Renesas RA Family

RA8 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 also discusses some design decisions from an application point of view.

Target Device

RA8 MCU Series

Contents

1. Power Supplies

The RA family has digital power supplies and analog power supplies. The power supplies use the following pins.

Table 1. Digital Power Supplies

Symbol	Function Name	Description
VCC, VCC2	Power supply	Power supply pin. Connect to the system power supply. Connect each pin to VSS via a 0.1 µF capacitor placed close to the VCC/VCC2 pin.
VCC_DCDC	Switching regulator (DCDC) Power supply	Switching regulator power supply input pin. Connect all VCC_DCDC pins together and connect to VSS_DCDC through a single 100 nF capacitor and a single 22 µF capacitor. Place the 100 nF capacitor closer to the VCC_DCDC pin.
VLO	Switching regulator (DCDC) Power supply	Switching regulator I/O pin. Connect to VCL through a 2.2 uH inductor. Connect the output of the inductor to VSS_DCDC through a 47 µF capacitor close to the inductor.
VSS	Ground	Ground pin. Connect it to the system power supply (0V).
VCL	Power supply	Connect each pin to VSS via a 0.22 µF smoothing capacitor used to stabilize the internal power supply. Place this capacitor close to the VCL pin.
VBATT	Backup power	Backup power pin. Supplies power to RTC, sub-clock oscillator, backup register, tapper detection, and VBATT_R voltage drop detection in the absence of VCC. When VBATT pin is not used, connect to VCC.
VSS_DCDC	Switching regulator (DCDC) Ground	Ground pin. Connect it to the system power supply (0 V) at a single point.
VCC_USB	USB FS power supply	USB Full-speed power supply pin. Connect this pin to VCC. Connect this pin to VSS_USB via a 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 at a single point.
VCC_USBHS[1]	USB HS power supply	USB High-speed power supply pin. Connect this pin to VCC. Connect this pin to VSS1_USBHS and VSS2_USBHS via a 0.01 µF 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 at a single point.
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 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.
VREFH0	12-bit ADC high reference voltage	Analog reference voltage supply pin for the ADC12 (unit 0). Connect this pin to VCC when not using the ADC12 (unit 0).
VREFL0	12-bit ADC low reference voltage	Analog reference ground pin for the ADC12. Connect this pin to VSS when not using the ADC12 (unit 0).
VREFH	12-bit ADC & DAC analog supply	Analog reference voltage supply pin for the ADC12 (unit 1) and D/A Converter. Connect this pin to VCC when not using the ADC12 (unit 1) and D/A Converter.
VREFL	12-bit ADC & DAC analog ground	Analog reference ground pin for the ADC12 and D/A Converter. Connect this pin to VSS when not using the ADC12 (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 to the 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	MIPI analog power supply	MIPI interface analog power supply. Connect this pin to the system power supply. Connect this pin to VSS through a 0.1 µF capacitor.

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

1.1 Dual VCC Power Domains

RA8 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.

The peripherals associated with VCC include both USB and SDRAM. When using either one of these peripherals, VCC should be connected to a nominal 3.3 V 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 and improved performance. 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 RA8 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 and VCC2 pins: Connect each pin to the relevant system power supply. Connect each pin to VSS 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.
- VCL pins: Connect each pin to VSS 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.

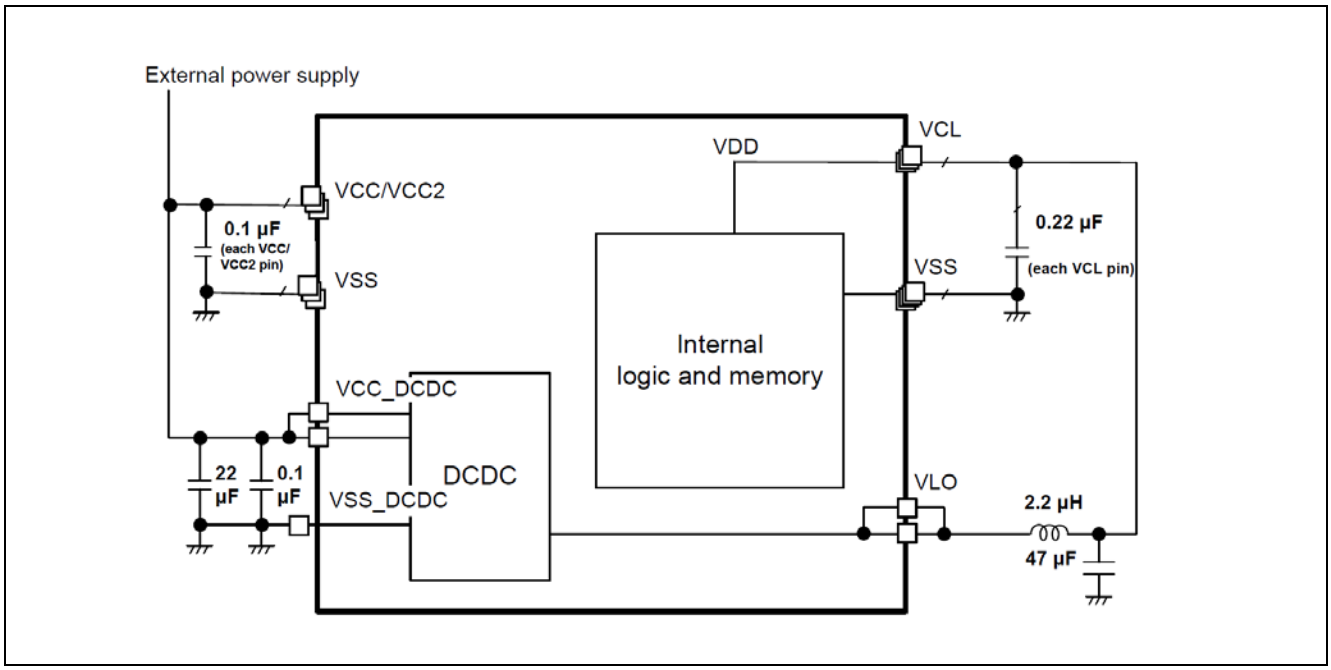


Figure 1. DCDC Mode Configuration

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. One of the following procedures must be followed:

- VCC and VCC2 must be raised to the minimum VCC voltage before VCL is raised. Refer to the Hardware User's Manual for the specific timing requirements for the selected device.
- Raise VCC and VCC2 voltage with RES pin low, and release RES pin after a delay when VCL voltage is raised. Refer to the Hardware User's Manual for the specific timing requirements for the selected device.

To implement External VDD mode:

- VCC and VCC2 pins: Connect each pin to the system power supply. Connect each pin to VSS 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.
- VCL pins: Connect each pin to the system power supply. Connect each pin to VSS 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. Low-speed mode, Software Standby mode, Deep Software Standby mode 1, 2, and 3, and Battery Backup Function are not supported when using External VDD mode.

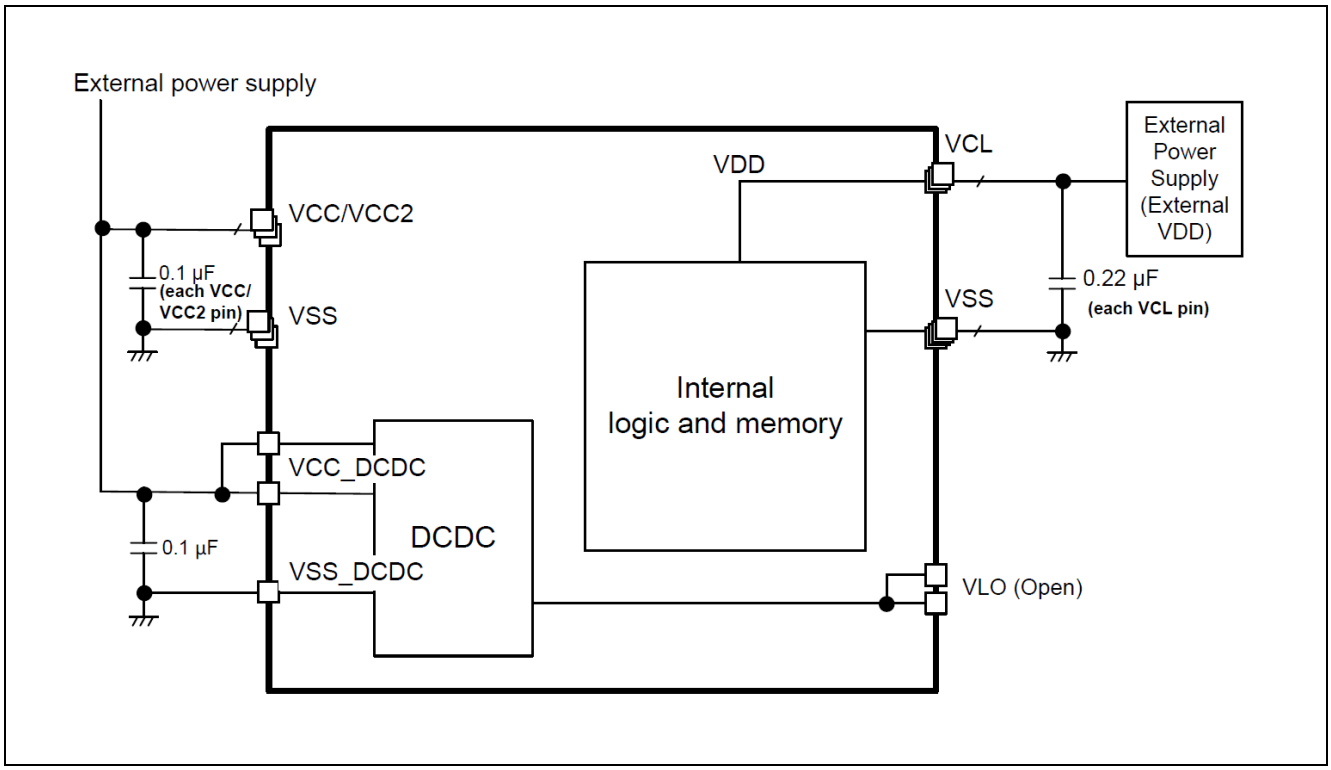


Figure 2. External VDD Mode Configuration

1.4 References

Further information regarding the power supply for the RA8 MCU Group can be found in the following documents:

R01UH0994EJ0100 RA8M1 Group, RA8M1 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 "12-Bit A/D Converter (ADC12)" and "12-Bit D/A Converter (DAC12)" for chapters in the respective Hardware User's Manuals for details.

Table 4. RA8 MCU Groups, 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
12-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 RA8, please reference section 7.1.2.

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, RA8 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.

