User's Manual

CONFIDENTIAL

Initial Program Loader

User's Manual: Software

R-Car H3/M3/M3N/E3/D3 Series

Renesas Electronics
www.renesas.com

Rev.3.0.3 Dec. 2021

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

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/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Trademark

How to Use This Manual

- **[Readers]**
  This manual is intended for engineers who develop products which use the R-Car H3/M3/M3N/E3/D3 processor.

- **[Purpose]**
  This manual is intended to give users an understanding of the functions of the R-Car H3/M3/M3N/E3/D3 processor device driver and to serve as a reference for developing hardware and software for systems that use this driver.

- **[How to Read This Manual]**
  It is assumed that the readers of this manual have general knowledge in the fields of electrical
  — engineering, logic circuits, microcontrollers, and Linux.
  → Read this manual in the order of the CONTENTS.
  — To understand the functions of a multimedia processor for R-Car H3/M3/M3N/E3/D3
  → See the R-Car H3/M3/M3N/E3/D3 User's Manual.
  — To know the electrical specifications of the multimedia processor for R-Car H3/M3/M3N/E3/D3
  → See the R-Car H3/M3/M3N/E3/D3 Data Sheet.

- **[Conventions]**
  The following symbols are used in this manual.
  Data significance: Higher digits on the left and lower digits on the right
  Note: Footnote for item marked with Note in the text
  Caution: Information requiring particular attention
  Remark: Supplementary information
  Numeric representation: Binary ... ××××, 0b××××, or ××××B
  Decimal ... ××××
  Hexadecimal ... 0x×××× or ××××H
  Data type: Double word … 64 bits
  Word … 32 bits
  Half word ... 16 bits
  Byte ... 8 bits

Table of Contents

1. Overview

1.1 Overview

This manual explains about the Initial Program Loader (hereafter referred to as “IPL”) for R-Car H3/M3/M3N/E3/D3/H3e/M3e/M3Ne/E3e/D3e System Evaluation Board. The IPL implementation is based on BL2 of Trusted Firmware-A.

1.2 Function

The following shows components of Trusted Firmware-A.
The IPL is designed based on the architecture of BL2 and the part of BL1. BL31, BL32 are included R-Car H3/M3/M3N/E3/D3 Security Board Support Package. BL33 is Linux.

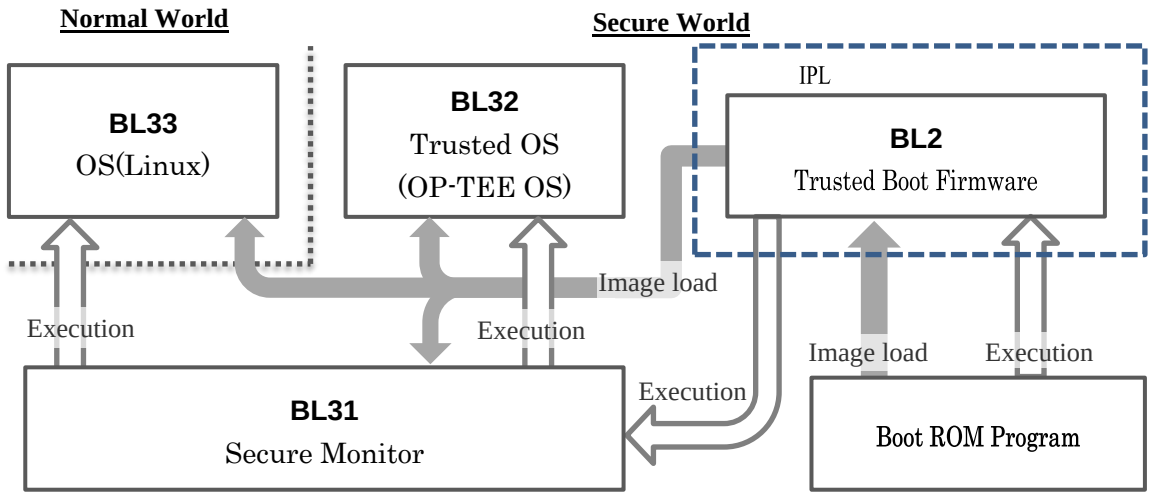


Figure 1.1 Scope of the IPL

The IPL provides environment that is customized source code distributed from the GitHub for the R-Car H3/M3/M3N/E3/D3 Hardware.
IPL (BL2) is running at the EL3 execution level. To avoid wasting memory on BL1, IPL is running BL2 at EL3 execution level using special mode.

The function of the IPL as follows.

Table 1.1 Function of the IPL

Function	Summary
PFC/GPIO setting	Set the PFC configuration. (Pin functions for LSI pins) Set the GPIO configuration. (I/O ports, GPIO interrupts)
SCIF2 console output setting	Set the UART configuration. (output for boot message and error message)
Process timeout detection setting	Set the IPL process timeout detection to the SWDT.
QoS arbitration setting	Set the AXI-bus configuration for the QoS arbitration. (guarantee for latency and bandwidth)
PMIC initial voltage setting	Set the initial voltage scaling to the PMIC. For details of this function, refer to “4.3.2 AVS” of "1.3.1 Related Document No.4". This function is implementation depending on the PMIC (ROHM BD9571) mounted on the evaluation board described in 3.1 hardware environment. Note) R-Car D3 is not supported.
Security access protection setting	Set the security attribute to the LifeC and the AXI-bus. (access authority for between modules, SRAM/SDRAM access protection)
SDRAM setting	Set the DDR configuration to the DBSC4 and the PHY. (Cache memory, Compression/Decompression, Split) Note) R-Car D3 does not have Compression/Decompression, Split.
Suspend to RAM	The Suspend to RAM is function that halt all power supplies except power supply for the SDRAM. At next booting, boot program skips image load and initialization of the system. The suspend state flag(BKUP_TRG) is held in the PMIC. That flag is referenced in the boot sequence, to determine the cold boot or warm boot. Since the mere power-on is a cold boot, after initializing the hardware, load the image. On the other hand, warm boot is different from the mere power-on. The suspend status keep the contents of the SDRAM. So, SDRAM initialization is performed the procedure to resume from suspend state. In addition, since the image is held, it does not load the image. After setting a flag indicating a warm boot, will jump to the entry point of the BL31. For details of this function, refer to “7.2.2 System Suspend to RAM sequence” of "1.3.1 Related Document No.4". This function is implementation depending on the PMIC mounted on the evaluation board described in 3.1 hardware environment. Note) R-Car D3 is not supported.
Initial clock supply setting	Set the clock supply state at modules to the CPG. (to supply module power)
Image loading	To load the program image from the Boot device. The Boot device is determined from HyperFlash / QSPI Flash / eMMC by the MD pin (refer to "1.3.1 Related Document No.2"). In the case of HyperFlash / QSPI Flash : Set the DMA transfer to the SYS-DMAC ch0.

	In the case of eMMC : Set the DMA transfer to eMMC through the SDHI. Initialization and configuration (frequency, bus width, etc.) of the eMMC and reading the program image from eMMC by issuing the command to eMMC.
Lossy register setting	Set the CMA region which supports Near Lossless data compression (FCNL) for media playback
Evaluation board identification	IPL gets the board identification of the R-Car H3/M3/M3N/E3/D3 System Evaluation Board and outputs that information to the log at startup. The format when outputting to the log is as follows. NOTICE: BL2: Board is YYYYYY Rev.ZZZ YYYYYY:Board Name ZZZ: Board Revision Evaluation board identification processing is realized by reading the board information written in the EEPROM mounted on the Evaluation board using IIC for DVFS. Therefore, this function works only with the combination of PMIC (ROHM BD9571) and EEPROM mounted on the Evaluation board described in 3.1 hardware environment. If reading value from EEPROM is 0xFF, processing judges that board identification is not written. The log displays the default string ' BL2: Board is salvator-X Rev.1.0'. Even if 'PMIC_ROHM_BD9571' option is set 0, the same default log message is displayed.
Start generic timer	Start the Arm Generic Timer. Set the base frequency by the evaluation board (Salvator-X, Salvator-XS) and MD pin (MD14, MD13). The Evaluation board identification processing depending on the PMIC mounted on the evaluation board described in 3.1 hardware environment. (If the evaluation board cannot be identified, Salvator-X is selected.) When the evaluation board is Ebisu, Set the base frequency is fixed clock.

1.3 References

1.3.1 Related Document

The following table shows the document related to this function.

Table 1.2 Related Document

Number	Issue	Title	Edition
1	Renesas Electronics	Linux Interface Specification Yocto recipe Start-Up Guide	#0
2	Renesas Electronics	R-Car Series, 3rd Generation User's Manual: Hardware	#0
3	JEDEC	JEDEC STANDARD Embedded Multi-Media Card (e•MMC) Electrical Standard (5.0)	JESD84-B50
4	Renesas Electronics	R-Car H3/M3/M3N/E3/D3 Series, Linux Interface Specification Power Management User's Manual: Software	#0
5	Renesas Electronics	R-CarH3-SiP System Evaluation Board Salvator-X Hardware Manual	#0
6	Renesas Electronics	R-CarM3-SiP System Evaluation Board Salvator-X Hardware Manual	#0
7	Renesas Electronics	R-CarH3-SiP System Evaluation Board Salvator-XS Hardware Manual	#0
8	Renesas Electronics	R-CarM3-SiP System Evaluation Board Salvator-XS Hardware Manual	#0
9	Renesas Electronics	R-CarM3N-SiP System Evaluation Board Salvator-XS Hardware Manual	#0
10	Renesas Electronics	R-CarE3 System Evaluation Board Ebisu Hardware Manual	#0
11	Renesas Electronics	R-CarE3 System Evaluation Board Ebisu-4D Hardware Manual	#0
12	Renesas Electronics	R-CarD3 System Evaluation Board Draak Hardware Manual	#0

#0 : This manual refers to the latest edition.

1.3.2 Related Site for original software

The following table shows the original software related to this function.

