Common memory error reset is an internal reset generated by SRAMs.

It is an integrated reset generated by SRAMs and consists of the following: SRAM error reset (ECC error).

Output of SRAM error reset can be selected by SRAMCRn.OAD.

6.12 Local Memory 0 / Local Memory 1 Error Resets

Local memory 0/ Local memory 1 error reset is an internal reset generated by SRAMs.

It is an integrated reset generated by SRAMs, and it consists of the following:

- Local Memory 0 error reset (ECC error of CPU0 Cache or TCM)
- Local Memory 1 error reset (ECC error of CPU1 Cache or TCM)

Output of Local Memory 1 error reset can be selected by CAPOAD.OAD and TCMCRC.OAD or TCMCRS.OAD

6.13 Other Resets

Most peripheral functions within the MCU can generate a reset under specific fault conditions. No hardware configuration is required to enable these resets. Refer to the relevant chapters in the Hardware User's Manual for details of the conditions that will generate a reset for each peripheral function.

6.14 Determination of Cold/Warm Start

The RA8x2 MCUs allows the user to determine the cause of the reset processing. The CWSF flag in RSTSR2 indicates whether a power on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start.)

The flag is set to 0 when a power on reset occurs. Otherwise, the flag is not set to 0. The flag is set to 1 when 1 is written to it through software. It is not set to 0 even on writing 0 to it.

6.15 Determining the Reset Source

The RA8x2 MCUs allows the user to determine the reset signal generation source. Read RSTSR0, RSTSR1, RSTSR2, and RSTSR3 to determine which reset was the source of the reset. Refer to the Hardware User's Manual section "Determination of Reset Generation Source" for the flow diagram.

The following code sample shows how to determine if a reset is caused by Software Reset Detect Flag, Deep Software Standby or Power On Reset using CMSIS based register structure in Renesas FSP.

```
/* Checking if RSTSR1 ≠ 0x00, RSTSR0.PVD1RF = 1 or RSTSR0.PVD2RF = 1 */
if((0x0!=R_SYSTEM->RSTSR1)||(0x1==R_SYSTEM->RSTSR0_b.LVD1RF)||
(0x1==R_SYSTEM->RSTSR0_b.LVD2RF))
    { /* Deep software standby reset */
        if(0x1 == R_SYSTEM->RSTSR0_b.DPSRSTF)
        {
            /* Do something */
        }
        else
        {   /* Core Voltage monitor reset */
            if (0x1 == R_SYSTEM->RSTSR3_b.CVMRF)
            {
                /* Do something*/
            }
            else
            {
                /* Continue checking other resets */

            }
        }
    }
    else
    {  /* Determine whether it caused the cancellation of Deep Software Standby mode */
        if(0x1 == R_SYSTEM->RSTSR0_b.DPSRSTF)
        {
        /*Deep Software Standby mode is canceled by IWDT reset or PVD1 and PVD2 Reset*/
        }
        else
        {
        /*Reset corresponding to each bit of RSTSR1 or RSTSR0.PVD1RF or RSTSR0.PVD2RF*/
        }
    }
```

7. Security Features

RA8x2 MCUs include advanced security features. This section will briefly introduce these features. For the operational flow on using these features, the best resources are the relevant application projects and example projects which are available on the Renesas GitHub.

Refer to the Security Features section in the appropriate Hardware User's Manual for more details.

7.1 Implementation of TrustZone Technology

RA8x2 MCUs include Arm® TrustZone® (TZ) security features. The two CPU cores each have their own TrustZone regions. Arm TrustZone technology divides the system and the application into secure and non-secure domains. A secure application can issue both secure and non-secure transactions, but a non-secure application can only issue non-secure transactions. Secure transactions can only access secure memory and resources, and non-secure transactions can only access non-secure memory and resources. Secure transactions can be issued only using secure region addresses and non-secure transactions can be issued only using non-secure region addresses.

Note that there is no Trust Zone filter in the external RAM and external device area. Therefore, access is not possible only if the External Memory area is set to Secure in the Secure Attribution Unit (SAU), and access is from CPU in Non-secure state. In all other cases, access to an external device area will be possible, for example:

- Access from bus master other than CPU
- Access from Secure CPU
- Access from Non-secure CPU when the external memory area is set to Non-secure attribute in SAU.

7.1.1 Arm Security Attribution

The TrustZone for Armv8-M implementation consists of the Security Attribution Unit (SAU) and Implementation Attribution Unit (IDAU).

The IDAU defines the code, SRAM and peripheral region into the secure alias region and non-secure alias region by the address bit [28]. The secure code region and secure SRAM region are assigned to the NSC security attributes. The security map defined by IDAU is fixed in hardware and cannot be changed by software. The Master Security Attribution Unit (MSAU) is the IDAU that defines system-specific security address map for bus masters other than the CPU. The Secure Attribution Unit (SAU) is a programmable unit that defines the security of an address. Renesas IDE provides a convenient way to set up the SAU (refer to section 7.1.2). If an address maps to regions defined by both Implementation Defined Attribution Unit (IDAU) and SAU, the region of the highest security level is selected.

7.1.2 Setting up the TrustZone Boundaries

The TrustZone boundaries can only be set up using the MCU boot mode. In the development stage, the TrustZone Secure boundary regions can be set using the following methods:

1. With e^2 studio, users can set up the TrustZone boundaries when starting the debugging connection. This feature is enabled in the default Debug connection.

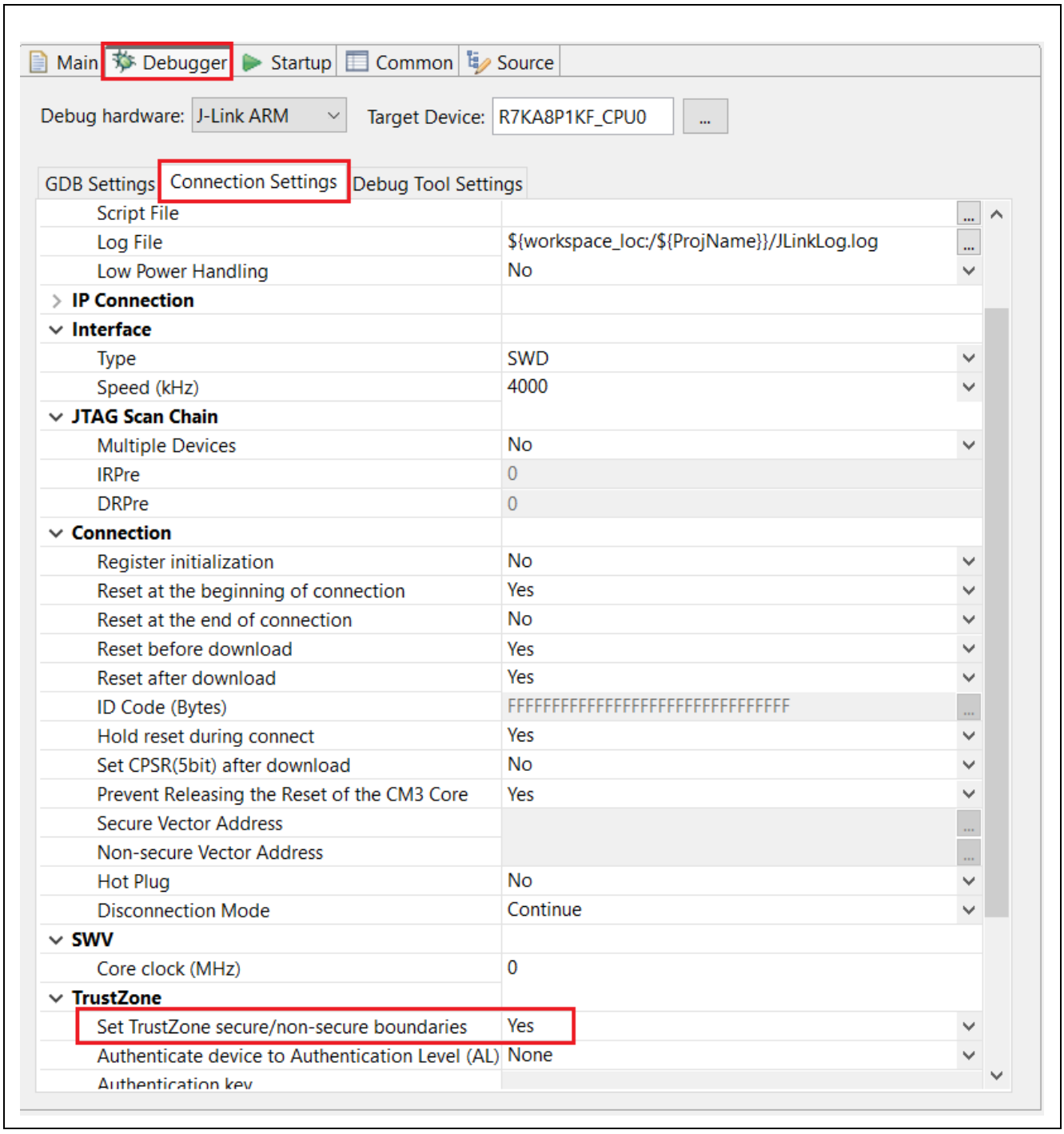


Figure 18. Set up the TrustZone® Region during the Debug Launch Session

The supported debuggers can access the MCU boot mode to set up the TrustZone boundary prior to downloading and starting a debugging session for the application images. For example, when using e² studio, the user can choose to set up the TrustZone boundary using any one of the three different debuggers as shown in Figure 19.

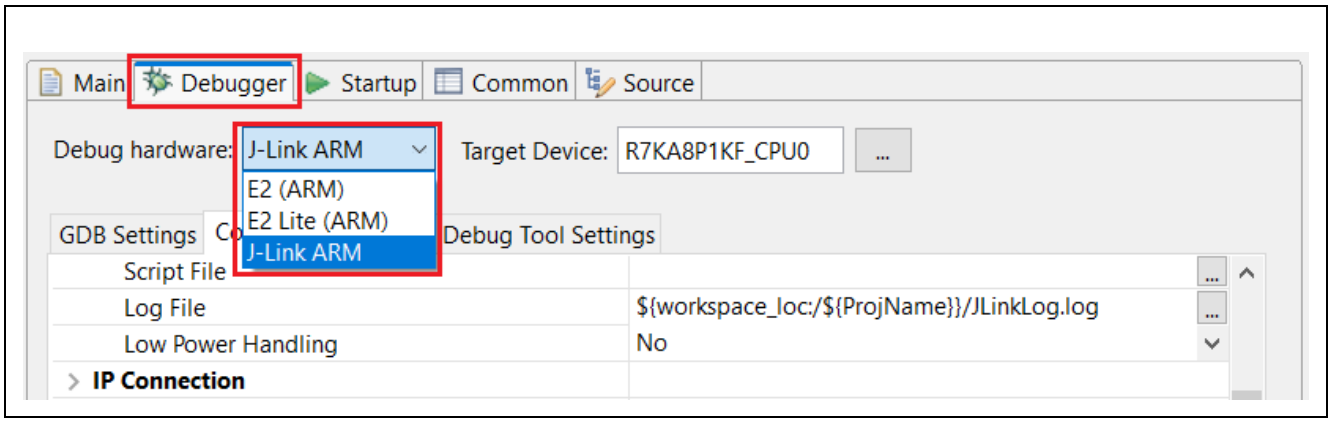


Figure 19. Debuggers Supported

The hardware setup requirement varies based on the selected debugger.

- If E2 and E2 Lite connections are selected, the standard ARM JTAG and SWD interface can be used to set up the TrustZone boundary using the E2 and E2 Lite emulator.
- If J-Link ARM connection is selected, both JTAG and SWD interface can be used to access the boot mode. Accessing the boot mode through JTAG and SWD interface is a new feature supported on RA8x2 MCUs. For a TrustZone® project, either interface can set up the TrustZone boundary prior to releasing the Reset pin and launch the user application.

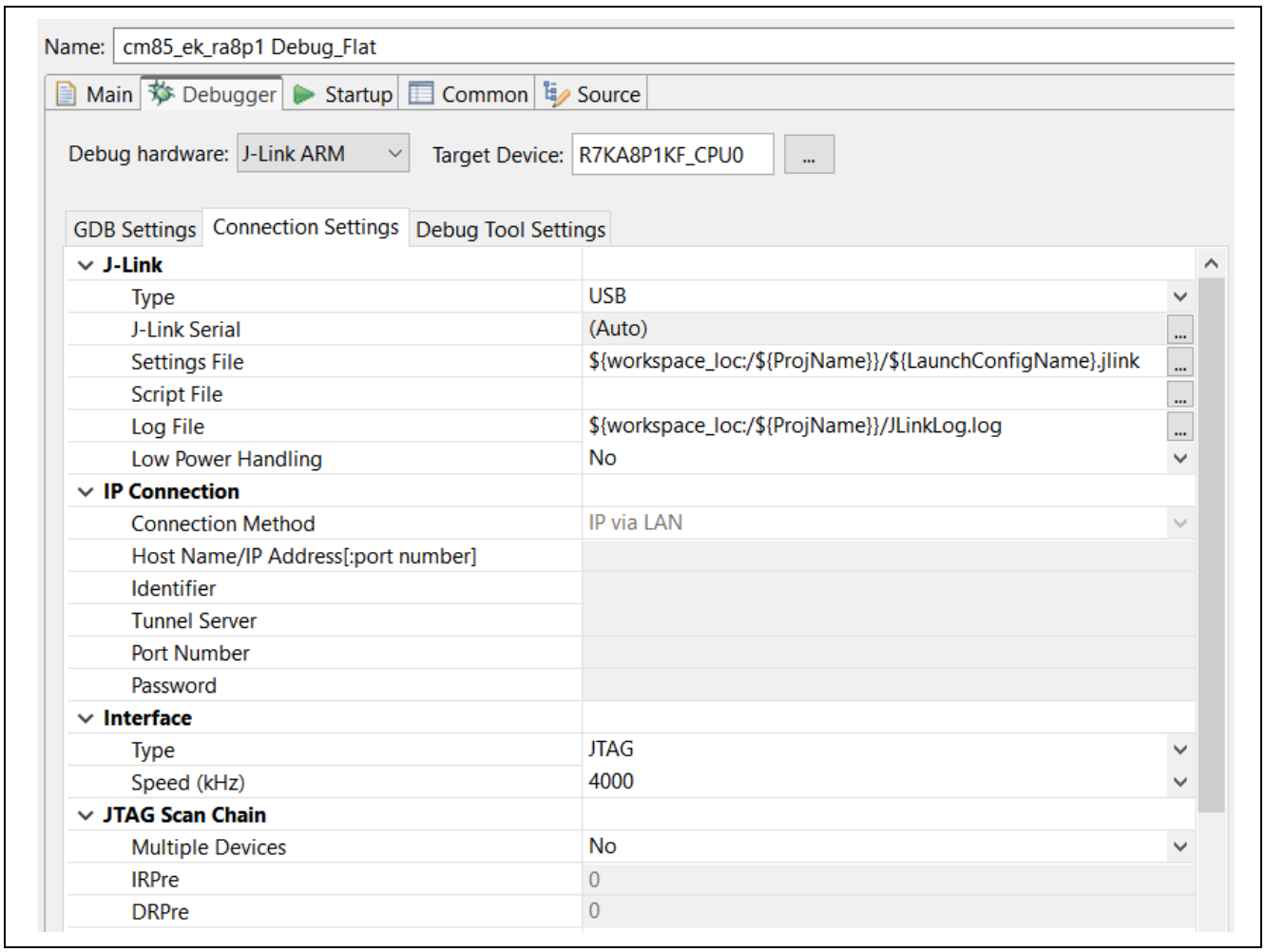


Figure 20. Connection Setting for JLink Debugger Example

- When using the J-Link debugger, the Renesas debug interface also supports boot mode access through the SCI boot mode interface. If this is desired, the MD pin and the SCI boot mode pins should be connected to the debugger interface following Table 12.

Table 12. Pin Assignments for Debugger Connection Supporting SCI Boot Mode

Pin No.	SWD	JTAG	Serial Programming Using SCI
1	VCC	VCC	VCC
2	P210/SWDIO	P210/TMS	NC
4	P211/SWCLK	P211/TCK	NC
6	P209/SWO/TXD9	P209/TDO	P209/TXD9
8	P208/RXD9	P208/TDI	P208/RXD9
9	GNDdetect	GNDdetect	GNDdetect
10	nRESET	nRESET	nRESET
12	P308/TCLK	P308/TCLK	NC
14	P307/TDATA[0]	P307/TDATA[0]	NC
16	P306/TDATA[1]	P306/TDATA[1]	NC
18	P305/TDATA[2]	P305/TDATA[2]	NC
20	P304/TDATA[3]	P304/TDATA[3]	NC
3,5,15,17,19	GND	GND	GND
7	NC	NC	NC
11,13	NC	NC	NC

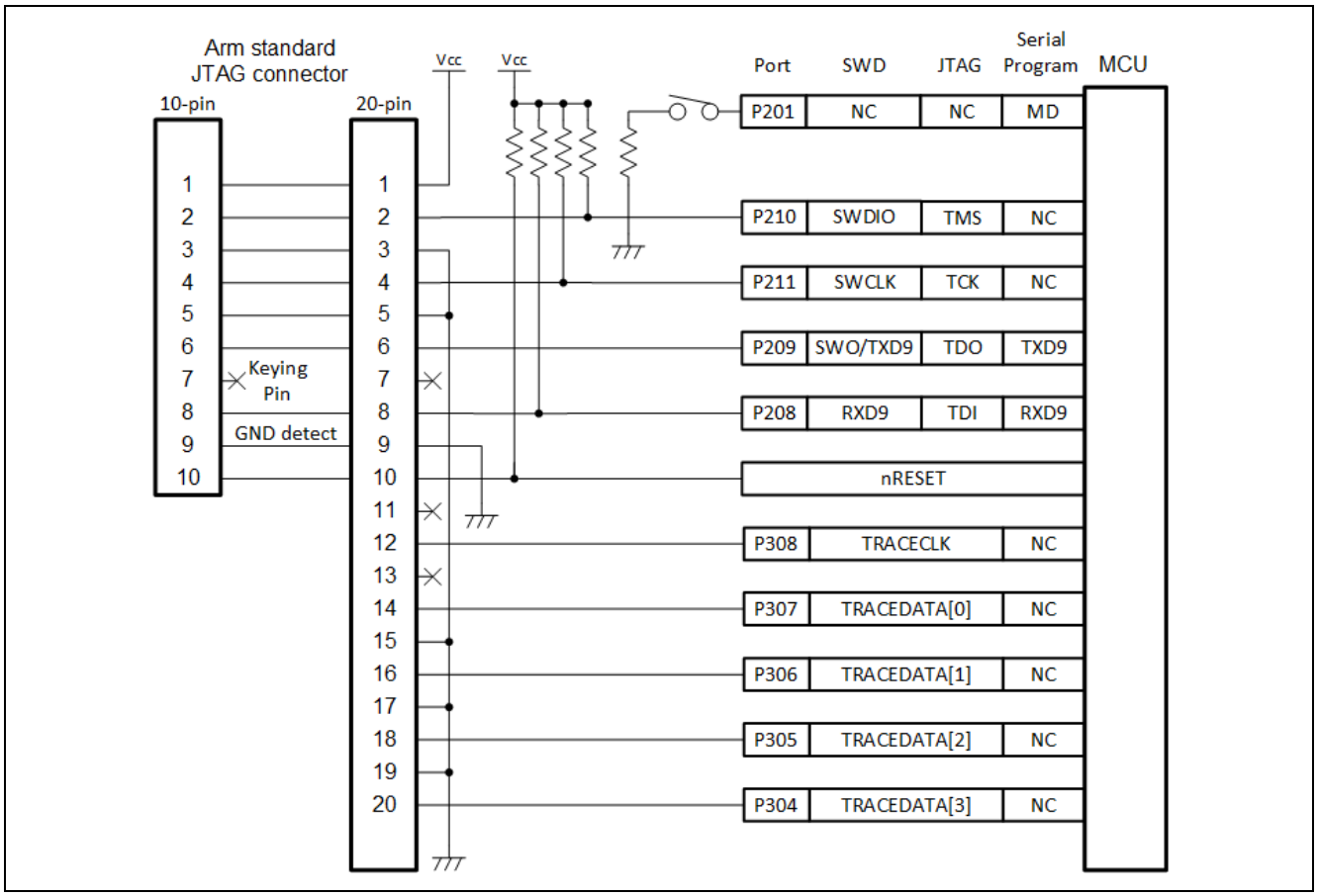


Figure 21. Emulator Connections for MCUs that Support Trustzone®

For Keil MDK and IAR EWARM, users need to use the Renesas Device Partition Manager (RDPM) to manually set up the TrustZone boundaries. The RDPM tool can be integrated into the Keil MDK and IAR EWARM following the descriptions provided in RASC Quick Start Guide which is installed when the RASC is installed.

The RDPM is automatically installed in e² studio when the e² studio is installed, users can also choose to set up the TrustZone boundaries using RDPM as a separate step when using e² studio. In this case, the user can disable the TrustZone setup options in the debug configuration. Reference Figure 18 for where this setting is located.

The RDPM supports two types of connections when accessing the boot mode as shown in Figure 22. If user wants to access the SCI connection through the debug header, user needs to provide the SCI connections as shown in Figure 21.

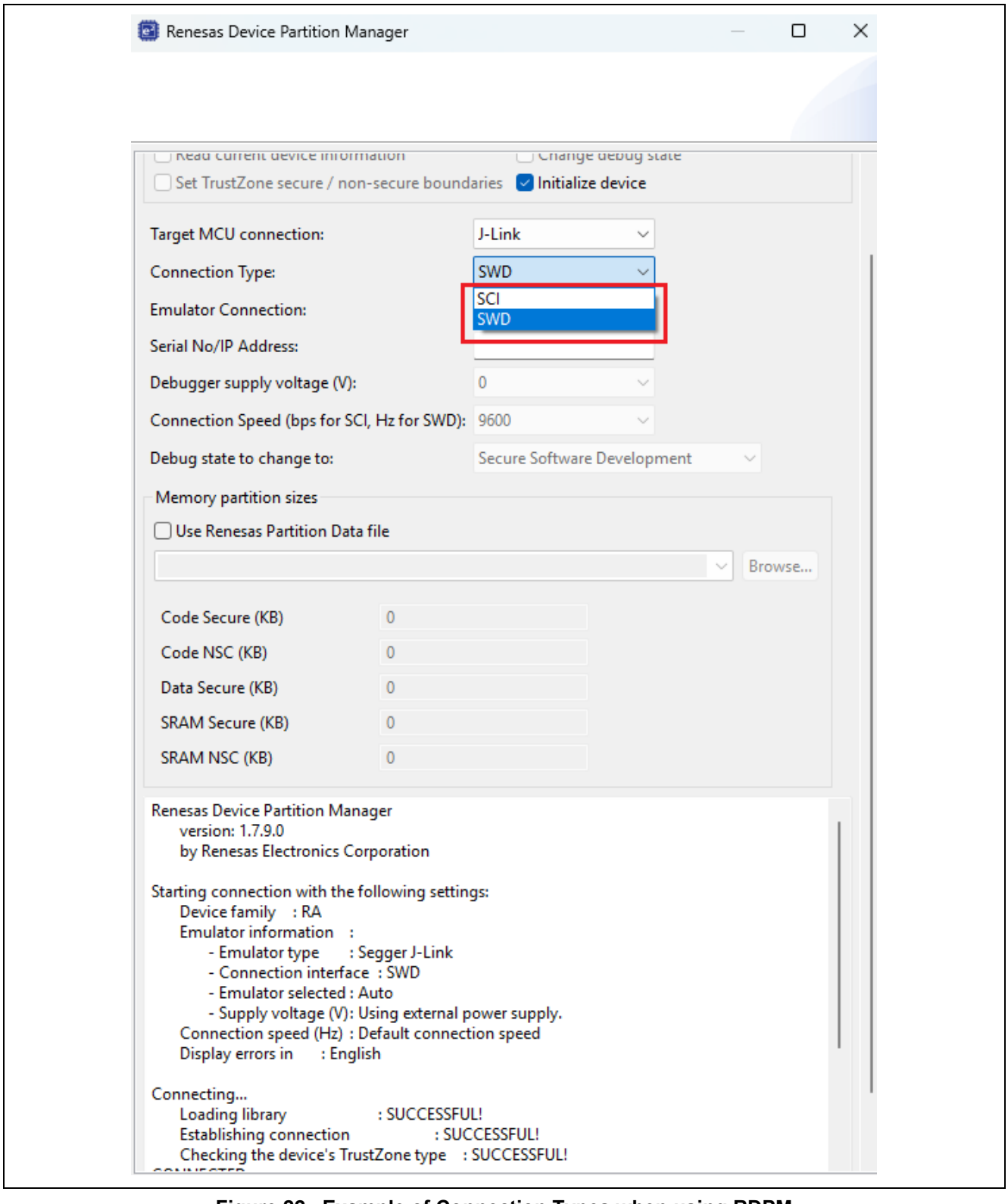


Figure 22. Example of Connection Types when using RDPM

7.2 Device Lifecycle Management

The RA8x2 Device Lifecycle Management is unique to the RA MCU Family. In the RA8x2 MCU Series, Device Lifecycle has been separated from authenticated debug.

7.3 First Stage Bootloader (FSBL) and Secure Boot.

The Renesas RA8x2 MCU provides an immutable first stage bootloader in the OTP memory area. When the FSBL is enabled, the primary CPU executes the FSBL code provided by Renesas before executing the user application code. During FSBL execution debug is prohibited regardless of Authentication Level. The Root of Trust for authenticating the application code is securely injected. The operation of the FSBL is managed by the Option-Setting Registers.

The FSBL, when enabled, can verify the integrity and authenticity of an OEM bootloader (OEM_BL) or a normal application (with no bootloader capability) starting at the application executable memory. The OEM_BL or a normal application, is verified when it is initially programmed as well as prior to execution.

For RA8x2 MCUs, a set of OEM_ROOT_PKs can be enabled or revoked by configuring the REVOKE register located in the Option-Setting Memory OTP region.

For specifications on secure boot operations, user can reference the RA8x2 User's Manual: Hardware section "Secure Boot".