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

2.1 SWD Interface

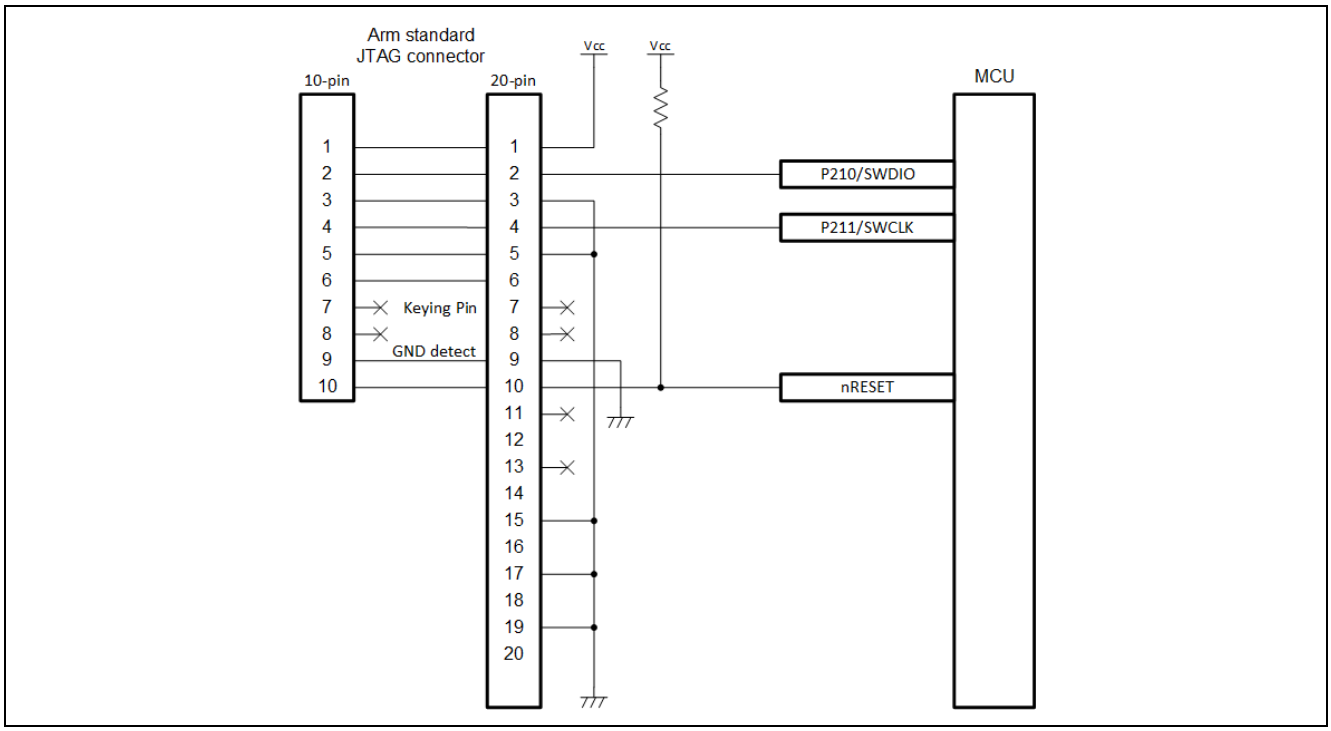


Figure 3. SWD Interface Connections

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

2.2 JTAG Interface

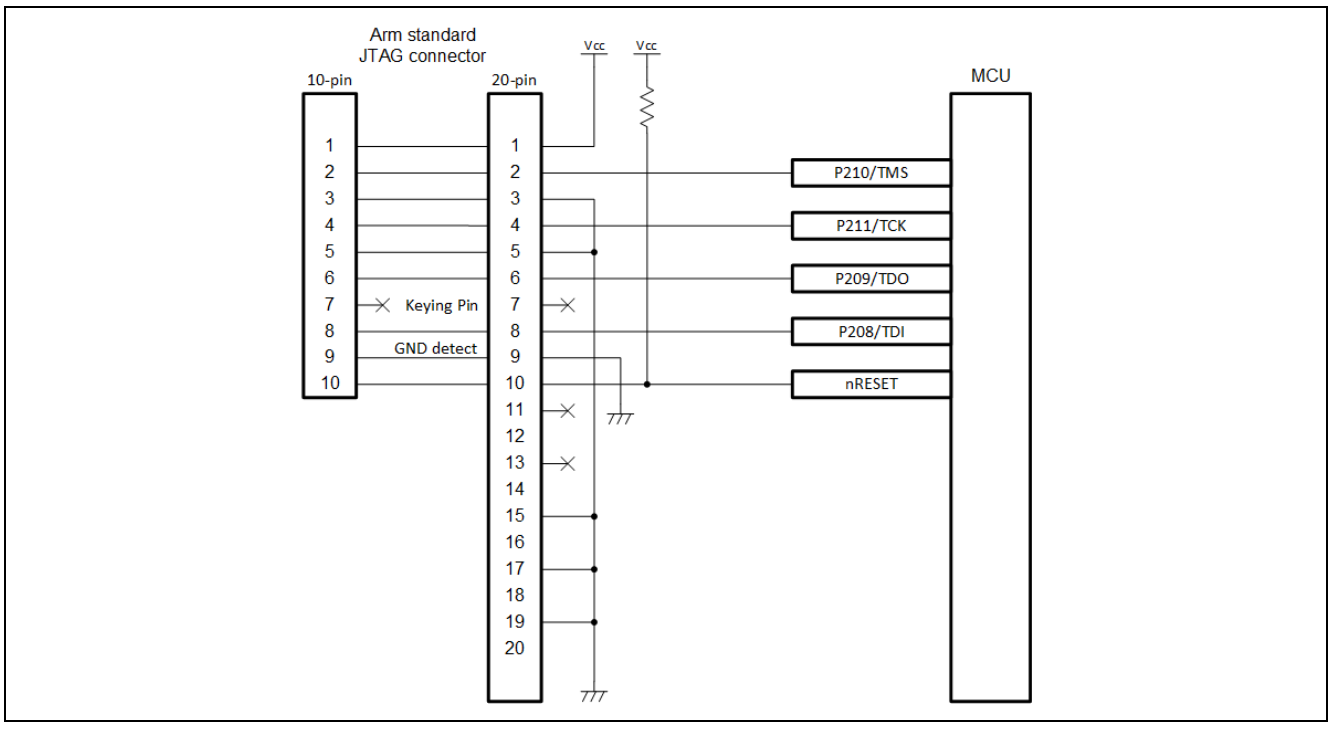


Figure 4. JTAG Interface Connections

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

2.3 Trace Data Interface

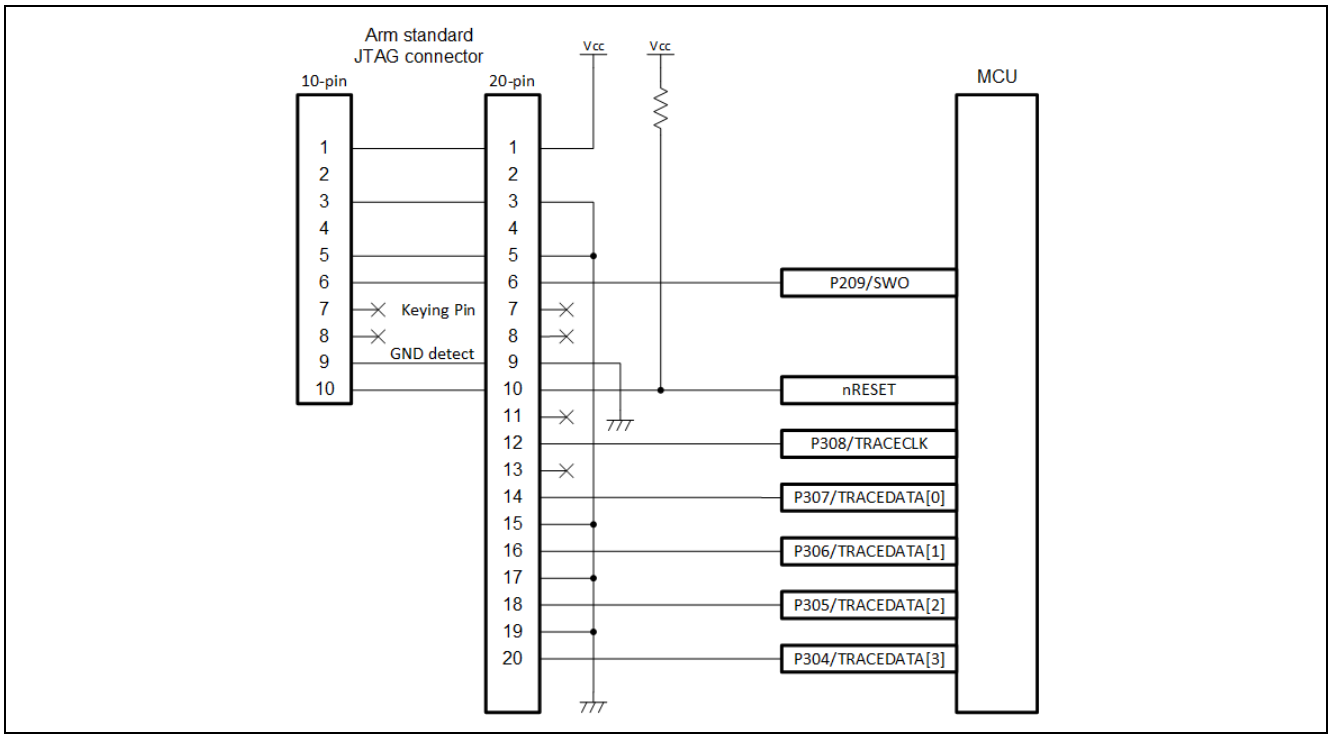


Figure 5. Trace Data Interface

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 Overview chapter of the Hardware User’s Manual for more information.

2.4 Using SCI Boot Mode over the Emulator Interface

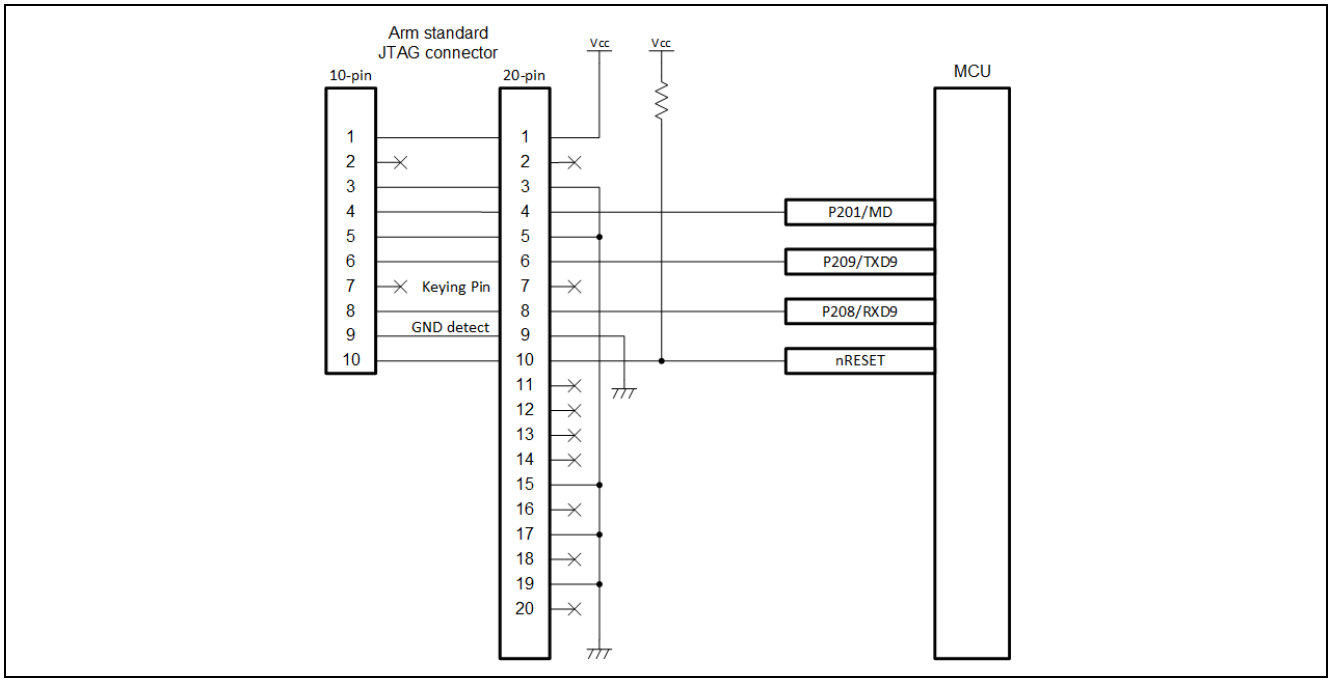


Figure 6. Serial Programming Interface using SCI Connections over Debug Connector

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 6.
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 RA8 devices, the IDAU 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

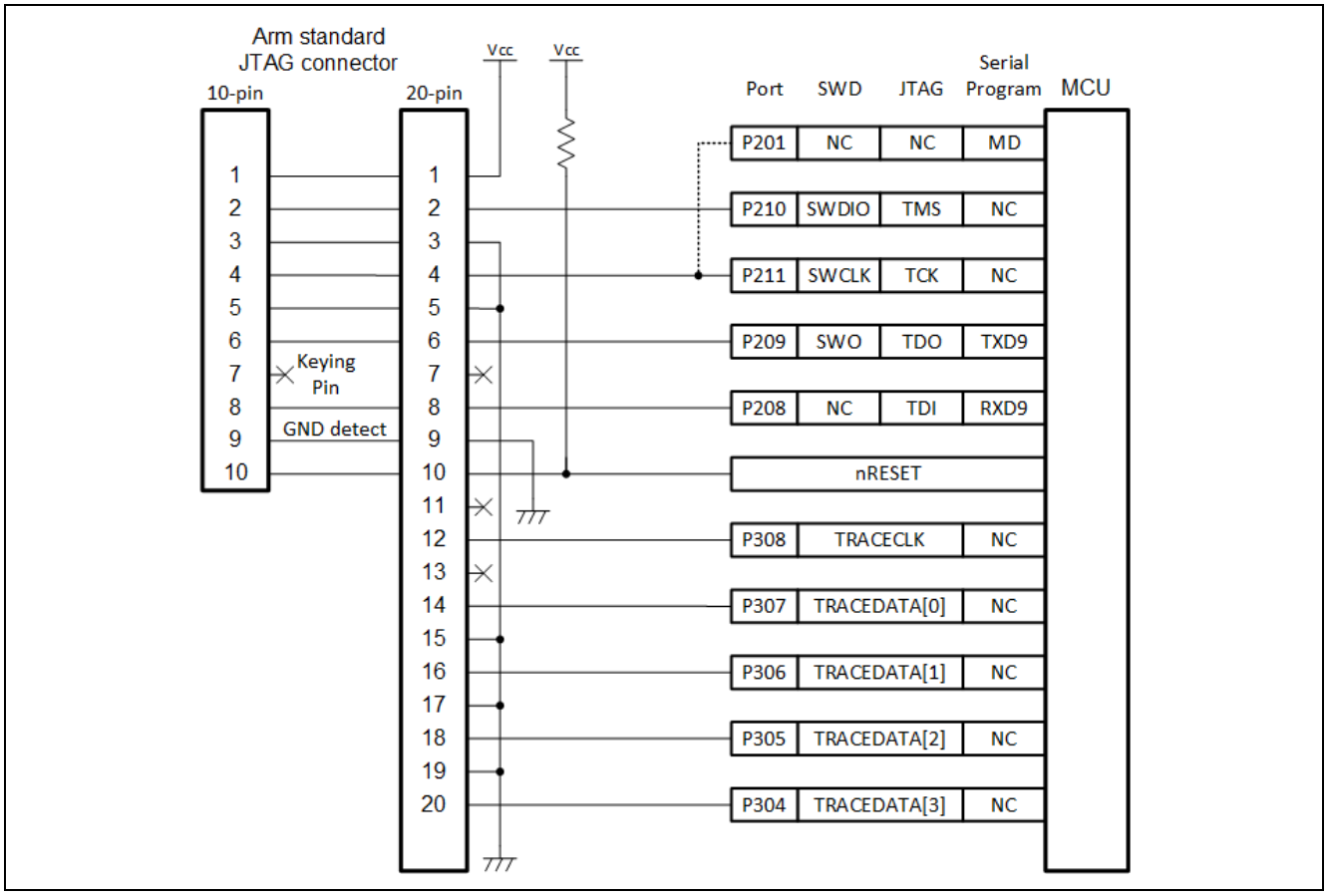


Figure 7. Multiple Emulator Interface Connections

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

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) and Serial Wire Output (SWO) provide trace output in RA8 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	P209
Function	TDATA3	TDATA2	TDATA1	TDATA0	TCLK	SWO

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.

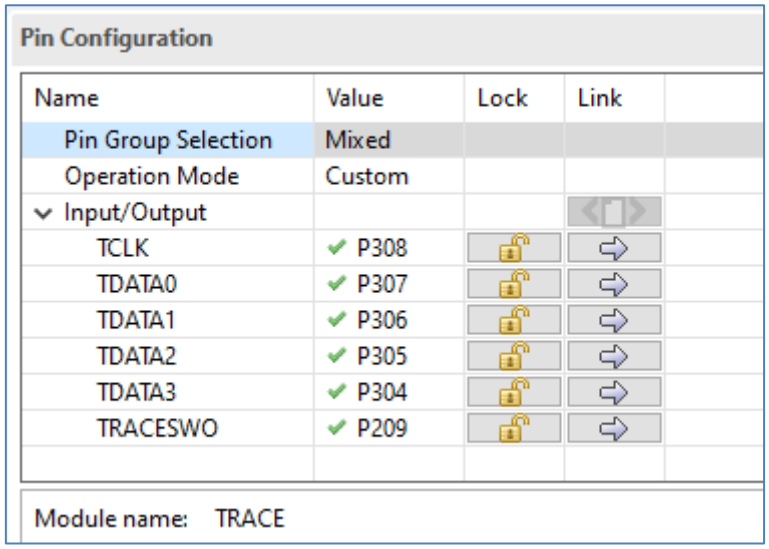


Figure 8. Enabling Trace Ports in Runtime Using FSP Configurator

3. MCU Operating Modes

The RA8 MCU series 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 Flash Memory	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.

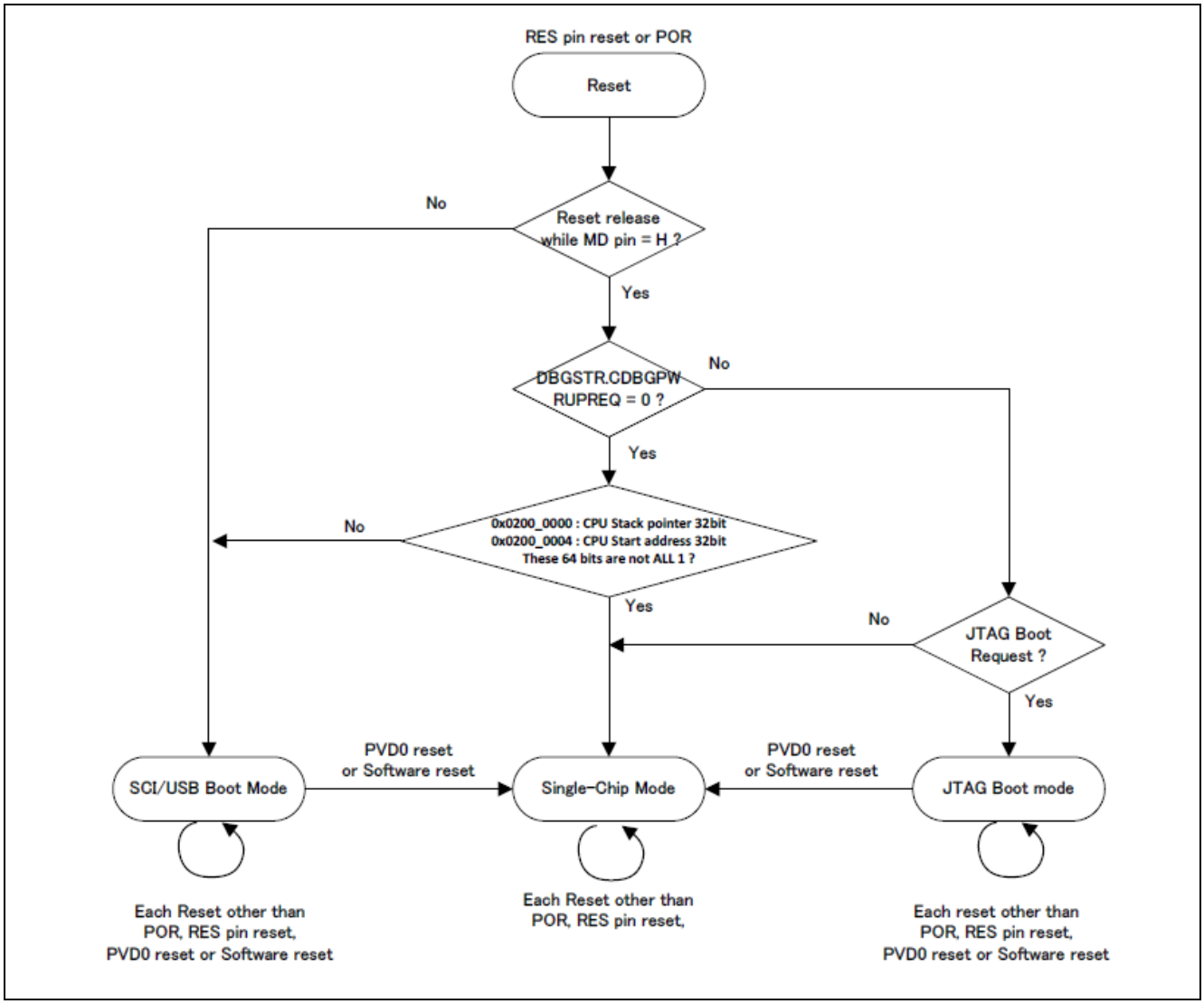


Figure 9. Mode Setting Pin level, Operating Modes and the relationship of mode transition

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.