Table 1.3 Related Site for original software

Number	Issue	Title and URL	Edition
1	Arm	Trusted Firmware-A https://github.com/ARM-software/arm-trusted-firmware	#1
2	Arm	Trusted Firmware-A Design https://github.com/ARM-software/arm-trusted-firmware/blob/master/docs/design/firmware-design.rst	#1
3	Arm	Trusted Firmware-A Porting Guide https://github.com/ARM-software/arm-trusted-firmware/blob/master/docs/getting_started/porting-guide.rst	#1
4	Arm	Trusted Firmware-A Reset Design https://github.com/ARM-software/arm-trusted-firmware/blob/master/docs/design/reset-design.rst	#1
5	Renesas Electronics	Flash Writer Application Note https://github.com/renesas-rcar/flash_writer	#0

#0 : This manual refers to the latest edition.
#1 : Editions of the current release, see IPL Release Note.

1.4 Restrictions

No.	Descriptions	Impact on customers	Note
1	SoC:D3 SuspendToRAM is not available in D3 because Draak board is not equipped with the PMIC.	SuspendToRAM is not available in D3.	-

2. Terminology and Abbreviation

The following table shows the terminology related and abbreviation to this function.

Table 2.1 Terminology and Abbreviation

Terms	Explanation
Secure World	It is one of the security states that defined Armv8-A architecture. When in this state, the CPU can access both the Secure and Non-secure space.
Normal World	It is one of the security states that defined Armv8-A architecture. When in this state, the CPU can access only Non-secure space.
Suspend to RAM	The function that halt all power supplies except power supply for DRAM. At next booting, boot program skips image load and initialization of the system.
SWDT	System Watch Dog Timer.
LifeC	Life Cycle. It has function for security access protection.
Lossy	Near Lossless data compression (FCNL) for media playback.
BL31	In the Yocto environment of R-Car H3 / M3 / M3N / E3 / D3, it shows the Boot Loader stage 3-1 (BL31) EL3 Runtime Software of Trusted Firmware-A. In this document, it is described as the “Secure Monitor”.
BL32	In the Yocto environment of R-Car H3 / M3 / M3N / E3 / D3, it shows the Boot Loader stage 3-2 (BL32) Secure-EL1 Payload (optional) of Trusted Firmware-A. In this document, it is described as the “OP-Tee”.
BL33	In the Yocto environment of R-Car H3 / M3 / M3N / E3 / D3, it shows the Boot Loader stage 3-2 (BL32) Secure-EL1 Payload (optional) of Trusted Firmware-A. In this document, it is described as the “U-boot”.
BL33x (BL332～BL338)	It is not defined in the Yocto environment of R-Car H3 / M3 / M3N / E3 / D3. It can be used to load programs and data that are not loaded by default, such as the Device Tree (dtb) and the Linux kernel (Image), from the Flash memory.

3. Operating Environment

3.1 Hardware Environment

The following table lists the hardware needed to use this function.

Table 3.1 Hardware environment (R-Car H3/M3/M3N/E3/D3)

Name	Explanation
Evaluation Board	R-CarH3-SiP System Evaluation Board Salvator-X R-CarM3-SiP System Evaluation Board Salvator-X R-CarH3-SiP System Evaluation Board Salvator-XS R-CarM3-SiP System Evaluation Board Salvator-XS R-CarM3N-SiP System Evaluation Board Salvator-XS R-CarE3 System Evaluation Board Ebisu R-CarE3 System Evaluation Board Ebisu-4D R-CarD3 System Evaluation Board Draak
Host PC 1	It is used as debugging environment. Terminal software is executed.
Host PC 2 (Linux)	TFTP server software It is used when Hyper Flash is written by U-Boot or Image is downloaded.
	NFS server software It is used when File system is mounted by NFS.

Refer to "1.3.1-related document No .5-9" for the evaluation board operation.

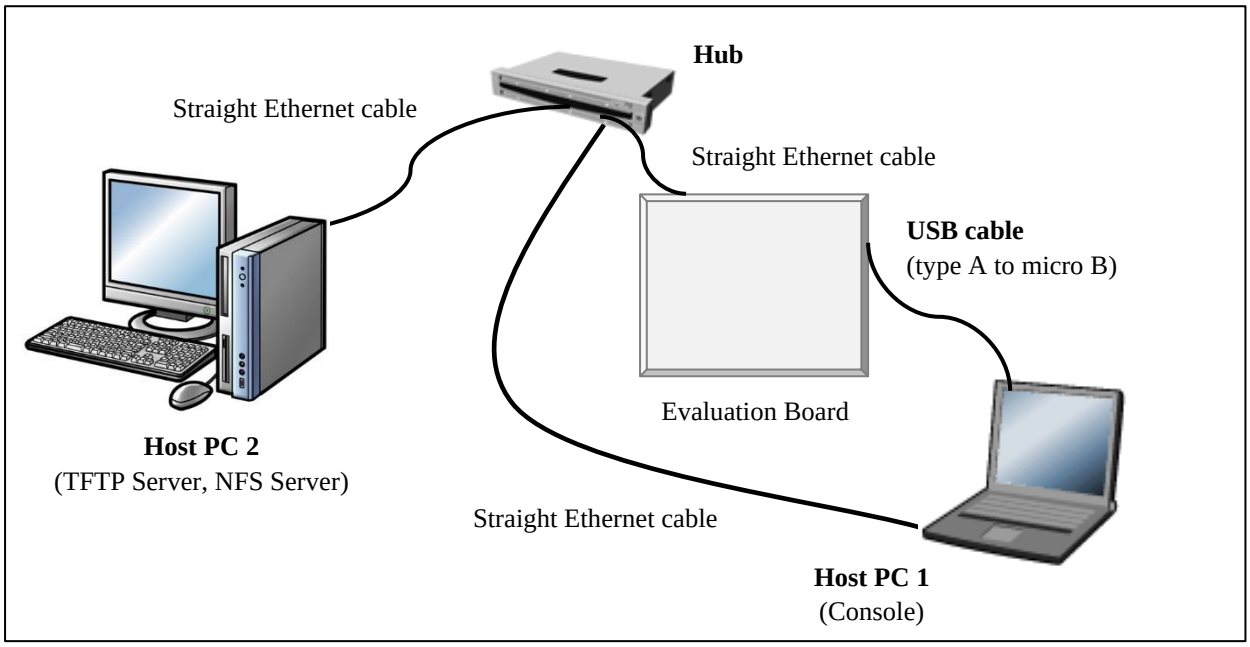
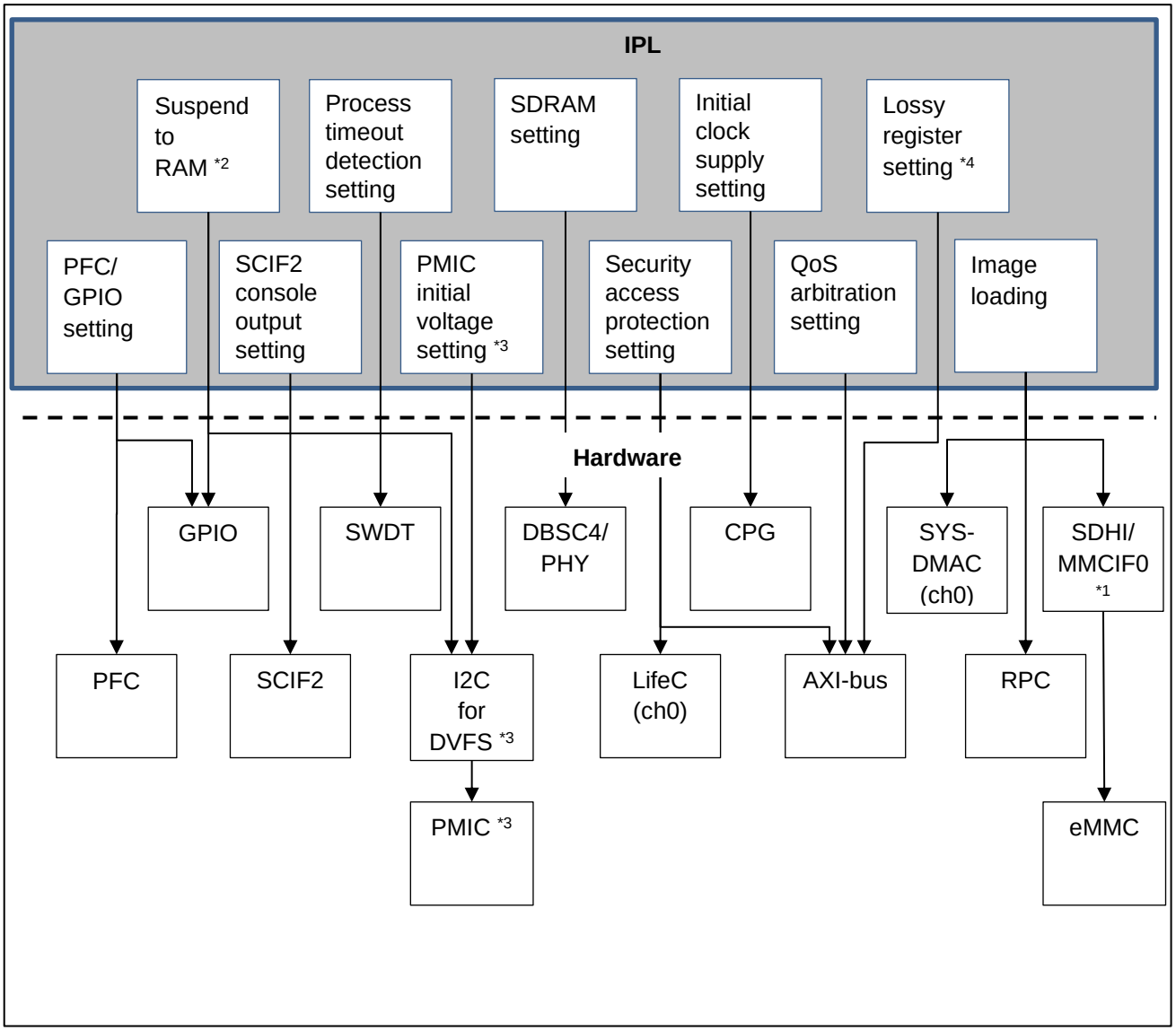


Figure 3.1 Recommended Environment

3.2 Module Configuration

This section explains module configurations.