7.4 Other Security Features

7.4.1 Secure Key Injection

The Secure Key Injection feature on RA8x2 MCUs is similar to the single core Cortex®-M85 and the single core Cortex®-M33 based Secure Key Injection with additional key types added to support the new DLM system, for example related with Secure boot. User can reference the Table "Keys that can be injected" in the Hardware User's Manual to understand the types of keys supported.

7.4.2 Secure Factory Programming

This feature supports programming an encrypted firmware image. The image is encrypted with an Image Encryption Key which is wrapped with the Renesas Key Wrap Service. This feature enables secure firmware programming in a non-secure environment.

7.4.3 Renesas Secure IP (RSIP-E50D)

The RA8x2 MCUs have the Renesas Secure IP (RSIP-E50D) for accelerated cryptographic operations. User can reference section "Renesas Secure IP (RSIP-E50D)" on the Hardware User's Manual for more details on the capability of the RSIP-E50D. Example project demonstrating the functionality of the RSIP will be provided in the Renesas GitHub repository. The RSIP-E50D supports two operating modes: Compatibility Mode and Protected Mode.

7.4.4 Application and OEM BL Anti-rollback

The RA8x2 MCU supports application firmware version Anti-rollback. Three Anti-rollback counters support three different use cases: OEM Bootloader (OEM BL), Secure and Non-secure application. See the Option-Setting Memory section 0 for the respective control registers and how to set them up.

7.4.5 Decryption On-The-Fly (DOTF)

Decryption on-the-fly is a security feature on RA8x2 MCUs. This feature allows confidential external code and/or data storage on external OSPI devices (also see section 8.2.2.3). The code or data is encrypted using a pre-stored known key or a generated key at run-time.

7.4.6 Tamper Pins

The tamper detection function detects the RTCICn (n = 0 to 2) pin input event. The input event is defined as a change of RTCICn (n = 0 to 2) pin input level. The tamper detection flag is set to 1 by the input event . When interrupt is enabled and flag is set to 1, tamper detection interrupt is generated. When backup registers clear is enabled and flag is set to 1, the data of backup registers is cleared. Time Capture Function can select this flag as the source of the time capture trigger.

When Tamper Detection Zeroization Enable is enabled and flag is set to 1, the HUK Zeroization request is outputted.

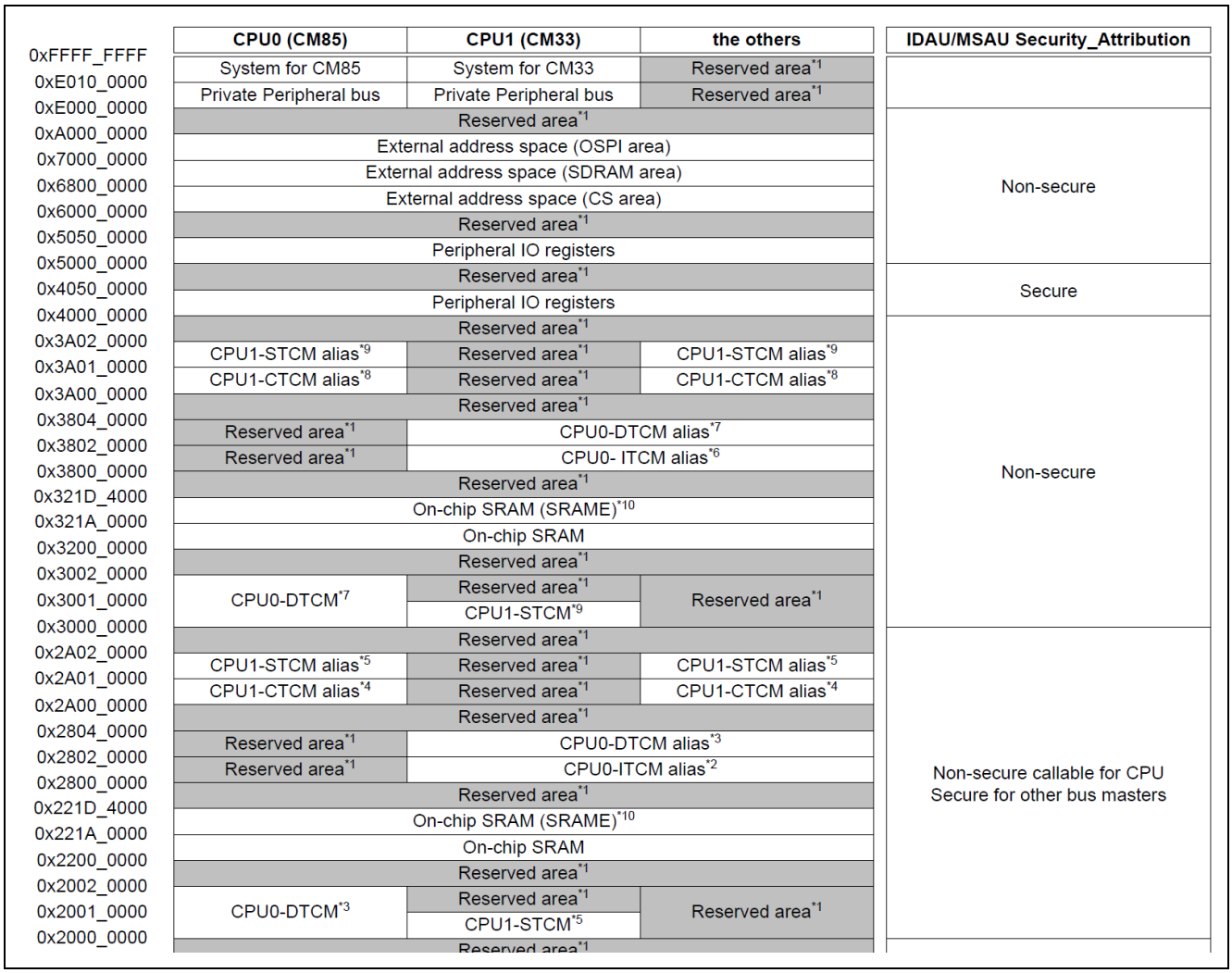
7.4.7 Pointer Authentication and Branch Target Identification (PACBTI)

This security feature is supported by the Cortex-M85, which is based on the Armv8.1-M architecture. When this feature is enabled, the return addresses from function calls are authenticated prior to return and valid destinations of indirect branch instructions are specified. Usage of this feature needs to be supported through the compilers used. By default, FSP does not enable this feature. Please refer to the IDE and compilers for the availability of this feature and how to enable it.

8. Memory

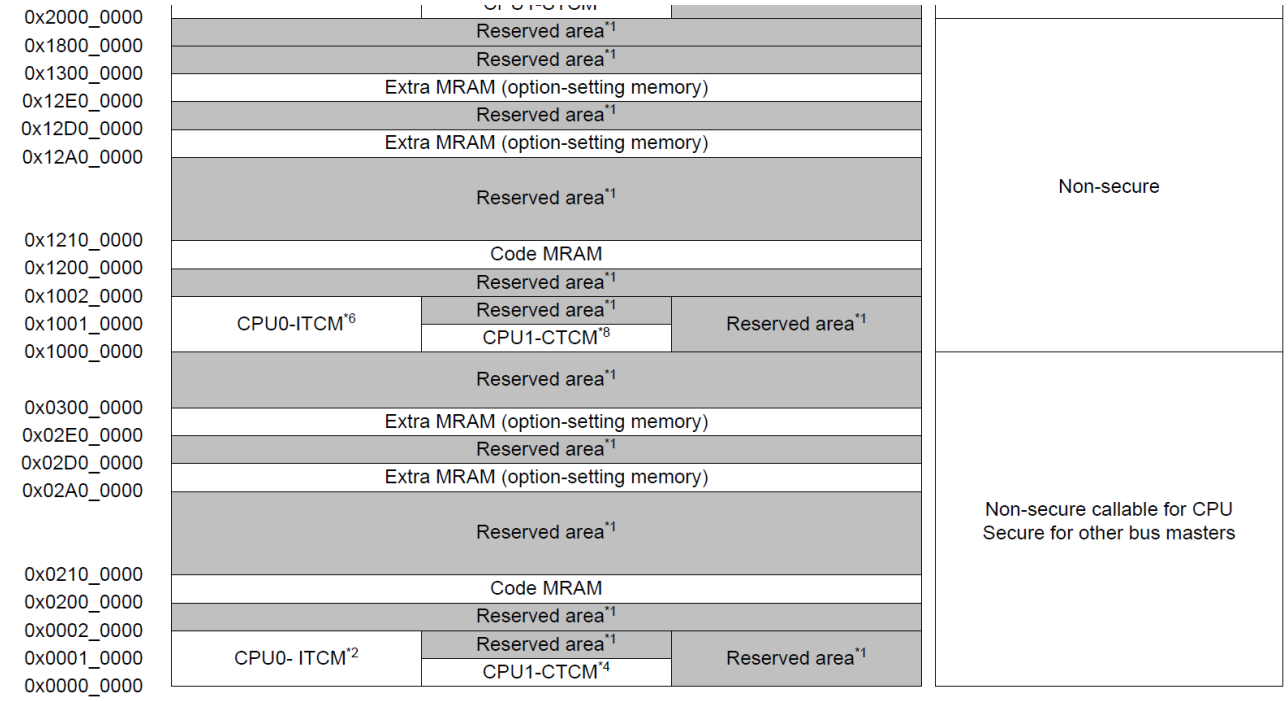
The RA8x2 MCUs support a 4 GB linear address space ranging from 0x0000_0000 to 0xFFFF_FFFF that can contain program, data, and external memory bus interface. Some members of the family include an SDRAM controller that allows access to an SDRAM device connected to external memory bus. Program and data memory share the address space; separate buses are used to access each, increasing performance and allowing same-cycle access of program and data. Contained within the memory map are regions for on-chip RAM, peripheral I/O registers and external memory.

For greater detail of the RA8x2 Memory usage, refer to the Application Note “RA8P1 Memory Architecture, document No. R01AN7880”.



	CPU0 (CM85)	CPU1 (CM33)	the others	IDAU/MSAU Security_Attribution
0xFFFF_FFFF	System for CM85	System for CM33	Reserved area*1	
0xE010_0000	Private Peripheral bus	Private Peripheral bus	Reserved area*1	
0xE000_0000	Reserved area*1			Non-secure
0xA000_0000	External address space (OSPI area)			
0x7000_0000	External address space (SDRAM area)			
0x6800_0000	External address space (CS area)			
0x6000_0000	Reserved area*1			
0x5050_0000	Peripheral IO registers			
0x5000_0000	Reserved area*1			Secure
0x4050_0000	Peripheral IO registers			
0x4000_0000	Reserved area*1			Non-secure
0x3A02_0000	CPU1-STCM alias*9	Reserved area*1	CPU1-STCM alias*9	
0x3A01_0000	CPU1-CTCM alias*8	Reserved area*1	CPU1-CTCM alias*8	
0x3A00_0000	Reserved area*1			
0x3804_0000	Reserved area*1	CPU0-DTCM alias*7		
0x3802_0000	Reserved area*1	CPU0- ITCM alias*6		
0x3800_0000	Reserved area*1			
0x321D_4000	On-chip SRAM (SRAME)*10			
0x321A_0000	On-chip SRAM			
0x3200_0000	Reserved area*1			
0x3002_0000	CPU0-DTCM*7	Reserved area*1	Reserved area*1	
0x3001_0000		CPU1-STCM*9		
0x3000_0000	Reserved area*1			
0x2A02_0000	CPU1-STCM alias*5	Reserved area*1	CPU1-STCM alias*5	Non-secure callable for CPU Secure for other bus masters
0x2A01_0000	CPU1-CTCM alias*4	Reserved area*1	CPU1-CTCM alias*4	
0x2A00_0000	Reserved area*1			
0x2804_0000	Reserved area*1	CPU0-DTCM alias*3		
0x2802_0000	Reserved area*1	CPU0-ITCM alias*2		
0x2800_0000	Reserved area*1			
0x221D_4000	On-chip SRAM (SRAME)*10			
0x221A_0000	On-chip SRAM			
0x2200_0000	Reserved area*1			
0x2002_0000	CPU0-DTCM*3	Reserved area*1	Reserved area*1	
0x2001_0000		CPU1-STCM*5		
0x2000_0000	Reserved area*1			

Figure 23. Example of RA8P1 Memory Map for 1MB MRAM Standard Product (part 1)



Note: See Table 5.1. The capacity of the Code MRAM, On-chip SRAM differs depending on the product.
Note 1. Reserved areas should not be accessed.
Note 2. CPU0-ITCM for Secure has different addresses for CPU0 and other bus masters.
Note 3. CPU0-DTCM for Secure has different addresses for CPU0 and other bus masters.
Note 4. CPU1-CTCM for Secure has different addresses for CPU1 and other bus masters.
Note 5. CPU1-STCM for Secure has different addresses for CPU1 and other bus masters.
Note 6. CPU0-ITCM for Non-Secure has different addresses for CPU0 and other bus masters.
Note 7. CPU0-DTCM for Non-Secure has different addresses for CPU0 and other bus masters.
Note 8. CPU1-CTCM for Non-Secure has different addresses for CPU1 and other bus masters.
Note 9. CPU1-STCM for Non-Secure has different addresses for CPU1 and other bus masters.
Note 10. These areas are accessible only when SRAM ECC function is disabled.

Figure 24. Example of RA8P1 Memory Map for 1MB MRAM Standard Product (part 2)

The address space of RA8x2 MCUs supports aliases. In the address from 0x0000_0000 to 0x5FFF_FFFF, secure and non-secure regions are isolated by using bit 28 of the address. Therefore, in these areas, the memory location can be addressed using two addresses. The validity of the address used depends on the relevant security attribution setting and the current security state of bus master.

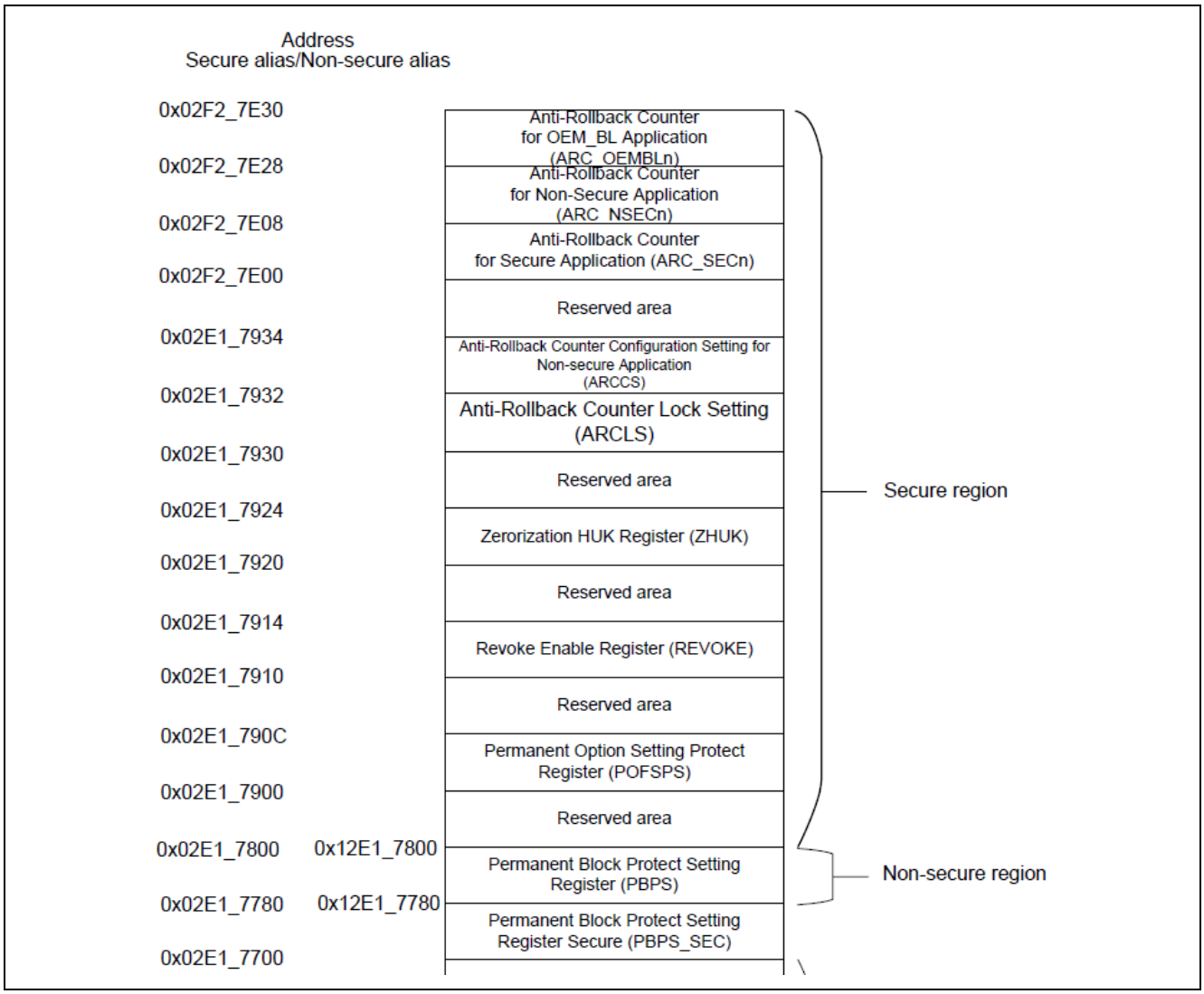


Figure 25. Example of RA8P1 Option Setting Memory Map with Alias (part 1)

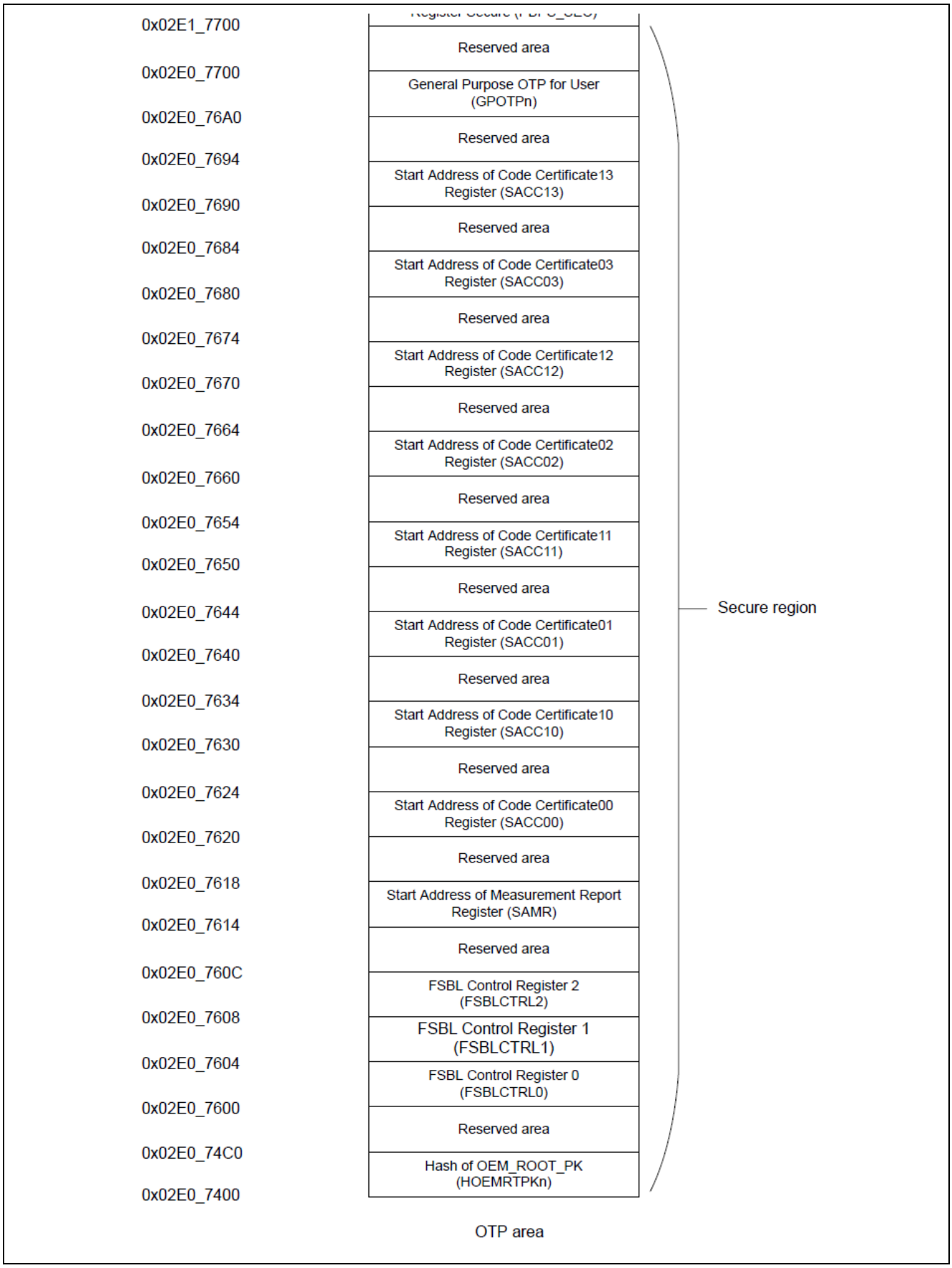


Figure 26. Example of RA8P1 Option Setting Memory Map with Alias (part 2)

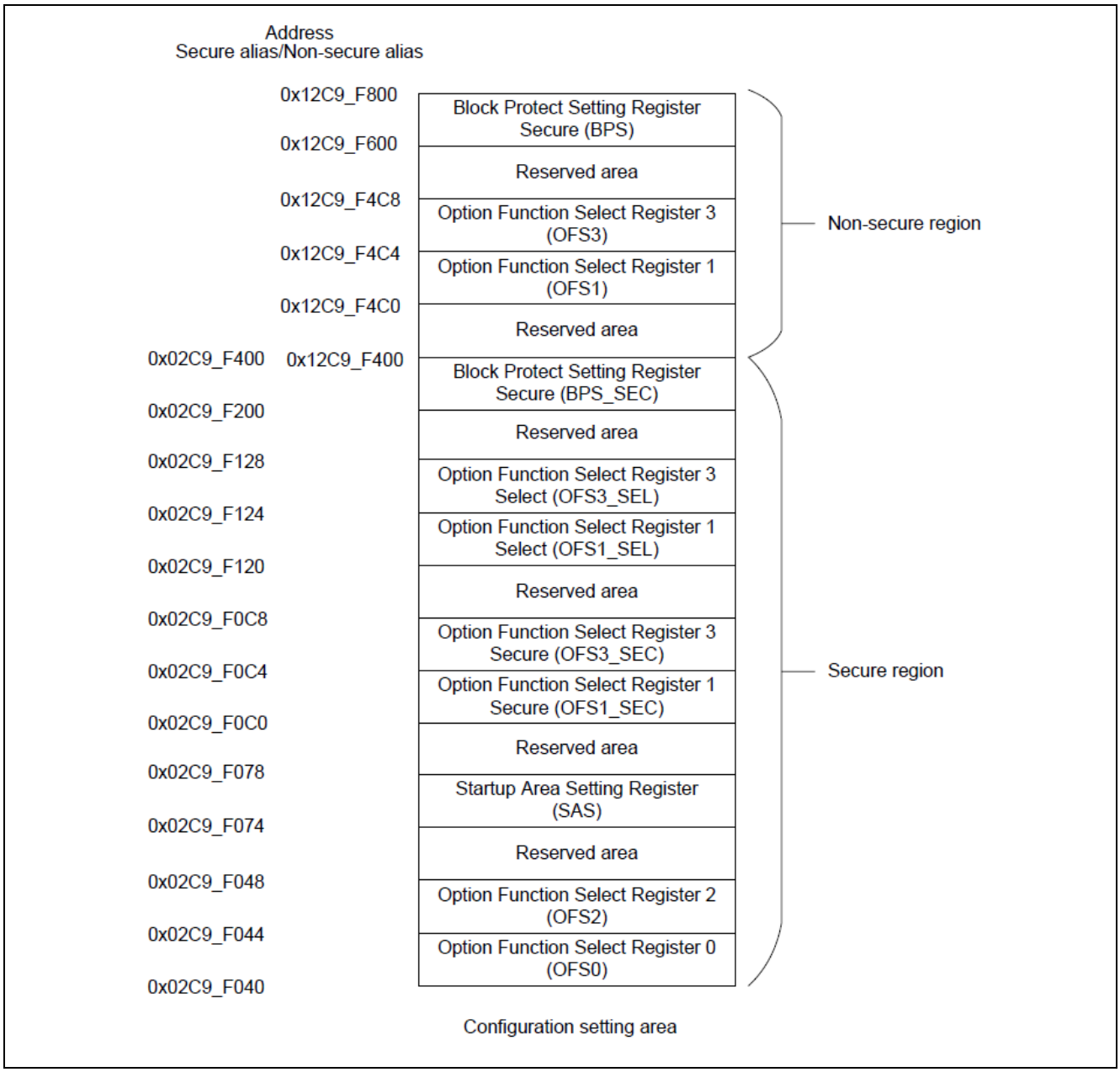


Figure 27. Example of RA8P1 Option Setting Memory Map with Alias (part 3)

8.1 Internal Memory

8.1.1 SRAM

The MCU provides an on-chip, high-density SRAM module with Error Correction Code (ECC).

In single-core, C-TCM and S-TCM at addresses 0x2A00_0000 to 0x2A01_FFFF can be used as SRAM.

See the section CPU in the MCU user manual for details on TCM.

Parameter		SRAM0	SRAM1	SRAM2	SRAM3
SRAM capacity		512 KB	512 KB	512 KB	128 KB
SRAM address	Secure alias	0x2200_0000 to 0x2207_FFFF	0x2208_0000 to 0x220F_FFFF	0x2210_0000 to 0x2217_FFFF	0x2218_0000 to 0x2219_FFFF
	Non-secure alias	0x3200_0000 to 0x3207_FFFF	0x3208_0000 to 0x320F_FFFF	0x3210_0000 to 0x3217_FFFF	0x3218_0000 to 0x3219_FFFF
ECC region*1	Secure alias	0x221A_0000 to 0x221A_FFFF	0x221B_0000 to 0x221B_FFFF	0x221C_0000 to 0x221C_FFFF	0x221D_0000 to 0x221D_3FFF
	Non-secure alias	0x321A_0000 to 0x321A_FFFF	0x321B_0000 to 0x321B_FFFF	0x321C_0000 to 0x321C_FFFF	0x321D_0000 to 0x321D_3FFF
Access		Wait states are not inserted into the access cycle by default. If the ICLK frequency is greater than half the maximum frequency in the electrical characteristics, a wait state is required. If the ICLK frequency is less than or equal half the maximum frequency in the electrical characteristics, a wait state is not required.			
Data retention function		Not available in Deep Software Standby mode			
Module-stop function		Module-stop state can be set to reduce power consumption			
Error checking		SEC-DED (Single-Error Correction and Double-Error Detection Code: 64-bit data with 8-bit ECC code)			
Security		TrustZone Filter is integrated for memory access and SFR access. Access to the memory space is controlled by setting the memory Security Attribution (SA). And access to I/O space (SFR) space is controlled by setting the register Security Attribution (SA). See section 59.3.5. TrustZone Filter Function.			