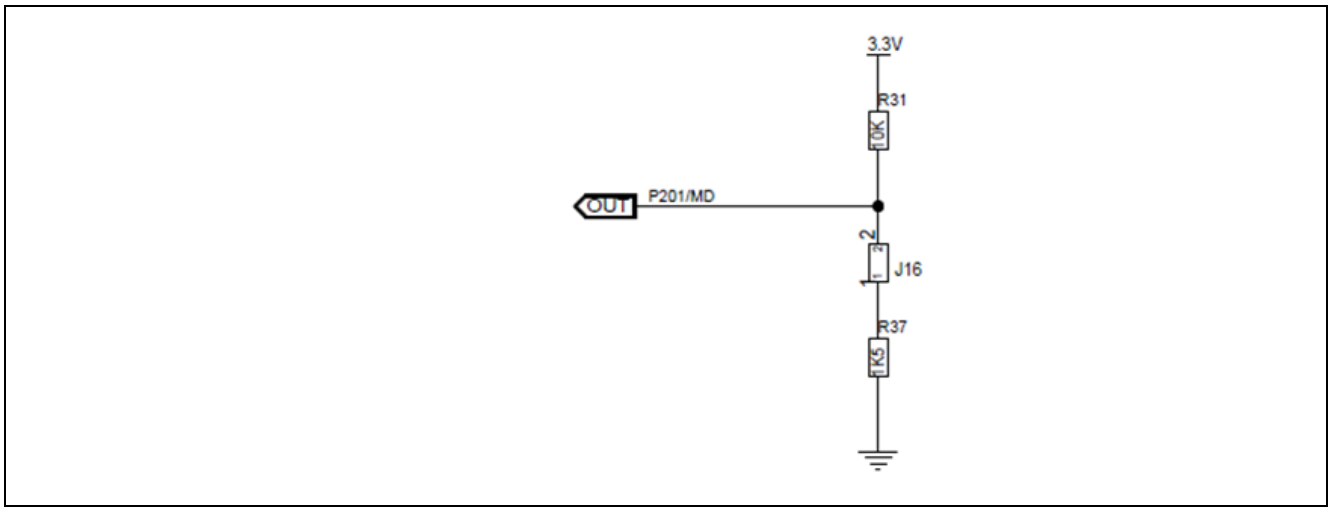


Figure 10. Typical Circuit for MCU Boot Mode Selection

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 and the program flash area of the flash memory. The available methods of setting are different for the two areas. Option setting memory may differ in size and layout for Cortex®-M85 based 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 code flash memory. Sometimes the registers may be located in a portion of the flash memory which is near reserved areas in the internal flash. It is possible that some customers may inadvertently store data in the registers of option-setting memory or write into the reserved area in internal flash. 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 internal flash by reviewing the map file or s-record files generated by the compiler upon creation of the binary. For instance, settings in the flash option registers can enable the Independent Watchdog Timer (IWDT) immediately after reset. If data stored in program ROM 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 code flash 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 the option setting memory which consists of the option function select registers on RA8M1, which is a Cortex®-M85 device.

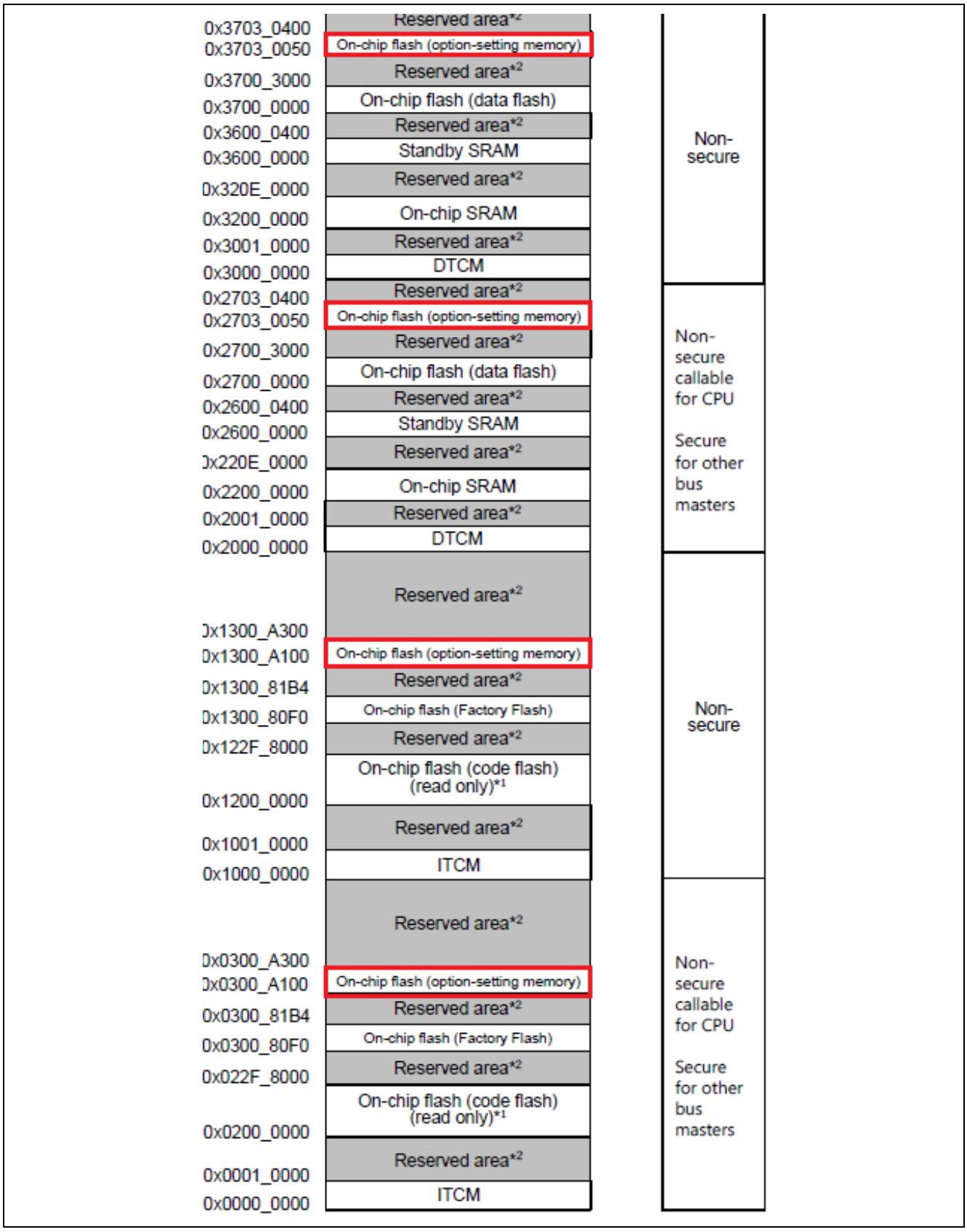


Figure 11. Option Function Select Registers Shown in Memory Map Example

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. 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. See Figure 12 on Memory Usage view.

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 (WDT) auto start
  - WDT timeout, frequency, windowing, interrupt type, and low power mode behavior
- OFS1 register
  - PVD0 startup settings after reset
  - HOCO startup settings after reset
  - Software Debug Control
  - Set ECC function of TCM and CACHE.
- OFS2 register
  - DCDC enable at reset.

Code Flash Memory Setting

- SAS register
  - Startup area selection.
- DUALSEL register
  - Code Flash memory mode upon reset.
- BANKSEL registers
  - Controls swapping of banks at the next reset
- BPS, PBPS registers
  - Turns off erasing and programming capability of selected blocks of code flash memory. Note that when PBPS are set to disable the flash erase and programming, it can never be reverted.

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 the code flash 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 13.
- SACCx register
  - Configure the locations in code flash 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
- HOEMRTPK 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.
- CFGDxLOCK register
  - Specify write protection for corresponding Lockable Configuration Data Areas in Data Flash. Note that when the protection is enabled, it can never be reverted.
- ARCLS register

— Controls Anti Rollback Counter Lock functionality. Note that 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

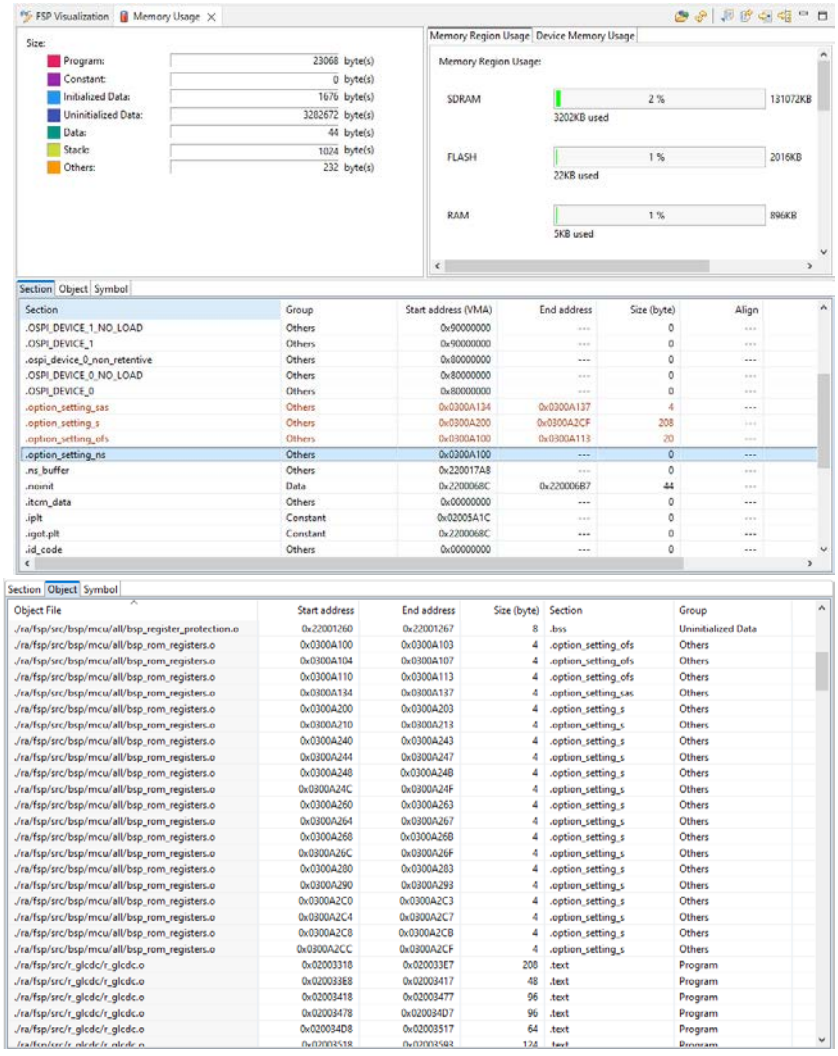
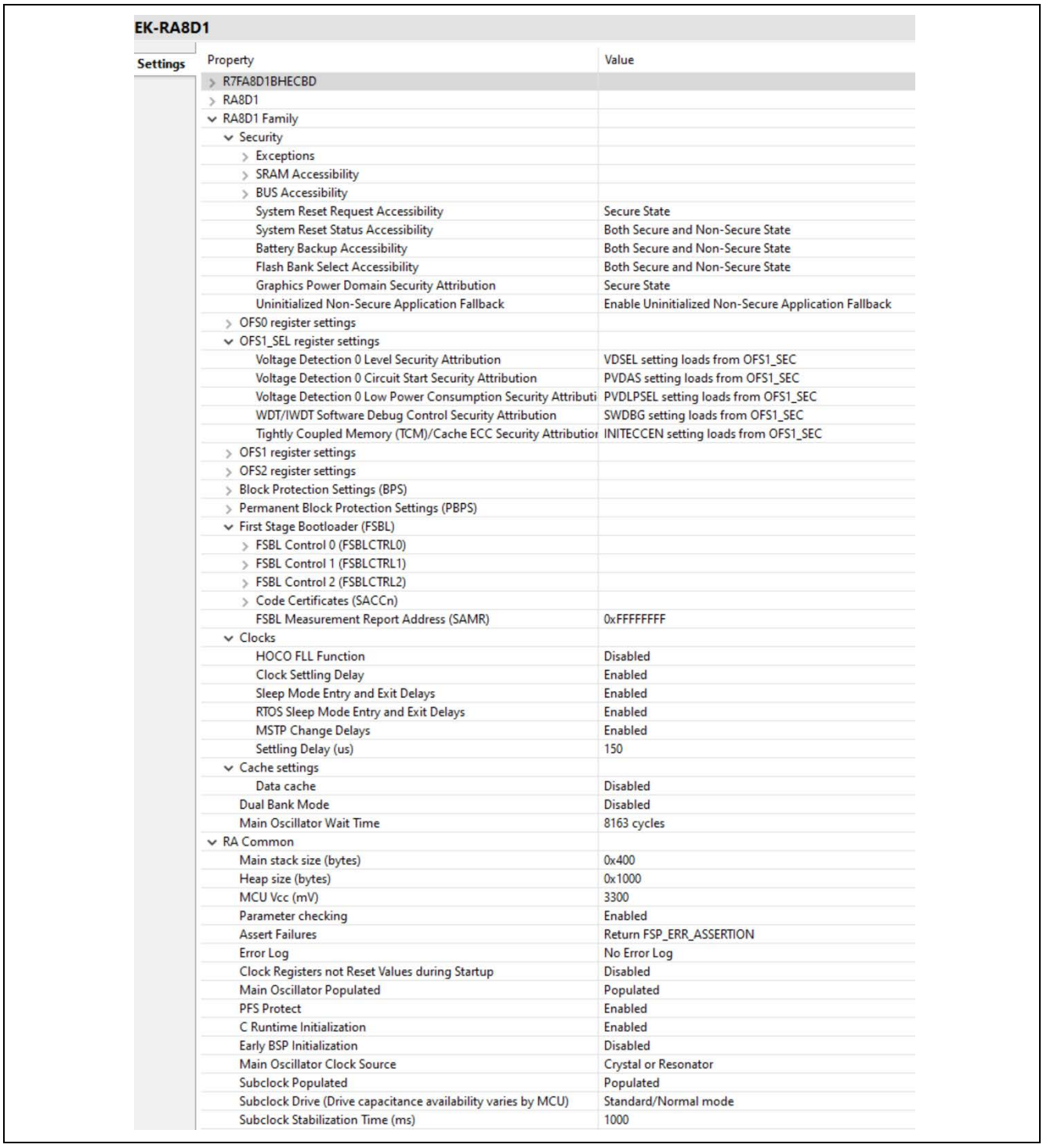


Figure 12. Using the Memory Usage View to Identify Option Setting Address Locations Programmed

You can also use the `objdump` command to check the settings of the option setting memory. Following is an example for dumping the option setting registers:

```
arm-none-eabi-objdump.exe -s --start-address=0x0300A100 --stop-
address=0x0300A300 elf_from_gcc.elf
```

Renesas FSP Configurator supports setting of option memory in BSP settings, as shown in the following figure for RA8M1 MCU.



EK-RA8D1

Settings

Property	Value
R7FA8D1BHECBD	
RA8D1	
RA8D1 Family	
Security	
Exceptions	
SRAM Accessibility	
BUS Accessibility	
System Reset Request Accessibility	Secure State
System Reset Status Accessibility	Both Secure and Non-Secure State
Battery Backup Accessibility	Both Secure and Non-Secure State
Flash Bank Select Accessibility	Both Secure and Non-Secure State
Graphics Power Domain Security Attribution	Secure State
Uninitialized Non-Secure Application Fallback	Enable Uninitialized Non-Secure Application Fallback
OFS0 register settings	
OFS1_SEL register settings	
Voltage Detection 0 Level Security Attribution	VDSEL setting loads from OFS1_SEC
Voltage Detection 0 Circuit Start Security Attribution	PVDAS setting loads from OFS1_SEC
Voltage Detection 0 Low Power Consumption Security Attributi	PVDLPSEL setting loads from OFS1_SEC
WDT/IWDT Software Debug Control Security Attribution	SWDBG setting loads from OFS1_SEC
Tightly Coupled Memory (TCM)/Cache ECC Security Attributior	INITECCEN setting loads from OFS1_SEC
OFS1 register settings	
OFS2 register settings	
Block Protection Settings (BPS)	
Permanent Block Protection Settings (PBPS)	
First Stage Bootloader (FSBL)	
FSBL Control 0 (FSBLCTRL0)	
FSBL Control 1 (FSBLCTRL1)	
FSBL Control 2 (FSBLCTRL2)	
Code Certificates (SACCn)	
FSBL Measurement Report Address (SAMR)	0xFFFFFFFF
Clocks	
HOCO FLL Function	Disabled
Clock Settling Delay	Enabled
Sleep Mode Entry and Exit Delays	Enabled
RTOS Sleep Mode Entry and Exit Delays	Enabled
MSTP Change Delays	Enabled
Settling Delay (us)	150
Cache settings	
Data cache	Disabled
Dual Bank Mode	Disabled
Main Oscillator Wait Time	8163 cycles
RA Common	
Main stack size (bytes)	0x400
Heap size (bytes)	0x1000
MCU Vcc (mV)	3300
Parameter checking	Enabled
Assert Failures	Return FSP_ERR_ASSERTION
Error Log	No Error Log
Clock Registers not Reset Values during Startup	Disabled
Main Oscillator Populated	Populated
PFS Protect	Enabled
C Runtime Initialization	Enabled
Early BSP Initialization	Disabled
Main Oscillator Clock Source	Crystal or Resonator
Subclock Populated	Populated
Subclock Drive (Drive capacitance availability varies by MCU)	Standard/Normal mode
Subclock Stabilization Time (ms)	1000

Figure 13. Option Memory Settings in FSP Configuration for RA8M1 MCU

5. Clock Circuits

The RA8 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 clock. This input is directed to PLLn (n=1,2) where it is eventually multiplied up to the PLL clock, then supplied into the CPU clock (CPUCLK), system clock (ICLK), flash clock (FCK), peripheral module clocks (PCLKn), external bus clock, and trace clock. Additional selectors and frequency dividers are used to generate the SCI clock, SPI clock, Octal SPI clock, CANFD clock, LCD clock, USB clocks, and I3C clock. 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. RA8 Oscillators

Oscillator	Input Source	Frequency	Primary Uses
Main clock (MOSC)	External crystal/resonator (EXTAL, XTAL)	8 MHz to 48 MHz	PLL1 input, PLL2 input, main system clock, CLKOUT, CAN clock, CAC clock, MIPI-PHY clock, USB-PHY clock
	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, RTC clock
	External clock (EXCIN)		
High-speed on-chip (HOCO)	On-chip oscillator	16/18/20/32/48 MHz	PLL1 input, PLL2 input, main system clock, CLKOUT, CAC clock
Middle-speed on-chip (MOCO)	On-chip oscillator	8 MHz	System clock at startup, CLKOUT, CAC 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, Watchdog Timer clock

5.1 Reset Conditions

After reset, RA8 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 4).