Figure 3.2 Module configuration



*1 Note) In case of R-Car H3/M3/M3N, SDHI2(MMC0) is used for SDHI and eMMC interface. In case of R-Car E3, SDHI3(MMC1) is used for SDHI and eMMC interface. In case of R-Car D3, MMCIF0 (MMC0) is used for eMMC interface.
*2, 3, 4 Note) In the case of R-Car D3, these modules do not exist.

3.3 State Transition Diagram

There is no state for these modules.

4. External Interface

There is no external interface for these modules.

5. Integration

5.1 Directory configuration

The directory configuration is shown below.

The IPL software environment of this package added directories to the original software directories configuration. At Yocto environment, IPL source code for R-Car H3/M3/M3N/E3/D3 are located on the following path.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/]

```
$WORK
|--build_<Building case>
|     |--tmp
|     |     |--work
|     |     |     |--<Board name>-poky-linux
|     |     |     |     |--arm-trusted-firmware
|     |     |     |     |     |--<Properties of yocto environment>
|     |     |     |     |     |     |--git
|     |     |     |     |     |     |     |--drivers
|     |     |     |     |     |     |     |     |--renesas
|     |     |     |     |     |     |     |     |     |--common
|     |     |     |     |     |     |     |     |     |     |--auth
|     |     |     |     |     |     |     |     |     |     |--avs
|     |     |     |     |     |     |     |     |     |     |--console
|     |     |     |     |     |     |     |     |     |     |--ddr
|     |     |     |     |     |     |     |     |     |     |     |--ddr_a
|     |     |     |     |     |     |     |     |     |     |     |--ddr_b
|     |     |     |     |     |     |     |     |     |     |--delay
|     |     |     |     |     |     |     |     |     |     |--dma
|     |     |     |     |     |     |     |     |     |     |--emmc
|     |     |     |     |     |     |     |     |     |     |--iic_dvfs
|     |     |     |     |     |     |     |     |     |     |--io
|     |     |     |     |     |     |     |     |     |     |--pwrc
|     |     |     |     |     |     |     |     |     |     |--rom
|     |     |     |     |     |     |     |     |     |     |--rpc
|     |     |     |     |     |     |     |     |     |     |--scif
|     |     |     |     |     |     |     |     |     |     |--watchdog
|     |     |     |     |     |     |     |     |     |--rcar
|     |     |     |     |     |     |     |     |     |     |--board
|     |     |     |     |     |     |     |     |     |     |--cpld
|     |     |     |     |     |     |     |     |     |     |--pfc
|     |     |     |     |     |     |     |     |     |     |     |--D3
|     |     |     |     |     |     |     |     |     |     |     |--E3
|     |     |     |     |     |     |     |     |     |     |     |--H3
|     |     |     |     |     |     |     |     |     |     |     |--M3
|     |     |     |     |     |     |     |     |     |     |     |--M3N
|     |     |     |     |     |     |     |     |     |     |--qos
|     |     |     |     |     |     |     |     |     |     |     |--D3
|     |     |     |     |     |     |     |     |     |     |     |--E3
|     |     |     |     |     |     |     |     |     |     |     |--H3
|     |     |     |     |     |     |     |     |     |     |     |--M3
|     |     |     |     |     |     |     |     |     |     |     |--M3N
|     |     |     |     |     |     |     |     |
|     |     |     |     |     |     |     |     |--plat
|     |     |     |     |     |     |     |     |     |--renesas
|     |     |     |     |     |     |     |     |     |     |--common
|     |     |     |     |     |     |     |     |     |     |     |--aarch64
|     |     |     |     |     |     |     |     |     |     |     |--include
|     |     |     |     |     |     |     |     |     |     |     |     |--registers
|     |     |     |     |     |     |     |     |     |     |--rcar
|     |     |     |     |     |     |     |     |
|     |     |     |     |     |     |     |     |--tools
|     |     |     |     |     |     |     |     |     |--renesas
|     |     |     |     |     |     |     |     |     |     |--rcar_layout_create
```

Figure 5.1 File tree for Loader of R-CarH3/M3/M3N/E3/D3

5.2 Integration Procedure

The IPL environment is included in Yocto Environment. To build these software images, refer to “1.3.1 Related Document No.1” Linux Interface Specification Yocto recipe Start-Up Guide documentation for the following function: 3. Building Instructions.

5.3 Option setting

The IPL support the following build options.

Options that are not described in this section have not been set to build command, default value has been assigned to options in building.

Note) Undefined value is treated as reservation. Please do not use undefined value.

• PLAT

Set string is “rcar”.
Any string other than above, please do not set.

• LOG_LEVEL

The IPL provides logging functions ERROR(), NOTICE(), WARN(), INFO() and VERBOSE().
Logging functions value in the following table are valid.
Set value is 0 to 50, 0 is no functions for output logs, 50 is all functions for output logs.

Table 5.1 Association table for the LOG_LEVEL value and valid logging functions

Log level	Valid logging function
0	No functions output logs
10	ERROR()
20	ERROR(), NOTICE()
30	ERROR(), NOTICE(), WARN()
40	ERROR(), NOTICE(), WARN(), INFO()
50	ERROR(), NOTICE(), WARN(), INFO(), VERBOSE()

• DEBUG

Set value is 0 or 1. 0 is a release build, and 1 is a debug build.
If LOG_LEVEL value hasn't been set, LOG_LEVEL is set as the DEBUG value in the following table.

Table 5.2 Association table for the DEBUG value and valid logging functions

DEBUG	build	LOG_LEVEL	Valid logging functions
0	release	20	ERROR(), NOTICE()
1	debug	40	ERROR(), NOTICE(), WARN(), INFO()

If this option is not set, the value is set 0 internally.

• SPD

The IPL provides configuration to load image of BL32.
Set string is "none" or "opteed".

Table 5.3 Association table for the SPD string and BL32 load setting

SPD	BL32 load setting
none	Does not load the image of BL32.
opteed	Load the image of BL32 from HyperFlash / QSPI Flash / eMMC.

If this option is not set, the string is set "opteed" internally.

• RCAR_QOS_TYPE

AXI-bus has the QoS arbitration to control latency and bandwidth.
Set value is 0 or 3.

Table 5.4 Association table for the RCAR_QOS_TYPE value and QoS arbitration setting

RCAR_QOS_TYPE	QoS arbitration setting
0	Enable the QoS arbitration setting (Default setting)
3	Disable the QoS arbitration setting

If this option is not set, the value is set 0 internally.
Note) If MD pins selection is other than MD7=0, MD6=0 or MD7=0, MD6=1, this option has been no effect.

• RCAR_DRAM_SPLIT

DRAM split setting in the SDRAM setting.
Set value is 0 to 3.

Table 5.5 Association table for the RCAR_DRAM_SPLIT value and DRAM split setting

RCAR_DRAM_SPLIT	DRAM split setting
0	Linear (No split)
1	4 channel split
2	2 channel split
3	R-Car H3: 4 channel split / R-Car M3: 2 channel split / R-Car M3N: Linear / R-Car E3: Linear

If this option is not set, the value is set 0 internally.

Note) The 4 channel split option is available only with R-Car H3.
Note) The 2 channel split option is available with R-Car H3 and R-Car M3.
Note) If MD pins selection is other than MD7=0, MD6=0 or MD7=0, MD6=1, this option has been no effect.

The following table shows the relationship between the build options LSI and RCAR_DRAM_SPLIT.

Table 5.6 Association table for the RCAR_DRAM_SPLIT and LSI

build option		RCAR_DRAM_SPLIT			
	value	0	1	2	3
LSI	H3	Linear (No split)	4 channel split	2 channel split	4 channel split
	M3	Linear (No split)	build error	2 channel split	2 channel split
	M3N	Linear (No split)	build error	build error	Linear (No split)
	E3	Linear (No split)	build error	build error	Linear (No split)
	D3	Linear (No split)	build error	build error	build error
	AUTO	Linear (No split)	*1	*1	*1

*1 DRAM split depends on the R-Car H3/M3/M3N/E3/D3 System Evaluation Board used.

The following is example of the build error.
When the build option LSI is M3 and RCAR_DRAM_SPLIT is 1.

```
#error "Don't set DRAM Split 4ch(M3)"
```

• PMIC_ROHM_BD9571

Set 1, when use the evaluation board described in the 3.1 Hardware environment.
This option represents the range of code implementation depending on the PMIC (ROHM BD9571) mounted on the evaluation board described in 3.1 hardware environment.
If this option is not set, the value is set 1 internally.

Note) If this option is set to 0, PMIC (ROHM BD9571) related functions will change as follows:
- Suspend To RAM ... The boot process always becomes Cold Boot.
- AVS initial voltage setting ... Initial voltage is not set for the PMIC.
- Evaluation board identification ... Evaluation board cannot be identified.

Note) This option must be set to disabled in the case of R-Car D3.

• RCAR_AVS_SETTING_ENABLE

Set value is 0 or 1.
0 is a disable the PMIC initial voltage setting, and 1 is an enable.

If this option is not set, the value is set 1 internally.

Note) When using other PMIC, it is necessary to invalidate the PMIC initial voltage setting, or adjust avs_driver according to your PMIC.
Note) This option must be set to disabled in the case of R-Car E3/D3.

• LSI

Target LSI product id number setting of IPL program.
Internal configuration changes are made to operate at the set product.
Set string is “H3”, “M3”, “M3N”, “E3” , “D3” or “AUTO”.

Table 5.7 Association table for the LSI string and Target LSI type setting

LSI	Target LSI type setting
H3	R-Car H3.
M3	R-Car M3.
M3N	R-Car M3N.
E3	R-Car E3.
D3	R-Car D3.
AUTO	R-Car H3, R-Car M3 and R-Car M3N. (not supported E3/D3.)

This is mandatory option. If not set it then the build error occurs.

• LSI_CUT

The CUT number setting of IPL program.
Internal configuration changes are made to operate at the set product.
Set value is 10, 11, 13, 20, or 30.

Table 5.8 Association table for the LSI_CUT value and CUT number setting.

LSI_CUT	CUT number setting of IPL program.
10	Ver.1.0
11	Ver.1.1
13	Ver.1.3
20	Ver.2.0
30	Ver.3.0

If this option is not set, the CUT number is auto detect internally.