Note 1. When ECC function is disabled, it is used as a data region and can be accessed directly. When ECC function is enabled, direct access is disabled. When bypass is enabled, direct access to ECC bit is possible.

Figure 28. Example of RA8P1 SRAM Specification

8.1.2 MRAM

RA8x2 MCUs incorporate code MRAM and option-setting memory. The code MRAM stores instructions and operands. For option-setting memory, see the section “Option-Setting Memory” in the Hardware User’s Manual.

The CPSEQ (Code MRAM programming sequencer) controls programming of the code MRAM. EPSEQ (Extra MRAM programming sequencer) controls programming of extra MRAM which includes option-setting memory. The MACI (MRAM application command interface) controls EPSEQ according to the specified MACI commands. For details of the code MRAM, see the “MRAM” section in the Hardware User’s Manual.

Item	Code MRAM	Option-setting memory in extra MRAM
Memory capacity	User area: 1 MB max	See section 7, Option-Setting Memory.
Read cycle	See section 60.5.2. MRCFREQ : Code MRAM Frequency Notifications Register	See section 60.5.3. MREFREQ : Extra MRAM Frequency Notifications Register
Programming method	Write page buffer	MACI command
Protection	Protects against erroneous rewriting of the MRAM	
Dual bank function	Not available	Not available
Block swap function	Not available	Not available
Background operations (BGOs)	• The code MRAM can be read while option-setting memory is being programmed • Option-setting memory can be read while the code MRAM is being programmed.	
Units of programming	Program data buffer: 1/2/4/8 bytes Code MRAM: 32 bytes	16 bytes
MACI command	Not available	Program: 16 bytes Forced stop Status clear Configuration set: 16 bytes Increment counter: 1 bit Read counter: 8 bytes
Security function	• Protect against illicit tampering with or reading out of data in MRAM • OFS area protection • Startup area select setting protection – BTFLG, BTSIZE, and MSUACR registers are protected by the FSPR bit. • Permanent block protect setting protection – User area is permanently protected from programming operation by the permanent block protect function. • MRAM protection for TrustZone – Protection for MRAM area (program) – Protection for MRAM area (read) – Protection for register • Extra MRAM program mode entry protection • Anti-rollback counter • HUK protection	
Safety function	Software protection • MACI command protection by MENTRYR register • User area protection by MRCP0 and MRCP1 registers • User area protection by the block protection setting. Error protection • Error is detected when unintended commands or prohibited settings occur. The MACI command is not accepted after an error detection. Boot area protection • The startup area select function allows the customer to safely update the boot firmware. The size of the startup area can be selected from 8 KB, 16 KB, or 32 KB. ECC support for double-bit error correction and triple-bit error detection.	

Item	Code MRAM	Option-setting memory in extra MRAM
Interrupt request	• MRAM_MRCRD (Code MRAM read error) Enabled by INTENBTC and INTENBDC bits. • MRAM_MRERD (Extra MRAM read error) Enabled by INTENBTE and INTENBDE bits. • MRAM_MRCPR (Code MRAM sequencer error) Enabled by MRCAEIE bits. • MRAM_MREPR (Extra MRAM sequencer error) Enabled by CMDLKIEE and MREAEIE bits. • MRAM_ENDOFPE (Extra MRAM sequencer ready (processing end)) Enabled by MRDYIE bit.	

Figure 29. Example of RA8P1 MRAM Specification

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Operation	Write state	G_{max_wr}	—	—	200	Gauss	—
	Read state	G_{max_rd}	—	—	200	Gauss	—
	No access state	G_{max_noac}	—	—	500	Gauss	—
Storage		G_{max_stg}	—	—	500	Gauss	The applied temperature is T_{stg}.

Figure 30. MRAM Magnetic Field Immunity Characteristics

Refer to the section “MRAM Characteristics” in the “Electrical Characteristics” section of the Hardware User’s Manual and RA/RX Family eMRAM Magnetic Immunity Guideline (R01AN7226) for more information.

8.1.3 SiP Flash

The system-in-package (SiP) products provide up to 8 MB of internal serial flash memory. Refer to SiP Product Configuration section in R8x2 MCU hardware manual for more information.

Flash memory ISSI IS25WX064-JWLE is packaged for SiP product. In the SiP products, connect VCC2_n to a power supply voltage ranging from 1.70 V to 2.00 V. The recommended voltage for VCC2_n is the 1.8V system power supply.

The following settings for OSPI channel 1 should be set before accessing the SiP Flash memory.

– The OM_1_DQS, OM_1_SIO7 to OM_1_SIO0 and OM_1_ECSINT1 pins are pull-up by PmnPFS.PCR bit before PmnPFS.PMR bit is set to 1.

– The LVOCR.LVO1E bit is set to 1.

8.1.4 Peripheral I/O Registers

Blocks of peripheral I/O registers appear at various locations in the memory map depending on the device and the current operating mode. The majority of peripheral I/O registers occupy a region from address 0x4000_0000 to 0x504F_FFFF. However, this may vary in location and size on a per device basis. Consult the Hardware User’s Manual for specifics. Details can be found in the “I/O Registers” appendix, and also in the register descriptions for each peripheral function. This region contains registers that are available at all times in all modes of operation.

The Renesas FSP contains C header files in CMSIS data structure that map all of the peripheral I/O registers for a specific device to easily accessible I/O data structures.

8.1.4.1 MRAM Block Protection

RA8x2 MCUs have a MRAM Block Protection feature that protects secure or non-secure code MRAM region from being erased or reprogrammed by secure or non-secure software. It is worth noting that the protection is for both Secure and Non-secure software accesses.

Each block in the user area has the block protect setting (BPS or BPS_SEC). When the MRCBPROT0 or MRCBPROT1 register is 0x0000 and the block protect bit is 0, program access to the non-secure user area of code MRAM is prohibited. When the MRCBPROT1 register is 0x0000 and the block protect bit is 0, program access to the secure user area of code MRAM is prohibited. To program the sector whose block protect bit is 0, set MRCBPROT0 or MRCBPROT1 register to 0x0001.

The block protect setting can be locked by the permanent block protect setting (PBPS or PBPS_SEC). When the permanent block protect setting and the block protect setting are 0, program access cannot be performed.

Valid block protect setting (BPS or BPS_SEC) depends on the Block Protect Select bit (BPS_SEL). See ‘Protection by Block Protect Setting’ in the MCU Hardware User’s Manual for more details.

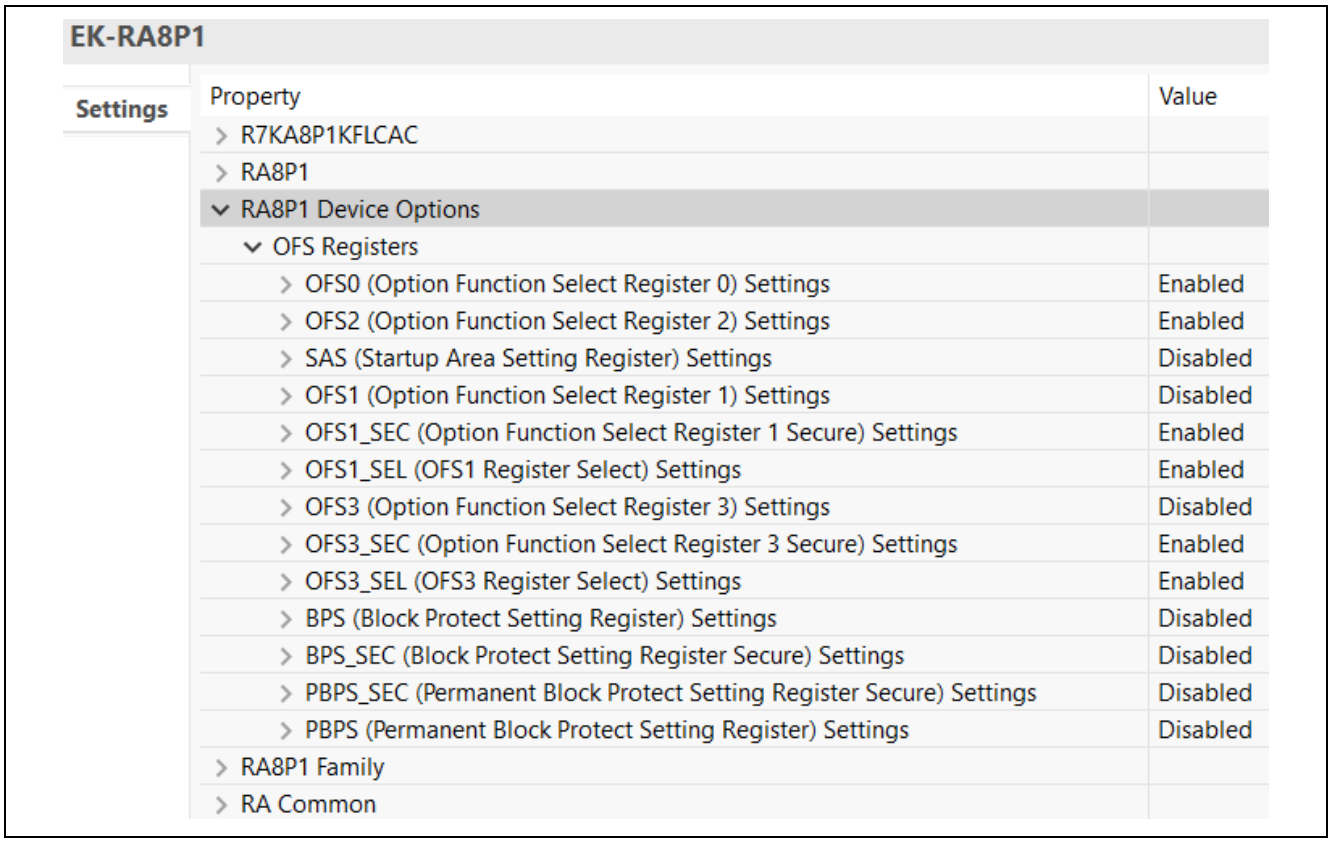


Figure 31. Example of Block Protect Settings in Renesas FSP Configurator

Note: Protection by Block Protect Setting must be handled carefully to prevent mistakes that may result in blocking accesses to an MCU region.

8.1.5 Tightly Coupled Memory (TCM)

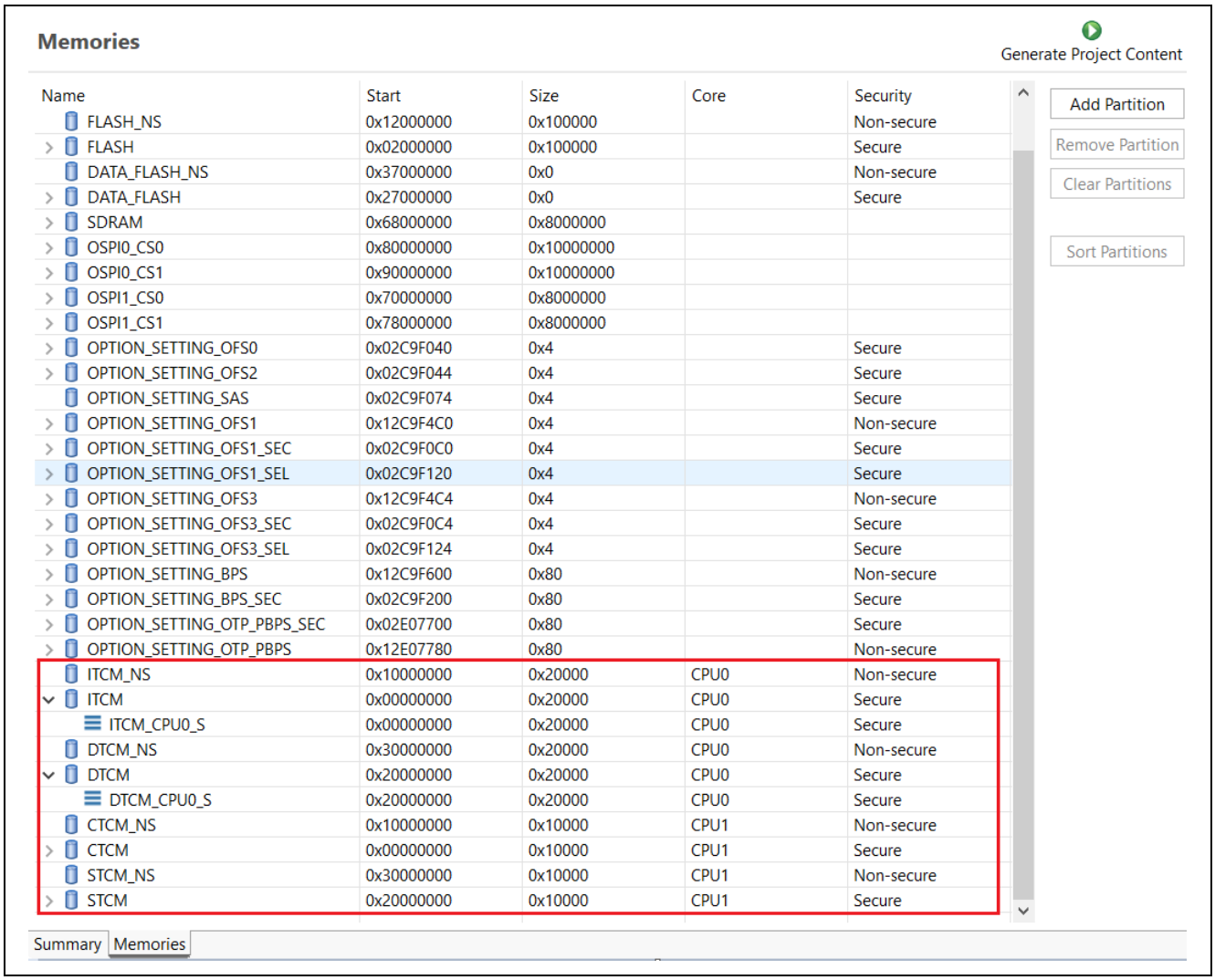
The RA8x2 family has TCM memory for each CPU. 128KB ITCM with ECC and 128 KB DTCM with ECC are associated with CPU0. 64KB CTCM with ECC and 64KB STCM with ECC are associated with CPU1.

Accessing TCM is not available in CPU Deep Sleep mode.

TCM is initialized by FSP. The linker script has defined sections for TCM. Users can choose the data and code to put in the corresponding regions. Refer to Application Note “RA8P1 Memory Architecture, document No. R01AN7880” for more details on the handling of the TCM regions.

For further details, refer to the TCM Interfaces section of the Arm® Cortex®-M85 Processor Technical Reference Manual.

The FSP provides a Memories configuration for customizing the TCM memory areas.



Name	Start	Size	Core	Security
FLASH_NS	0x12000000	0x100000		Non-secure
FLASH	0x02000000	0x100000		Secure
DATA_FLASH_NS	0x37000000	0x0		Non-secure
DATA_FLASH	0x27000000	0x0		Secure
SDRAM	0x68000000	0x8000000		
OSPI0_CS0	0x80000000	0x10000000		
OSPI0_CS1	0x90000000	0x10000000		
OSPI1_CS0	0x70000000	0x8000000		
OSPI1_CS1	0x78000000	0x8000000		
OPTION_SETTING_OFS0	0x02C9F040	0x4		Secure
OPTION_SETTING_OFS2	0x02C9F044	0x4		Secure
OPTION_SETTING_SAS	0x02C9F074	0x4		Secure
OPTION_SETTING_OFS1	0x12C9F4C0	0x4		Non-secure
OPTION_SETTING_OFS1_SEC	0x02C9F0C0	0x4		Secure
OPTION_SETTING_OFS1_SEL	0x02C9F120	0x4		Secure
OPTION_SETTING_OFS3	0x12C9F4C4	0x4		Non-secure
OPTION_SETTING_OFS3_SEC	0x02C9F0C4	0x4		Secure
OPTION_SETTING_OFS3_SEL	0x02C9F124	0x4		Secure
OPTION_SETTING_BPS	0x12C9F600	0x80		Non-secure
OPTION_SETTING_BPS_SEC	0x02C9F200	0x80		Secure
OPTION_SETTING_OTP_PBPS_SEC	0x02E07700	0x80		Secure
OPTION_SETTING_OTP_PBPS	0x12E07780	0x80		Non-secure
ITCM_NS	0x10000000	0x20000	CPU0	Non-secure
ITCM	0x00000000	0x20000	CPU0	Secure
ITCM_CPU0_S	0x00000000	0x20000	CPU0	Secure
DTCM_NS	0x30000000	0x20000	CPU0	Non-secure
DTCM	0x20000000	0x20000	CPU0	Secure
DTCM_CPU0_S	0x20000000	0x20000	CPU0	Secure
CTCM_NS	0x10000000	0x10000	CPU1	Non-secure
CTCM	0x00000000	0x10000	CPU1	Secure
STCM_NS	0x30000000	0x10000	CPU1	Non-secure
STCM	0x20000000	0x10000	CPU1	Secure

Figure 32. Example of TCM Memory Configuration in Renesas FSP

8.2 External Memory

The external address space is divided into CS areas (CS0 to CS7), SDRAM area (SDCS), and OSPI areas (CS0 and CS1).The eight CS areas (CS0 to CS7) each correspond to the CSn signal output from a CSn (n = 0 to 7) pin. The two OSPI areas (CS0 and CS1) each correspond to the OM_CSn signal output from an OM_CSn (n = 0, 1) pin. In standard products, two more OSPI areas (CS0 and CS1) marked as channel 1 (OSPI1) are available for external OSPI memory connection. Figure 33 and Figure 34 show the example address ranges associated with the individual CS areas (CS0 to CS7), SDRAM area (SDCS), and OSPI areas (CS0 and CS1).

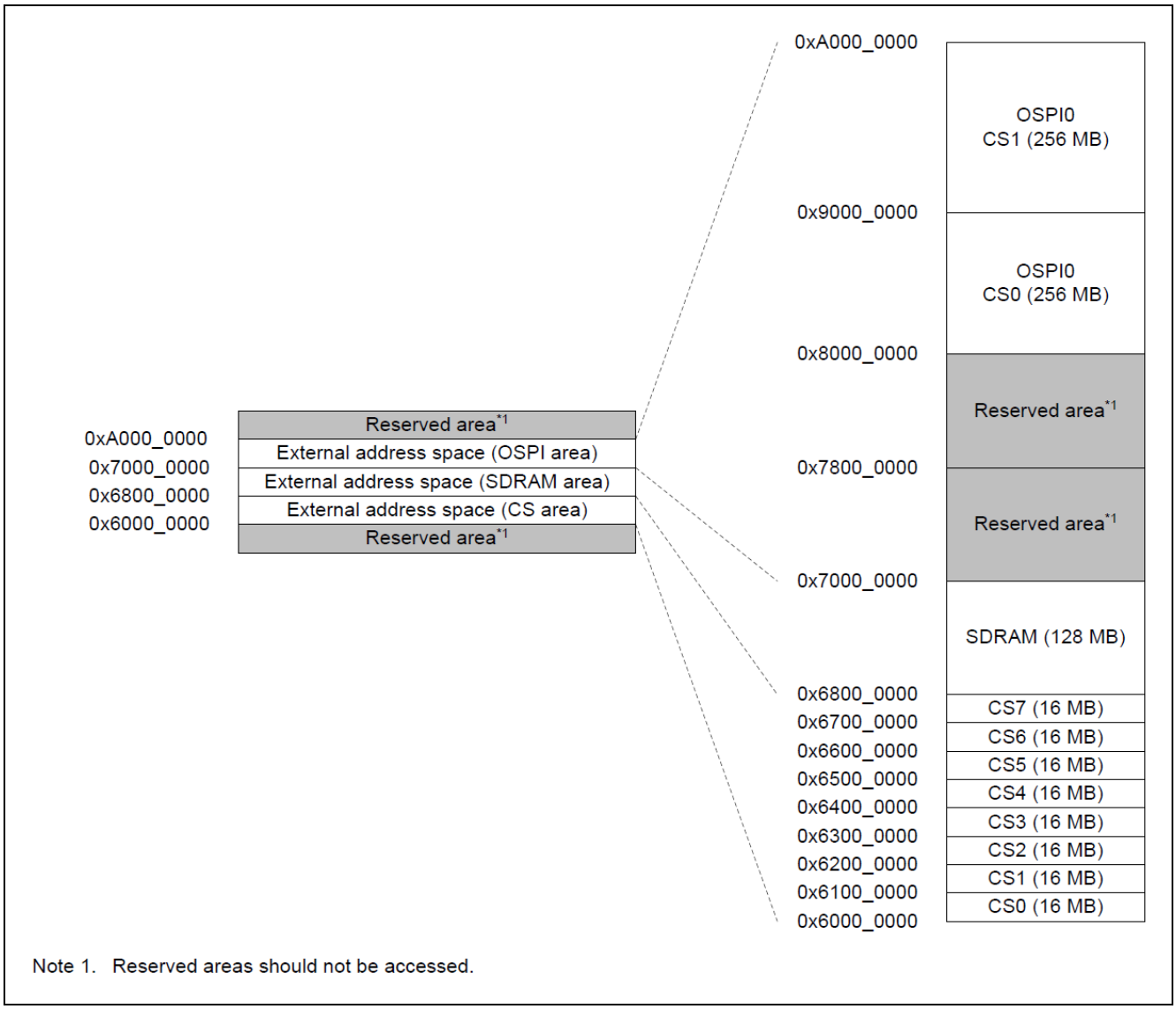


Figure 33. Example of Detailed address map for CS area, SDRAM area, and OSPI area for SiP Product

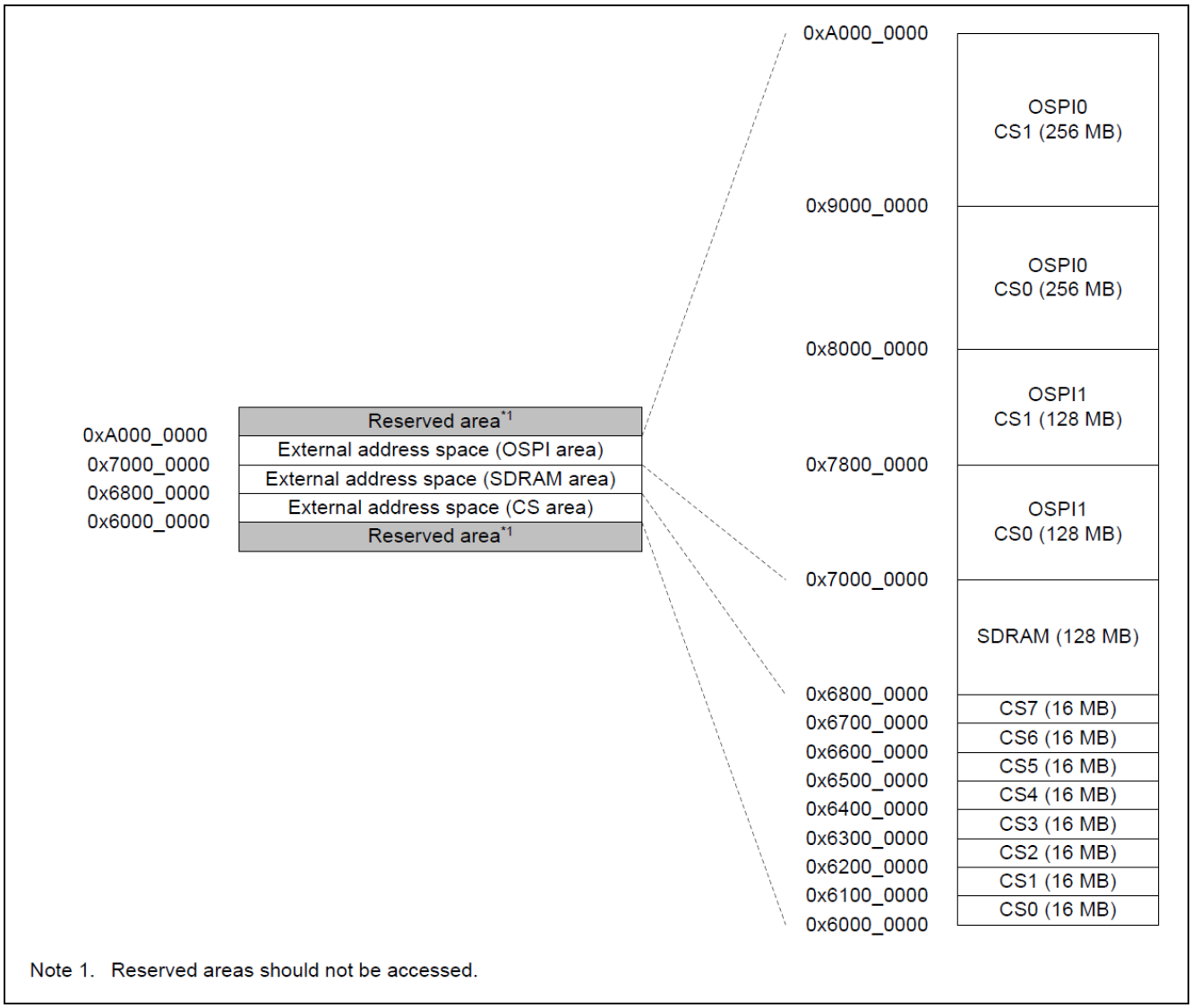


Figure 34. Example of Detailed address map for CS area, SDRAM area, and OSPI area for Standard Product

8.2.1 Using External 32 or 16-bit Memory Devices

When connecting an external 32-bit or 16-bit memory device that has a byte select line, connect A1 of the MCU to A0 of the memory and A0 of the MCU to the byte select line.