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 setup 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. Example Frequency Range for RA8M1 MCU Internal Clocks

Clock Frequency [MHz]	**CPUCLK** [2]	**ICLK** [1]	**PCLKA** [1]	**PCLKB**	**PCLKC**	**PCLKD**	**PCLKE**
Maximum	480	240	120	60	60	120	240
Minimum		—	—	—	—	—	—

1. The ICLK and PCLKA frequencies must be the same and at least 12.5 MHz if the Ethernet controller is in use.
2. Maximum CPUCLK frequency depends on the package type and operating junction temperature.
   When CPUCLK is over 360 MHz, MOSC must be used for PLL1 clock source.

Clock Frequency [MHz]	**FCLK** [1]	**BCLK**	**EBCLK**	**SDCLK**	**UCLK**	**USB60 CLK**	**DCLK**
Maximum	60	120	60	120	48	60	120
Minimum	4	—	—	—	48	60	

1. The FCLK must run at a frequency of at least 4 MHz when writing or erasing ROM or data flash.

Clock Frequency [MHz]	**OCTACLK**	**CANFD CLK**	**LCDCLK**	**I3CCLK**	**MIPICLK**	**SCI CLK**	**SPICLK**
Maximum	200	80	240	200	48	120	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 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 controller (ETHERC) and Ethernet DMA Controller (EDMAC) are used, PCLKA must be in the range 12.5 MHz to 120 MHz.

5.2.3 Requirements for Programming and Erasing ROM or Data Flash

The FCLK must be at least 4 MHz to perform programming and erasing on internal ROM and data flash.

5.2.4 Requirements for SDRAM Controller

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

5.2.5 Requirements for MIPI D-PHY

The clock for the MIPI PHY 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 RA8 MCUs shown as follows. The tool will highlight if setting conditions violate specifications for the device.

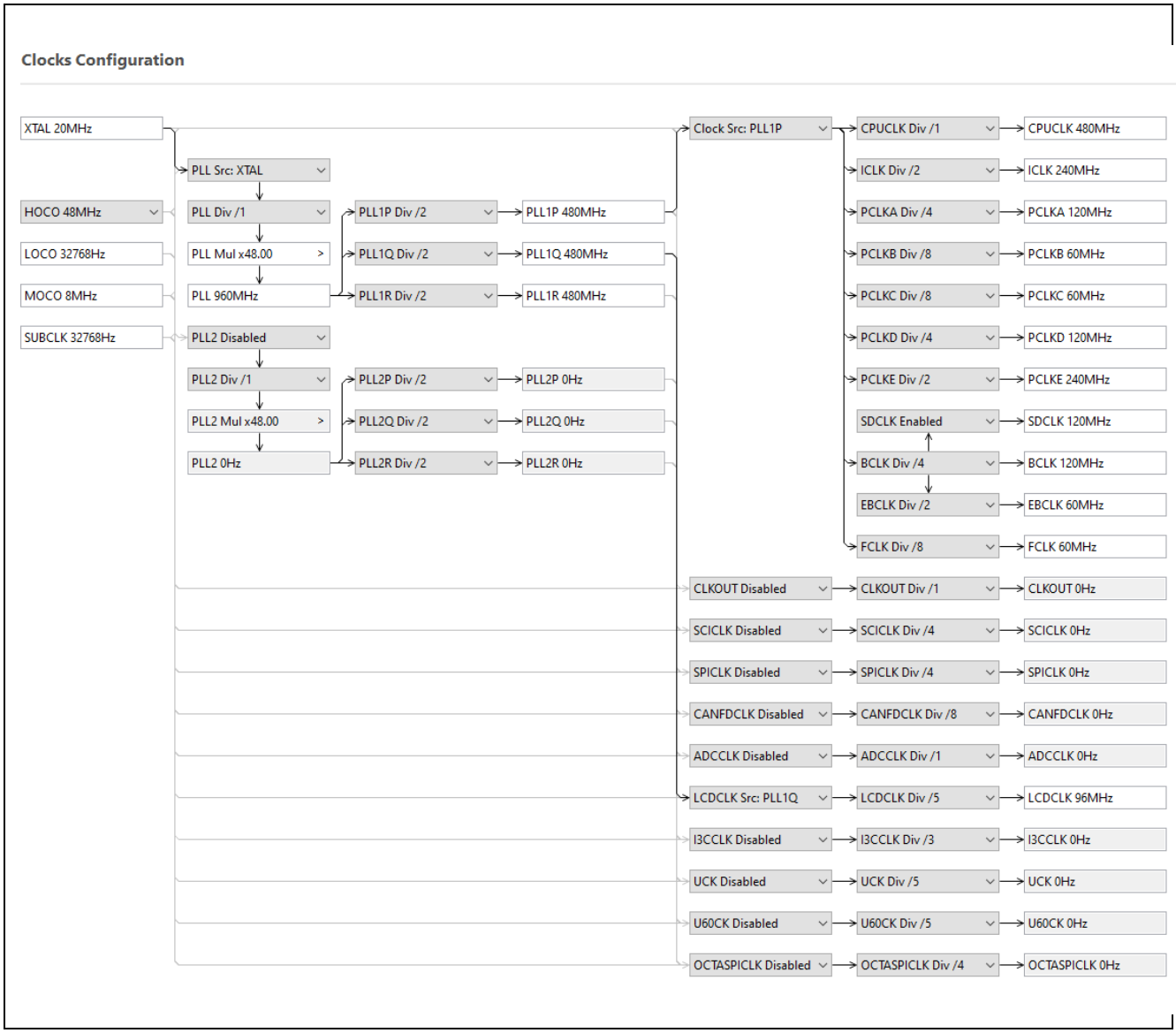


Figure 14. Clock Settings Using Renesas FSP Configurator

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 Flash Interface Clock

The Flash interface Clock (FCLK) is used as the operating clock when programming and erasing internal flash (Code Flash and Data Flash) and for reading from the data flash. Therefore, the frequency setting of the FCLK will have a direct impact on the amount of time it takes to read from the data flash. If the user's program is reading from the data flash, or performing programming or erasures on internal flash, then using the maximum FCLK frequency is recommended.

Please note that the FCLK frequency does not have any impact upon reading from Code Flash or reading and writing to SRAM.

5.8 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.

5.9 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 RA8 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.

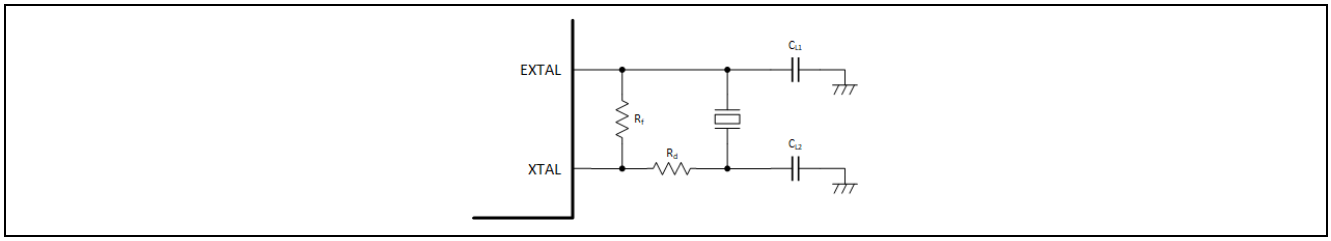


Figure 15. Example of Crystal Resonator Connection

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.

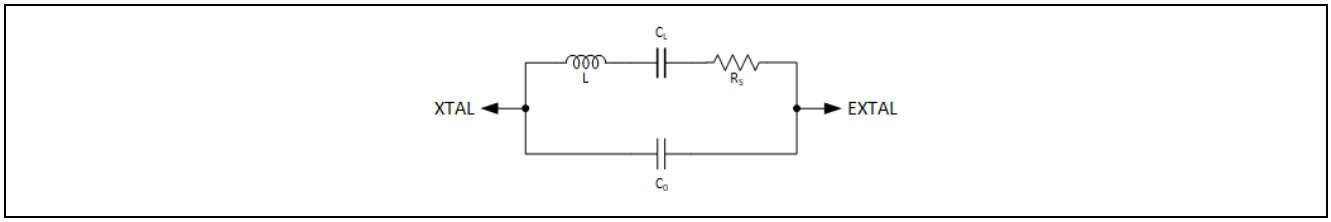


Figure 16. Equivalent Circuit of the Crystal Resonator

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 in to 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 “RA Family Design Guide for Sub-Clock Circuits” available at www.renesas.com.

6. Reset Requirements and the Reset Circuit

There are fourteen types of resets.

Table 11. Arm® Cortex®-M85 Device Resets

Reset Name	Source
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)
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh occurs
Watchdog timer reset	The watchdog timer underflows, or a refresh error for CPU occurs
CPU Lockup reset	This reset is generated when CPU encounters lockup
Bus Error reset	A bus error occurred (MSAU, MMPU, Illegal address, Slave TrustZone® Filter, Slave Bus, or Bufferable write error)
Common Memory Error reset	SRAM error (ECC error or Parity error of SRAM and Standby SRAM)
VBATT_POR reset	VBATT_R voltage drop detection below $V_{PORBATT}$
Deep software standby reset	Deep software standby mode is canceled by an interrupt
Software reset	Register setting

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 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-Monitoring Resets

The RA8 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, 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} and V_{det2} are each selectable from 13 different levels.

When VCC subsequently rises above V_{det0}, V_{det1}, or V_{det2}, release from the voltage-monitoring 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.

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

6.4 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.

6.5 Watchdog Timer Reset

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

When the 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.

6.6 Deep Software Standby Reset

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

When Deep Software Standby mode is canceled, a deep software standby reset is generated, and clock oscillation starts. On receiving the interrupt, after the Deep Standby Cancellation Wait Time (tDSBYWT) has elapsed, reset is canceled, and normal processing starts. 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.

6.7 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.

6.8 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.

6.9 Determination of Cold/Warm Start

The RA8 MCUs allow 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.

6.10 Determining the Reset Source

The RA8 MCUs allow the user to determine the reset signal generation source. Read RSTSR0, RSTSR1, 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, Deep Software Standby or Power On Reset using CMSIS based register structure in Renesas FSP.

```
/* Deep Software Standby Reset */
if(1 == R_SYSTEM->RSTSR0_b.DPSRSTF)
{
  /* Do something */
}

/* Power on Reset */
if(1 == R_SYSTEM->RSTSR0_b.PORF)
{
  /* Do something */
}

/* Software Reset */
if(1 == R_SYSTEM->RSTSR1_b.SWRF)
{
  /* Do something */
}
```

7. Security Features

RA8 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

RA8 MCUs include Arm® TrustZone® (TZ) security features. 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. For full details of TZ implementation, refer to Arm documentation, the Security Features section in the appropriate Hardware User's Manual, and the Reference documents in this section. RA8 MCUs implemented new ARM Security Features compared with the other RA6/RA4 Cortex®-M33 MCUs which supports Trust Zone. For example, the inclusion of the Secure Attribution Unit (SAU).

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 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 Armv8.1-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 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 setup using the MCU boot mode. In development stage, the TrustZone Secure boundary regions can be set using the following methods:

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

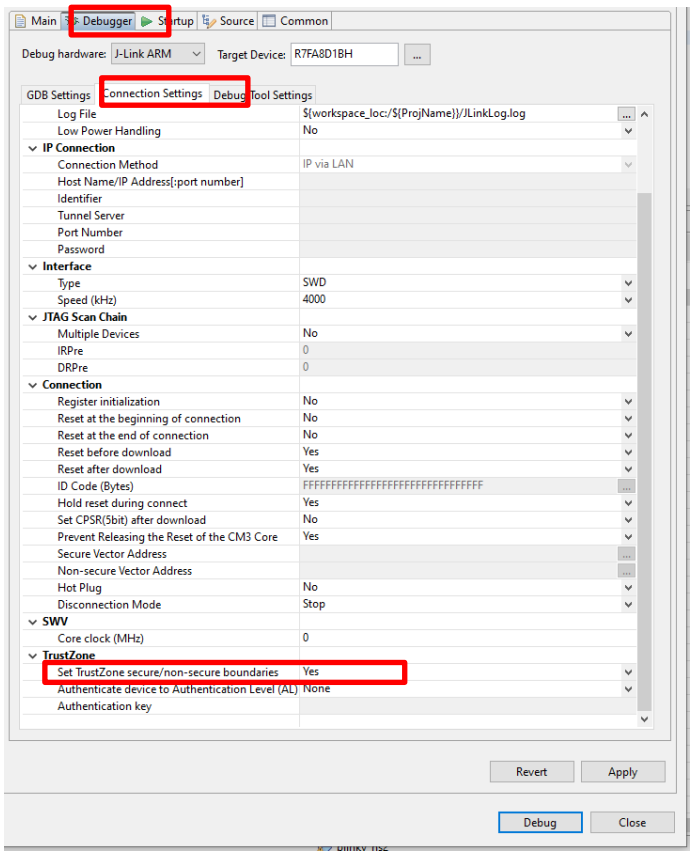


Figure 17. Set up the TrustZone® Region during the Debug Launch Session

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 e^2 studio, the user can choose to set up the TrustZone boundary using any one of the three different debuggers as shown in Figure 18.

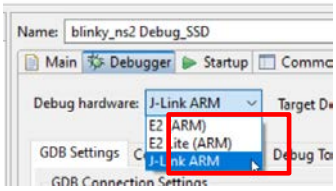


Figure 18. Debuggers Supported

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 RA8 MCUs. For a TrustZone® project, either interface can set up the TrustZone boundary prior to releasing the Reset pin and launch the user application.

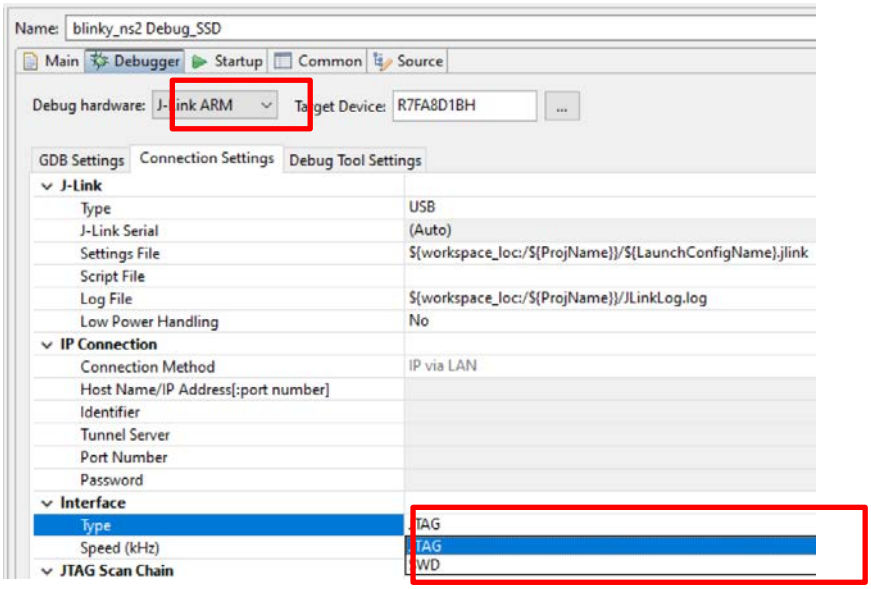


Figure 19. Connection Setting for JLink Debugger

- 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 Wired OR with P201/MD	P211/TCK Wired OR with P201/MD	P201/MD
6	P209/SWO	P209/TDO	P209/TXD9
8	NC	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