Note) When setting LSI_CUT with R-Car M3 Ver.1.2, it becomes the same as Ver.1.1, please set up Ver.1.1.
Note) Ver.1.3 is available with R-Car M3.
Note) When setting LSI_CUT with R-Car D3 Ver.1.1, it becomes the same as Ver.2.0, please set up Ver.2.0.

• LIFEC_DBSC_PROTECT_ENABLE

Settings of the Security access protection to DBSC.
Enable the Security access protection to DBSC, when LIFEC_DBSC_PROTECT_ENABLE is 1.
Set LIFEC_DBSC_PROTECT_ENABLE=0 at the time of build, if want to change disable.

If this option is not set, the value is set 1 internally.

Note) If MD pins selection is MD7 = 1, MD6 = 1, this option has been no effect.

• RCAR_SA6_TYPE

Switch the information to load the image by the boot mode.
Set value is 0 or 1.

Table 5.9 Association table for the RCAR_SA6_TYPE value and the image load information

RCAR_SA6_TYPE	Image load information
0	for Hyper Flash / QPSI Flash boot (Default setting)
1	for eMMC boot

If this option is not set, the value is set 0 internally.

Note) RCAR_SA6_TYPE is the build option for the layout_create.

• RCAR_BL33_EXECUTION_EL

Set the exception level of BL33.
Set value is 0 or 1.

Table 5.10 Association table for the RCAR_BL33_EXECUTION_EL value and Exception level of BL33

RCAR_BL33_EXECUTION_EL	Exception level of BL33
0	EL1 (Default setting)
1	EL2

If this option is not set, the value is set 0 internally.

• RCAR_SYSTEM_SUSPEND

Set 1, when use the evaluation board described in the 3.1 Hardware environment.
If this option is not set, the value is set 1 internally.

Note) The System Suspend process cannot be invalidated, but invalidate the Suspend To RAM correspondence of the IPL, the boot process always becomes Cold Boot, so Suspend To RAM is invalid for the entire system.
Note) This option must be set to disabled in the case of R-Car D3.

• RCAR_REF_INT

Select DRAM refresh interval. If this option is not set, the value is set 0 internally.

Table 5.11 Association table for the RCAR_REF_INT value and DRAM refresh interval

RCAR_REF_INT	DRAM refresh interval
0	Default setting (H3, M3, M3N 1.95us / E3, D3 3.90us)
1	Optional setting (H3, M3, M3N 3.90us / E3 7.80us)

Note) The option is available with R-Car H3 Ver.3.0/Ver.2.0, M3 Ver.3.0/Ver.1.3/Ver.1.2/Ver.1.1, M3N and E3.
Note) Except for the above chips, the value must not be set 1.
Note) If MD pins selection is other than MD7=0, MD6=0 or MD7=0, MD6=1, this option has been no effect.
Note) In the case of R-Car D3, theDRAM refresh interval is fixed at 3.90us.

• RCAR_REWT_TRAINING

Select "periodic write DQ training" mode. If this option is not set, the value is set 1 internally.
"periodic write DQ training" adjusts write signal timings for LPDDR4 skew correction.
For details, refer to the appendix.

Table 5.12 Association table for the RCAR_REWT_TRAINING value and Periodic write DQ training

RCAR_REWT_TRAINING	Periodic write DQ training
0	not available
1	available (Default setting)

Note) The option is available with R-Car H3 Ver.3.0/Ver.2.0, M3 Ver.3.0/Ver.1.3/Ver.1.2/Ver.1.1 and M3N, and when using LPDDR4, RCAR_REWT_TRAINING=0 is prohibited setting.
Note) Except above conditions, the value must not be set 1.
Note) If MD pins selection is other than MD7=0, MD6=0 or MD7=0, MD6=1, this option has been no effect.

• RCAR_BL2_DCACHE

Set this option enables D-cache (L1 data cache / L2 cache) on IPL.
Set value is 0 or 1.
0 is a disable the D-cache setting, and 1 is an enable.

Table 5.13 Association table for the RCAR_BL2_DCACHE value and D-cache setting

RCAR_BL2_DCACHE	D-Cache setting
0	Disable the D-cache setting (Default setting)
1	Enable the D-cache setting

If this option is not set, the value is set 0 internally.

The memory mapping is as follows.
Address translation by MMU is not performed, so the physical address and virtual address are the same value.

Table 5.14 Association table for the memory mapping

address range	memory types and attributes	access	description
0x08000000-0x0BFFFFFF	Normal memory, write-back/read-write-allocate cacheable	RO	External Address Space area for RPC
0x40000000-0xBFFFFFFF	Normal memory, write-back/read-write-allocate cacheable	RW	DRAM area(Legacy)
0xE6000000-0xE62FFFFF	Device memory, nGnRE	RW	SoC register area
0xE6300000-0xE6303FFF	Normal memory, write-back/read-write-allocate cacheable	RW	System RAM area
0xE6304000-0xE63xxxxx	Normal memory, write-back/read-write-allocate cacheable	RO	System RAM area(IPL code area)
0xE63xxxxx-0xE63yyyyy	Normal memory, write-back/read-write-allocate cacheable	RW	System RAM area(IPL data area)
0xE6400000-0xEAFFFFFF	Device memory, nGnRE	RW	SoC register area
0xEB100000-0xEB127FFF	Normal memory, write-back/read-write-allocate cacheable	RO	boot ROM area
0xEC000000-0xFFFFFFFF	Device memory, nGnRE	RW	SoC register area
0x0400000000-0x07FFFFFFF	Normal memory, write-back/read-write-allocate cacheable	RW	DRAM area(4GB over)

xxxxx : depends on the size of IPL code.
yyyyy : depends on the size of IPL data (The end of BL2 given by the linker is set).

• RCAR_SA0_SIZE

Switch the IPL size of dummy certificate.
Set value is 0 or 1.

Table 5.15 Association table for the RCAR_SA0_SIZE value and the IPL size information

RCAR_SA0_SIZE	IPL size information
0	For R-Car E3/D3, IPL size is 80KB.
1	For R-Car H3/M3/M3N, IPL size is 170KB. (Default setting)

If this option is not set, the value is set 1 internally.

Note) RCAR_SA0_SIZE is the build option for the layout_create.

• RCAR_DRAM_DDR3L_MEMCONF

Select DRAM memory size. If this option is not set, the value is set 1 internally.

Table 5.16 Association table for the RCAR_DRAM_DDR3L_MEMCONF value and the DRAM memory size

RCAR_DRAM_DDR3L_MEMCONF	DRAM memory size
0	1G Byte
1	2G Byte (Default setting)

Note) The option is available only with R-Car E3.
Note) Except for the above chips, the value is not available.

• RCAR_DRAM_DDR3L_MEMDUAL

Select the SoC output (the number of connected SDRAM on board), If this option is not set, the value is set 1, internally.

Table 5.17 Association table for the RCAR_DRAM_DDR3L_MEMDUAL value and the SoC output

RCAR_DRAM_DDR3L_MEMDUAL	SoC output (the number of connected SDRAM on board)
0	CS0, ODT0 enable and CS1, ODT1 disable (SDRAM 2pieces)
1	CS0, ODT0, CS1, ODT1 enable (SDRAM 4pieces) (Default setting)

Note) The option is available only with R-Car E3.
Note) Except for the above chips, the value is not available.

• RCAR_DRAM_LPDDR4_MEMCONF

The LPDDR4 settings code included in the IPL release supports 3 types of memory configurations for R-Car H3 Ver.3.0: 4GB (1GB x 4ch) and 8GB (2GB x 4ch) and R-Car M3N: 2GB (2GB x 1ch) and 4GB (4GB x 1ch). Please refer to chapter 7 in [1] about both memory maps.
The option RCAR_DRAM_LPDDR4_MEMCONF shows a capacity of LPDDR4 module per channel. By default, it is defined to '1' (2GB/ch).

Table 5.18 Association table for the RCAR_DRAM_LPDDR4_MEMCONF value and the LPDDR4 configuration

RCAR_DRAM_LPDDR4_MEMCONF	LPDDR4 configuration
0	1G Byte x 4 channel : 4GB
1	2G Byte x 4 channel : 8GB for H3 Ver.3.0 (Default setting) 2G Byte x 1 channel : 2GB for M3N
2	4G Byte x 1 channel : 4GB for M3N
Others	Prohibited

Note) This option is only available for R-Car H3 Ver.3.0 and R-Car M3N, so it does not affect both older-version R-Car H3 and other products' LPDDR4 settings.

• RCAR_DRAM_MEMRANK

IPL supports 1 rank and 2 rank DDR of Salvator-XS with H3 later Ver.3.0. This setting is enabled when RCAR_DRAM_LPDDR4_MEMCONF is 1.

Table 5.19 Association table for the RCAR_DRAM_MEMRANK value and the DDR rank configuration

RCAR_DRAM_MEMRANK	DDR rank configuration
0	The rank is automatically determined from the board ID and DDR is initialized (default setting).
1	Initialize DDR assuming 1 rank DDR is installed.
2	Initialize DDR assuming 2 rank DDR is installed.
Others	Prohibited

5.4 How to

5.4.1 Build option

The following shows the path of recipe file, and the sample for adding or changing build option.
When users change building option, please modify a recipe file to add option.

• Trusted Firmware-A (IPL)

Example for R-Car H3.

On $WORK/meta-renesas/meta-rcar-gen3/recipes-bsp/arm-trusted-firmware/arm-trusted-firmware_git.bb

```
COMPATIBLE_MACHINE = "salvator-x"
PLATFORM = "rcar"
ATFW_OPT_r8a7795 = "LSI=H3 RCAR_DRAM_SPLIT=1 XXXXX=YY"
ATFW_OPT_r8a7796 = "LSI=M3 RCAR_DRAM_SPLIT=2"
```

Example for R-Car M3.

On $WORK/meta-renesas/meta-rcar-gen3/recipes-bsp/arm-trusted-firmware/arm-trusted-firmware_git.bb

```
COMPATIBLE_MACHINE = "salvator-x"
PLATFORM = "rcar"
ATFW_OPT_r8a7795 = "LSI=H3 RCAR_DRAM_SPLIT=1"
ATFW_OPT_r8a7796 = "LSI=M3 RCAR_DRAM_SPLIT=2 XXXXX=YY"
```

DEBUG option

When users add option DEBUG and set it to 1, additionally need to modify like the following steps.