The Renesas FSP provides data structures (R_BUS) in C header files through the board support package which allows access to all the external bus control registers. Any device which supports a parallel interface can be mapped into the RA8x2 External address space (CS0 to CS7).

8.2.1.1 Example of SDRAM Configuration and Initialization

The Renesas FSP provides a SDRAM configuration in BSP settings to initialize SDRAM memory controller, as shown in Figure 35. Look for the *R_BSP_SdramInit* function in the file *ra>fsp>src>bsp>mcu>all >bsp_sdram.c* for more details.

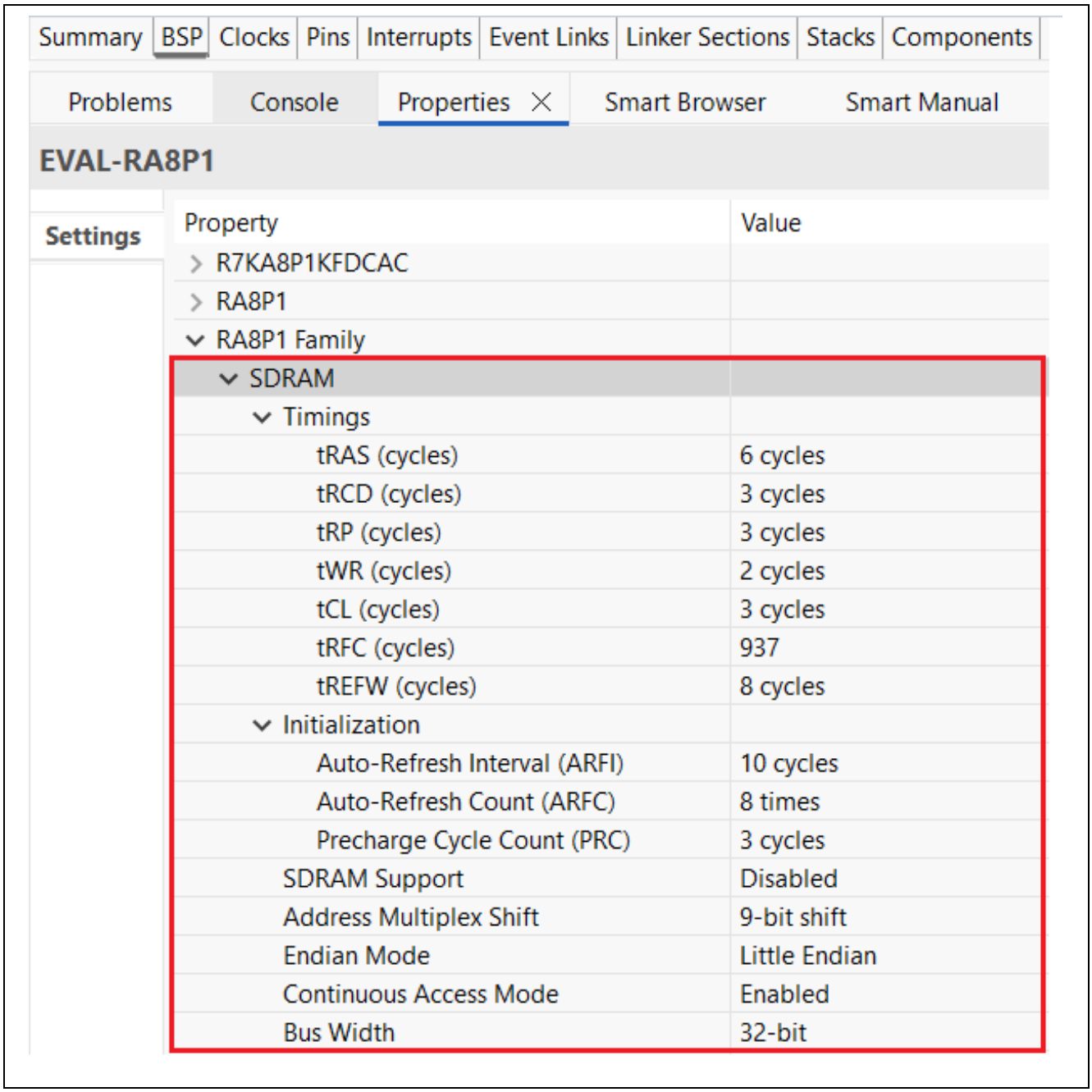


Figure 35. Example of SDRAM Settings in Renesas FSP

8.2.2 Using External Octal SPI Devices

RA8x2 microcontrollers include peripheral interfaces to connect Serial Peripheral Interface (SPI) devices, including memory devices. External OSPI devices are supported on OSPI channel 0 for SiP products, and on both OSPI channel 0 and OSPI channel 1 for standard products. The OSPI peripheral supports single-, quad- and octal-bit data widths. Refer to the Octal Serial Peripheral Interface (OSPI) section of the MCU Hardware User's Manual for details on configuration and implementation.

Careful consideration should be made when connecting to an OSPI device. These devices are usually higher speed than other SPI devices, so they may be subject to PCB routing limitations that are not normally required for other SPI devices.

The digital signals include the SPI Chip Select signal, the SPI clock, the Read Data Strobe, the SPI Reset signal and the data signals.

OSPI signals should be routed with 50Ω ±10% single-ended characteristic impedance. All data lines should be matched length within ±50 mils (1.27 mm) relative to the DQS signal. The DQS signal should match the length of the clock signal. Keep the total routing length under 2000 mils (50.8mm), but the total routing length should also be kept as short as possible. The clock signal should be spaced apart from other signals by at least 3 times the clock trace width. Minimize the vias to as few as possible for the entire signal path. Avoid serpentine routing on the clock signal.

Additional implementation guidelines may be available from the OSPI device manufacturer.

Renesas FSP provides support for communicating with SPI devices by providing initialization routines, pin and timing configurations.

The OSPI device can be erased and programmed using the OSPI APIs from the OSPI module support. In addition, the FSP linker script has provided sections to support placing OSPI data to the device. The J-Link driver that is integrated with the IDE support can program these sections to the OSPI device while programing the MCU.

The RA8x2 OSPI peripheral is compatible with all flash devices that are JESD xSPI standard compliant. It guarantees operation with JESD251 (xSPI for Non Volatile Memory) compliant memory.

The following table lists some of the OSPI devices that are compatible with RA8x2. Compatibility has been determined by simulation.

Table 13. RA8x2 Compatible xSPI Devices

Category	Supplier	Part Number
RAM (Octo SPI)	JSC	JSC28SSU8AGDY
	Cypress	S27KL0641
Flash (Octa SPI)	Infineon	S28HS512TGABHI01
	ISSI	IS25LX032/64/128
	Macronix	MX25LM51245G
	Cypress	S26KL512S
	Micron (Xccela Flash)	MT35XL512ABA
Flash (Quad SPI)	Macronix	MX25UW512454G
	Micron (Xccela Flash)	MT25QL128ABA
	Cypress	S25FS512S
	Macronix	MX25R1635F

8.2.2.1 OSPI Master Functionalities

The summary of OSPI specifications supported by RA8x2 MCU family is as follows.

Item	Description
Number of units	2 units
Protocol	Compliant with the xSPI protocol*1
Data transmission and reception	Issue transaction for up to 2 slaves as masters. Only one of the memory devices can operate at a time.
Transfer speed	Support the transfer at xSPI333
Mode	• Support the following protocol modes: – 1-, 4-, 8-pin with SDR/DDR (1S-1S-1S, 4S-4D-4D, 8D-8D-8D) – 2-, 4-pin with SDR (1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S) • Configurable address length • Configurable initial access latency cycle • Support XiP mode
OSPI function	• Support write data mask • Support in-band reset • Memory-mapping – Unit 0 supports up to 256 MB address space each CS – Unit 1 supports up to 128 MB address space each CS – 2 control channels per unit for the bus masters • ch 1: GLCDC1 bus master • ch 0: Other bus master – Prefetch function for burst-read with low latency – Outstanding buffer for burst-write with high throughput • Manual command – Configurable up to 4 commands – Status register polling function • Input strobe port timing shift
Decryption function	Decryption on-the-fly is available for memory-map read
Interrupt source	• Error interrupt • Completion interrupt
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	• Security attribution can be set for the IO register area • External address space is defined as non-secure

Note 1. The OSPI is compliant with JEDEC standard JESD251 (profiles 1.0 and 2.0), JESD251-1 and JESD252.

Figure 36. Example of RA8P1 OSPI Specifications Example

Renesas RA8x2 MCUs supports memory mapping mode that automatically converts system bus access for pre-configured memory area into xSPI transaction.

In this operation, the payload of address and data field are delivered from system bus signals. The information on command field and size are delivered from the configured register bits. When using FSP OSPI driver, after R_OSPI_B_Open API is executed successfully, access to the OSPI data area will be performed in a memory mapped manner. Note that each memory mapped region has an associated CS (channel selection on the OSPI FSP stack), it is important to use the correct channel when calling the R_OSPI_B_Open.

8.2.2.2 Octal SPI Initialization Process

By default, most of the flash devices are in SPI mode, so it is necessary to open the OSPI module in SPI mode before initializing both the OSPI module and the flash device. Refer to the OSPI example project (available for download on Renesas GitHub). The process to initialize the OSPI is as follows.

- Reset and initialize to put both the OSPI module and the flash device in SPI mode.
- Transition the OSPI flash device to OSPI mode.
- Transition the OSPI module to OSPI mode.
- Start OSPI transaction.

8.2.2.3 Encrypting Data in External OSPI devices

The OSPI interface provides for decryption on-the-fly (DOTF) when configured in memory map mode. This provides a strong layer of protection when storing information in the external SPI devices. DOTF functionality supports encrypted data and code storage on the OSPI device. The data or code can be encrypted using a pre-stored known key or a run-time generated key. A dedicated AES engine supports transparent OSPI

operation for data read and code execution. For the details on the operational flow of using this feature, user can reference application note “Application Design using RA8 Series MCU Decrypt on the Fly for OSPI” , document No. R11AN0773.

8.3 Data Alignment

There are no restrictions on data alignment in external memory aligning data. The external bus in the RA8x2 MCU can perform 8-bit, 16-bit, and 32-bit accesses on odd memory locations as well. While it is still optimal to align data access, it is not required.

8.4 Restriction on Endian

The external bus can be configured for either little endian or big endian. However, if the user intends to execute instructions stored in external memory, then the external bus must be configured as little endian.

8.5 Memory Protection Unit

All bus masters have Memory Protection Units (MPUs) to prevent unprivileged access. When unprivileged access occurs the MPU blocks the address and may notify the CPU using a Non Maskable Interrupt or Reset Handler. It is recommended to set up the MPU to improve the security of the application. Users can reference the ARM® Cortex®-CM85 and ARM® Cortex®-CM33 Technical Reference Manuals to understand more on the Arm MPU configurations. For the RA8x2 Bus master MPU, user can reference the Hardware User’s Manal for more details. FSP BSP stack enables user to perform configurations to these Bus master security attributes.

The tables below list the MPU specifications and show the behavior on detection of each MPU error.

Table 14. MPU Specifications

Classification	Module/Function	Specifications
Memory protection	Arm MPU	Memory protection function for the CPU: • CPU0: Secure MPU 8 regions and Non-secure MPU 8 regions • CPU1: Secure MPU 8 regions and Non-secure MPU 8 regions
	Bus master MPU	Memory protection function for each bus master except for the CPU: • NPU: 5 regions • DMAC0 (DMAC/DTC0): 8 regions • DMAC1 (DMAC/DTC1): 8 regions • EDMAC (Ether-DMAC): 5 regions • GLCDC: 2 regions • DRW: 3 regions • MIPI-DSI: 1 region • MIPI-CSI by VIN: 3 regions • CEU: 2 regions

Table 15. Behavior on MPU error detection

MPU type	Access permissions setting	Boundary address setting minimum unit	Error response for the MPU error notification	Bus access at error detection	Hold the information of error access
Arm MPU	Read access Write access Execution	32 bytes	Supported*1	• Incorrectly write access • Incorrectly read access	Hold in CPU
Bus master MPU	Read access Write access Privileged access (DMAC/DTC only)	NPU: 4 KB DMAC0/1: 32 bytes EDMAC: 32 bytes GLCDC: 1 KB DRW: 1 KB MIPI-DSI: 4 KB MIPI-CSI: 4 KB CEU: 4 KB	Supported	• Write access ignored • Read access is read as 0	Hold

Note 1. A privileged DAP request through the unprivileged debug extension mechanism is demoted to an unprivileged access and is subject to MPU checks. Both privileged and unprivileged requests are subject to MPU checks.

The register definitions for the Bus Master MPU are provided by FSP to use in your project through the data structure R_MPU_MMPU. The Arm MPU access is provided through the CMSIS pack from ARM. User can call the CMSIS APIs to set up MPU region with desired security attribute. The ARM CMSIS API also support enable and disable the MPU with the required instruction barrier calls.

8.6 Cortex®-M85 Cache and Cortex®-M33 Cache

The RA8x2 MCU has Caches for each CPU. 16KB of L1 Instruction Cache and 16 KB of Data Cache, both with ECC support are associated with CPU0. 16KB Code-Bus Cache (C-Cache) with ECC and 16KB System-Bus Cache (S-Cache) with ECC are associated with CPU1. The Cache usage is strongly recommended for applications that use data with good locality, for example Artificial Intelligence (AI) and graphic applications with data declaration that are constant or not changing frequently.

- Setting up the CM85 I-Cache and D-Cache can be done using the CMSIS API which is included in the FSP packs. These APIs take care the memory operation barriers that need to be used when updating Cache controls and configurations, for example, the usage of __DMB(), __DSB() and __ISB() calls. The FSP does not provide support for this feature on the CM33 C-Cache and S-Cache. Users must implement their code for this functionality.
- The processors do not support hardware coherency for the L1 instruction and data caches. Coherency can only be maintained at the system level. Typically, invalidating the cache prior to disabling the Cache or CPU access after a bus master access and using MPU to protect regions from be cached are the common methods to maintain Cache Coherency.

9. Register Write Protection

The register write protection function protects important registers from being overwritten because of software errors. The registers to be protected are set with the Protect Register (PRCR_S and PRCR_NS). Table 16 lists the association between the PRCR bits and the registers to be protected.

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PRKEY[7:0]								—	—	PRC5	PRC4	PRC3	—	PRC1	PRC0

Figure 37. PRCR_S Register

Table 16. PRCR Protection Bits

PRCR bit	Description
PRC0	• Registers related to the clock generation circuit: SCKDIVCR, SCKDIVCR2, SCKSCR, PLLCCR, PLLCR, BCKCR, MOSCCR, HOCOCR, HOCOCR2, MOCOCR, FLLCR1, FLLCR2, CKOCR, OSTDCR, OSTDSR, SOSTDCR, SOSTDSR, PLL2CCR, PLL2CR, PLLCCR2, PLL2CCR2, EBCKOCR, SDCKOCR, SCICKDIVCR, SCICKCR, SPICKDIVCR, SPICKCR, ADCCKDIVCR, ADCCKCR, GPTCKDIVCR, GPTCKCR, LCDCKDIVCR, LCDCKCR, BCKADIVCR, ESWCKDIVCR, ESWPCKDIVCR, ETHPCKDIVCR, BCKACR, ESWCKCR, ESWPCKCR, ETHPCKCR, MOCOUTCR, HOCOUTCR, USBCKDIVCR, OCTACKDIVCR, CANFDCKDIVCR, USB60CKDIVCR, I3CCKDIVCR, USBCKCR, OCTACKCR, CANFDCKCR, USB60CKCR, I3CCKCR, MOSCSCR, HOCOSCR, PLLSCR, PLL2SCR, MOCOSCR, MOSCWTCR, LOCOCR, LOCOUTCR, MOMCR, MOMCR2, SOSCCR, SOMCR, SYRACCR
PRC1	• Registers related to the low power modes: OPCCR, PGSCR, PSSTCR0, PSSTCR1, PSSTCR2, PSSTCR3, PSSTCR4, PSSTCR5, PDCTRGD, PDCTRNPU, PDCTRESWM, PDRAMSCR0, PDRAMSCR1, SBYCR, SSCR1, LPSCR, DPSBYCR, DPSWCR, DPSIER0-5, DPSIFR0-5, DPSIEGR0-4, LDOSCR, PLL1LDOCR, PLL2LDOCR, HOCOLDOCR, LVOCR, SCR, SVSCR, MLSMCR, CPUDSCR, MWMCR • Registers related to the battery backup function: VBTBER, VBTICTLR, VBTBKRn (n = 0 to 127), VBTBPCR1, VBTBPCR2, VBTBPSR, VBTADSR, VBTADCR1, VBTADCR2, VBTICTLR2, VBTADCR3, VBTNCWCR
PRC3	• Registers related to the PVD: PVD1CMPCR, PVD2CMPCR, PVD4CMPCR, PVD5CMPCR, PVD1CR0, PVD2CR0, PVD4CR0, PVD5CR0, PVD1CR1, PVD1SR, PVD2CR1, PVD2SR, PVD1FCR, PVD2FCR, PVD4FCR, PVD5FCR, PVDLR, VBATTMNSELR
PRC4	• Registers related to the Security and Privilege setting registers: CGFSAR, RSTSAR, LPMSAR, PVDSAR, BBFSAR, DPFSAR, DPFSAR1, RSCSAR, PGCSAR, VBRSABAR, VBRPABARS, VBRPABARNS, CPUSAR, DEBUGSAR, CACHESAR, TCMSAR, TCMSABARC, TCMSABARS, IPCSAR, IPCPAR, ICUSARx (x = A, B, E to M), TEVTRCR, BUSSARx (x = A, B, C), BUSPARC, MMPUSARx (x = A, B), DMACCHSAR, DMACCHPAR, DMACSAR, DTCSAR, ELCSARx (x = A, B, C), ELCPARx (x = A, B, C), PmSAR(m = 0 to 9, A, B, C, D), PSARx (x = B, C, D, E), MSSAR, PPARBx (x = B, C, D, E), MSPAR, SRAMSABARn (n = 0 to 3), SRAMSAR, SRAMESAR
PRC5	• Registers related to the reset control: SYRSTMSK0-2, TEMPRCR, TEMPRLR
PRKEY[7:0]	These bits control write access to the PRCR register. To modify the PRCR register, write A5h to the eight higher-order bits and the wanted value to the eight lower-order bits as a 16-bit unit.

Renesas FSP supplies two APIs (R_BSP_RegisterProtectEnable and R_BSP_RegisterProtectDisable) to simplify modifying Register Write Protection.

10. I/O Port Configuration

The “I/O Ports” section of the Hardware User’s Manual describes exact pin configurations based on peripheral selection and other register settings. Some general information is listed as follows.

It is important to note that after a reset, each pin will be in the default state for that pin until the configuration is applied. There may be a small period where some pins may be in an undesirable state. This will be true regardless of what configuration approach is used. The user should consider the impact this may have for each application, including how this may affect other system features.

One important aspect of the configuration of each I/O Port is the drive strength, which is adjusted using the Drive Capacity Control (DSCR) bits in the PSEL register for each port. It is important to identify what drive strength is required for the specific user application, then select I/O ports that meet those requirements. Pay attention to the drive strength options and limitations for each port when selecting the function for that port. Some ports have limited drive strength options, while many ports have a wide range of drive strength options.

Renesas FSP provides a convenient way to configure each I/O port without needing to explicitly write to each register bit.

10.1 Multifunction Pin Selection Design Strategies

Most ports on the RA8x2 products are capable of multiple peripheral functions. Tools, such as the pin configurator in FSP, are available from Renesas to assist with port selection for each RA8x2 device. When several peripheral functions are needed, use the following design strategies to help with port function selection.

- Assign peripheral functions with only one port option first. For example, there is only one port option for each Trace Data signal in the debug function. When this function is needed, assign these ports first.
- Assign peripheral functions with limited port options next. For example, devices that support the OSPI peripheral typically only have two options for each OSPI signal.
- Assign peripheral functions with multiple port options last. One example would be the Serial Communications Interface (SCI), which typically has many available port options.
- Some peripheral function port options are interchangeable, while others must be assigned in logical groups. For example, the IIC peripheral has some ports with the suffix “_A” while others have the suffix “_B” in the signal name. Ports should be selected to have the same suffix for the peripheral function. Other peripheral functions do not have this type of suffix, and ports may be assigned interchangeably, such as the USB_VBUSEN signal for the USBFS peripheral function. Also see Section 21.4 in this document.

10.2 Setting Up and Using a Port as GPIO

There are two methods for setting up and using a port as GPIO, either using the Port Control Register (PCNTR1), or the PmnPFS registers.

Method 1: Port Control Register (PCNTR1)

- Select a pin as an output by writing a “1” to the Port Direction bit (PDRn) in Port Control Register 1 (PCNTR1).
- The Port Direction bits (PDRn) are read/write. Setting the value to a “1” selects the pin as an output. Default state for I/O Ports is “0” (input). The port direction registers can be read on the RA8x2 MCUs.
- The Port Output Data bits (PODRn) in the corresponding Port Control Register (PCNTR1) are read/write. When the PODR is read the state of the output data latch (not the pin level) is read.
- The Port Input bits (PIDRn) in Port Control Register 2 (PCNTR2) are read only. Read the PIDRn bit in the PCNTR2 register to read the pin state.

Method 2: Port mn Pin Function Select (PmnPFS) registers

- The Port Mode Register (PMR) is read/write and is used to specify whether individual pins function as GPIO or as peripheral pins. Out of reset all PMR registers are set to 0 which sets all pins to work as GPIO. If a PMR register is set to 1 then that corresponding pin will be used for peripheral functions. The peripheral function is defined by that pin's PSEL setting.
- When setting a pin as an output it is recommended that the desired output value of the port be written to the data latch first, then the direction register is set to an output. Though not important in all systems, this prevents an unintended output glitch on the port being set up.

In general, using PCNTR1 to configure a port will provide faster access, but will have fewer configuration features available. Using the PmnPFS registers will have more configuration features available but will have slower access.

Renesas FSP provides Pin Configuration to configure GPIO pin after reset as shown below. After the GPIO is configured, it can be controlled using HAL layer APIs in FSP.

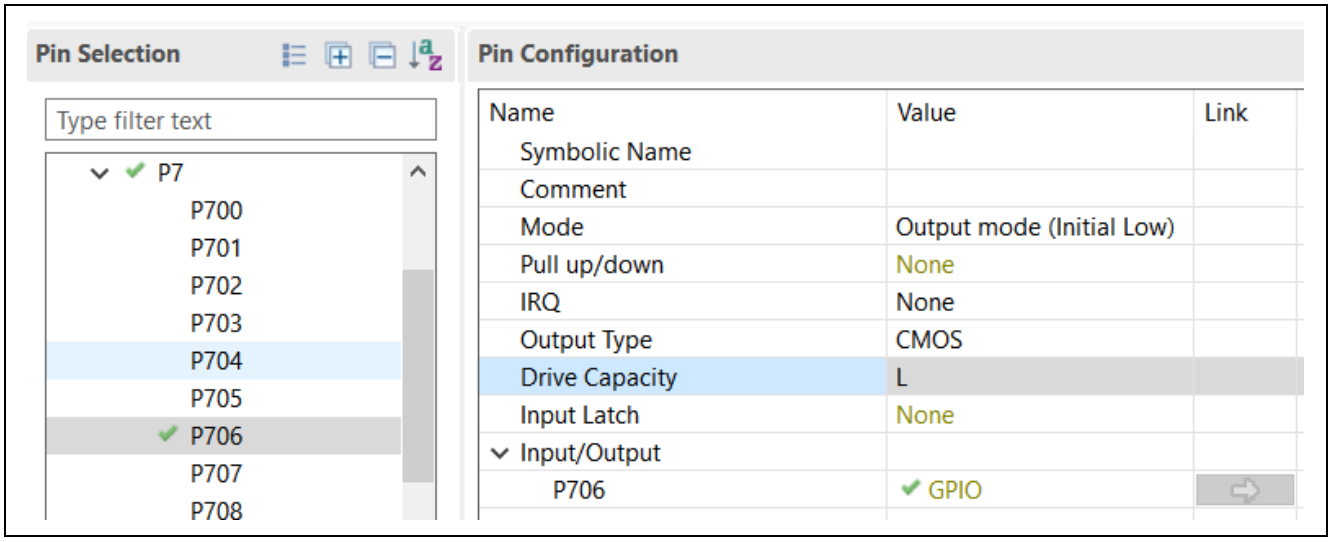


Figure 38. Configuring P706 as Output and Low using FSP Configurator

10.2.1 Internal Pull-Ups

- Most pins on ports 0 through D have the option of enabling a pull-up resistor. Most pins on ports 0 through 9, A through D have the option of enabling a pull-up resistor. The pull-up is controlled by the Pull-Up bit (PCR) bit in each Port mn Pin Function Select (PmnPFS) Register. The PCR bit in each PmnPFS register controls the corresponding pin on the port.
- The pin must first be set as an input with the associated bit in the PmnPFS register. Set the PCR bit to "1" to enable the pull-up and to "0" to disable it.
- Out of reset all PCR registers are cleared to 0, therefore all pull-up resistors are disabled.
- The pull-up is automatically turned off whenever a pin is designated as an external bus pin, a GPIO output, or a peripheral function output pin.

10.2.2 Open-Drain Output

- Pins configured as outputs normally operate as CMOS outputs.
- Most pins on ports 0 through D have the option of being configured as an NMOS open-drain output.Most pins on ports 0 through 9, A through D have the option of being configured as an NMOS open-drain output.
- The N-channel open-drain control (NCODR) bit in each Port mn Pin Function Select (PmnPFS) Register controls which pins operate in open-drain mode. Setting the applicable bit in each register to a "1" makes the output open-drain. Setting the applicable bit in each register to a "0" sets the port to CMOS output.

10.2.3 Drive Capacity

- The drive capacity switching is controlled by the Drive Capacity Control Register (DSCR) bits in each Port mn Pin Function Select (PmnPFS) register.
- Most port pins have the option of enabling low-, middle-, high-, or high-speed high-drive output . Refer to the “Peripheral Select Settings for Each Product” section in the Hardware User’s Manual for details of the options for each port.
- Port0 and P201 are limited to low-drive output only.
- P200 is input only
- P214, P215 drive capacity can not be controlled.
- Out of reset all DSCR registers are cleared to 0 therefore all pins are set to low drive output. Setting a value other than “00” will change the drive capacity of the output for the selected pin.
- The maximum total output of all pins summed together must not exceed the capabilities of the specific device. Refer to the Electrical Characteristics section of the Hardware User's Manual, specifically the sub-section "I/O I_{OH}, I_{OL}".
- The differences between the drive levels are shown in the following table.