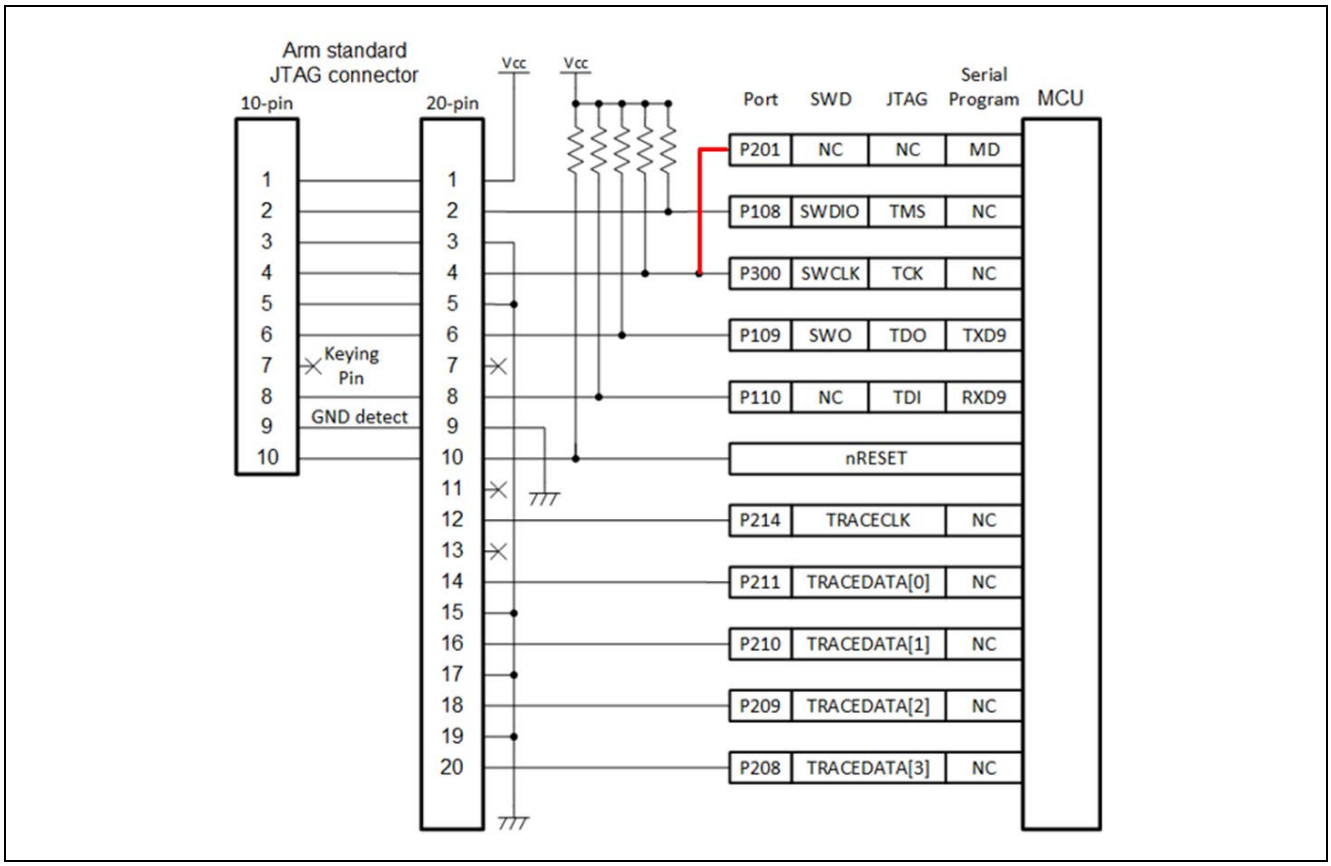


Figure 20. Emulator Connections for MCUs that Support Trustzone®

2. For Keil MDK and IAR EWARM, user needs 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 e^2 studio when the e^2 studio is installed, user can also choose to set up the TrustZone boundaries using RDPM as a separate step when using e^2 studio. In this case, user can disable the TrustZone setup options in the debug configuration. Reference Figure 17 for where this setting is located.

   The RDPM supports two types of connections when accessing the boot mode as shown in Figure 21. If user wants to access the SCI connection through the debug header, user needs to provide the SCI connections as shown in Figure 20.

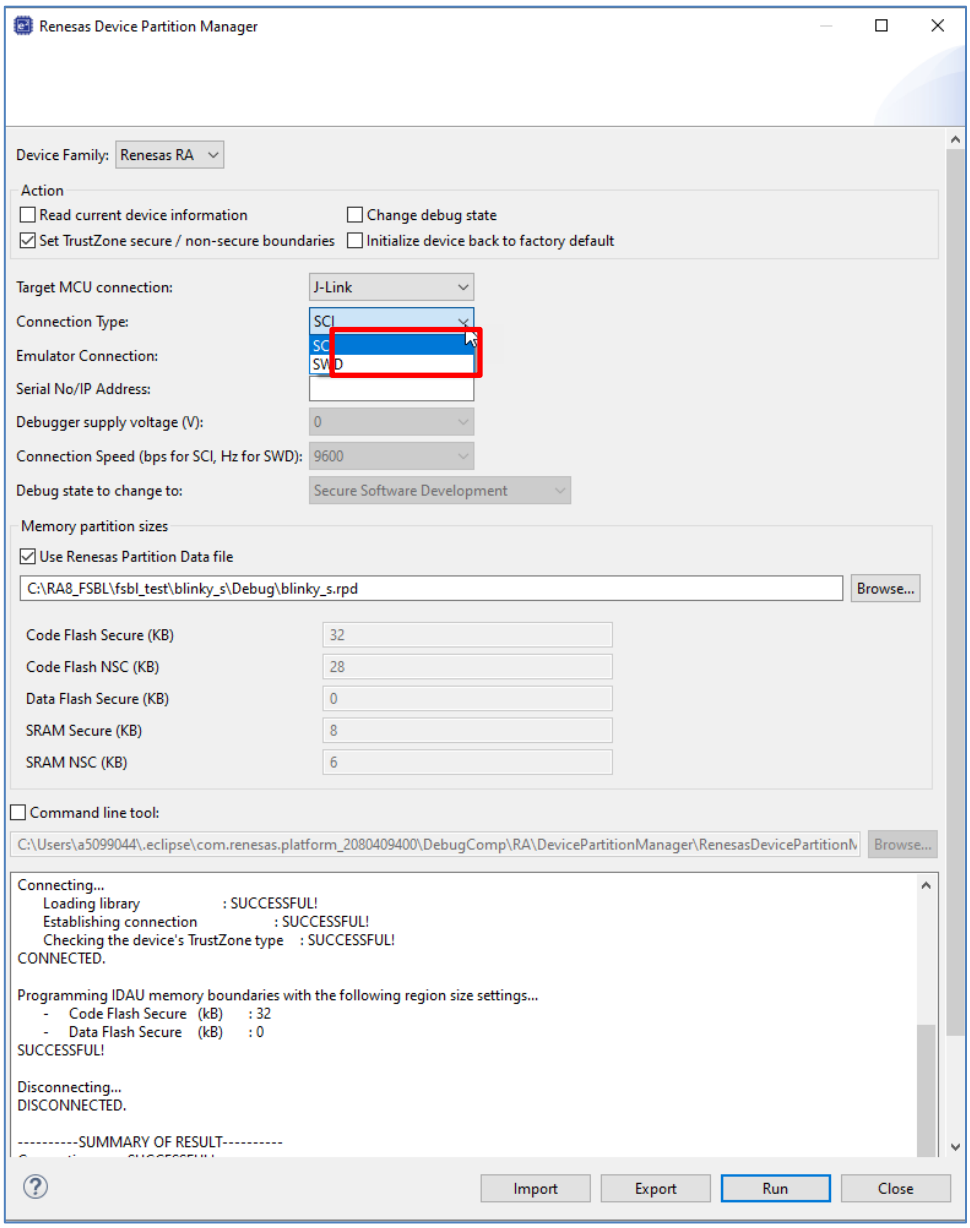


Figure 21. Connection Types when using RDPM

7.2 Device Lifecycle Management

The RA8 Device Lifecycle Management is unique to the RA MCU Family. In the RA8 MCU Series, Device Lifecycle has been separated from authenticated debug, which is different from the DLM used in the RA Cortex®-M33 MCUs.

For the concepts on the RA8 DLM system, user can reference the RA8 User's Manual: Hardware section "Device Lifecycle Management". To utilize the RA8 DLM system, user can follow the instructions provided in the application project R11AN0785 (Renesas RA Family Device Lifecycle Management for RA8 MCUs).

7.3 First Stage Bootloader (FSBL) and Secure Boot.

The Renesas RA8 MCU provides an on-chip immutable First Stage Bootloader (FSBL). 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 registers used during development phase can be configured using the BSP tab property setting.

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 RA8 MCUs, user can use the FSP MCUboot module to establish the OEM_BL or use their own custom bootloader.

For specifications on secure boot operations, user can reference the RA8 User's Manual: Hardware section "Secure Boot". For examples on how to use the Secure Boot feature, user can reference the application project R11AN0774 (Application Design Using the RA8 MCU Series First Stage Bootloader).

7.4 Other Security Features

7.4.1 Secure Key Injection

The Secure Key Injection feature on RA8 MCUs is similar to the 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. Renesas will provide application projects to guide user on the generation, injection, and usage of these keys.

7.4.2 Secure Factory Programming

The Secure Factory Programming is a new feature for RA8 MCUs. This feature supports programming an encrypted firmware image. The image is encrypted with an Image Encryption Key which is wrapped with the Renesas DLM server. This feature enables secure firmware programming in a non-secure environment. Renesas will provide an application project to support user utilizing this feature.

7.4.3 Renesas Secure IP (RSIP-E51A)

The RA8 MCUs have the Renesas Secure IP (RSIP-E51A) for accelerated cryptographic operations. User can reference section "Renesas Secure IP (RSIP-E51A)" on the Hardware User's Manual for more details on the capability of the RAIP-E51A. Example project demonstrating the functionality of the RSIP will be provided in the Renesas GitHub repository. The RSIP-E51A supports two operating mode: Compatibility Mode and Protected Mode. User can reference the application note R11AN0498 to understand these two different modes. Application projects will also be provided to guide user on how to utilize these two operating modes.

7.4.4 Application and OEM BL Anti-rollback

The RA8 MCU supports application firmware version Anti-rollback. Three Anti-rollback counters supports three different use cases: OEM Bootloader (OEM BL), Secure and Non-secure application. See the Option-Setting Memory section 4 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 new security feature on RA8 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. User can reference application project R11AN0773 for the operational details on this feature.

7.4.6 Tamper Pins

Up to three "tamper pins" may exist on a particular RA8 MCUs. When the pin is triggered, the current time can be stored in the RTC and an interrupt can be generated. The VBATT backup registers are zeroized following the detection of a tamper event.

7.4.7 Pointer Authentication and Branch Target Identification (PACBTI)

This is a security feature supported by 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 RA8 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, program ROM, data flash, and external memory.

For greater detail of the RA8 Memory usage, refer to the Application Note “Getting Started with RA8x1 Memory Architecture, Configurations and Topologies”.

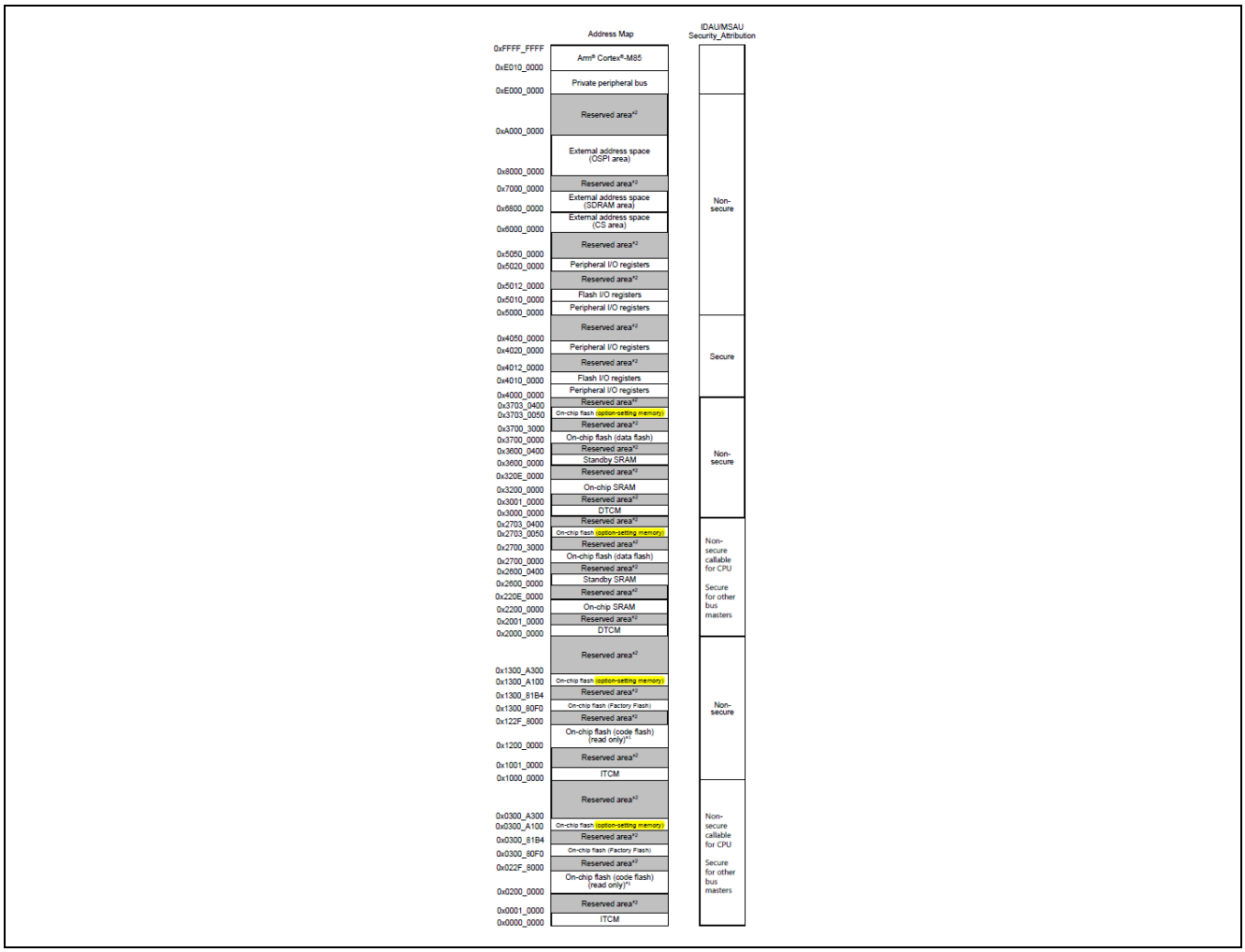


Figure 22. RA8M1 Memory Map

The address space of RA8 MCUs supports aliases. In the address from 0x0000_0000 to 0x5FFF_FFFF, secure and non-secure region 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 used address depends on the relevant security attribution setting and the current security state of bus master.

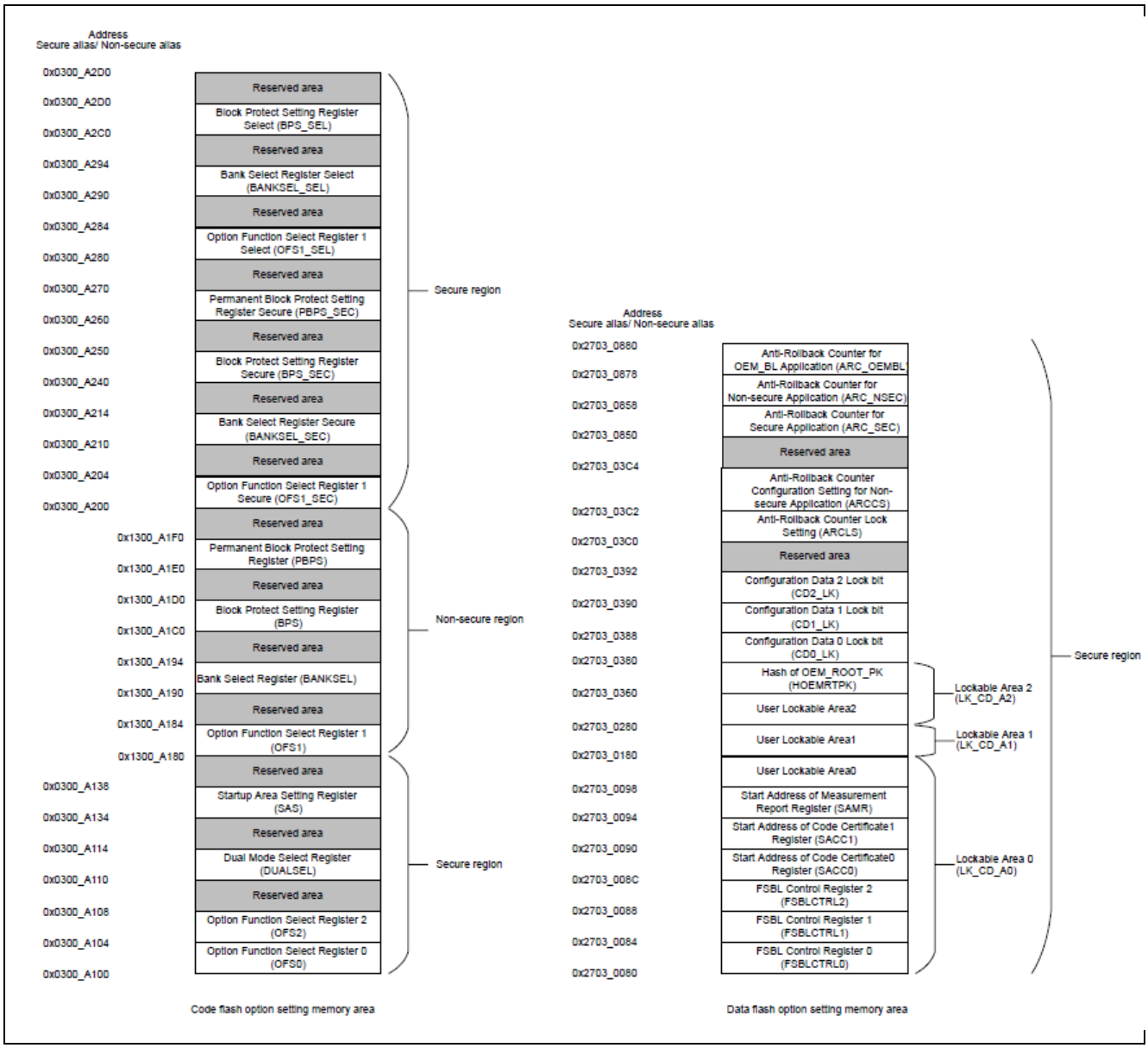


Figure 23. RA8M1 Option Setting Memory Map with alias

8.1 Internal Memory

8.1.1 SRAM

The RA8 MCUs provide on-chip high-density SRAM modules with either parity-bit checking or ECC (Error Correction Code). The area of the first 384 KB of SRAM0 is subject to Single-Error Correction and Double-

Error Detection Code. Parity check is performed on other areas. The following table lists the SRAM specifications. The number of SRAM modules and capacity vary by device. Consult the Hardware User's Manual for specifics.