Step1: modify the recipe file

Add the patch file name in the recipe file and change output directory as shown below.

On $WORK/meta-renesas/meta-rcar-gen3/recipes-bsp/arm-trusted-firmware/arm-trusted-firmware_git.bb

```
SRC_URI = \
    "git://github.com/renesas-rcar/arm-trusted-firmware.git;branch=${BRANCH}"
SRCREV = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"


SRC_URI += " \
    file://sample.patch \
"

PV = "v1.1+renesas+git${SRCPV}"

do_deploy() {
    install -d ${DEPLOYDIR}
    install -m 0644 ${S}/build/${PLATFORM}/debug/bl2/bl2.elf
 ${DEPLOYDIR}/bl2-${MACHINE}.elf
    install -m 0644 ${S}/build/${PLATFORM}/debug/bl2.bin ${DEPLOYDIR}/bl2-
 ${MACHINE}.bin
    install -m 0644 ${S}/build/${PLATFORM}/debug/bl2.srec ${DEPLOYDIR}/bl2-
 ${MACHINE}.srec
    install -m 0644 ${S}/build/${PLATFORM}/debug/bl31/bl31.elf
 ${DEPLOYDIR}/bl31-${MACHINE}.elf
    install -m 0644 ${S}/build/${PLATFORM}/debug/bl31.bin
 ${DEPLOYDIR}/bl31-${MACHINE}.bin
    install -m 0644 ${S}/build/${PLATFORM}/debug/bl31.srec
 ${DEPLOYDIR}/bl31-${MACHINE}.srec
. . .
}
```

5.4.2 Import change code

When users want to import edited code into Yocto Environment, need to create a patch file.
Please create a patch file in accordance with the following steps.

Step1: copy & edit source file
Once building Yocto Environment, IPL codes appear under the git directory(see5.1).
Copy the data under the git directory to the user's two directories (e.g. copy to both org and mod directories).
Then edit the source code only in one directory (e.g. mod directory only).

Step2: create patch file (*.patch)
Use the diff or git diff command between two directories to create a patch.
The followings are examples of command.

```
$diff -uprN unedited_directory_path edited_directory_path > sample.patch
```

```
$git diff unedited_directory_path edited_directory_path > sample.patch
```

Patch file is output under the directory in which user enters the command.

Step3: put created patch file in appropriate directory
Put a patch file in the directory of the same level as the recipe file.
(In most cases, the folder's name is the same name as removing the " _git " from the recipe file name.)

Step4: edit recipe file
Edit the recipe file to import a created patch file into Yocto Environment.
The following is example that user applies a created patch file(sample.patch) to build IPL.

On $WORK/meta-renesas/meta-rcar-gen3/recipes-bsp/ arm-trusted-firmware/ arm-trusted-firmware_git.bb

```
SRC_URI = \
    "git://github.com/renesas-rcar/arm-trusted-firmware.git;branch=${BRANCH}"
SRCREV = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"


SRC_URI += " \
    file://sample.patch \
"
```

Step5: building with bitbake

IPL can be built individually.

The following bitbake command is a case to build individually by specifying the recipe file (e.g.XXXX_git.bb).

```
cd $WORK/build_<Building case>
$ bitbake XXXX
```

If the build is successful, the image will be created in $WORK/build_<Building case>/tmp/deploy/images/salvator-x/directory.

If user wants to clean environment, user can run command “bitbake XXXX -c clean”.

For various settings before user run the bitbake command, refer to “1.3.1 Related Document No.1”.

5.4.3 How to customize

The following shows a method of customize.

5.4.3.1 PFC/GPIO setting

Performs the PFC/GPIO setting for the R-Car H3/M3/M3N/E3/D3 System Evaluation Board.

The following shows file and directory structure of the PFC/GPIO setting.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/drivers/renesas/rcar]

```
rcar
 └──pfc
      │   pfc.mk
      │   pfc_init.c
      │
      ├──H3
      │       pfc_init_h3_v1.c
      │       pfc_init_h3_v1.h
      │       pfc_init_h3_v2.c
      │       pfc_init_h3_v2.h
      │
      ├──M3
      │       pfc_init_m3.c
      │       pfc_init_m3.h
      │
      ├──M3N
      │       pfc_init_m3n.c
      │       pfc_init_m3n.h
      │
      ├──E3
      │       pfc_init_e3.c
      │       pfc_init_e3.h
      │
      └──D3
              pfc_init_e3.c
              pfc_init_e3.h
```

pfc_init() switches the initialization function to call for perform a different setting by the product id. Initialization function and implemented file is shown in the Table 5.20 below.

Table 5.20 File and Function list of PFC/GPIO setting

Product ID	CUT number	Filename	Function
H3	Ver.3.0 or later	pfc_init_h3_v2.c	pfc_init_h3_v2()
M3	-	pfc_init_m3.c	pfc_init_m3()
M3N	-	pfc_init_m3n.c	pfc_init_m3n()
E3	-	pfc_init_e3.c	pfc_init_e3()
D3	-	pfc_init_d3.c	pfc_init_d3()

Note) If MD pins selection is other than MD7=0, MD6=0 or MD7=0, MD6=1, register setting of PFC/GPIO is skipped.

5.4.3.2 Process timeout detection setting

Process timeout detection setting specifies the time to detect a state where IPL stopped due to an unexpected problem.

The following shows file and directory structure of the process timeout detection setting.

[$ WORK/arm-trusted-firmware/drivers/renesas/common]

```
watchdog
  | swdt.c
```

Process timeout detection setting is performed by bl2_swdt_init() function implemented in bl2_swdt.c.

bl2_swdt_init() function is selected the setting value of wait time by the product id and MD pins.

In Table 5.21, three definitions are prepared to make the same waiting time for all combinations of Product ID, CUT number, and MD pins.

Table 5.21 Setting value list of Process timeout detection setting

Product ID	MD13 and MD14 setting			
	MD13=0 MD14=0	MD13=0 MD14=1	MD13=1 MD14=0	MD13=1 MD14=1
H3	WTCNT_COUNT_8p13k		WTCNT_COUNT_8p22k	WTCNT_COUNT_8p13k
M3				
M3N				
E3	WTCNT_COUNT_7p81k			
D3	WTCNT_COUNT_8p13k			

Setting value of wait time definition in bl2_swdt.c is shown in the Figure 5.2. By this setting value, wait time becomes 5 [sec] for all combinations of Table 5.21.

```
#define WTCNT_COUNT_8p13k       (0x10000U - 40687U)
#define WTCNT_COUNT_8p22k       (0x10000U - 41115U)
#define WTCNT_COUNT_7p81k       (0x10000U - 39062U)
```

Figure 5.2 Setting value of wait time

Note) Setting value of wait time of R-Car D3.

```
#define WTCNT_COUNT_8p13k       (0x10000U - 40760U)
```

This setting value is calculated by the Figure 5.3.

```
e.g.) wait time            : 5 [sec]
      Product ID           : H3
      CUT number           : Ver.3.0
      MD13=1, MD14=1 : OSCCLK=130.20 [kHz]

#define WTCNT_COUNT_8p13k              (0x10000U - WTCNT_COUNT)

WTCNT_COUNT = ROUND DOWN ( wait time[sec] * ( OSCCLK[Hz] / 16 ) )
            = ROUND DOWN ( 5 * (130200 / 16 ) )
            = 40687
```

Figure 5.3 Calculation formula.

To change the waiting time, refer to "1.3.1 Related Document No.2" and change the definition value of Figure 5.2 so that it becomes the desired waiting time.

5.4.3.3 SDRAM setting

Setting for access to the LPDDR4-SDRAM implemented in the R-Car H3/M3/M3N System Evaluation Board, and the DDR3L-SDRAM implemented in the R-Car E3/D3 System Evaluation Board.

The following shows file and directory structure of the SDRAM setting.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/drivers/renesas/common]

```
common
└──ddr
    │   dram_sub_func.c
    │   dram_sub_func.h
    │   boot_init_dram.h
    │   ddr.mk
    │
    ├──ddr_a
    │       boot_init_dram_regdef.h
    │       ddr_init_e3.c
    │       ddr_init_d3.c
    │       ddr_a.mk
    │
    └──ddr_b
            boot_init_dram.c
            boot_init_dram_config.c
            boot_init_dram_regdef.h
            ddr_regdef.h
            init_dram_tbl_h3.h
            init_dram_tbl_h3ver2.h
            init_dram_tbl_m3.h
            init_dram_tbl_m3n.h
            ddr_b.mk
```

SDRAM settings is performed by rcar_dram_init() function implemented in boot_init_dram.c. rcar_dram_init() performs initialization suitable for both H3 and M3.

boot_init_dram_config.c is used to configure the board.

dram_sub_func.c is implements a function that determines cold boot and warm boot.

Set the SDRAM bus clock setting corresponding to product id. SDRAM bus clock is switched by MD pin.

Shown in Table 5.22.

Table 5.22 SDRAM bus clock

Product ID	CUT number	MD19	MD17	SDRAM bus clock
H3	Ver.3.0	0	0	1600MHz(LPDDR4-3200)
M3	-	0	0	1600MHz(LPDDR4-3200)
M3N	-	0	0	1600MHz(LPDDR4-3200)
E3	-	0	-	792MHz(DDR3L-1600) *1
	-	1	-	928MHz(DDR3L-1856)
D3	-	0	-	792MHz(DDR3L-1600) *1
	-	1	-	928MHz(DDR3L-1856)

*1 1584Mbps is recommended for DDR3L-1600.

If customization is required, please based on the above.
Note) If MD pins selection is other than MD7=0, MD6=0 or MD7=0, MD6=1, SDRAM setting is skipped.

The following table shows the combinations between the DRAM configuration related build options.