Table 17. Output Pin Drive Capacity Levels

Typical output pins	DSCR[1:0]	Drive Capacity	Average (mA)	Max (mA)
Permissible output current per pin	0 0	Low Drive	2.0	4.0
Permissible output current per pin	0 1	Middle Drive	4.0	8.0
Permissible output current per pin	1 0	High-speed high-drive	20.0	40.0
Permissible output current per pin	1 1	High Drive	16	32

Output drive capacity varies depending on the port. Please refer to the Electrical Characteristics section of the Hardware User’s Manual for details of the output current capabilities for each port pin.

Output drive capacity can have a significant impact on overall performance of a board design. The following points should be considered when selecting the drive capacity for each output.

- It is recommended to start with all pins set to low-drive capacity (default) and evaluate the performance.
- Depending on the board layout, pins set to middle- or high-drive capacity may result in higher EMI radiation.
- Long traces may require higher drive capacity for signals to propagate correctly to the receiver.

10.3 Setting Up and Using Port Peripheral Functions

The Port mn Pin Function Select Registers (PmnPFS) are used to configure the characteristics of each port. The PSEL bits select the peripheral function selected for each port.

- Since most pins have multiple functions the RA8x2 MCUs have Pin Function Control Registers (PmnPFS) that allow you to change the function assigned to a pin.
- Each pin has its own PmnPFS register.
- Each PmnPFS register allows a pin to be used for peripheral function (PSEL bits), as an IRQ input pin (ISEL bit), or as an analog input pin (ASEL bit). If the ASEL bit is set to “1” (use pin as analog input pin) then the pin’s PMR bit should be set for GPIO use and the pin’s PDR bit should be set for input.
- Refer to the “Peripheral Select Settings for each Product” section in the “I/O Ports” chapter of the Hardware User’s Manual.
- In order to prevent unintentional voltage levels on peripheral pins, make sure to clear the Port Mode Control (PMR) bit for the targeted pin before modifying the pin’s PmnPFS register.
- All PmnPFS registers are write protected out of reset. In order to write to these registers, the Write-Protect Register (PWPR) must first be used to enable writing.
- Care should be taken when setting PmnPFS registers such that a single function is not assigned to multiple pins. The user should not do this, but the MCU will allow it. If this occurs the function on the pins will be undefined.
- The figure below shows an example of enabling OSPI pins using FSP Pin configuration.

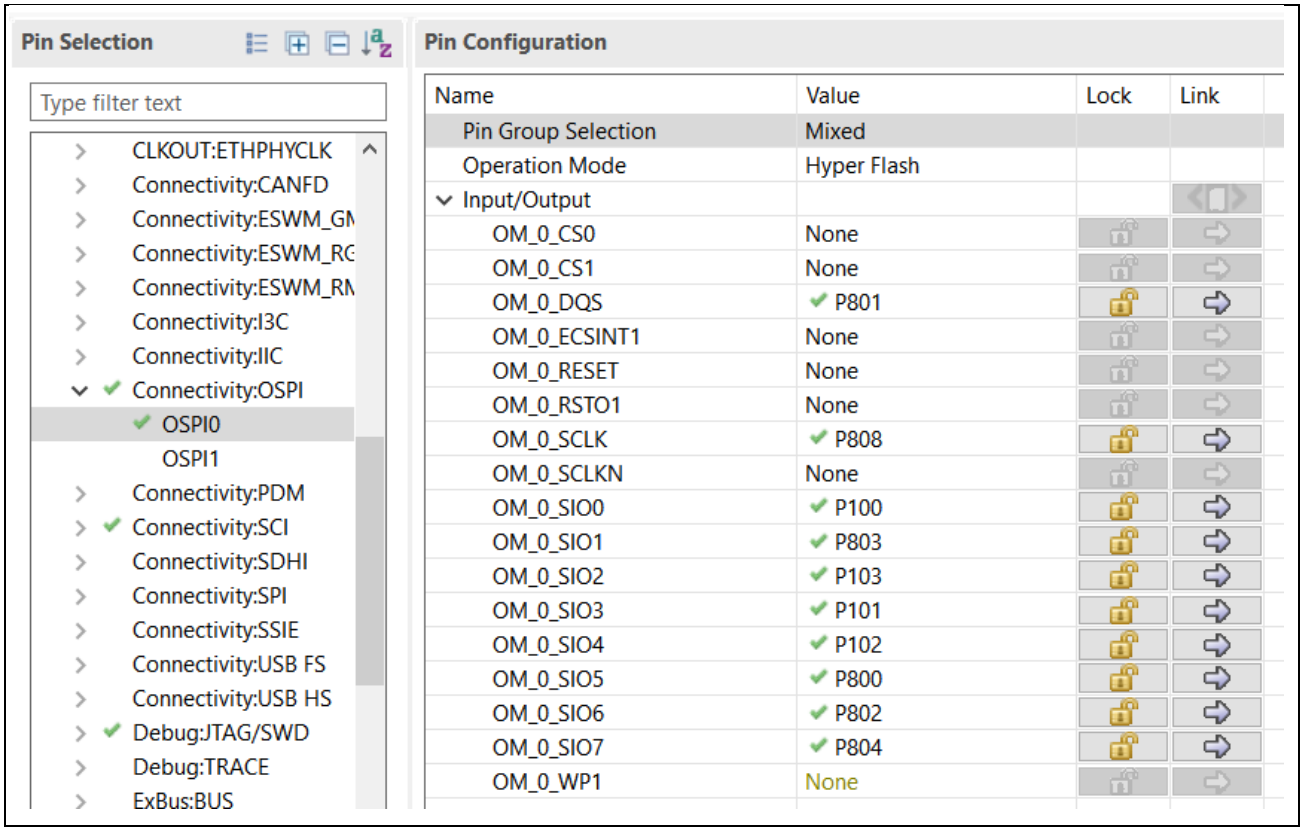


Figure 39. Example of Enabling OSPI pins using Pin Configurator in Renesas FSP

10.4 Setting Up and Using IRQ Pins

- Certain port pins can be used as hardware interrupt lines (IRQ). See the “Peripheral Select Settings for each Product” section in the “I/O Ports” chapter of the Hardware User’s Manual for information on which pins are available for your MCU.
- Some IRQ pins have a “-DS” suffix (e.g. IRQ1-DS). The “-DS” designates that this pin can be used to wake the MCU out of deep software standby mode.
- Note: It is not possible to use IRQn and IRQn-DS at the same time. Same number interrupts with the -DS and without the -DS suffix connect to the same interrupt internally, even though they use different external pin connections.
- To set a port pin to be used as an IRQ pin, the Interrupt Input Function Select bit (ISEL) in the pin’s PFS register must be set to “1”.
- Pins can be used for both IRQ and peripheral functions simultaneously. To enable this the user should set both the ISEL and PSEL bits in the pin’s PFS register.
- IRQ functions of the same number must only be enabled on one pin.
- IRQ pins can trigger interrupts on detection of:
  - Low level
  - Falling edge
  - Rising edge
  - Rising and falling edges

  Which trigger is selected is chosen using the IRQ Control Registers (IRQCRi).
- Digital filtering is available for IRQ pins. The filters are based on repetitive sampling of the signal at one of four selectable clock rates (PCLK, PCLK/8, PCLK/32, PCLK/64). They filter out short pulses: any high or low pulse less than 3 samples at the filter rate. The filters are useful for filtering out ringing and noise in

these lines, but are much too quick for filtering out long events like mechanical switch bounce. Enabling filtering adds a short bit of latency (the filter time) to the hardware IRQ lines.

- Digital filtering can be enabled for each IRQ pin independently. This is done by setting the IRQ Pin Digital Filter Enable (FLTEN) bit in the IRQCRi register for each IRQ.
- The clock rate for digital filtering is configurable for each IRQ pin independently. This is done by setting the IRQ Pin Digital Filter Setting (FCLKSEL[1:0]) bits in the IRQCRi register for each IRQ.
- Figure 40 and Figure 41 show examples of enabling and configuring IRQ pins using Renesas FSP.

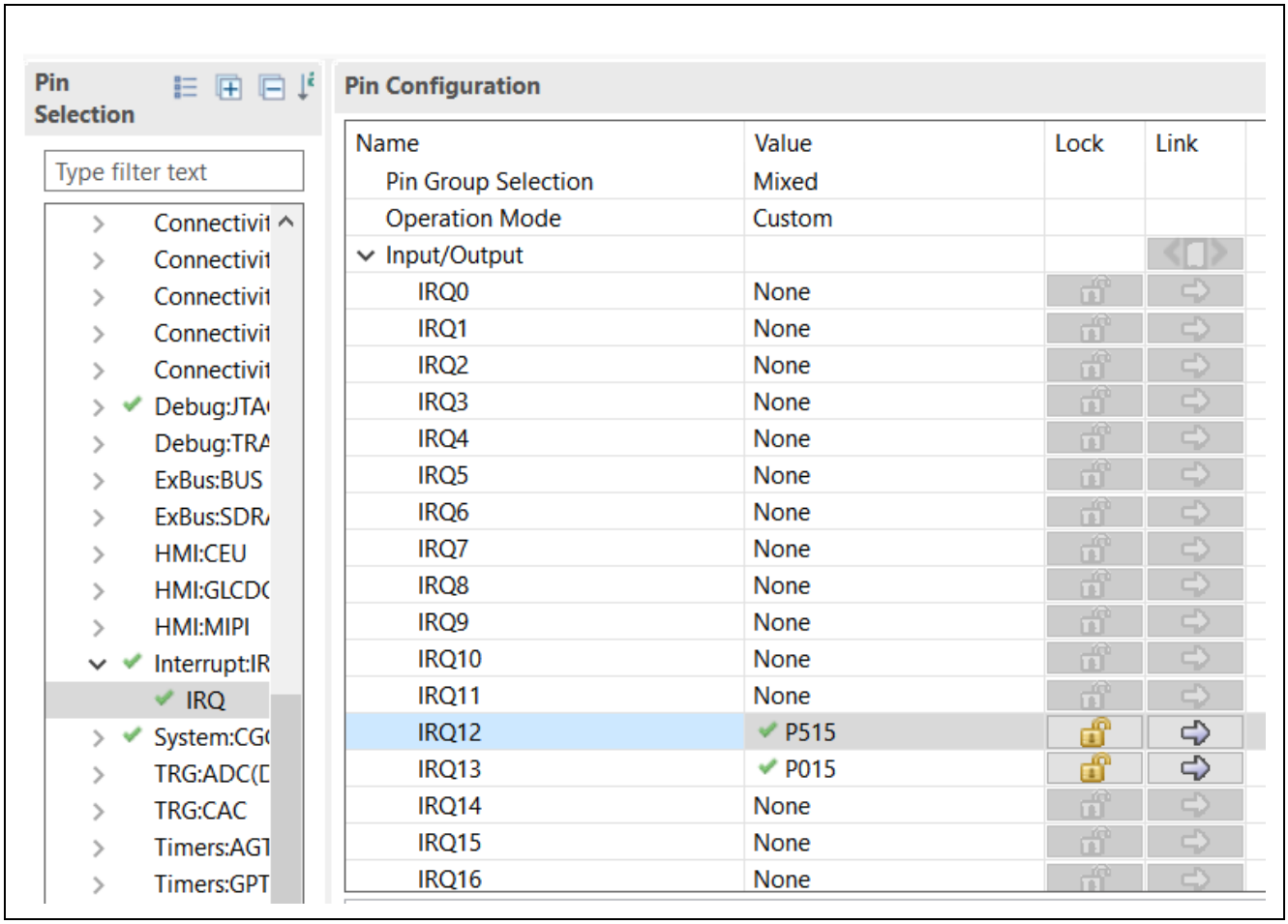


Figure 40. Enable P515, P015 as IRQ12, IRQ13 inputs Respectively using Pin Configurator in Renesas FSP

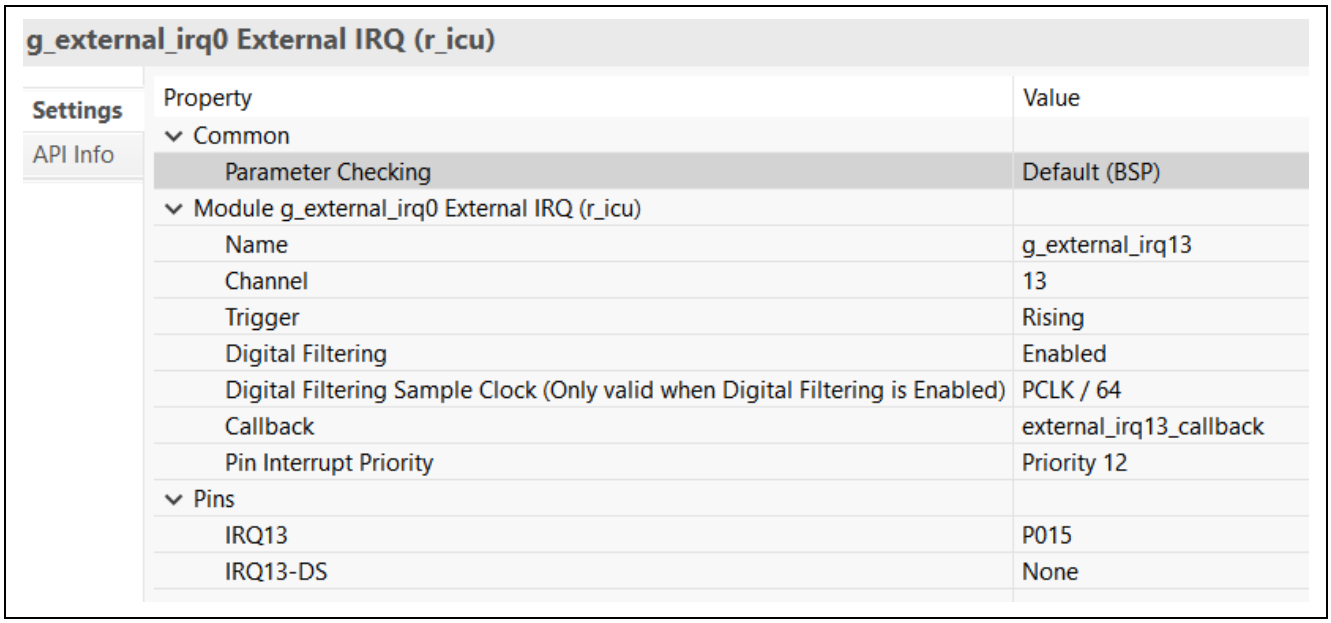


Figure 41. Configure IRQ13 using Renesas FSP Configurator

10.5 Unused Pins

Note: Some pins require specific termination: See the “Handling of Unused Pins” section of the Hardware User’s Manual for specific recommendations.

Unused pins that are left floating can consume extra power and leave the system more susceptible to noise problems. Terminate unused pins with one of the methods detailed here:

1. The first option is to set the pin to an input (the default state after Reset) and connect the pin to VCC or VSS using a resistor. There is no difference to the MCU between one connection or another; however, there may be an advantage from a system noise perspective. VSS is probably the most typical choice. Avoid connecting a pin directly to VCC or VSS since an accidental write to the port’s direction register that sets the pin to an output could create a shorted output.
2. A second method is to set the pin to an output. It does not matter whether the pin level is set high or low; however, setting the pin as an output and making the output low connects the pin internally to the ground plane. This may help with overall system noise concerns. A disadvantage of setting unused pins to outputs is that the configuration of the port must be done via software control. While the MCU is held in Reset and until the direction register is set for output, the pin will be a floating input and may draw extra current. If the extra current can be tolerated during this time, this method eliminates the external resistors required in the first method.
3. A variation on leaving the pins as inputs and terminating them with external resistors uses the internal pull-ups available on many ports of the MCU. This has the same limitation as setting the pins to outputs (requires the program to set up the port) but it does limit the effect of accidental pin shorts to ground, adjacent pins or VCC since the device will not be driving the pin.

10.6 Nonexistent Pins

Each RA8x2 MCU group is available in multiple package sizes, with different total pin counts. For any package smaller than the largest package for that MCU group, set the corresponding bits of nonexistent ports in the PDR register to “1” (output) and in the PODR register to “0”. The user can see which ports are available on each MCU package by reviewing the “Specifications of I/O Ports” table in the I/O Ports section of the Hardware User’s Manual. Note that no additional handling of nonexistent pins is required.

10.7 Electrical Characteristics

Normal GPIO ports typically require CMOS level inputs (High ≥ 0.8 * VCC, Low ≤ 0.2 * VCC). Some GPIO ports have Schmitt Trigger inputs, which have slightly different input requirements. See the Hardware User’s Manual section “Electrical Characteristics” for more information.

11. Module Stop Function

To maximize power efficiency, the RA8x2 MCUs allow on-chip peripherals to be stopped individually by writing to the Module Stop Control Registers (MSTPCRi, I = A, B, C, D, E). Once a module stops, access to the registers associated with the module is not possible.

After a reset, most of the modules are placed in module-stop state, except for DMAC, DTC, and SRAM. See Hardware User's Manual for details.

Before accessing any of the registers for a peripheral, it must be enabled by taking it out of stop mode by writing a '0' to the corresponding bit in the MSTPCRi register.

Peripherals may be stopped by writing a '1' to the proper bit in the MSTPCRi register.

HAL drivers in Renesas FSP handle module start/stop function automatically.

12. Interrupt Control Unit

The Interrupt Controller Unit (ICU) controls which event signals are linked to the NVIC, DTC, and DMAC modules. The ICU also controls non-maskable interrupts.

The maskable interrupts in RA8x2 MCUs can all be used to wake either CPU from Deep Sleep Mode. The non-maskable interrupts have specific functions that may apply to only one CPU. Refer to the sections "Maskable Interrupt Setting Procedure" and "Non-Maskable Interrupt Operation" in the Hardware User's Manual for details.

Table 18 and Table 19 show an example of the ICU specifications, and Table 20 shows an example of the ability to raise the IRQi event from the I/O pins. Refer to the Hardware User's Manual for details for each RA8x2 MCU Group.

Table 18. Example of RA8x2 ICU Specification (1 of 2)

Parameter		Description
Maskable interrupts	Peripheral function interrupts	Interrupts from peripheral modules Number of sources: 497 (a factor is chosen with an event list number 33 to 1023)
	External pin interrupts	• Interrupt detection on low level*4, falling edge, rising edge, rising and falling edges. One of these detection methods can be set for each source • Digital filter function supported • 32 sources, with interrupts from IRQi (i = 0 to 31) pins.
	CPU mutual interrupt	• IPC CPU mutual interrupt 0 (IPC_IRQ0) • IPC CPU mutual interrupt 1 (IPC_IRQ1)
	Interrupt requests to CPU (NVIC)	96 interrupt requests are output to NVIC.
	DMAC control	• The DMAC can be activated using interrupt sources*1 • The target interrupt source can be selected individually for every DMAC channels.
	DTC control	• The DTC can be activated using interrupt sources*1 • The method for selecting an interrupt source is the same as that of the interrupt request to NVIC.
Non-maskable interrupts*2	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection on falling edge or rising edge • Digital filter function supported
	Oscillation stop detection interrupt*3	Interrupt on detection of main oscillation is stopped
	Sub Oscillation stop detection interrupt*3	Interrupt on detection of sub oscillation is stopped
	WDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error (CPU0, CPU1)
	IWDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Voltage-monitor 1 interrupt*3	Voltage monitor interrupt of the voltage detection circuit 1 (PVD_PVD1)
	Voltage-monitor 2 interrupt*3	Voltage monitor interrupt of the voltage detection circuit 2 (PVD_PVD2)
	Common memory error interrupt	Common memory error means SRAM ECC error.
	Local memory error interrupt	Local memory error interrupt includes Cache and TCM ECC Error. (CPU1 only)
	Bus error Interrupt	Bus error interrupt includes MPU and TZF error
	Lockup error interrupt	Lockup error interrupt (CPU0, CPU1)
	CPU mutual Interrupt	IPC NMI CPU mutual interrupt (CPU0, CPU1)
	FPU Exception interrupt	FPU Exception interrupt (CPU0, CPU1)
	MRAM read error interrupt	• MRAM read error interrupt for MRC • MRAM read error interrupt for MRE
Security	Secure	Some registers have Security Attribution.
	Privilege	Each register of the ICU can only be accessed with Privilege access.

Table 19. Example of RA8x2 ICU Specification (2 of 2)

Parameter	Description
Return from low power mode	• CPU Sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source. • CPU Deep Sleep mode: Return is initiated by non-maskable interrupts. Interrupt can be selected as WUPEN0, WUPEN1, DSLPWUPIRQENj register (j = 0 to 2). See section 14.2.4. ICUSARF : Interrupt Controller Unit Security Attribution Register F, section 14.2.9. ICUSARK : Interrupt Controller Unit Security Attribution Register K, section 14.2.10. ICUSARL : Interrupt Controller Unit Security Attribution Register L, section 14.2.11. TEVTRCR : Trusted Event Route Control Register, section 14.9.2. Return from CPU Deep Sleep Mode. • Software Standby mode: Return is initiated by non-maskable interrupts. Interrupt can be selected as WUPEN0, WUPEN1 register. See section 14.2.4. ICUSARF : Interrupt Controller Unit Security Attribution Register F, section 14.2.9. ICUSARK : Interrupt Controller Unit Security Attribution Register K, section 14.2.10. ICUSARL : Interrupt Controller Unit Security Attribution Register L, section 14.9.2. Return from CPU Deep Sleep Mode.

Note 1. For the DMAC and DTC activation sources, see Table 14.4.
Note 2. Non-maskable interrupts can be enabled only once after a reset release.
Note 3. These non-maskable interrupts can also be used as interrupts in general. When used as maskable interrupts, do not change the value of the NMIER register from the reset state. To enable voltage monitor 1 to 2 interrupts, set the PVD1CR1.PVD1IRQSEL and PVD2CR1.PVD2IRQSEL bits to 1.
Note 4. Low level: interrupt detection is not canceled if you do not clear it after a detection.

Table 20. Example of RA8x2 ICU I/O Pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQi (i = 0 to IRQ31)	Input	External interrupt request pins

The following figure is an example of using Renesas FSP configurator to enable and configure an interrupt using Renesas FSP. The ICU and interrupts are configured as part of the HAL driver configuration through FSP.

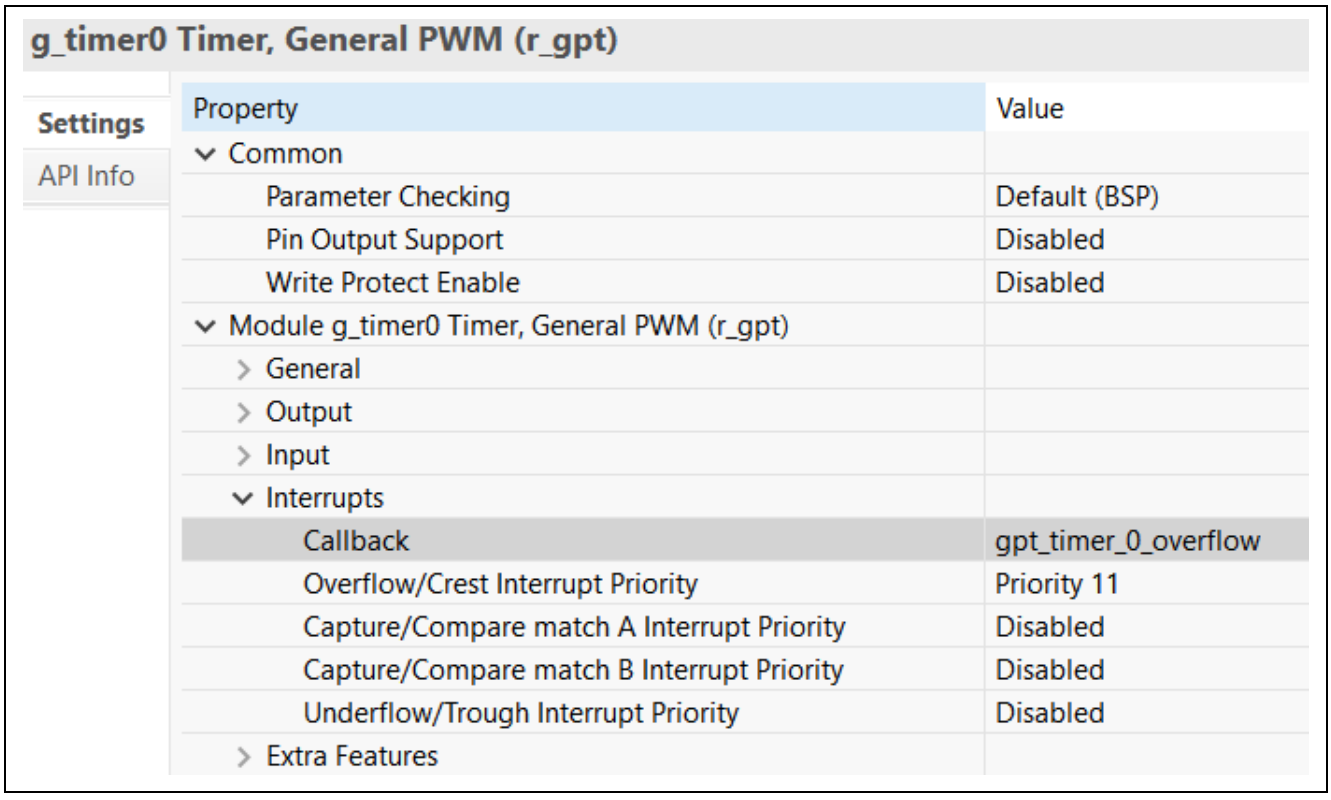


Figure 42. Enable GTP0 Overflow Interrupt and Set User Callback Functions which will be Invoked by Interrupt Service Routine

13. Pulse Density Modulation Interface (PDM-IF)

The Pulse Density Modulation Interface (PDM-IF) has a maximum of three channels that are connectable with external microphone which outputs the pulse density modulated signal. PDM-IF is connectable with up to three external microphones. PDM-IF can filter and convert 1-bit digital data streams that were pulse density modulated at a high sampling rate into 20-bit or 16-bit digital data at a lower sampling rate.