Table 50.1 SRAM specifications

Parameter	SRAM0	SRAM1
SRAM capacity	384 KB	512 KB
SRAM address	0x2200_0000 to 0x2205_FFFF (Secure alias), 0x3200_0000 to 0x3205_FFFF (Non-secure alias)	0x2206_0000 to 0x220D_FFFF (Secure alias), 0x3206_0000 to 0x320D_FFFF (Non-secure alias)
Access	Wait states are inserted into the read cycle by default. If the ICLK frequency is higher than 120 MHz, a wait state is required. If the ICLK frequency is 120 MHz or less, 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)	Even-parity (Data: 8 bits, Parity: 1 bit)
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 SA. See section 50.3.6. TrustZone Filter function.	

Figure 24. Example of RA8M1 SRAM Specification

8.1.2 Standby SRAM

RA8 MCUs may provide an on-chip SRAM to retain data in (Deep) Software Standby mode. The table below lists the Standby SRAM specifications.

The power supply to the Standby SRAM in (Deep) Software Standby mode is enabled by the DPSBYCR.SRKEEP bit. If the DPSBYCR.SRKEEP bit is set to 1, data in the Standby SRAM is retained in (Deep) Software Standby mode. See section 11, Low Power Modes, for details on the DPSBYCR.SRKEEP bit.

Table 51.1 Standby SRAM specifications

Item	Description
SRAM capacity	1 KB
SRAM address	0x2600_0000 to 0x2600_03FF (Secure alias), 0x3600_0000 to 0x3600_03FF (Non-secure alias)
Access	Wait states are inserted into the access cycle by default. If the ICLK frequency is higher than 120 MHz, a wait state is required. If the ICLK frequency is 120 MHz or less, a wait state is not required.
Data retention function	Data can be retained in Deep Software Standby mode 1. In Deep Software Standby mode 2 and 3, data cannot be retained. See section 51.3.1. Data Retention for details.
parity	Even parity (data: 8 bits, parity: 1 bit)
Module-stop function	Module-stop state can be set to reduce power consumption. See section 51.3.2. Module-stop Function for details.
Security	Permits the read and write operations to Standby RAM following TrustZone Filter function. See section 51.3.4. TrustZone Filter function for details.

Figure 25. Example of RA8M1 Standby SRAM Specification

The LPM (Low Power Mode) driver in Renesas FSP provides an option to cut or keep power to different areas in Standby SRAM as shown in the following figure. The LPM driver's APIs still needs to be invoked to apply the selected settings to the MCU registers.

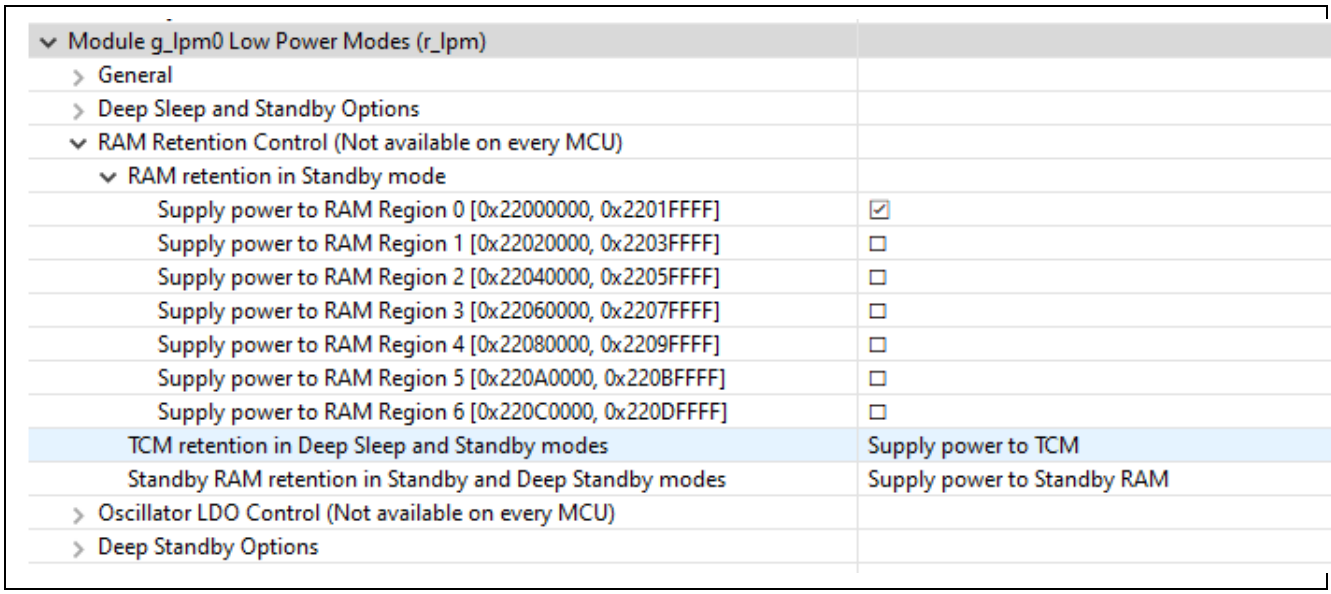


Figure 26. Enable/Disable Power Supply to Standby SRAM Using Renesas FSP Configurator

8.1.3 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. Flash I/O registers to control access flash memory occupy two regions, 0x4010_0000 to 0x401F_FFFF and 0x5010_0000 to 0x5012_0000.

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 On-Chip Flash Memory

The RA8 MCUs feature two flash memory sections: code flash and data flash, which vary in size and programmable cycle capacity. The Flash Control Unit (FCU) controls programming and erasure of the flash memory. The Flash Application Command Interface (FACI) controls the FCU in accordance with the specified FACI commands.

The code flash is designed to store user application code and constant data. The data flash is designed to store information that may be updated from time to time such as configuration parameters, user settings, or logged data. The units of programming and erasure in the data flash area are much smaller than that of the code flash (4 bytes for data flash versus 128 bytes for code flash).

Both the data flash and code flash areas can be programmed or erased by application code i.e., self-programming. This enables field firmware updates without having to connect an external programming tool.

Renesas FSP provides HAL layer drivers for both code flash memory and data flash memory.

The following figure shows example specifications of code flash memory and data flash memory.

Table 52.1 Specifications of flash memory (1 of 2)

Item	Code flash memory	Data flash memory
Memory capacity	User area: 2 Mbytes max	Data area: 12 Kbytes
Read cycle	See section 52.16.3. Access Cycle	See section 52.16.3. Access Cycle
Value after erasure	0xFF	Undefined
Programming/erasing method	• Programming and erasing the code flash memory and data flash memory, and programming the option-setting memory are handled by the FACI commands specified in the FACI command issuing area (Secure: 0x4010_0000, Non-secure: 0x5010_0000) (self-programming). • Programming/erasure through transfer by a serial-programmer via a serial interface (serial programming)	
Protection	Protects against erroneous rewriting of the flash memory	
Dual bank function	The dual-bank structure makes a safe update possible in cases where programming is suspended. • Linear mode: the code flash memory is used as one area. • Dual mode: the code flash memory is divided into two areas.	Not available
Block swap function	The block swap structure makes a safe update for a part of Non-secure application possible in case where programming is suspended.	Not available
Background operations (BGOs)	• The code flash memory can be read while the code flash memory is being programmed or erased. • The data flash memory can be read while the code flash memory is being programmed or erased. • The code flash memory can be read while the data flash memory is being programmed or erased.	
Units of programming and erasure	• Units of programming for the user area: 128 bytes • Units of erasure for the user area: Block units	• Unit of programming for the data area:4/8/16 bytes • Unit of erasure for the data area:64/128/256 bytes
Other functions	Interrupts can be accepted during self-programming. In the initial settings of this MCU, an expansion area of the option-setting memory can be set.	

Figure 27. Specifications of Code Flash Memory and Data Flash Memory on RA8 MCU

Note: Erase state of code flash is FFh but erase state of data flash is undefined.

8.1.4.1 Background Operation

RA8 MCUs support background operations for code flash and data flash. This means that when a program or erase is started, the user can keep executing and accessing memory from memory areas other than the one being operated on. For example, the CPU can execute application code from code flash while the data flash memory is being erased or programmed. Also, the CPU can execute application code from SRAM while the code flash memory is being erased or programmed. The only exception to this rule is that the data flash cannot be accessed during data flash programming or erasing. When using Dual Bank function, the code flash memory can be read while the code flash memory is being programmed or erased.

8.1.4.2 Flash Block Protection

RA8 MCUs with Cortex®-M85 core have a Flash Block Protection feature that protects secure or non-secure flash 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 user area has the block protect setting (BPS or BPS_SEC). When the FBPROT0 or FBPROT1 register is 0x0000 and the block protect bit is 0, issuing the Program or Block Erase command to user area of the code flash causes the command-locked state. To program or erase the block whose block protect bit is 0, set the FBPROT0 or FBPROT1 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, issuing a Program or Block erase

command to user area of the code flash causes the flash sequencer to enter the command-locked state regardless of the FBPROT0 and FBPROT1 register settings.

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.

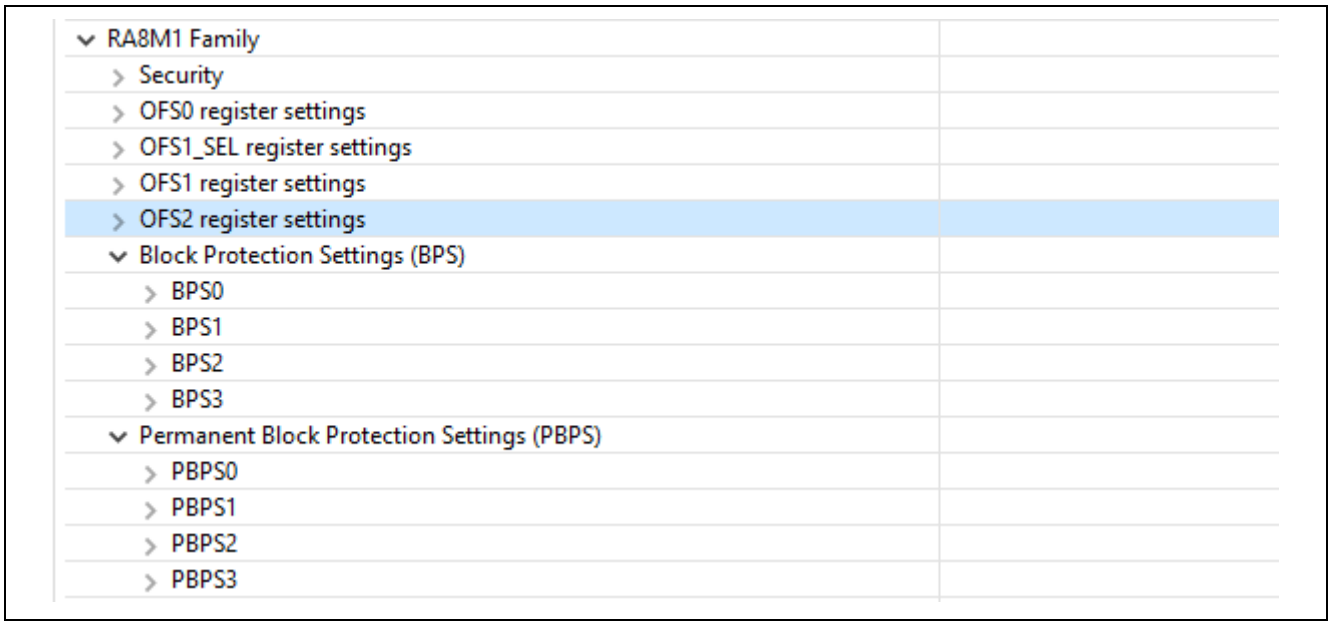


Figure 28. Protection by Block Protect Setting Using Renesas FSP Configurator

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 RA8 family has 128 KB TCM memory that consists of 64 KB ITCM (Instruction TCM) with ECC (8 KB x 8 block) and 64 KB DTCM (Data TCM) with ECC (8 KB x 8 block). Accessing to TCM is not available in CPU Deep Sleep mode.

Both ITCM and DTCM are initialized by FSP. The linker script has defined sections for ITCM and DTCM. User can choose the data and code to put in the corresponding regions. Refer to Application Note “Getting Started with RA8x1 Memory Architecture, Configurations and Topologies” 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.

8.2 External Memory

The RA8 MCUs include a function module for connecting to memory and devices. Some MCUs include a built-in SDRAM controller that allows the use of up to 128 Mbytes of external SDRAM. Eight programmable chip selects provide several options that are settable on a per-chip select basis to allow connection to a wide range of external devices. The external chip select area of the memory map begins at address 0x60000000. Some RA MCUs also have a xSPI (eXpanded Serial Peripheral Interface) which allows interfacing to volatile and non-volatile memory devices. Such devices provide high data throughput, low signal count, and limited backward compatibility with legacy SPI devices. The electrical interface can deliver up to 200 Mbytes per second raw data throughput. The OSPI is compliant with JEDEC standard JESD251(Profile 1.0 and 2.0), JESD251-1 and JESD252. Refer the Hardware User’s Manual for more details.

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 RA8 External address space (CS0 to CS7).

8.2.1.1 Example of SDRAM Initialization

The Renesas FSP provides an example to initialize SDRAM memory controller using direct register access for boards which interface the MCU and an SDRAM. Look for the *bsp_sdram_init* function in the file *ra* > *board* > *board_name* > *board_sdram.c*

8.2.2 Using External Octal SPI Devices

RA8 microcontrollers include peripheral interfaces to connect Serial Peripheral Interface (SPI) devices, including memory devices. 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 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 supporting 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 RA8 OSPI peripheral is compatible with all flash devices that are JESD xSPI standard compliant. RA8 xSPI guarantees operation with JESD251 (xSPI for Non Volatile Memory) compliant memory.

The following table lists some of the OSPI devices that are compatible with RA8M1. Compatibility has been determined by simulation.

Table 13: RA8M1 Compatible OSPI Devices

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

8.2.2.1 OSPI Master Functionalities

The summary of OSPI specifications supported by RA8 MCU family is as follows.

Item	Description
Protocol	Compliant with the xSPI protocol*1
Data transmission and reception	Issue the transaction for up to 2 Slave as Master Only one of the memory devices can operate at a time.
Transfer speed	Support the transfer at xSPI200
Mode	• Support Protocol modes below – 1/4/8pin with SDR/DDR (1S-1S-1S, 4S-4D-4D, 8D-8D-8D) – 2/4pin 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 – Support up to 256 MB address space each CS – 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
Transfer target	• ch1 : GLCDC1 bus master • ch0 : Other bus master
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
Trust Zone Filter	Security attribution can be set for IO register area External address space is defined as Non-secure

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

Figure 29. OSPI Specifications

Renesas RA8 MCUs support 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 of command field and size are delivered from the configured register bits. When using FSP OSPI driver, after R_OSPI_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_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. The 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 Decrypt on the Fly for OSPI” (R11AN0773).

8.3 Data Alignment

There are no restrictions on data alignment in external memory aligning data. The external bus in the RA8 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 accesses, it is not required.

8.4 Restriction on Endian

The external bus can be configured for either little endian or big endian. However, if 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. User can reference the ARM® Cortex®-M85 Technical Reference Manuals to understand more on the Arm MPU configurations. For the RA8 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.