Table 5.23 Combination of DRAM configuration related build options

SoC	version	DRAM config	build options (RCAR_DRAM_XXXX)			
			SPLIT	LPDDR4_MEM CONF	DDR3L_MEM CONF	DDR3L_MEM DUAL
R-Car H3	Ver.3.0	8GB (2GB x4)	1 or 3 *1 (4 channel split)	1 (2G Byte)	-	-
		4GB (1GB x4)		0 (1G Byte)	-	-
R-Car M3	Ver.3.0	8GB (4GB x2)	2 or 3 *1 (2 channel split)	-	-	-
	Ver.1.2 /1.3	4GB (2GB x2)				
R-Car M3N	Ver.1.1	4GB (4GB x1)	0 or 3 *1 (No split)	2 (4G Byte)	-	-
		2GB (2GB x1)	0 or 3 *1 (No split)	1 (2G Byte)	-	-
R-Car E3	Ver.1.1	2GB (512MB x4)		-	1 (2G Byte)	1 (4 SDRAM)
		1GB (512MB x2)		-	0 (1G Byte)	0 (2 SDRAM)
R-Car D3	Ver. 1.1	512MB (512MB x1)	0	-	-	-

*1 Refer to value 3 of RCAR_DRAM_SPLIT in Table 5.5

Supported memory configurations are determined by a combination of build options.
Please refer to chapter 7 in [1] about memory maps.
The combinations that are not explicitly listed above are not supported.

5.4.3.4 Suspend to RAM

The Suspend to RAM is implemented in the rcar_dram_init(), the bl2_early_platform_setup(), and the bl2_plat_get_bl31_ep_info().

The following shows file and directory structure of the Suspend to RAM.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/drivers/renesas/rcar]

```
rcar
 |
 ├──ddr
 |    |    dram_sub_func.c
 |    |
 |    ├──ddr_a
 |    |         ddr_init_e3.c
 |    |
 |    └──ddr_b
 |              boot_init_dram.c
 |
 | bl2_plat_setup.c
```

Immediately after booting, IPL refers to the BKUP_TRG, to check whether it is cold boot or warm boot (Suspend to RAM). The BKUP_TRG is held on PMIC, and indicates whether the system is in the Suspend to RAM state or not. Access to the BKUP_TRG is done via the GPIO and the I2C for DVFS.

In order to judge the Suspend to RAM, three functions are used: is_ddr_backup_mode(), rcar_dram_get_boot_status(), and dram_update_boot_status().

Function is_ddr_backup_mode() reads the BKUP_TRG from GPIO on the first call, a second call returns a value that was held for the first time.

Function rcar_dram_get_boot_status() looks like function as is_ddr_backup_mode(), but it reads the BKUP_TRG from GPIO by every time.

Function dram_update_boot_status() assumes that status (Cold boot = 0 / Warm boot = 1) acquired by rcar_dram_get_boot_status() is set with arguments. If 0 (Cold boot) is specified for status, it will not process anything. If 1 (Warm boot) is specified for status, change so that the BKUP_TRG indicates the Cold boot.

The BKUP_TRG is referenced from code of the SDRAM setting. it can be found out by keyword "DDR_BACKUPMODE".

Note) When using the different PMIC, then need to change this implementation. However, since this function is not subject to customization, there is no description of the procedure to change the implementation.

Note) If MD pins selection is other than MD7=0, MD6=0 or MD7=0, MD6=1, the Suspend to RAM does not work.

Note) R-Car D3 is not supported.

5.4.3.5 QoS arbitration setting

QoS arbitration setting is possible to select enabled or disabled by build option. In this clause describes how to change the Default setting.

The following shows file and directory structure of the QoS arbitration setting.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/drivers/renesas]

```
renesas
├──common
│       qos_reg.h
└──rcar
    └──qos
        │   qos.mk
        │   qos_common.h
        │   qos_init.c
        │   qos_init.h
        │
        ├──H3
        │       qos_init_h3_v10.c
        │       qos_init_h3_v10.h
        │       qos_init_h3_v11.c
        │       qos_init_h3_v11.h
        │       qos_init_h3_v20.c
        │       qos_init_h3_v20.h
        │       qos_init_h3_v20_mstat195.h
        │       qos_init_h3_v20_mstat390.h
        │       qos_init_h3_v20_qoswt195.h
        │       qos_init_h3_v20_qoswt390.h
        │       qos_init_h3_v30.c
        │       qos_init_h3_v30.h
        │       qos_init_h3_v30_mstat195.h
        │       qos_init_h3_v30_mstat390.h
        │       qos_init_h3_v30_qoswt195.h
        │       qos_init_h3_v30_qoswt390.h
        │
        ├──M3
        │       qos_init_m3_v10.c
        │       qos_init_m3_v10.h
        │       qos_init_m3_v11.c
        │       qos_init_m3_v11.h
        │       qos_init_m3_v11_mstat195.h
        │       qos_init_m3_v11_mstat390.h
        │       qos_init_m3_v11_qoswt195.h
        │       qos_init_m3_v11_qoswt390.h
        │       qos_init_m3_v30.c
        │       qos_init_m3_v30.h
        │
        ├──M3N
        │       qos_init_m3n_v10.c
        │       qos_init_m3n_v10.h
        │       qos_init_m3n_v10_mstat195.h
        │       qos_init_m3n_v10_mstat390.h
        │       qos_init_m3n_v10_qoswt195.h
        │       qos_init_m3n_v10_qoswt390.h
        │
        ├──E3
        │       qos_init_e3_v10.c
        │       qos_init_e3_v10.h
```

```
|       qos_init_e3_v10_mstat390.h
|       qos_init_e3_v10_mstat780.h
|
└──D3
        qos_init_d3.c
        qos_init_d3.h
        qos_init_d3_mstat.h
```

QoS arbitration setting is performed by qos_init() function implemented in qos_init.c.

qos_init() function is selected the initialization function by the product id. Initialization function and implemented file is shown in the Table 5.24 below.

Table 5.24 File and Function list of QoS access arbitration setting

Product ID	CUT number	Filename	Function
H3	Ver.3.0	qos_init_h3_v30.c	qos_init_h3_v30()
M3	Ver3.0	qos_init_m3_v30.c	qos_init_m3_v30()
	Ver1.3/Ver.1.2	qos_init_m3_v11.c	qos_init_m3_v11()
M3N	-	qos_init_m3n_v10.c	qos_init_m3n_v10()
E3	-	qos_init_e3_v10.c	qos_init_e3_v10()
D3	-	qos_init_d3.c	qos_init_d3()

For H3, M3, M3N, E3 and D3, the QoS data tables shown in the Figure 5.4 in respectively file of the Table 5.24. It is selected to correspond to the Build Option RCAR_QOS_TYPE and RCAR_REF_INT.

```
#if RCAR_QOS_TYPE   == RCAR_QOS_TYPE_DEFAULT
#if RCAR_REF_INT == RCAR_REF_DEFAULT
        :
#else
        :
#endif
#endif
```

Figure 5.4 QoS data tables (H3, M3, M3N, E3 and D3)

Data in the table are respectively set to the address 0xE67E0000- 0xE67E033F, 0xE67E1000- 0xE67E133F, 0xE67E2000- 0xE67E233F, 0xE67E3000- 0xE67E333F. To change the QoS arbitration setting (the Default setting), please change the file in accordance with the LSI / CUT.

Note) If MD pins selection is other than MD7=0, MD6=0 or MD7=0, MD6=1, QoS arbitration setting is skipped.

5.4.3.6 Security access protection setting

The Security protection setting determines whether to protect the access from Normal world to SRAM, SDRAM, and IPs.

The following shows file and directory structure of the Security protection setting.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/plat/renesas/common]

```
common
├──include
│     │
│     └──bl2_lifec_register.h
│
└──bl2_secure_setting.c
```

The default of the Security access protection setting to the LifeC is shown below.
Bits that has been changed from the Hardware initial value are set by the IPL to restrict access from Normal world.
For example, setting to 0 for bit19 of SEC_SEL3 means that will not be able to write from Normal world to the main memory.

Table 5.25 List of the Security access protection setting to the LifeC

Module	Register Name	Setting value	Bits that has been changed from the Hardware initial value.
LifeC	SEC_SRC	0x0000001E	Bit 0: Arm realtime core (Cortex-R7) master port.
	SEC_SEL3	0xFFF7FDFF	Bit19: AXI-Bus (Main Memory domain AXI) slave ports. Bit 9: DBSC4 register access slave ports. Note) Setting value is changed to 0x00000000, when build option is LIFEC_DBSC_PROTECT_ENABLE=0.
	SEC_SEL5	0xFFFFFFBF	Bit 6: Boot ROM slave ports.
	SEC_SEL6	0xFFFFCBFF [R-CarH3/M3/M3N] 0xFFFFFBFF [R-CarE3/D3]	Bit13: SCEG PKA (secure APB) slave ports. [R-CarH3/M3/M3N] Bit12: SCEG PKA (public APB) slave ports. [R-CarH3/M3/M3N] Bit10: SCEG Secure Core slave ports.
	SEC_SEL12	0xFFFFFFFF	Bit 3: SD host interface 3 slave ports. Bit 2: SD host interface 2 slave ports. Bit 1: SD host interface 1 slave ports. Bit 0: SD host interface 0 slave ports.
	SEC_SEL13	0xFFBFFFFF	Bit22: RPC slave ports.
	SEC_SEL14	0xF3FFFFFF	Bit27: System Timer (SCMT) slave ports. Bit26: System Watchdog Timer (SWDT) slave ports.
	SEC_SEL15	0xFFFFFF3F	Bit13: RST slave ports. Bit 7: Life Cycle 0 slave ports.
	SECGRP0CR2	0x00020000	Bit17: SCEG Secure Core master ports. (Security Group 3)
	SECGRP1CR2	0x00020000	Bit17: SCEG Secure Core master ports. (Security Group 3)
	SECGRP0CR3	0x00003780	Bit13: SYS-DMAC (0) (MEM) master ports. (Security Group 3) Bit12: SYS-DMAC (0) (PERI, BUS) master ports. (Security Group 3) Bit10: SYS-DMAC (2) (PERI, BUS) master ports. (Security Group 3) Bit 9: SYS-DMAC (1) (PERI, BUS) master ports. (Security Group 3) Bit 7: SYS-DMAC (2) (MEM) master ports. (Security Group 3) Bit 8: SYS-DMAC (2) (MEM) master ports. (Security Group 3)
	SECGRP1CR3	0x00003780	Bit13: SYS-DMAC (0) (MEM) master ports. (Security Group 3) Bit12: SYS-DMAC (0) (PERI, BUS) master ports. (Security Group 3) Bit10: SYS-DMAC (2) (PERI, BUS) master ports. (Security Group 3) Bit 9: SYS-DMAC (1) (PERI, BUS) master ports. (Security Group 3) Bit 7: SYS-DMAC (2) (MEM) master ports. (Security Group 3) Bit 8: SYS-DMAC (2) (MEM) master ports. (Security Group 3)