Table 21. PDM-IF Specification

Item	Description
Number of channels	3 channels
Functions	• Capable of filtering 1-bit digital input data PDM_DATAn (n = 0 to 2) and converting them into 20-bit or 16-bit digital data. The bit order is little-endian. • IP supports stereo microphone (L/R sampling by rising/falling clock edge). • IP supports sound activity detector. • Each channel of IP includes programmable filters: 4th order sinc filter, high-pass filter (for suppression of DC bias), correction filter (for sinc passband distortion), half-band decimation filter (for aliasing distortion). – IP supports programmable and flexible decimation ratios. – The sinc filter is selectable as first-, second-, third-, fourth-order. • IP supports DMA operation through APB. • Each channel of IP has an internal buffer. – Buffer size as 32 stages at design time – Capable of storing voice data during low power mode • Error detection functions can be used for debugging.
Interrupt sources	Maximum 7 sources 1. Data reception interrupt per channel (Maximum 3 interrupts) 2. Sound detection interrupt (shared between channels) 3. Error detection interrupt per channel (Maximum 3 interrupts)
Low power consumption function	• IP supports microphone low power configuration (slower clock). IP can switch to higher clock speed after sound detection recognized. • MCU can stop bus clock during low power mode.
Bus interface	• IP has an APB interface which is confirmed to APB4.
TrustZone filter	• Security and Privilege attribution can be set.

Figure 43 and Figure 44 show examples of PDM-Microphone design and PDM-IF settings in FSP.

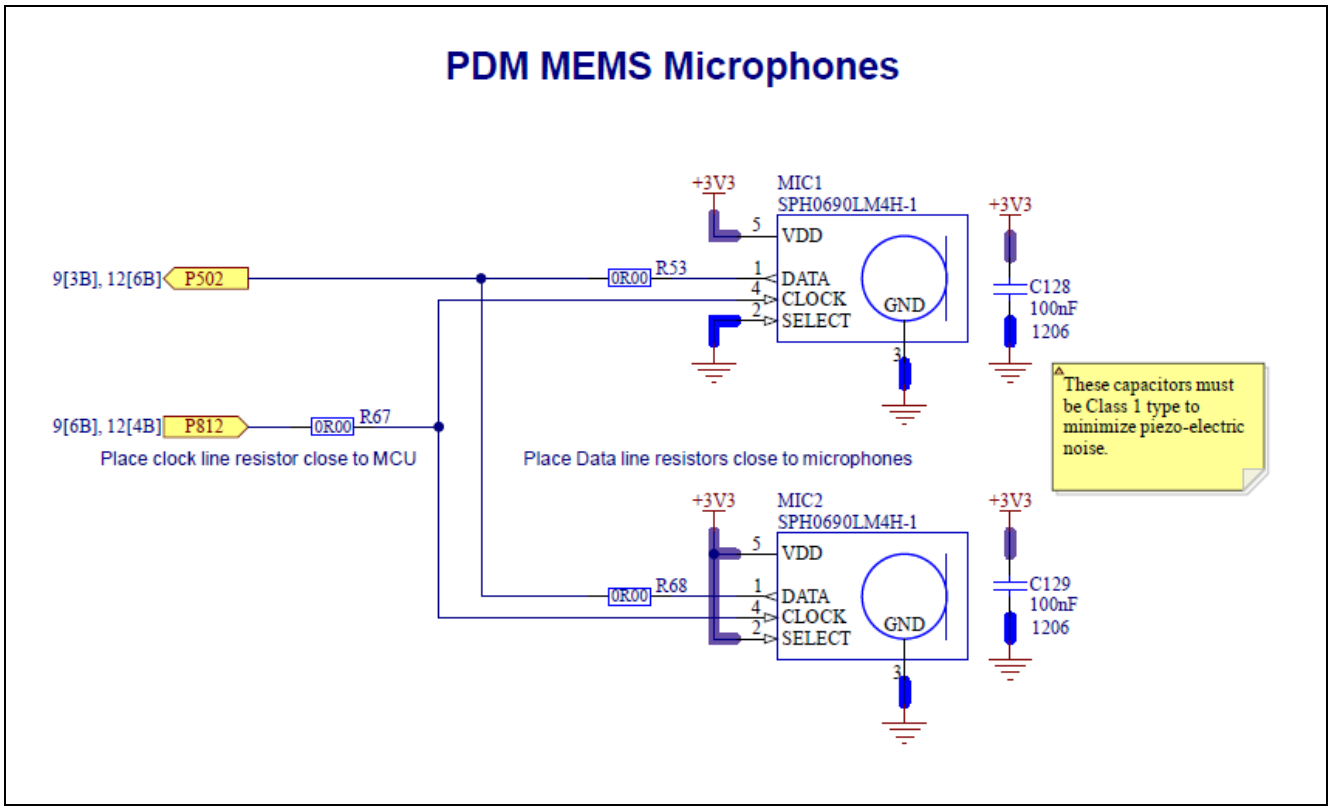


Figure 43. Example of PDM-Microphone Design

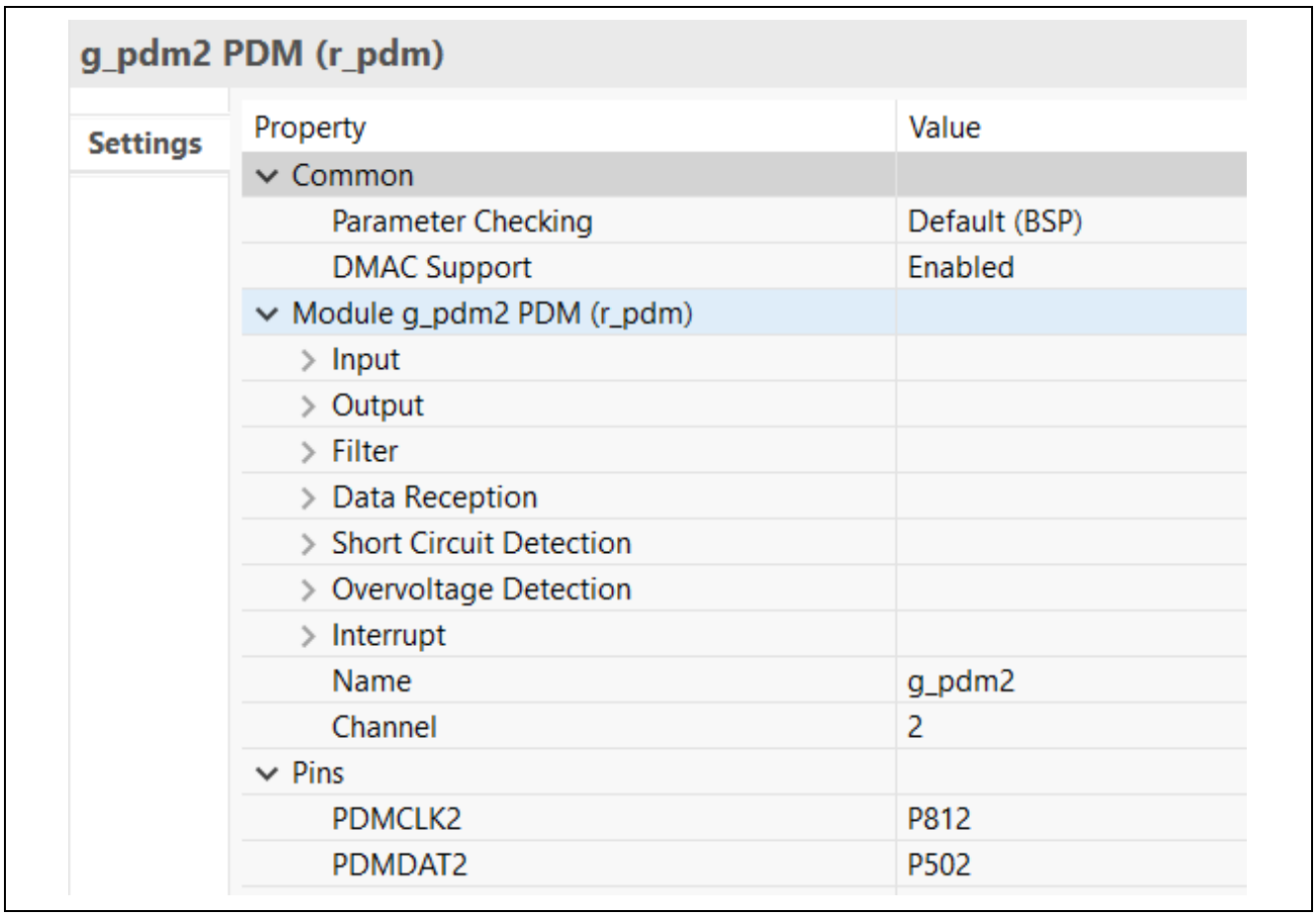


Figure 44. Example of PDM Module Settings in FSP

14. Inter-Processor Communication (IPC)

Inter-Processor Communication (IPC) supports hardware sharing and communication between two processors inside the CPUs. IPC can generate interrupt events to support communication between each processor.

Several semaphore mechanisms are required to achieve mutually exclusive access between processors. To achieve this, IPC implements multiple locks. Application developers choose to use separate or shared hardware semaphores between the two CPUs. A mechanism is also required for simple data communication between CPUs. For example, a FIFO that CPU0 uses for write-only data and CPU1 uses for read-only data. In addition, for mutual use, a FIFO is required with CPU1 as write-only data and CPU0 as read-only data.

This section does not apply to single-core versions of these MCU devices.

Table 22. IPC Specification

Item		Description
Semaphore	Lock information	• A total of 16 factors • Security attributes and privileged attributes are selected every 8 factors
Inter-Processor interrupt	Non-maskable	• IPC0: one factor, one non-maskable interrupt, security attributes and privileged attributes are selected. • IPC1: one factor, one non-maskable interrupt, security attributes and privileged attributes are selected.
	Maskable	• IPC00: 8 factors, one maskable interrupt, security attributes and privileged attributes are selected*1 • IPC01: 8 factors, one maskable interrupt, security attributes and privileged attributes are selected*1 • IPC10: 8 factors, one maskable interrupt, security attributes and privileged attributes are selected*1 • IPC11: 8 factors, one maskable interrupt, security attributes and privileged attributes are selected*1
Message-FIFO	FIFO	• Message FIFO 00 (CPU1 -> CPU0), one maskable interrupt, security attributes and privileged attributes are selected*1 • Message FIFO 01 (CPU1 -> CPU0), one maskable interrupt, security attributes and privileged attributes are selected*1 • Message FIFO 10 (CPU0 -> CPU1), one maskable interrupt, security attributes and privileged attributes are selected*1 • Message FIFO 11 (CPU0 -> CPU1), one maskable interrupt, security attributes and privileged attributes are selected*1 • 4 FIFO stages • Transfer data size 32 bits
	Error information	• Write with FIFO FULL*1 • Read with FIFO Empty*1

Note 1. The interrupts for IPC00, Message FIFO 00 and Error information are common. The same applies to others.

14.1 Using IPC In Application

Renesas FSP configurator to enable and configure an IPC using Renesas FSP. The IPC and its interrupts behaviors are configured as part of the HAL driver configuration through FSP.

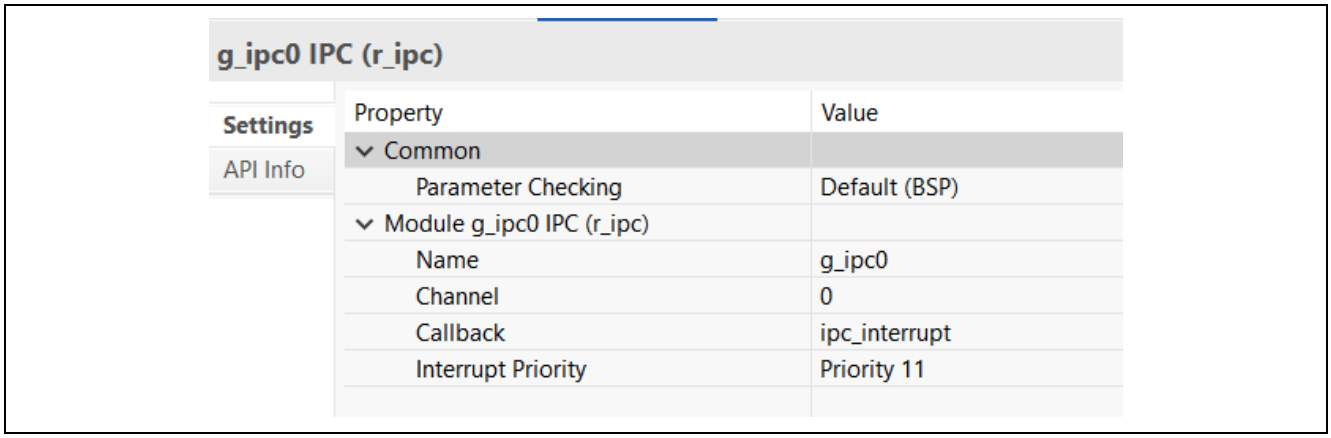


Figure 45. Example of IPC Configuration

IPC can generate interrupt events to support communication between CPU0 and CPU1, including Maskable and Non-Maskable interrupt.

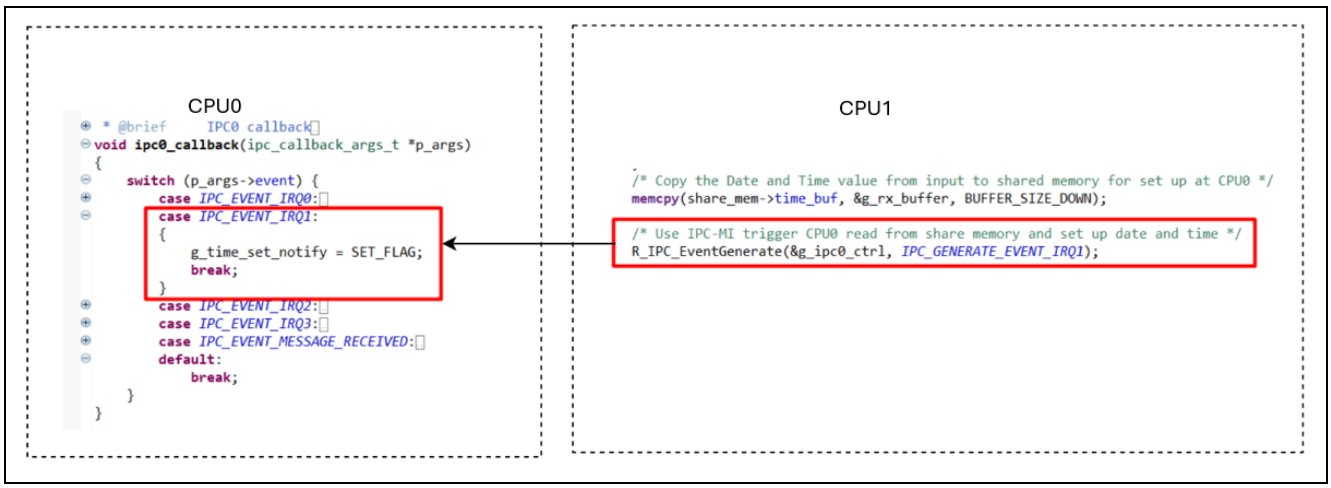


Figure 46. Example of IPC Interrupt Use with Renesas FSP APIs

Refer to MCU hardware manual and the Application Note Developing with RA8x2 MCU, document No. R01AN788 for more details.

15. Low Power Consumption

The RA8x2 devices have several functions for reducing power consumption. These include setting clock dividers, EBCLK and SDCLK output controls, stopping modules, power gating control, selecting operating power control mode in Normal mode, and transitions to low power modes. Table 23 lists the specifications of functions to reduce power consumption. See the “Low Power Mode” section in the Hardware User’s Manual, which lists the conditions to shift to low power modes, states of the CPU and peripheral modules, and the method for cancelling each mode. After a reset, the MCU enters the program execution state, but modules, except for the DMAC, DTC, and SRAM, do not operate.

RA8x2 MCUs support four different types of Low Power Consumption depending on the MCU Group. These are:

- CPU Sleep mode and CPU Deep Sleep mode
- Software Standby mode
- Deep Software Standby mode 1, 2, 3

The following table is an overview of the functions available for reducing power consumption.

Table 23. Specifications of the Lower Power Mode Functions

Item	Specification
Reducing power consumption by modifying clock signals	The frequency division ratio can be selected independently for the system clock, peripheral module clock, external bus clock, and MRAM. *1
EBCLK output control	BCLK output or high-level output can be selected.
SDCLK output control	SDCLK output or high-level output can be selected.
Module stop	Functions can be stopped independently for each peripheral module.
Power gating control	This function can be used to control the power state of the power domain. • Control the turning On/OFF for the power domain • Control the retention of specific circuits during power gating
Processor low power modes	• CPU Sleep mode • CPU Deep Sleep mode
Low-power modes	• Software Standby mode*2 • Deep Software Standby mode 1, 2, 3*2

Notes: 1. For details, see the chapter “Clock Generation Circuit” in the Hardware User’s Manual.
2. This mode is not supported in external VDD mode.

Sleep Mode and Deep Sleep Mode for both CPU0 and CPU1 can be cancelled by any available interrupt.

Table 24. Operating State of Processor Low Power Mode

Item	CPU0 Sleep Mode	CPU1 Sleep Mode	CPU0 Deep Sleep Mode	CPU1 Deep Sleep Mode
Transition Condition	WFI instruction after set CPU0.SCR. SLEEPDEEP= 0.	WFI instruction after set CPU1.SCR. SLEEPDEEP= 0.	WFI instruction after set CPU0.SCR. SLEEPDEEP= 1	WFI instruction after set CPU1.SCR. SLEEPDEEP= 1
Canceling method	All interrupts. Any reset available in the mode.	All interrupts. Any reset available in the mode.	All interrupts. Any reset available in the mode.	All interrupts. Any reset available in the mode.
State after cancellation by an interrupt	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)
State after cancellation by	Reset state	Reset state	Reset state	Reset state

a reset				

Notes: 1. Refer to the table “Operating state of processor low power mode” in the Hardware User’s Manual for additional details.

The “Operating state of processor low power mode” and “Operating state of each low power mode” table in MCU Hardware User’s Manual list the conditions to transition to low power modes, the states of the CPU and the peripheral modules, and the method for cancelling each mode.

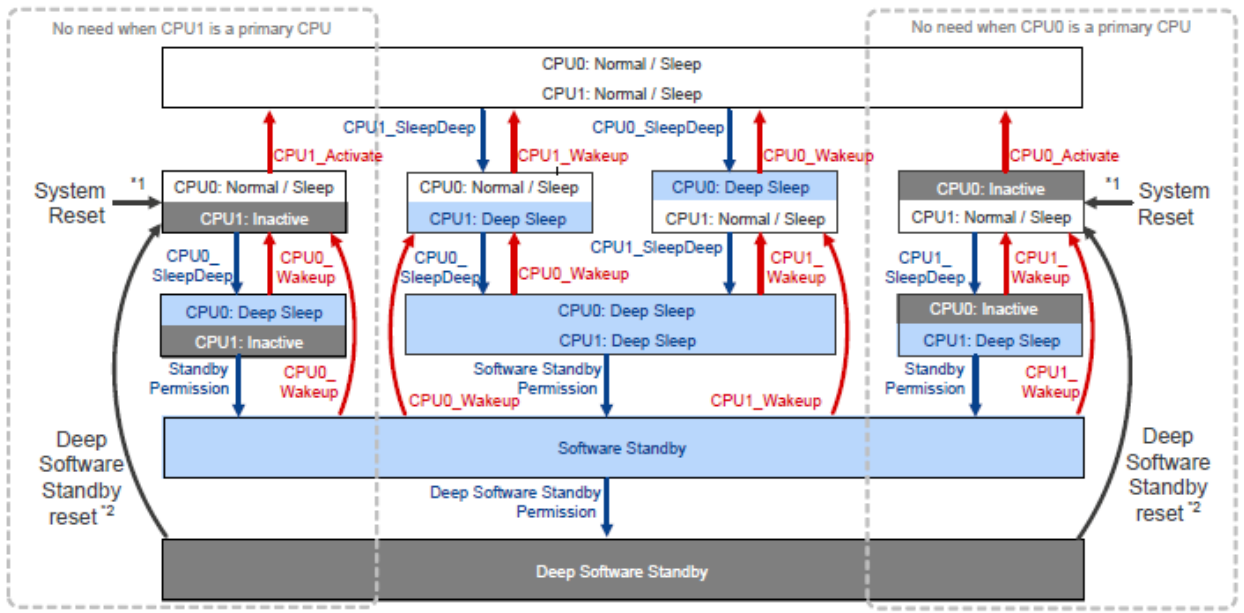
Table 25. Low Power Consumption Modes for CPUx(x=0,1)

State of operation*1	**Software Standby Mode**	**Deep Software Standby Mode 1, 2, 3**
Transition condition	WFI instruction after set LPSCR=0x5 and CPUx.SCR.SLEEPDEEP=1	WFI instruction after set 0x8 ≤ LPSCR ≤ 0xA and CPUx.SCR.SLEEPDEEP=1.
Canceling method	Interrupts shown in table “Interrupt source for canceling Software Standby and Deep Software Standby modes”. Any reset available in the mode.	Interrupts shown in table “Interrupt source for canceling Software Standby and Deep Software Standby modes”. Any reset available in the mode.
State after cancellation by an interrupt	Program execution state (interrupt processing)	Reset state
State after cancellation by a reset	Reset state	Reset state

Notes: 1. Refer to the table “Operating conditions of each low power mode” in the Hardware User’s Manual for additional details.

To achieve the lowest power numbers, use the maximum possible dividers in the clock generation circuits.

Figure 47 show LPM transitions in both CPU0 and CPU1.



Note: When an interrupt as a trigger for cancel is received during a transition to the program-stopped state after the execution of a WFI instruction, the CPU that receives the interrupt executes interrupt exception handling instead of transitioning to low power mode. When entering to Deep Software Standby mode, if an interrupt occurs after the state has transitioned, after mode transition is completed, returning with Deep Software Standby reset.

Note: When the transition to Normal mode is done because of an interrupt from CPU Sleep mode, CPU Deep Sleep mode or Software Standby mode, the clock source is the same as before entering each low power modes.

Note: For interrupt sources for canceling CPU Sleep mode and CPU Deep Sleep mode, see section 2.7. Sleep Modes for Low Power States.

Note: For interrupt sources for canceling Software Standby mode and Deep Software Standby mode, see Table 11.4.

Note 1. The MOCO clock is the source of the operating clock following a transition from the reset state to Normal mode.

Note 2. When an available interrupt request is generated, an internal reset (Deep Software Standby reset) is generated over a fixed period. Canceling of Deep Software Standby mode accompanies release from the internal reset state, and then the primary CPU transitions to Normal mode and executes reset exception processing with the MOCO clock as the source of the operating clock. The secondary CPU becomes inactive.

Figure 47. Low power mode transitions

Low power modes are cancelled by various interrupt sources such as RES pin reset, power-on reset, voltage monitor reset, and peripheral interrupts. Refer to the Low Power Modes section in MCU Hardware User's Manual for a list of interrupt sources for different LPMs.

Renesas FSP provides a low power mode (LPM) driver and driver configurator to set up low power mode, wake source/cancel source, and so forth.

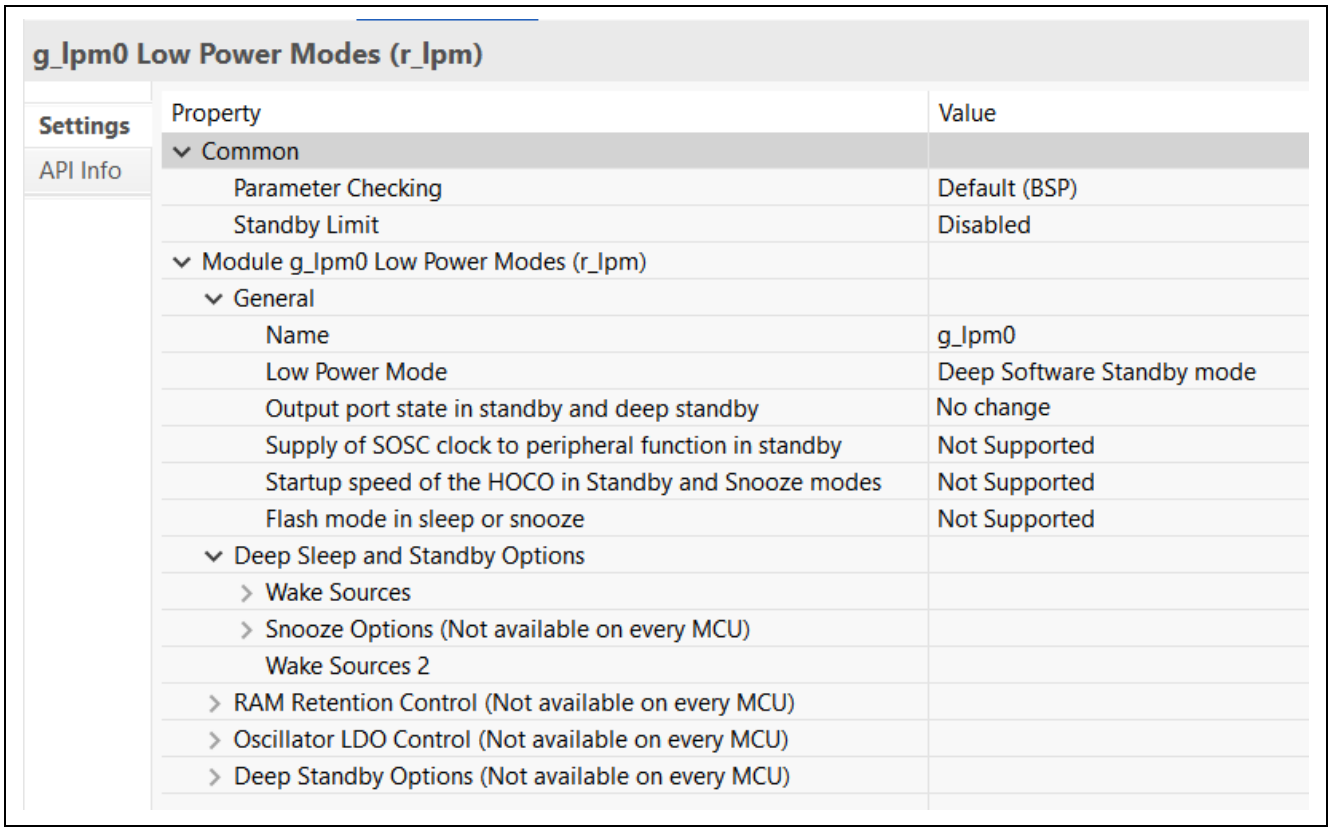


Figure 48. Set up Low Power Mode Using Renesas FSP Configurator

After a specific LPM mode is set up by FSP Configurator, LPM driver's API can be used to initialize LPM driver and place MCU in configured LPM mode:

```
/* Open LPM driver and initialize LPM mode */
err = R_LPM_Open(&g_lpm_ctrl_instance_ctrls[g_lpm_transition_pos],
       &g_lpm_ctrl_instance_cfgs[g_lpm_transition_pos]);
/* Handle error */
if (FSP_SUCCESS != err)
  {
      return (err);
  }
/* Transition to configured LPM mode */
err = lpm_mode_enter(g_lpm_transition_sequence[g_lpm_transition_pos]);
/* Handle error */
if (FSP_SUCCESS != err)
  {
      return (err);
  }
```

16. External Buses

RA8x2 devices include an external bus controller. Some RA8x2 devices have built-in SDRAM controllers.