MPU specifications

Classification	Module/Function	Specifications
Memory protection	Arm MPU	Memory protection function for the CPU: • CPU: Secure MPU 8 regions and Non-secure MPU 8 regions
	Bus master MPU	Memory protection function for each bus master except for the CPU: • DMAC (DMAC/DTC): 8 regions • EDMAC (Ether-DMAC): 4 regions • GLCDC (GLCDC0/GLCDC1): 2 regions • DRW (DRW0/DRW1): 3 regions • MIPI DSI:1 region • CEU: 2 regions

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)	DMAC: 32 bytes EDMAC: 32 bytes GLCDC: 1 KB DRW: 1 KB MIPI DSI: 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

The RA8 MCUs include 16KB of L1 Instruction Cache and 16 kB of Data Cache, both with ECC support. The Cache usage is strongly recommended for applications that uses 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 Caches 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 Cortex®-M85 processor does 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 14 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 30. PRCR_S Register

Table 14. PRCR Protection Bits

PRCR bit	Description
PRC0	• Registers related to the clock generation circuit: SCKDIVCR, SCKDIVCR2, SCKSCR, PLLCCR, PLLCR, BCKCR, MOSCCR, HOCOCR, MOCOCR, FLLCR1, FLLCR2, CKOCR, OSTDCR, OSTDSR, PLL2CCR, PLL2CR, PLLCCR2, PLL2CCR2, EBCKOCR, SDCKOCR, SCICKDIVCR, SCICKCR, SPICKDIVCR, SPICKCR, LCDCKDIVCR, LCDCKCR, MOCOUTCR, HOCOUTCR, USBCKDIVCR, OCTACKDIVCR, CANFDCKDIVCR, USB60CKDIVCR, I3CCKDIVCR, USBCKCR, OCTACKCR, CANFDCKCR, USB60CKCR, I3CCKCR, MOSCSCR, HOCOSCR, MOSCWTCR, MOMCR, SOSCCR, SOMCR, LOCOCR, LOCOUTCR, SYRACCR
PRC1	• Registers related to the low power modes: SBYCR, OPCCR, PDCTRGD, PDRAMSCR0, PDRAMSCR1, SSCR1, LPSCR, DPSBYCR, DPSWCR, DPSIER0-3, DPSIFR0-3, DPSIEGR0-2, PLL1LDOCR, PLL2LDOCR, HOCOLDOCR, LVOCR • Registers related to the battery backup function: VBTBER, VBTICTLR, VBTBKRn (n = 0 to 127), VBTBPCR1, VBTBPCR2, VBTBPSR, VBTADSR, VBTADCR1, VBTADCR2, VBTICTLR2
PRC3	• Registers related to the PVD: PVD1CR1, PVD1SR, PVD2CR1, PVD2SR, PVD1CMPCR, PVD2CMPCR, PVD1FCR, PVD1CR0, PVD2CR0, PVD2FCR, VBATTMNSELR
PRC4	• Registers related to the Security and Privilege setting registers: • ELCSARx (x=A,B)*1, ELCPARx (x=A,B), PSARx (x=A to E), MSSAR, PPARx (x=A to E), MSPAR, PmSAR (m=0 to 9, A to G), CPUSAR, DEBUGSAR, ICUSARx (x=A,B,E to I), SRAMSAR, BUSSARx(x=A to C), BUSPARC, MMPUSARx (x=A,B), DTCSAR, DMACSAR, DMACCHSAR, DMACCHPAR, TEVTRCR, SRAMSABAR0-1, STBRAMSABAR, STBRAMPABAR_NS, STBRAMPABAR_S, FSAR, CGFSAR, RSTSAR, LPMSAR, PVDSAR, BBFSAR, DPFSAR, RSCSAR, PGCSAR, VBRSABAR, VBRPABARS, VBRPABARNS
PRC5*1	• Registers related to the reset control: SYRSTMSK0, SYRSTMSK2
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.

Note 1. Only PRCR_S is supported

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 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 RA8 Series of MCUs 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 RA8 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 16.4 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 RA8 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 MPC 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 setup.

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.

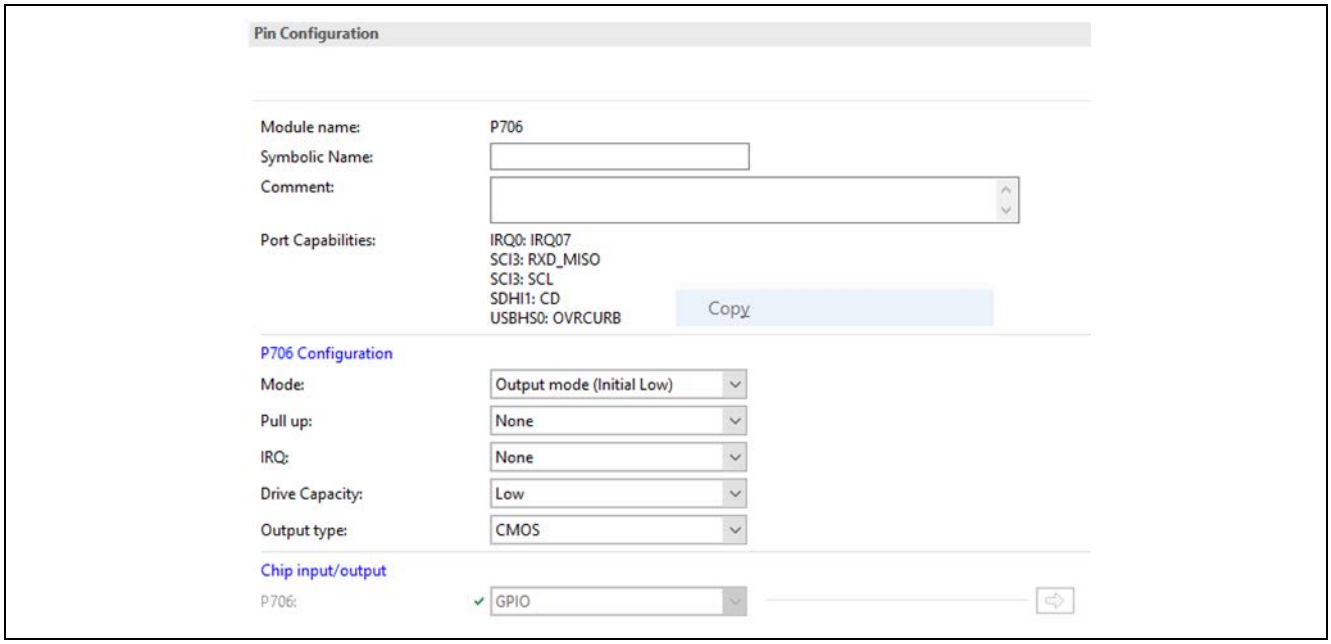


Figure 31. Configuring P706 as Output and Low using FSP Configurator

10.2.1 Internal Pull-Ups

- Most pins on ports 0 through 9, A and B 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 9, A and B 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-, or high-drive output.
- Some ports 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
- 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 is 80 mA.
- The differences between the drive levels are shown is the following table.

Table 15. 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 RA8 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.
- If you are using the external bus, the Ethernet controller, or USB, there are additional registers in the MPC that must be configured before using these peripherals.
- The figure below shows an example of enabling OSPI pins using FSP Pin configuration.

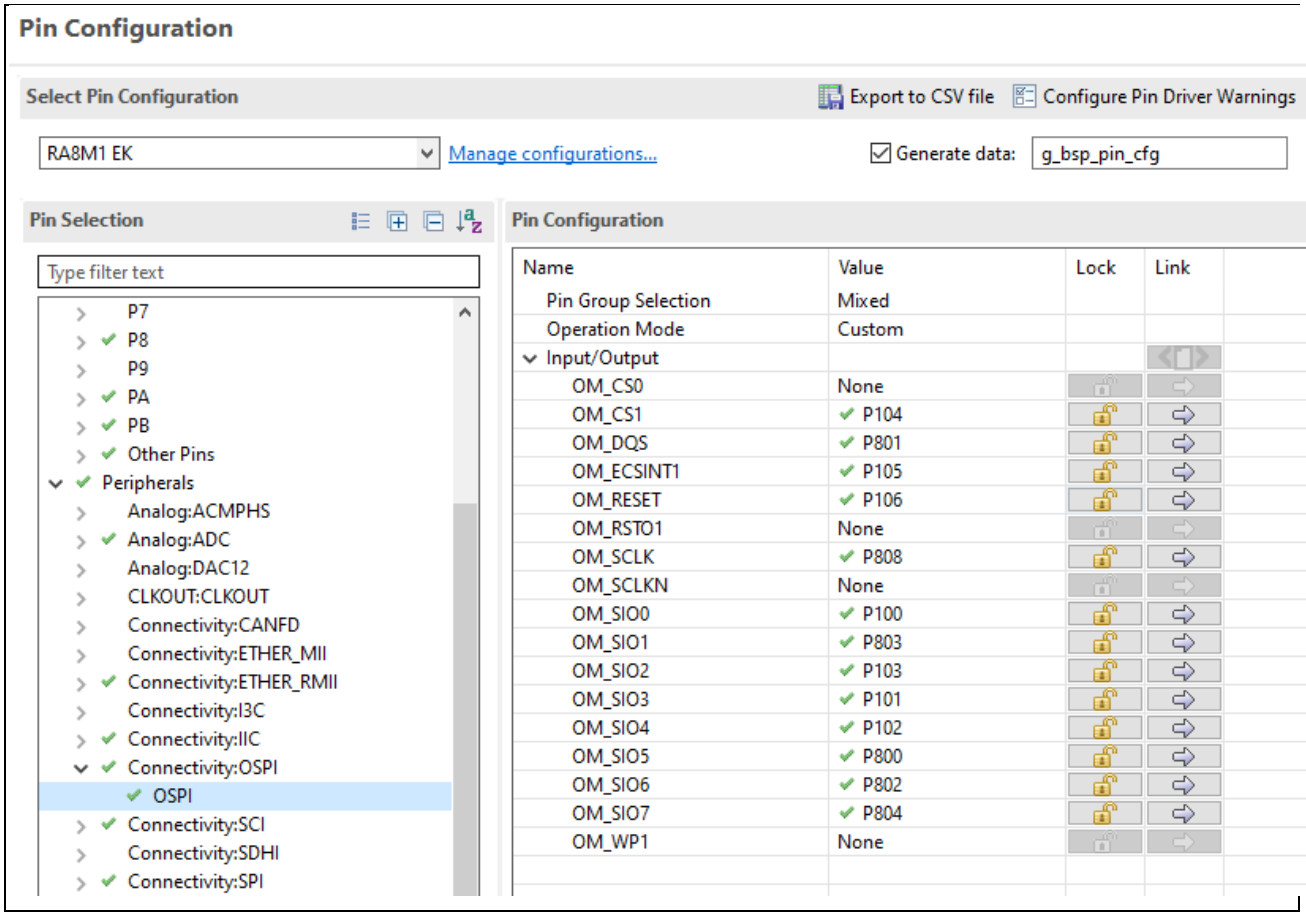


Figure 32. Enabling OSPI pins using Pin Configurator in Renesas FSP

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 33and Figure 34 show examples of enabling and configuring IRQ pins using Renesas FSP.

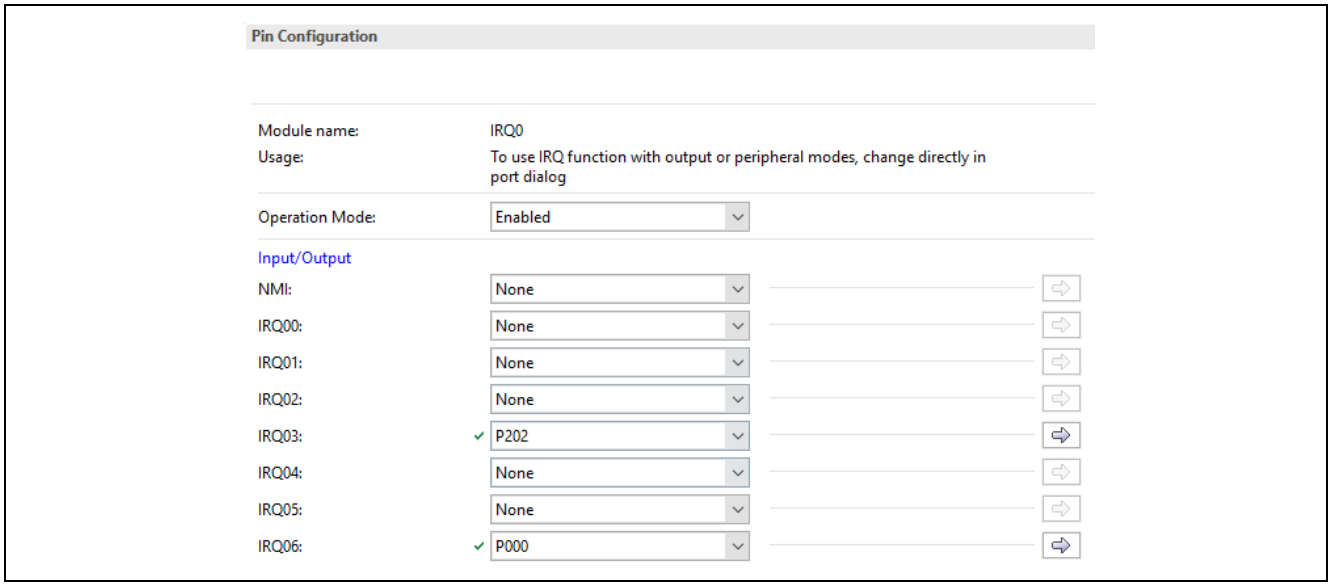


Figure 33. Enable P202, P000 as IRQ03, IRQ06 inputs Respectively using Pin Configurator in Renesas FSP

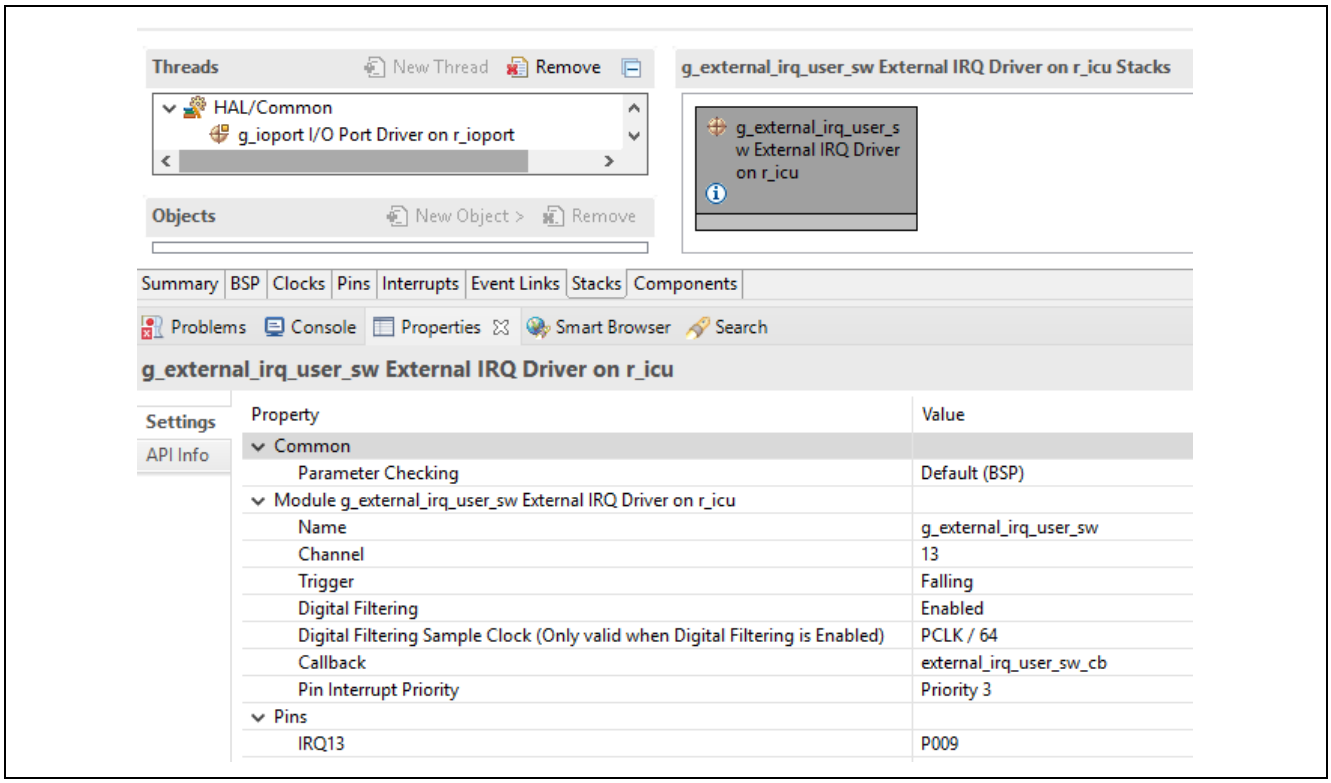


Figure 34. Configure IRQ13 using Renesas FSP Configurator

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 RA8 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 RA8 Series of 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 to 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. Figure 35 shows an example of the ICU specifications, and Figure 36 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 RA8 MCU Group.