Module	Register Name	Setting value	Bits that has been changed from the Hardware initial value.
	SECGRP0COND3	0x00080200	Bit19: AXI-Bus (Main Memory domain AXI) slave ports. (Security Group 3) Bit 9: DBSC4 register access slave ports. (Security Group 3) Note) Setting value is changed to 0x00000000, when build option is LIFEC_DBSC_PROTECT_ENABLE=0.
	SECGRP1COND3	0x00080200	Bit19: AXI-Bus (Main Memory domain AXI) slave ports. (Security Group 3) Bit 9: DBSC4 register access slave ports. (Security Group 3) Note) Setting value is changed to 0x00000000, when build option is LIFEC_DBSC_PROTECT_ENABLE=0.
	SECGRP0COND5	0x00000040	Bit 6: Boot ROM slave ports. (Security Group 3)
	SECGRP1COND5	0x00000040	Bit 6: Boot ROM slave ports. (Security Group 3)
	SECGRP0COND6	0x00003400 [R-CarH3/M3/M3N] 0x00000400 [R-CarE3/D3]	Bit13: SCEG PKA (secure APB) slave ports. (Security Group 3) [R-CarH3/M3/M3N] Bit12: SCEG PKA (public APB) slave ports. (Security Group 3) [R-CarH3/M3/M3N] Bit10: SCEG Secure Core slave ports. (Security Group 3)
	SECGRP1COND6	0x00003400 [R-CarH3/M3/M3N] 0x00000400 [R-CarE3/D3]	Bit13: SCEG PKA (secure APB) slave ports. (Security Group 3) [R-CarH3/M3/M3N] Bit12: SCEG PKA (public APB) slave ports. (Security Group 3) [R-CarH3/M3/M3N] Bit10: SCEG Secure Core slave ports. (Security Group 3)
	SECGRP0COND13	0x00400000	Bit22: RPC slave ports. (Security Group 3)
	SECGRP1COND13	0x00400000	Bit22: RPC slave ports. (Security Group 3)
	SECGRP0COND14	0x0C000000	Bit26: System Timer (SCMT) slave ports. (Security Group 3) Bit27: System Watchdog Timer (SWDT) slave ports. (Security Group 3)
	SECGRP1COND14	0x0C000000	Bit26: System Timer (SCMT) slave ports. (Security Group 3) Bit27: System Watchdog Timer (SWDT) slave ports. (Security Group 3)
	SECGRP0COND15	0x000000C0	Bit13: RST slave ports. (Security Group 3) Bit 7: Life Cycle 0 slave ports. (Security Group 3) Bit 6: TDBG (is described in Debug and Trace section) slave ports. (Security Group 3)
	SECGRP1COND15	0x000000C0	Bit13: RST slave ports. (Security Group 3) Bit 7: Life Cycle 0 slave ports. (Security Group 3) Bit 6: TDBG (is described in Debug and Trace section) slave ports. (Security Group 3)

Note) If MD pins selection is MD7 = 1, MD6 = 1, register setting of LifeC is skipped.

The default of the Security access protection setting to the AXI-bus is shown below.
These are set by the IPL to restrict access from Normal world to the SRAM protected area / the SDRAM protected area.
The SRAM protected area is set to 0xE6303FFF from 0xE6300000. (Logical address)
And the SDRAM protected area is set to 0x47DFFFFF from 0x43F00000. (Logical address)

Table 5.26 List of the Security access protection setting for AXI-bus

Module	Register Name	Setting value	Comment
AXI-bus	DPTDIVCR0	0x0E0403F0	DRAM Protected Area Division #0 Protection area division physical address is H'04_03F00000
	DPTDIVCR1	0x0E0407E0	DRAM Protected Area Division #1 Protection area division physical address is H'04_07E00000
	DPTDIVCR2	0x0E080000	DRAM Protected Area Division #2 Protection area division physical address is H'08_00000000
	DPTDIVCR3	0x0E080000	DRAM Protected Area Division #3 Protection area division physical address is H'08_00000000
	DPTDIVCR4	0x0E080000	DRAM Protected Area Division #4 Protection area division physical address is H'08_00000000

Module	Register Name	Setting value	Comment
	DPTDIVCR5	0x0E080000	DRAM Protected Area Division #5 Protection area division physical address is H'08_00000000
	DPTDIVCR6	0x0E080000	DRAM Protected Area Division #6 Protection area division physical address is H'08_00000000
	DPTDIVCR7	0x0E080000	DRAM Protected Area Division #7 Protection area division physical address is H'08_00000000
	DPTDIVCR8	0x0E080000	DRAM Protected Area Division #8 Protection area division physical address is H'08_00000000
	DPTDIVCR9	0x0E080000	DRAM Protected Area Division #9 Protection area division physical address is H'08_00000000
	DPTDIVCR10	0x0E080000	DRAM Protected Area Division #10 Protection area division physical address is H'08_00000000
	DPTDIVCR11	0x0E080000	DRAM Protected Area Division #11 Protection area division physical address is H'08_00000000
	DPTDIVCR12	0x0E080000	DRAM Protected Area Division #12 Protection area division physical address is H'08_00000000
	DPTDIVCR13	0x0E080000	DRAM Protected Area Division #13 Protection area division physical address is H'08_00000000
	DPTDIVCR14	0x0E080000	DRAM Protected Area Division #14 Protection area division physical address is H'08_00000000
	DPTCR0	0x0E000000	DRAM Protected Area Setting #0 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR1	0x0E000E0E	DRAM Protected Area Setting #1 The Security Group 3 has the privilege to write the relevant DRAM area.
	DPTCR2	0x0E000000	DRAM Protected Area Setting #2 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR3	0x0E000000	DRAM Protected Area Setting #3 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR4	0x0E000000	DRAM Protected Area Setting #4 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR5	0x0E000000	DRAM Protected Area Setting #5 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR6	0x0E000000	DRAM Protected Area Setting #6 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR7	0x0E000000	DRAM Protected Area Setting #7 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR8	0x0E000000	DRAM Protected Area Setting #8 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR9	0x0E000000	DRAM Protected Area Setting #9 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR10	0x0E000000	DRAM Protected Area Setting #10 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR11	0x0E000000	DRAM Protected Area Setting #11 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR12	0x0E000000	DRAM Protected Area Setting #12 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR13	0x0E000000	DRAM Protected Area Setting #13 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR14	0x0E000000	DRAM Protected Area Setting #14

Module	Register Name	Setting value	Comment
			All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR15	0x0E000000	DRAM Protected Area Setting #15 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	SPTDIVCR0	0x0E0E6304	System RAM Protected Area Division #0 Protection area division physical address is H'E6304000
	SPTDIVCR1	0x0E0E6360	System RAM Protected Area Division #1 Protection area division physical address is H'E6360000
	SPTDIVCR2	0x0E0E6360	System RAM Protected Area Division #2 Protection area division physical address is H'E6360000
	SPTDIVCR3	0x0E0E6360	System RAM Protected Area Division #3 Protection area division physical address is H'E6360000
	SPTDIVCR4	0x0E0E6360	System RAM Protected Area Division #4 Protection area division physical address is H'E6360000
	SPTDIVCR5	0x0E0E6360	System RAM Protected Area Division #5 Protection area division physical address is H'E6360000
	SPTDIVCR6	0x0E0E6360	System RAM Protected Area Division #6 Protection area division physical address is H'E6360000
	SPTDIVCR7	0x0E0E6360	System RAM Protected Area Division #7 Protection area division physical address is H'E6360000
	SPTDIVCR8	0x0E0E6360	System RAM Protected Area Division #8 Protection area division physical address is H'E6360000
	SPTDIVCR9	0x0E0E6360	System RAM Protected Area Division #9 Protection area division physical address is H'E6360000
	SPTDIVCR10	0x0E0E6360	System RAM Protected Area Division #10 Protection area division physical address is H'E6360000
	SPTDIVCR11	0x0E0E6360	System RAM Protected Area Division #11 Protection area division physical address is H'E6360000
	SPTDIVCR12	0x0E0E6360	System RAM Protected Area Division #12 Protection area division physical address is H'E6360000
	SPTDIVCR13	0x0E0E6360	System RAM Protected Area Division #13 Protection area division physical address is H'E6360000
	SPTDIVCR14	0x0E0E6360	System RAM Protected Area Division #14 Protection area division physical address is H'E6360000
	SPTCR0	0x0E000E0E	System RAM Protected Area Setting #0 The Security Group 3 has the privilege to write the relevant System RAM area.
	SPTCR1	0x0E000000	System RAM Protected Area Setting #1 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR2	0x0E000000	System RAM Protected Area Setting #2 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR3	0x0E000000	System RAM Protected Area Setting #3 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR4	0x0E000000	System RAM Protected Area Setting #4 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR5	0x0E000000	System RAM Protected Area Setting #5 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR6	0x0E000000	System RAM Protected Area Setting #6 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR7	0x0E000000	System RAM Protected Area Setting #7 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR8	0x0E000000	System RAM Protected Area Setting #8 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR9	0x0E000000	System RAM Protected Area Setting #9

Module	Register Name	Setting value	Comment
			All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR10	0x0E000000	System RAM Protected Area Setting #10 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR11	0x0E000000	System RAM Protected Area Setting #11 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR12	0x0E000000	System RAM Protected Area Setting #12 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR13	0x0E000000	System RAM Protected Area Setting #13 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR14	0x0E000000	System RAM Protected Area Setting #14 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.
	SPTCR15	0x0E000000	System RAM Protected Area Setting #15 All of the Security Group (0-3) has the privilege to write the relevant System RAM area.

Note) If MD pins selection is MD7 = 1, MD6 = 1, register setting of AXI-bus is skipped.