16.1 Bus Width and Multiplexing

The access width of external memory areas can be set to 8-bit, 16-bit, or 32-bit. Width settings are set on a per-chip-select basis by setting the BSIZE bits in the CSnCR register or the SDC Control Register (SDCCR). The address and data lines of chip-select regions can be multiplexed by setting the MPXEN bit in the CSnCR register.

16.2 Drive Strength for Bus Signals

When an external memory area is used, pins that control the bus signals should be set for high-drive capacity output in high-speed setting. See the section "Port mn Pin Function Select Register" in the "I/O

Ports” chapter, and the “Electrical Characteristics” chapter in MCU Hardware User’s Manual for more information on setting the drive capacity of a pin.

16.3 Bus Errors

The following types of errors can occur on each bus:

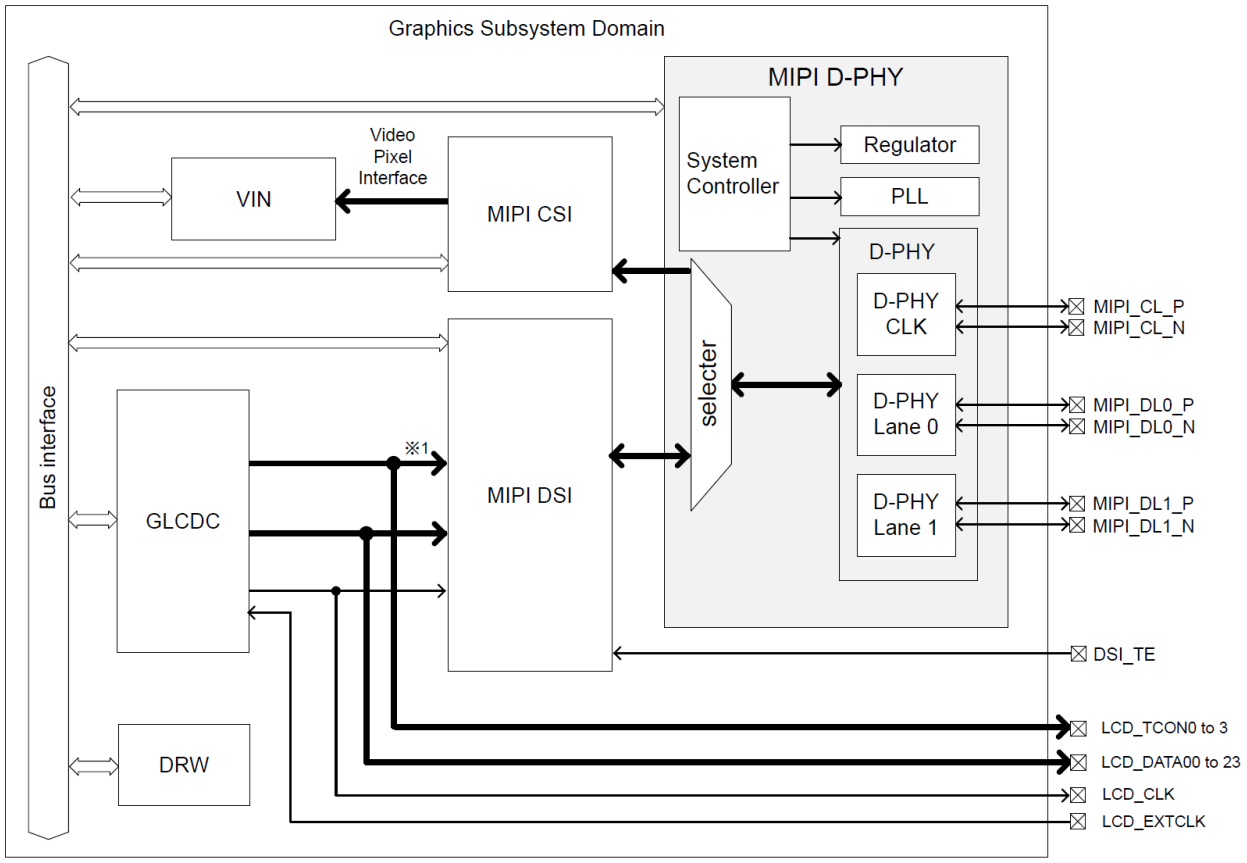
- Slave Bus Error
- Master MPU Error
- Illegal Address Access Error
- Master Security Attribution Unit Error

When a bus error occurs, the operation is not guaranteed, and the error is returned to the requesting master IP. The bus errors that occur for each master are stored in the BUSERRSTAT<Master Name> registers. These registers must only be cleared by a reset. For more information, see section “BUSERRSTAT<Master Name> : BUS Error Status Register” in the Hardware User’s Manual.

17. Graphics Subsystem

The graphic domain has the following features.

- Graphics LCD Controller (GLCDC): 8/16/18/24-bit RGB LCD interface supported.
- 2D Drawing Engine (DRW): Raster and vector graphics supported.
- MIPI D-PHY controller (MIPI PHY): Up to 2Lanes.
- Max rate: 720 Mbps/Lane
- MIPI Display Serial Interface (MIPI DSI): Display Serial interface 2 supported.
- MIPI Serial Interface (MIPI CSI):
- MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2) Ver 3.0
- MIPI Alliance Specification for D-PHY Ver 2.5 (Supported functions are equivalent to Ver 2.1).
- Video Input Module (VIN): Stores YCbCr-422 data and RGB data in memory through MIPI CSI interface.



Note 1. MIPI DSI is connected to only LCDC_TCON0 to 2.

Figure 49. Graphics Domain

17.1 MIPI Subsystem

Some RA8x2 MCUs have MIPI-DSI and MIPI PHY integrated as parts of the graphics domain. You can utilize the MIPI-DSI and MIPI PHY for graphics design with low pin count. Only one of either MIPI-CSI or MIPI-DSI can be used at a time.

Specifications for the MIPI interface are governed by the MIPI Alliance. MIPI DSI-2℠ is the Display Serial Interface specification. MIPI CSI-2® is the Camera Serial Interface specification. MIPI D-PHY℠ is the Physical Layer specification, which applies to both MIPI DSI-2 and MIPI CSI-2. These specifications are available from the MIPI Alliance. (www.mipi.org) Note that membership in the MIPI Alliance may be required to obtain the latest versions of these specifications.

17.2 MIPI DSI

The MIPI subsystem on some RA8x2 devices incorporates a MIPI DSI-2 Host module. The DSI-2 Host module has a transmitter function for MIPI Alliance Specification for Display Serial Interface 2 (DSI-2). The related D-PHY module supports MIPI Alliance Specification Version 2.1 for D-PHY Specification.

Table 26. MIPI DSI Specifications

Parameter		Specifications
Compliant specification		• MIPI Alliance Specification for Display Serial Interface 2 Version 1.1 (with Errata 01) • MIPI Alliance Specification for D-PHY Version 2.1 (with Errata 01) (80 Mbps to 720 Mbps/lane and up to 2 lanes).
Video mode operation	Available input video format from GLCDC	• Parallel RGB888 (24 bits), little endian • Parallel RGB666 (18 bits), little endian • Parallel RGB565 (16 bits), little endian
	Available output format	• RGB (16 bits, 18 bits, 24 bits)
	Available video mode packet sequence	• Non-Burst mode with sync pulse • Non-Burst mode with sync event • Burst mode
	Others	• Selectable blanking packet or LP-11 during each of blanking interval of HSA, HBP, and HFP
Command mode operation*1	Sequence operation channel 0	LP-only packet generation and LP packet reception from descriptor list
	Sequence operation channel 1	HS or LP packet generation and LP packet reception from descriptor list
DSI Link support functions		• 1 and 2 lane configurations • Unidirectional High-Speed mode transfer (HS-TX) • Bidirectional LP mode transfer/receipt (LP-TX/LP-RX) (Only Lane 0) • ECC/Checksum generation for TX packet • ECC/Checksum verification and ECC error correction for RX packet • Ultra-Low Power mode (ULPS) • Automated power change to LP mode and return to HS mode • Automated clock stop and resume (Non-Continuous Clock mode) • Assignment for virtual channel in Video mode • Assignment for individual virtual channel for each packet in Command mode • Detection for PHY contention error and timeout error • Generation of scrambled packets • Input of TE signal
Module-stop function		Module-stop state can be set to reduce power consumption.
TrustZone Filter		Security and Privilege attribution can be set.

Note 1. Ch0 and Ch1 are exclusive and cannot be used simultaneously.

Refer to the MIPI DSI section in the RA8x2 MCU Hardware User's Manual for more details.

The DSI-2 host module supports Ultra-Low Power mode (ULPS) for conserving energy. FSP provides APIs to enter and exit ULPS as follows.

Enter ULPS:

```
fsp_err_t err = FSP_SUCCESS;
/* Enter Ultra-low Power mode ULPS) */
err = R_MIPI_DSI_UlpsEnter (&g_mipi_dsi_ctrl,(mipi_dsi_lane_t)
(MIPI_DSI_LANE_DATA_ALL));
```

Exit ULPS:

```
fsp_err_t err = FSP_SUCCESS;
/* Exit Ultra-low Power mode (ULPS) */
err = R_MIPI_DSI_UlpsExit (&g_mipi_dsi_ctrl, (mipi_dsi_lane_t)
(MIPI_DSI_LANE_DATA_ALL));
```

17.3 MIPI CSI

The MIPI subsystem on some RA8 devices incorporates a MIPI CSI-2 module. The CSI-2 module has a receiver function for MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2). The related D-PHY module supports MIPI Alliance Specification Version 2.1 for D-PHY Specification.

17.4 MIPI PHY

The MIPI interface is composed of a Clock differential pair and one or more Data differential pairs. Each differential pair is referred to as a Lane. (One clock Lane, one or more data Lanes.) The Renesas RA8x2 MIPI interface includes one clock lane and 2 data lanes.

The MIPI D-PHY specification provides the following key characteristics for MIPI signals.

- The reference characteristic impedance level is 100Ω ±20% differential, 50Ω ±20% single-ended per line, and 25Ω common-mode for both lines together.
- The signal lines within a lane should be length matched.
- The lanes within the interface should be length matched.

The following routing guidelines should also be considered.

- Signals should be routed with the minimum length possible.
- Signals should be referenced to a solid ground or power plane over the entire routed length.
- Layer transitions should be avoided if possible. If layer transitions must be used, ground stitching vias should be added immediately next to the signal layer transition vias.
- Lanes should be routed with a minimum spacing between lanes of 3 times the differential pair spacing.

Item		Specifications
D-PHY	Number of lanes	Up to 2 lanes
	Maximum rate	720 Mbps/lane

MIPI subsystem I/O pins.

Pin name	I/O	Function
MIPI_CL_P	DSI: Output CSI: Input	DSI/CSI Clock Lane positive pin
MIPI_CL_N	DSI: Output CSI: Input	DSI/CSI Clock Lane negative pin

Pin name	I/O	Function
MIPI_DL0_P	DSI: I/O CSI: Input	DSI/CSI Data Lane 0 positive pin
MIPI_DL0_N	DSI: I/O CSI: Input	DSI/CSI Data Lane 0 negative pin
MIPI_DL1_P	DSI: Output CSI: Input	DSI/CSI Data Lane 1 positive pin
MIPI_DL1_N	DSI: Output CSI: Input	DSI/CSI Data Lane 1 negative pin
DSI_TE	DSI: Input	DSI Tearing Effect pin
AVCC_MIPI	Power	D-PHY Analog Power Connect this pin to VSS_MIPI by a 0.1-µF capacitor. The capacitor should be placed close to the pin.
VCC18_MIPI	Power	D-PHY I/O Power Connect this pin to VSS_MIPI by a 0.1-µF capacitor. The capacitor should be placed close to the pin.
VSS_MIPI	Power	D-PHY GND

Refer to section MIPI PHY in each RA8x2 MCU hardware manual for more details.

18. ΔΣ Interface (DSMIF)

RA8T2 family has the ΔΣ interfaces (DSMIF). It can be connected with up to three external ΔΣ modulators. It is capable of filtering delta-sigma modulated 1-bit digital input data and converting it into 16-bit digital data.

Figure 50 shows a block diagram of DSMIF.

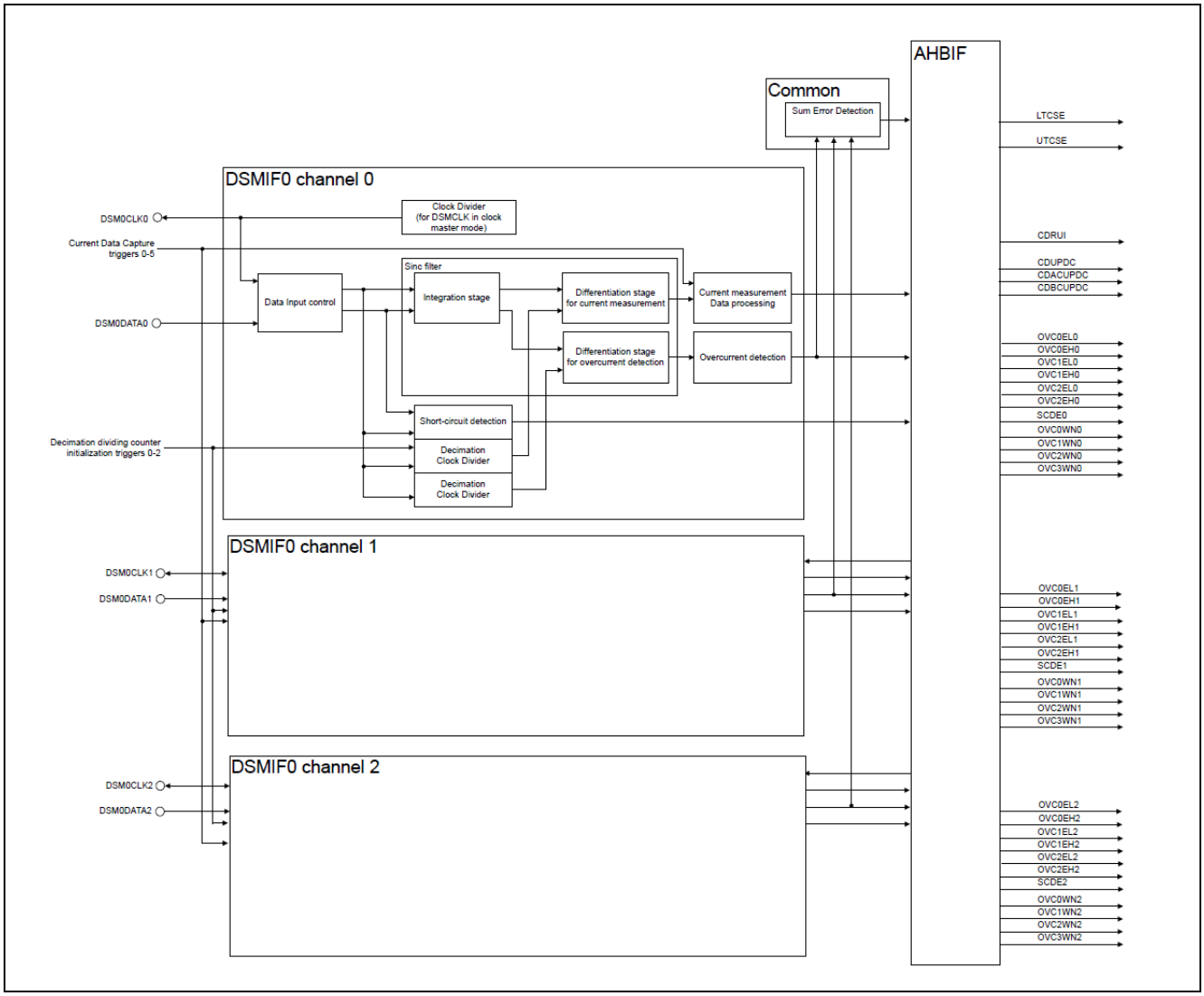


Figure 50. DSMIF0 Block Diagram

The Delta-Sigma Modulator interface (DSMIF) functions similarly to a normal ADC, however, instead of sampling an analog signal directly, DSMIF samples a PDM signal that is generated by an externally connected 1-bit Delta-Sigma Modulator.

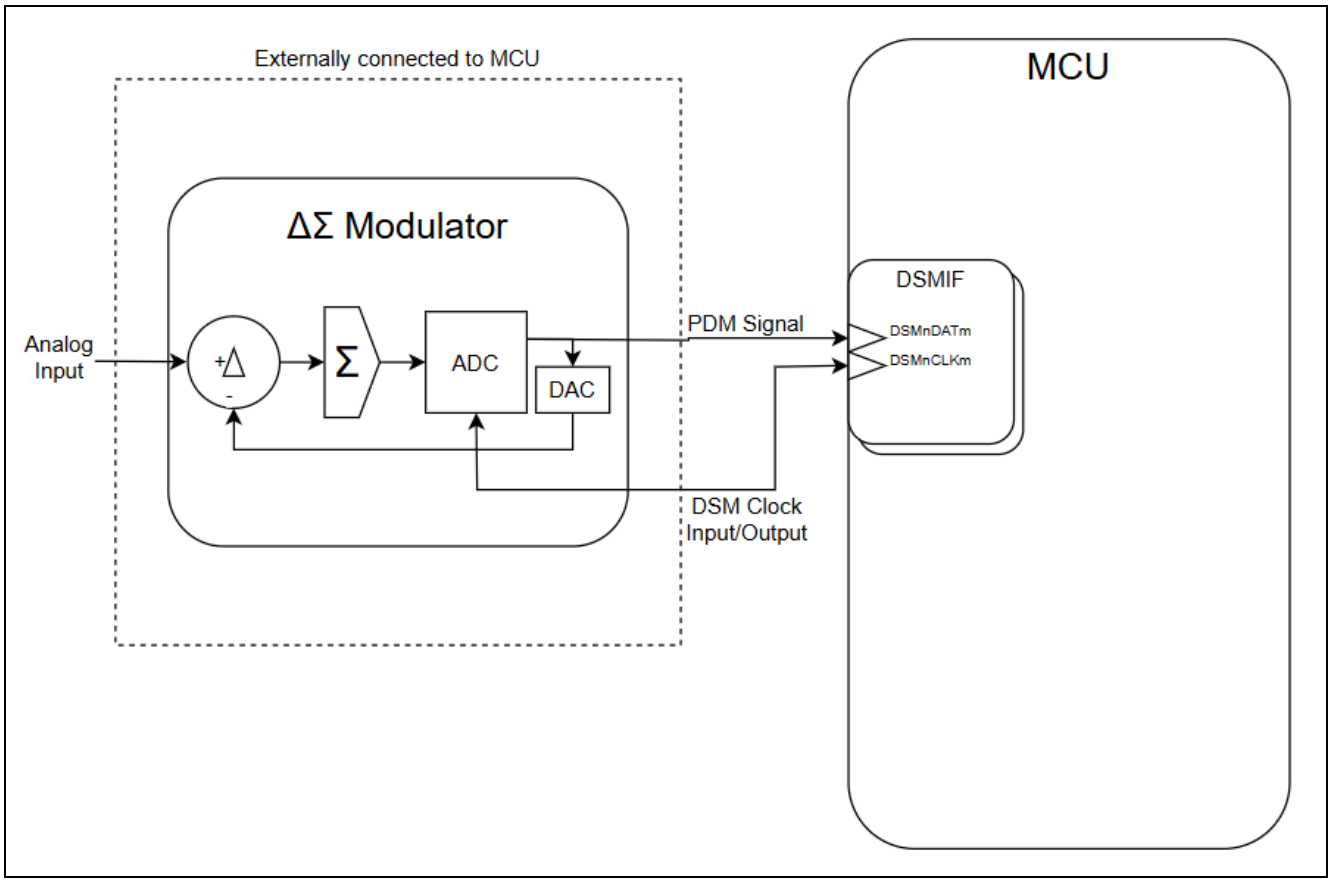


Figure 51. An Example of DSMIF Use

Renesas FSP offers an easy method for configuring the DSMIF module along with its channels, as Shown in Figure 52 and Figure 53.

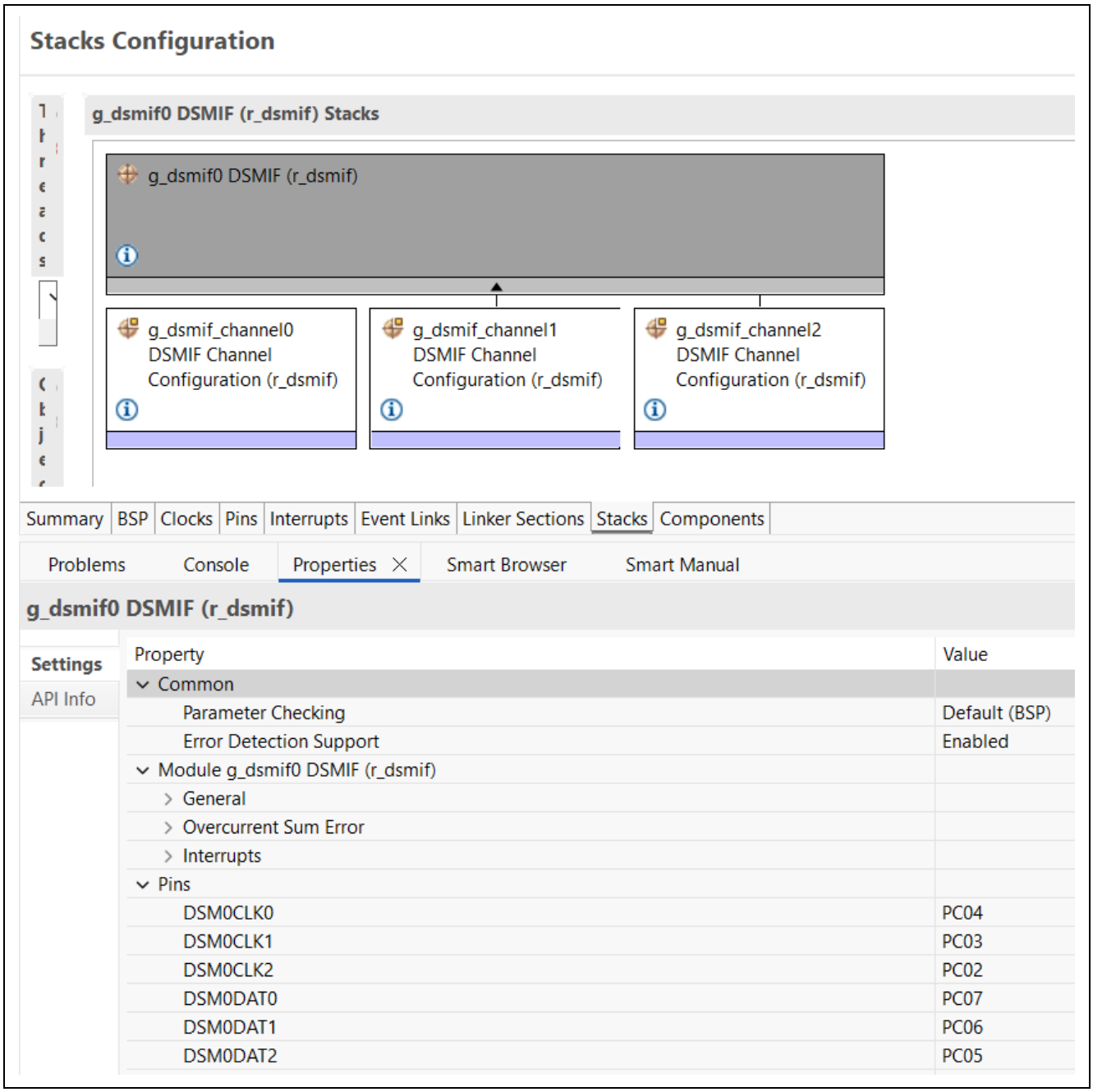


Figure 52. An Example of DSMIF Module Configuration

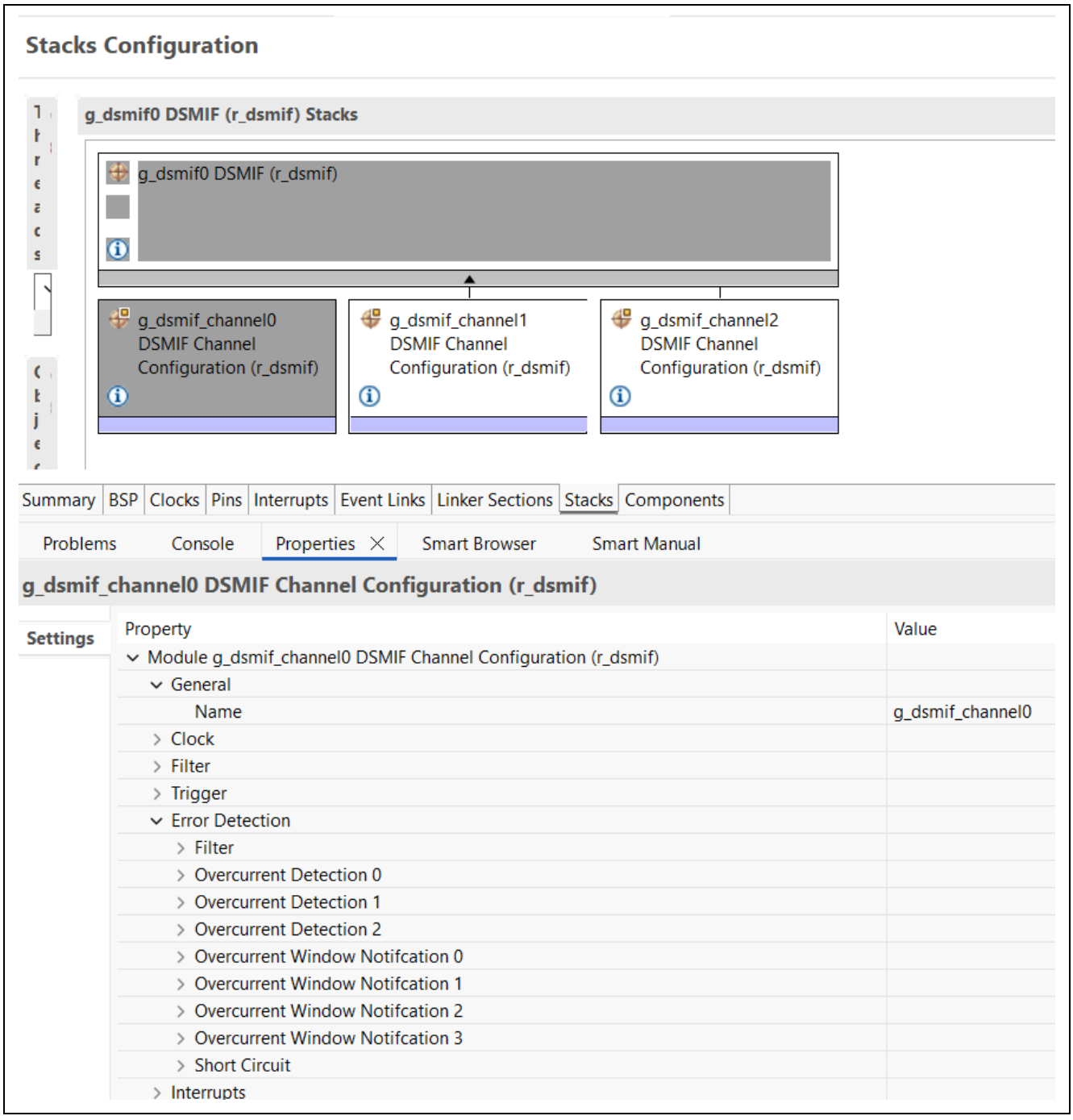


Figure 53. An Example of DSMIF Channel Configuration

When the frequency of the core clock and bus clock is changed, change it after stopping the DSMIF according to Stop flow. It is prohibited to change the clock frequency while DSMIF is operating.