Parameter		Description
Maskable interrupts	Peripheral function interrupts	• Interrupts from peripheral modules Number of sources: 306 (select factor within event list numbers 17 to 511)
	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 • 16 sources, with interrupts from IRQi (i = 0 to 15) pins.
	Interrupt requests to CPU (NVIC)	• 96 interrupt requests are output to NVIC.*5
	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 detecting that the main oscillation has stopped
	WDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	IWDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Voltage-monitoring 1 interrupt*3	Voltage monitor 1 interrupt of the voltage monitor 1 circuit (PVD_PVD1)
	Voltage-monitoring 2 interrupt*3	Voltage monitor 2 interrupt of the voltage monitor 2 circuit (PVD_PVD2)
	Common memory error interrupt	Common memory errors include SRAM ECC error, SRAM parity error, or Standby SRAM parity error
	Bus error Interrupt	Bus error includes MPU and TZF error
	Lock up error interrupt	Lock up interrupt
Security	Secure	Some registers have Security Attribution
	Privilege	Each register of the ICU can only be accessed with Privilege access
Low power modes		• CPU Sleep mode: return is initiated by non-maskable interrupts or any other interrupt source • Deep Sleep and Software Standby mode: Return is initiated by non-maskable interrupts. Interrupt can be selected as WUPEN register. See section 13.2.15. WUPEN0 : Wake Up Interrupt Enable Register 0, section 13.2.16. WUPEN1 : Wake Up interrupt enable register 1.
TrustZone Filter		Available

Figure 35. Example of RA8M1 ICU Specification

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQi (i = 0 to 15)	Input	External interrupt request pins

Figure 36. Example of RA8M1 ICU I/O 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.

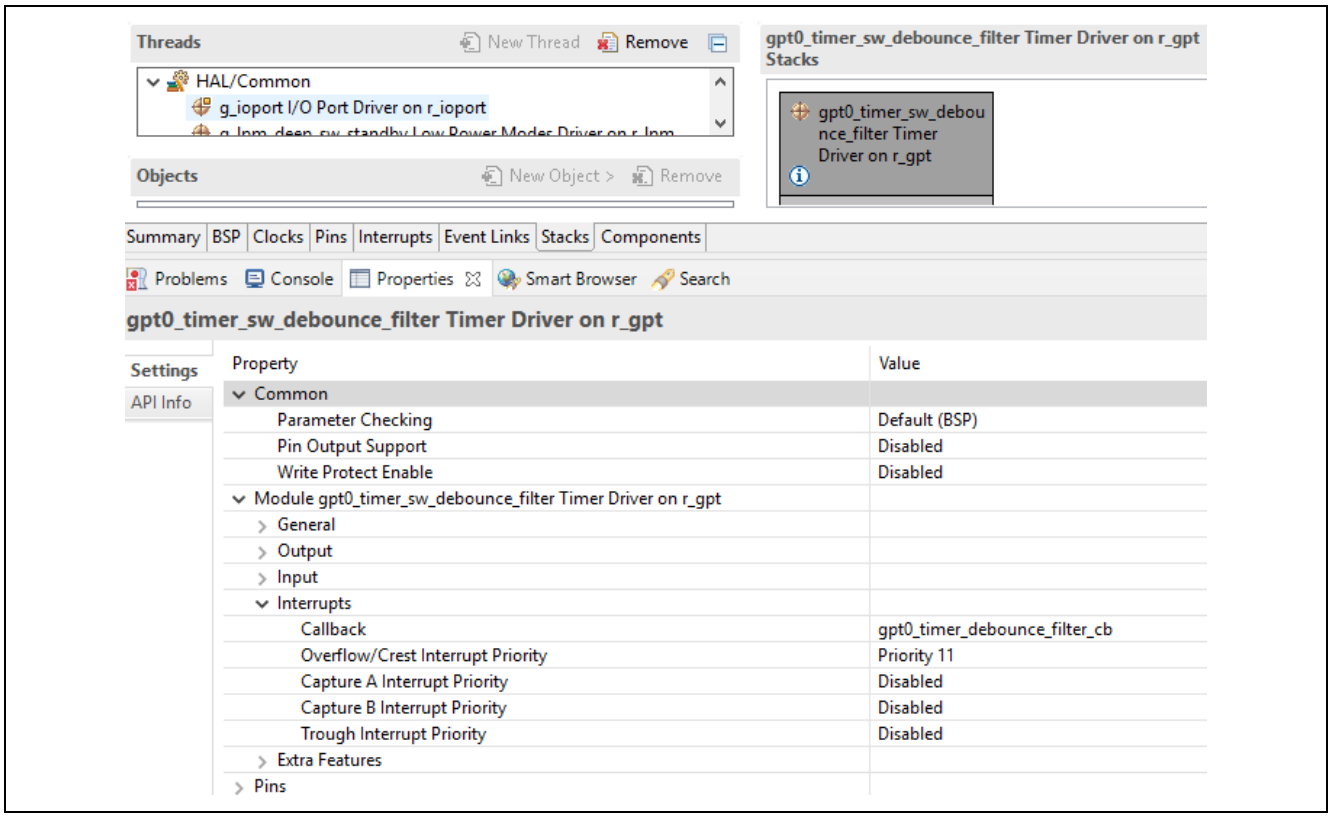


Figure 37. Enable GTP0 Overflow Interrupt and Set User Callback Functions which will be Invoked by Interrupt Service Routine

13. Low Power Consumption

The RA8 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. Refer to the chapter “Low Power Modes” in the Hardware User’s Manual for more details.

RA8 MCUs support four different types of LPM 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 16. 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 CPU clock (CPUCLK), system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE), external bus clock (BCLK), and flash interface clock (FCLK). *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

Item	Specification
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
Power control modes	• Power consumption can be reduced in Normal and Processor low power mode by selecting an appropriate operating power control mode according to the operating frequency. • Two operating power control modes are available: High-speed mode Low-speed mode *2
TrustZone® Filter	Accessibility and configuration of Low Power Mode functions can be restricted by using the security and privilege attribution for multiple LPM registers.

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.

The “Operating state of processor low power mode” and “ Operating state of each low power mode” table lists 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 17. Low Power Consumption Modes

State of operation*1	Software Standby Mode	Deep Software Standby Mode 1, 2, 3
Transition condition	WFI instruction after set LPSCR=0x4 and CPU0.SCR.SLEEPDEEP=1	WFI instruction after set 0x8 ≤ LPSCR ≤ 0xA and CPU0.SCR.SLEEPDEEP=1.
Canceling method	Interrupts shown in table “Interrupt source for canceling CPU Deep Sleep, Software Standby and Deep Software Standby Modes”. Any reset available in the mode.	Interrupts shown in table “Interrupt source for canceling CPU Deep Sleep, 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.

RA8 devices include register settings that allow the MCU to operate with lower power consumption in Normal mode and Sleep mode. These modes are referred to as the Operating Power Control Modes and are controlled by the OPCCR register.

The following is a summary of the Operating Power Consumption Control modes and available oscillators under each mode.

Table 18. Available oscillators in each Operating Power Consumption Control mode

Mode	Oscillator					
	PLL1, PLL2	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	Main clock oscillator	Sub-clock oscillator
High-speed	Available	Available	Available	Available	Available	Available
Low-speed	Not Available	Available	Available	Available	Available	Available

Note: While it may be possible to set the value in the OPCCR register to any of the low power operating modes, clock frequencies must also be set to meet the requirements of the desired mode. Otherwise, the settings in the OPCCR register will not have any effect in lowering power consumption.

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

In Low-speed Mode, all of the available clocks are limited to 1 MHz maximum frequency. Refer to the tables “Operation frequency value in high-speed mode” and “Operation frequency value in low-speed mode” in the Electrical Characteristics chapter of the Hardware User’s Manual for details and notes regarding clock frequency limits.

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.

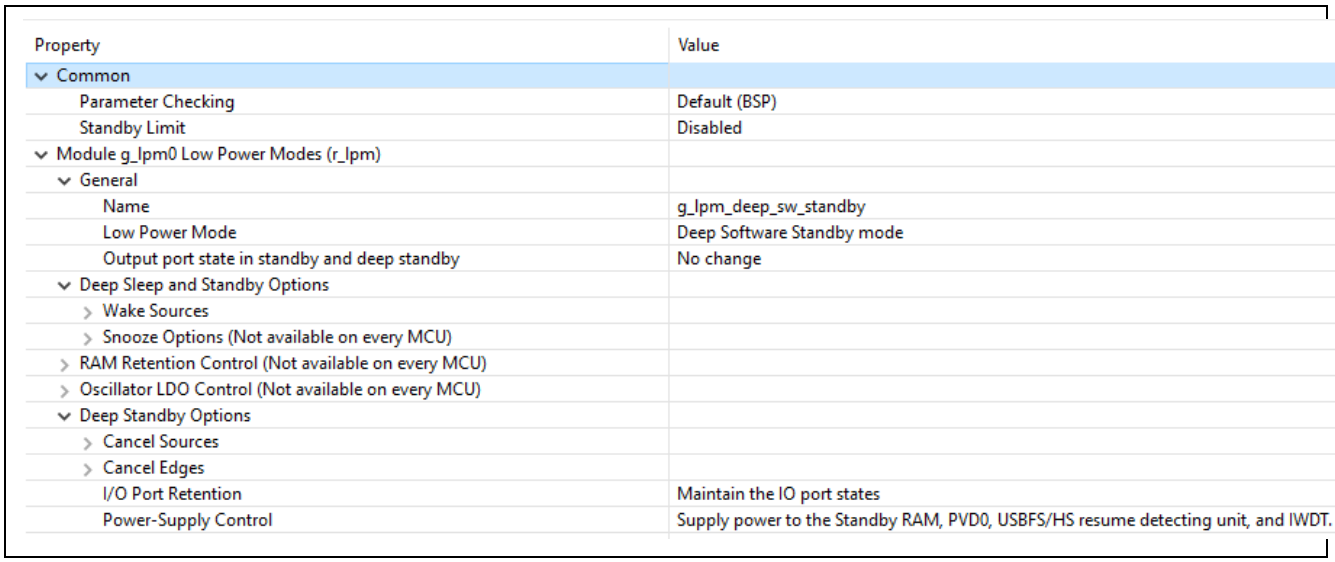


Figure 38. Set up Low Power Mode Using Renesas FSP Configurator

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);
  }
```

14. External Buses

RA8 devices include an external bus controller. Some RA8 devices have built-in SDRAM controllers.

14.1 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.

14.2 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.

14.3 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, 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 BUSnERRADD and BUSnERRSTAT registers. These registers must only be cleared by a reset. For more information, see section “Bus Error Address Register (BUSnERRADD)” and “Bus Error Status Register (BUSnERRSTAT)” in the Hardware User’s Manual.

Note: The DMAC and DTC do not receive bus errors, so their operation is not affected by bus errors.

15. MIPI Subsystem

Some RA8 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.

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.

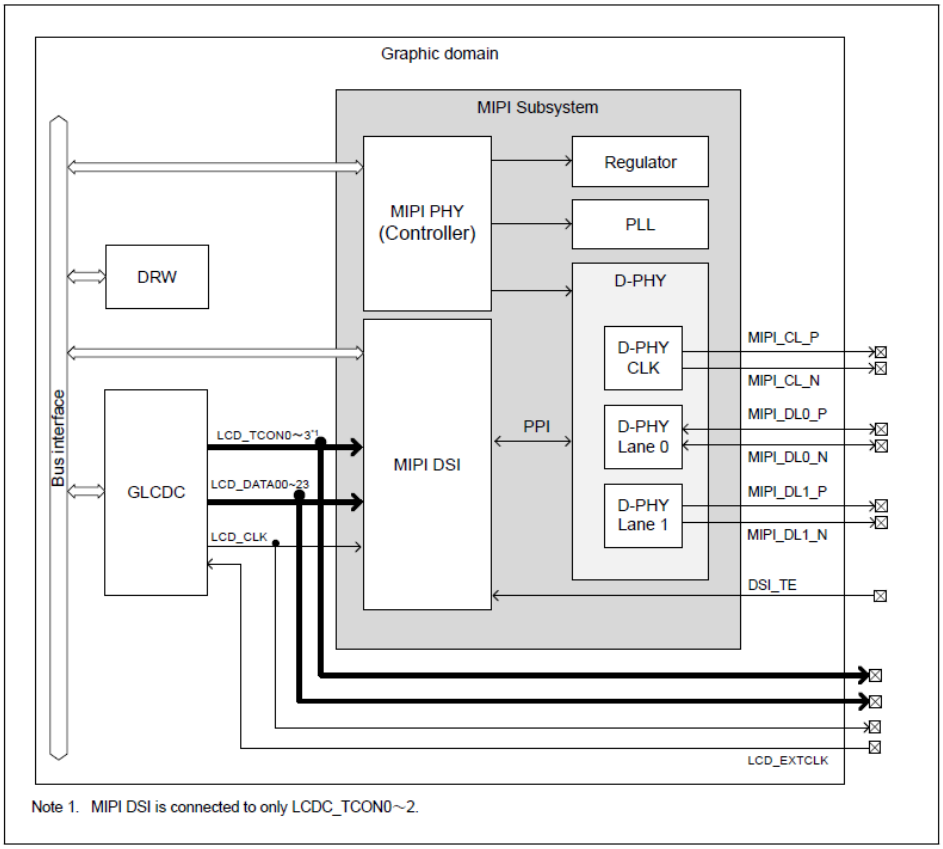


Figure 39. Graphics Domain Block Diagram with MIPI DSI and MIPI PHY

15.1 MIPI DSI

The MIPI subsystem on some RA8 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.

Parameter		Specifications
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	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 WRITE packet • ECC/Checksum verification and ECC error correction for READ 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.

Figure 40. Graphics Specifications

Refer to the MIPI DSI section in the RA8 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));
```

15.2 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 RA8 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.

Parameter		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	Output	DSI Clock Lane positive pin
MIPI_CL_N	Output	DSI Clock Lane negative pin
MIPI_DL0_P	I/O	DSI Data Lane 0 positive pin

Pin name	I/O	Function
MIPI_DL0_N	I/O	DSI Data Lane 0 negative pin
MIPI_DL1_P	Output	DSI Data Lane 1 positive pin
MIPI_DL1_N	Output	DSI Data Lane 1 negative pin
DSI_TE	Input	DSI Tearing Effect pin
AVCC_MIPI	Power	D-PHY Analog Power
VCC18_MIPI	Power	D-PHY I/O Power
VSS_MIPI	Power	D-PHY GND

Refer to section MIPI PHY in RA8 MCU hardware manual for more details.

16. General Layout Practices

16.1 Digital Domain vs. Analog Domain

Renesas RA8 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 RA8 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.

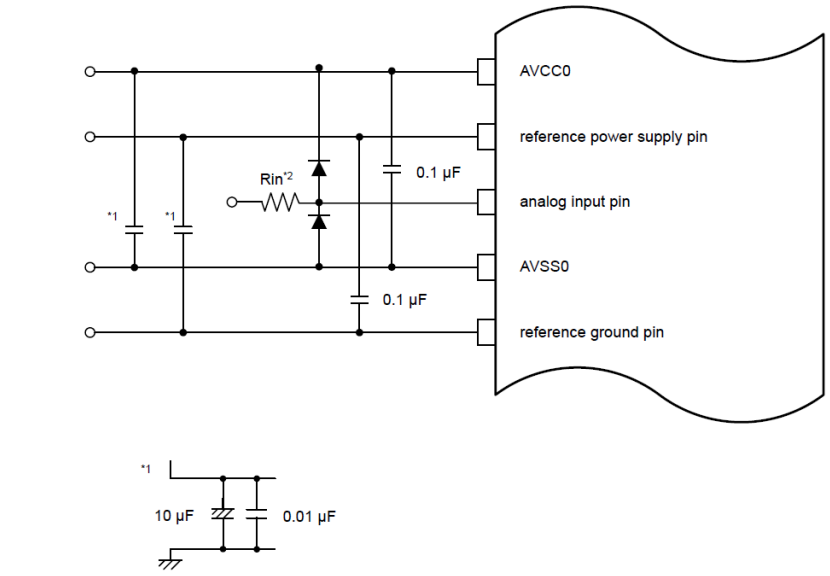
- When assigning I/O pin functions, select pin functions such that analog pins and digital pins are physically separated as much as possible.
- Each analog signal should be separated from all other signals as much as possible.
- PCB routing should isolate each analog signal as much as possible. Avoid 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.

16.2 Precautions for Analog Signals

All RA MCUs include 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 41. 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.

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 RA8 Group microcontrollers, the design concepts and recommendations apply to both groups.

16.3 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 RA8 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 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.

16.4 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, ETHERC, 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.

17. References

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

- Device Lifecycle Management for Renesas RA8 MCUs, document No. R11AN0785
- Renesas Flash Programmer, document No. R20UT5312EJ0
- Security Key Management Tool, document No. R20UT5254EJ
- RA8M1 MCU Hardware User’s Manual, document No. R01UH0994.
- Getting Started with RA8x1 Memory Architecture, Configurations and Topologies, document No. R01AN7088EU0100
- Application Design using RA8 Decrypt on the Fly for OSPI, document No. R11AN0773
- Arm Cortex®-M85 Processor Technical Reference Manual, document No. 101924, available from Arm
- Application Design using RA8 MCU Series First Stage Bootloader No. R11AN0774EU
- E2 Emulator, E2 Emulator Lite Additional Document for User’s Manual, Notes on Connection of RA Devices, document No. R20UT4686EJ0320

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	Oct.24.23	—	First release document
1.01	May.02.24	—	Update sections Power Supplies, On-chip Flash Memory and Register Write Protection

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/.