Before transition to software standby mode, stop operation according to Stop Flow. After return from software standby mode, start operation according to Start Flow. Register setting before transition is retained. Therefore, you can omit the setting that is not needed to be changed.

19. Layer 3 Ethernet Switch Module (ESWM)

The Layer 3 Ethernet Switch Module (ESWM) consists of an Ethernet switch with higher-level routing capability and multi-protocol interface support. It allows autonomous frame routing within the same and between different network interface protocols (for now only Ethernet) for optimized gateway applications.

The PHY interface includes 2 ports with multiple switch functions, including Hub, Layer 2 Switch, VLAN Aware Layer 2 Switch, and Layer 3 Switch.

Figure 54 shows the ESWM block diagram.

Item	Function		Specification
Basic function	PHY interface	Number or port	2 ports
		Type	MII, RMII, GMII, RGMII
		Speed	10 Mbps,100 Mbps, 1Gbps
		Mode	Full Duplex
		Management IF(MDIO)	Yes
		Energy Efficient Ethernet (low power idle)	Yes
		Magic Packet™	Yes
	Switch function	Layer 3 support	Yes
		Cut through	Yes
		VLAN	Yes
		Jumbo frame support	Max 64 KB
		MAC table	2048
		VLAN table	4096
		Stream (perfect filter, layer 3, layer 2) table	256
		Layer 2/Layer 3 update rule table	256
	gPTP	Number of channels	2
	PTP pulse generator	Number of channels	4
Ethernet AVB	IEEE802.1BA	The AVB profile	Yes
	IEEE1722-2011	Media transport protocol for audio and video	Yes
	IEEE 802.1AS-2011	gPTP (a profile of IEEE1588 for AVB systems)	Yes
	IEEE 802.1Qav-2009	Credit based shaper for bandwidth reservation	Yes
Ethernet TSN	IEEE802.1AS-Rev	Adds redundancy and enhancements to gPTP	Yes
	IEEE 802.1Qbv-2015	Time aware shaper	Yes
	IEEE 802.1Qbu-2016 (802.3br)	Frame preemption	Yes
	IEEE 802.1Qch-2017	Cyclic queuing and forwarding	Yes
	IEEE 802.1Qcr	Asynchronous traffic shaping	Yes
	IEEE 802.1Qci-2017	Stream based filtering and policing	Yes (without gate filter)
	IEEE 802.1CB-2017	Frame replication and elimination for reliability	Yes
TrustZone Filter	Security attribution and privilege attribution can be set		

Note: Please contact a Renesas Electronics sales office for network support.

Figure 54. ESWM specifications

Figure 55 shows the ESWM block diagram, Table 27 and Table 28 shows ESWM functional blocks.

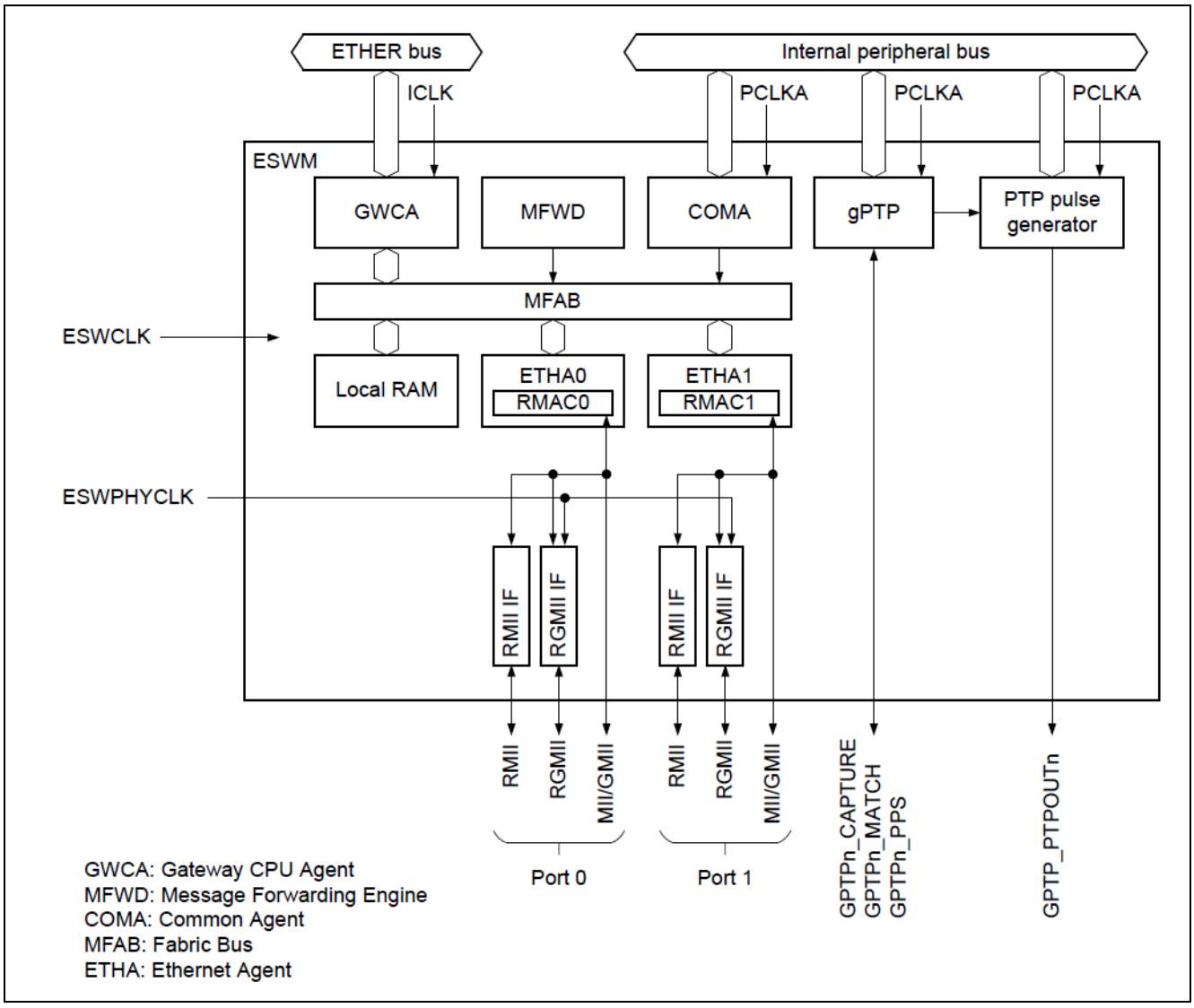


Figure 55. ESWM Block Diagram

Table 27. ESWM functional blocks (1 of 2)

Block name	Function
MFWD	Ethernet Message Forwarding Engine (MFWD) is part of ESWM and aims to filter, forward, and route the switch frames. MFWD snoops the frame information from fabric to redistribute them to the agent after filtering, forwarding, and routing mechanisms.
MFAB	Fabric Bus (MFAB) is an internal bus used to exchange data between Ethernet Agent [ETHA], Ethernet Common Agent [COMA], Ethernet CPU Agent [GWCA], local RAM, TAG RAM, pointer RAM, and Ethernet Message Forwarding Engine [MFWD].
ETHA	Ethernet Agent (ETHA) consists of an agent interface module to allow communication within ESWM. ETHA handles data exchange between ESWM and an Ethernet PHY.
RMAC	Ethernet MAC (RMAC) is an Ethernet controller that conforms to the definition of the MAC (Media Access Control) layer for Ethernet in the IEEE 802.3 standard. By connecting with a physical-layer LSI chip (PHY-LSI) that complies with the standard, RMAC can transmit and receive Ethernet (IEEE 802.3) frames. RMAC has a single MAC layer interface.
GWCA	Ethernet CPU Agent (GWCA) is a CPU interface that consists of an agent interface module allowing communication within ESWM. GWCA handles data exchange between ESWM and the GWCPU subsystem. Software is required for the configuration of Gateway AXI bus interface.
COMA	Ethernet Common Agent (COMA) is an agent of ESWM which aims at gathering common functionalities for other agents. COMA handles the APB to SFR bus conversion, the buffer pointer release for rejected frames and the pointer handling for local RAM. Software is required for the configuration of Ethernet Common Agent.

Table 28. ESWM functional blocks (2 of 2)

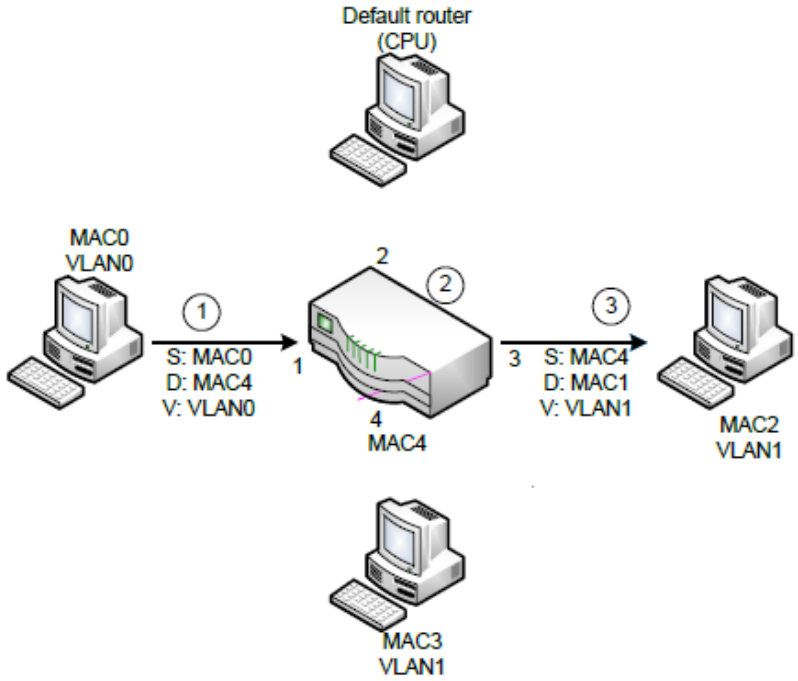
Block name	Function
gPTP	Ethernet Generic PTP timer (GPTP) enables accurate synchronization of the clock in the control system.
plsotim	The pulse output timer module (plsotim), a submodule of gPTP, produces a pulsed output according to the base period of a gPTP timer.

Refer Gateway CPU Agent, Message Forwarding Engine, Common Agent, Ethernet Agent in the MCU hardware user manual for more details.

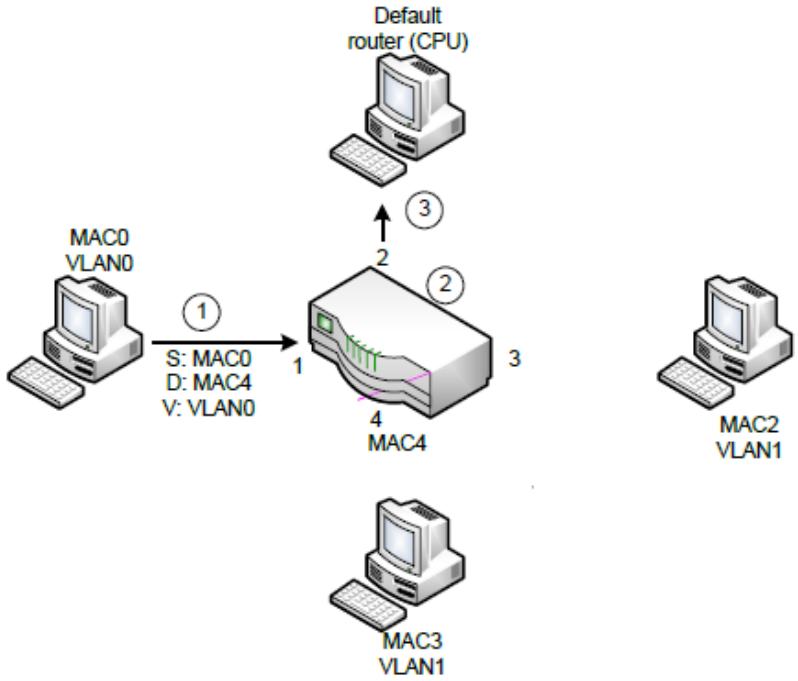
19.1 Layer 3 Switch

A layer 3 switch is an ethernet network interconnection system (OSI) which works on layer 3 using IP addresses. Because ESWM can only perform perfect matches by stream forwarding, when developing a layer 3 switch, it is recommended to use perfect filters [FWD].

Figure 56 shows the layer 3 switch operation.



1	The switch receives a packet from port 1 with MAC0 as source MAC address, MAC4 as destination MAC address and VLAN0 as VLAN.
2	The packet IP address matches the cascade filter and should be routed to port 3.
3	The switch updates MAC destination to MAC1, MAC source to MAC4 and VLAN to VLAN1 and sends the frame to port 3.



1	The switch receives a packet from port 1 with MAC0 as source MAC address, MAC4 as destination MAC address and VLAN0 as VLAN.
2	The packet IP address is unknown so routing cannot happen.
3	The switch forwards the frame to the default router (CPU) without update for routing.

Figure 56. Example of Layer 3 Switch Operation

20. Single Core Version MCU

The RA8x2 family comprises both Dual-Core and Single-Core series. Table 29 outlines the key differences between the two MCU versions within this family. While there are differences in the number of CPU cores and peripherals, the guidelines in this document apply to both MCU series.

Table 29. Example of Main Differences between RA8P1 Dual-Core and Single-Core MCUs for 2 MB Products

Item	Dual	Single
Core	CM85 + CM33	CM85
Startup core	Selectable by Boot firmware	Not selectable
Communication between cores	Yes (IPC)	No
SRAM size	1,664KB	1,792KB (1,664KB + 128KB) *1
INT resource	Core distribution possible	Core distribution impossible
SSTBY move condition	Both cores need Deep sleep or Inactive	No restriction

Note 1. 128 KB of TCM memory in CPU1 in the Dual-Core MCU version is available for use as additional SRAM in the Single-Core MCU version.

21. General Layout Practices

21.1 Digital Domain vs. Analog Domain

Renesas RA8x2 Microcontroller devices have three primary types of pin functions: Power, Digital, and Analog.

Generally, power pins are dedicated for voltage and reference input and do not have multiple functions. Power pins are typically dedicated to specific portions, or domains, within the MCU. For example, the main supply voltage for the MCU will provide power to the digital core, many of the digital peripheral functions and many of the digital I/O pins. The digital domain can be defined as the digital circuitry, digital I/O pins, and the related power pins. Power pins which are designated for analog functions (such as AVCC0 and the associated AVSS0) supply specific analog circuitry within the MCU, which is separate from the digital domain circuitry. The analog domain can be defined as the analog circuitry, analog I/O pins, and the related power pins.

Digital signals are typically repetitive, switched patterns that are associated with periodic clocks. The transitions on digital signals tend to be relatively sharp edges, with stable levels of high or low between the transitions. Each signal must be stable at an acceptable voltage level, referred to as a logic state, within a specified timeframe. The state of the signal is typically sampled at predetermined clock intervals, using the edge transition of a clock to evaluate the associated data signals. Small variations in the voltage level of digital signals are typically acceptable, as long as the level remains within a specified range. However, large external influences on digital signals can have an acute influence on a digital signal, which can result in an incorrect logic state at the moment when the data is sampled.

Analog signals are usually quite different. Analog signals may be periodic, but the evaluation of an analog signal is typically a measurement of voltage over a range instead of logic state. The voltage level of an analog signal is sampled based on a specific trigger event, and the resulting measurement is processed using the analog circuitry in the MCU. The accuracy of an analog measurement is directly related to the accuracy of the sampled voltage level. Any unwanted external influence which may change the voltage level of an analog input signal, even slightly, can influence the accuracy of the measurement.

Due to the highly multiplexed nature of the I/O pins on Renesas RA8x2 MCU devices, many I/O pins can be used for either Analog or Digital functions. This can result in situations where digital and analog functions may overlap and result in data errors.

To minimize potential problems between digital and analog signal domains, consider the following guidelines.

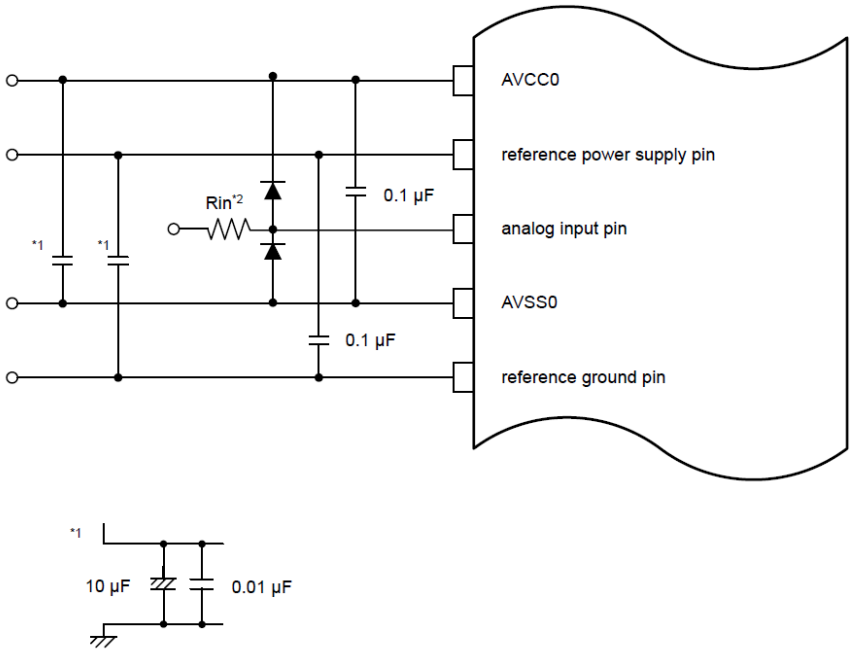
- When assigning I/O pin functions, select pin functions such that analog pins and digital pins are physically separated as much as possible.
- Each analog signal should be separated from all other signals as much as possible.
- PCB routing should isolate each analog signal as much as possible. Avoid routing any other signals, either analog or digital, in the same area.
- Ensure that analog supply voltages and analog reference voltages include appropriate AC filters. This may be in the form of recommended capacitors located near the MCU voltage pin, or appropriate inductive filters. The goal is to provide voltage supply and reference voltage with little or no voltage ripple.
- When using dedicated power layers in a PCB design, avoid routing digital signals in the areas of analog voltages, and avoid routing analog signals in the areas of digital voltages.

For highly sensitive applications, it is highly recommended to evaluate the specific design using simulation tools to understand the effect that circuit design has on the performance. For example, this may include applications such as precision sensor designs, or very high-speed digital bus interfaces. Refer to the “Electrical Characteristics” chapter in the Hardware User’s Manual for the specific requirements for each peripheral function.

21.2 Precautions for Analog Signals

All RA MCUs includes peripherals to process analog signals. These include Analog to Digital Converters (ADC), Digital to Analog Converters (DAC), and High-Speed Analog Comparators (ACMPHS).

Analog input pins can be destroyed by abnormal input voltage, such as an excessive surge. To protect the analog input pins, it is recommended to include a protection circuit and capacitors to the analog voltage supply pins, analog reference pins, and analog input pins.



Note 1. The values shown here are reference values.
Note 2. Rin: Signal source impedance

Figure 57. Analog Input Pin Protection

Refer to the A/D Converter chapter and the High-Speed Analog Comparator chapter of the Hardware User's Manual for details of these analog input peripherals. Pay close attention to the Usage Notes sections for specific details of using these inputs.

Each analog voltage supply pin and analog reference power supply pin have recommended capacitors that should be placed close to the voltage supply or reference power pins. Follow the recommendations in the Hardware User's Manual for the specific device to ensure correct operation of the analog peripheral.

An additional resource for guidelines on analog circuit design using Renesas RA microcontroller is the Application Note "Best Practices for Analog PCB Layout Using RA2A1 MCU" (document number R01AN5287EU0100) available from Renesas. While the analog peripherals between RA2A1 and RA8x2 Group microcontrollers may be different, the design concepts and recommendations apply to both groups.

21.3 High Speed Signal Design Considerations

As clock speeds for digital signals increase, the influence of external stimuli on those signals can become more significant. Some peripheral functions can be classified as "High Speed" digital signals. Additional design considerations should be made for high speed digital signals.

Crosstalk is a condition where transitions on one signal have an inductive influence on another nearby signal. When this crosstalk effect is strong enough, the first signal may cause errors on the second signal. To reduce the effects of crosstalk, use the following general PCB routing guidelines.

- Provide sufficient space between routed signals on the same routing layer. Generally, keep a minimum of one trace width space between signals of the same digital group, and a minimum of 3-5 trace widths space between signals of different digital groups.
- Provide extra space between clock signals and data signals on the same routing layer. Generally, keep a minimum of 3-5 trace widths space between clocks and any other digital signals.
- Avoid parallel routing of digital signals on any adjacent routing layers. If signals must be routed on adjacent signals layers, try to use only orthogonal crossings wherever possible.

If possible, separate PCB signal layers using power or ground layers between signal layers. The solid copper of the power or ground layer can act as a "shield" for the digital signals.

Peripherals on RA8x2 microcontrollers that should be treated with high-speed design considerations include Octal SPI, LCD, I3C, Gigabit Ethernet, MIPI, CANFD, SPI, and CEU. These peripherals include clocks that can be classified as high-speed. In addition, there are peripheral functions that may not be classified as high-speed but should be considered for similar design practices. These include the External Bus, when used for SDRAM, SDHI and USB.

Each standardized interface will have specific requirements. To ensure that the PCB is designed to avoid signal crosstalk problems, it is strongly suggested to refer to the relevant standards for each interface in the design.

21.4 Signal Group Selections

Some pin names have an added _A, _B, or _C suffix to indicate signal groups. When assigning certain peripheral functions, such as IIC, SPI, SSIE, ESWM, and SDHI, select the functional pins having the same suffix. In some cases, the AC timing characteristics shown in the "Electrical Characteristics" chapter of the Hardware User's Manual are measured for each signal group. If the signal groups are mixed, the peripheral is not guaranteed to function, and the stated AC timing characteristics may not apply.

If the pin names for a peripheral function to not have a signal group suffix, it is safe to select the most convenient pin assignment for each function signal.

Refer to the sections "Peripheral Select Settings for each Product" and "Notes on the PmnPFS Register Setting" in the "I/O Ports" chapter of the Hardware User's Manual.

22. References

The following documents were used in creating this Quick Design Guide:

- RA8P1 Group User's Manual: Hardware, document No. R01UH1064
- RA8D2 Group User's Manual: Hardware, document No. R01UH1065
- RA8M2 Group User's Manual: Hardware, document No. R01UH1066
- RA8T2 Group User's Manual: Hardware, document No. R01UH1067
- RA8P1 Memory Architecture App Note, document No. R01AN7880
- Developing with Dual-Core RA8P1 MCU, document No. R01AN7881
- RA/RX Family eMRAM Magnetic Immunity Guideline, document No. R01AN7226
- Application Design using RA8 Series MCU Decrypt on the Fly for OSPI, document No. R11AN0773
- Arm Cortex®-M85 Processor Technical Reference Manual, document No. 101924, available from Arm
- E2 Emulator, E2 Emulator Lite Additional Document for User's Manual, Notes on Connection of RA Devices, document No. R20UT4686

Website and Support

Visit the following vanity URLs to learn about key elements of the RA family, download components and related documentation, and get support.

RA Product Information	www.renesas.com/ra
RA Product Support Forum	www.renesas.com/ra/forum
RA Flexible Software Package	www.renesas.com/FSP
Renesas Support	www.renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun.30.25	—	First release document
1.10	Oct.21.25	—	Update for RA8D2, RA8M2, RA8T2
		—	

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/.