The Security access protection setting parameter tables are shown below.

```
static const struct {
        uint32_t adr;
        uint32_t val;
} lifec [] = {
        /** LIFEC0 (SECURITY) settings                                  */
        /* Security attribute setting for master ports                  */
        /* Bit 0: ARM realtime core (Cortex-R7) master port             */
        /*         0: Non-Secure                                        */
        {
        SEC_SRC, 0x0000001EU}, …
};
/* AXI settings */

static const struct {
        uint32_t adr;
        uint32_t val;
};
static const struct axi_t axi[] = {
            /* SRAM ptotection                          */
            /* AXI sram protected area division        */
        {
        AXI_SPTDIVCR0,   0x0E0E6304U}, {
        AXI_SPTDIVCR1,   0x0E0E6360U}, {
        AXI_SPTDIVCR2,   0x0E0E6360U}, {
        AXI_SPTDIVCR3,   0x0E0E6360U}, {
        AXI_SPTDIVCR4,   0x0E0E6360U}, {
        AXI_SPTDIVCR5,   0x0E0E6360U}, {
        AXI_SPTDIVCR6,   0x0E0E6360U}, {
        AXI_SPTDIVCR7,   0x0E0E6360U}, {
        AXI_SPTDIVCR8,   0x0E0E6360U}, {
        AXI_SPTDIVCR9,   0x0E0E6360U}, …
};
```

Figure 5.5 Table of the Security access protection setting

If want to change the Security access protection setting, please refer to clause 15.3.4 -Access protection for secure/safety regions-, and section 68.3 -Operation- in “1.3.1 Related Document No.2” R-Car Series, 3rd Generation User's Manual: Hardware, and please edit the bl2_secure_setting.c.

5.4.3.7 Image loading

In an environment that provided by R-Car H3/M3/M3N/E3/D3, IPL is loading image of BL31 and BL32, BL33#1-#8 from HyperFlash / QSPI Flash / eMMC to SDRAM only.

In this clause, describes the image load from HyperFlash / QSPI Flash shown in the following Figure 5.6.

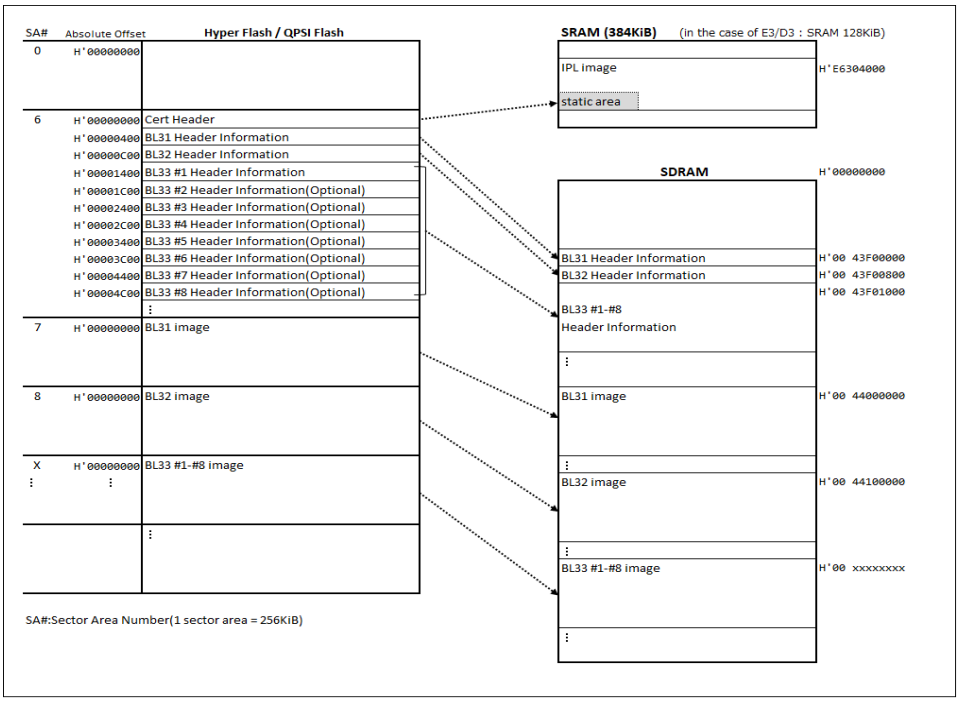


Figure 5.6 Memory Map

IPL supports the following features.

- Load data from HyperFlash / QSPI Flash using DMA.
- Unit of transfer size is 256bytes. It is necessary to align the address of the load image to 256bytes.

This function is supported that the source address and destination address of the image loading is variably. IPL getting the source address and destination address and image size by reading the image load information (Cert Header and Header information).

The load image information is stored in SA6 of flash.

The load image information is created by the layout_create (see clause 5.4.3.9).

Note) In the Yocto 2.12.0 or later, changed the format of the load image information (memory map on SA6 area). Therefore, the image that was created, please use a combination of the same version.

Address on the Flash Memory of BL31 and BL32, BL33(BL33 #1 - #8) refers to the “Offset of address” of Cert Header. Details of the Cert Header shown in the following Figure 5.7.

The “Offset of address” is setting the offset from the beginning of SA0 of the flash memory.

Each field of the Cert Header is 64-bit little endian.

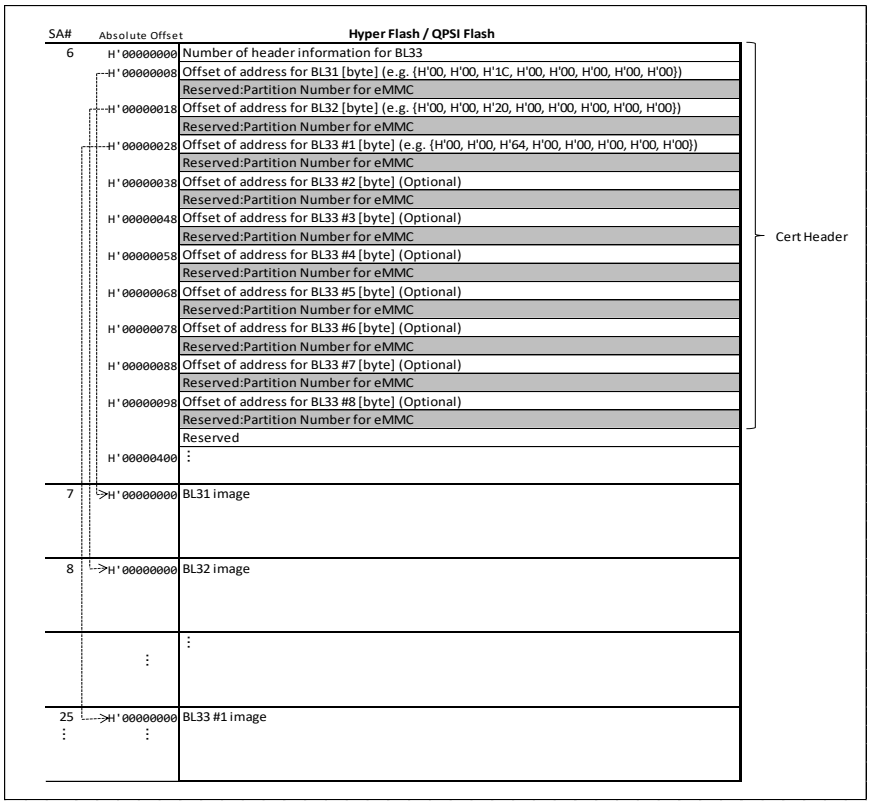


Figure 5.7 Cert Header

The destination address and image size getting from Header Information. The structure of the Header Information for BL31 and BL32, BL33(BL33 #1-#8) shown in the following Figure 5.8.

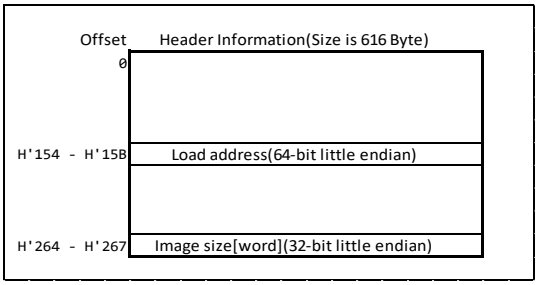


Figure 5.8 Header Information

IPL checks the range of source address (“Offset of address” of Figure 5.7) and size (“Image size” of Figure 5.8) of the load image.

To execute the range check, set the Flash Memory capacity to FLASH_MEMORY_SIZE defined in rcar_def.h.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/plat/renesas/common/include]

```
include
  rcar_def.h
```

rcar_def.h

```
#define FLASH_MEMORY_SIZE              U(0x04000000)    /* hyper flash */
```

The definition of the boundary on the flash memory is shown in Figure 5.9.

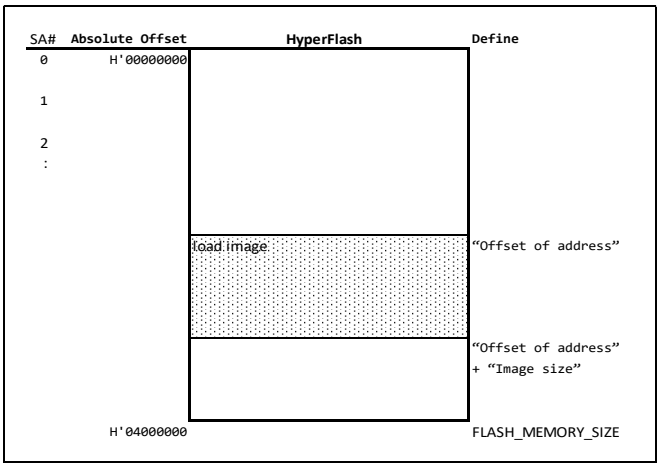


Figure 5.9 Defining bounds of range check

IPL executes the loading in the following condition.

“Offset of address” + “Image size” – 1 < FLASH_MEMORY_SIZE

If the load image is out of the range, the loading is not executed and it is an error.

IPL checks the range of destination address (“Load address” of Figure 5.8) and size (“Image size” of Figure 5.8) of the load image.

The following explains the range check of BL31, BL32 and BL33#1-#8.

The range in which BL31, BL32 and BL33 #1-#8 can be loaded is shown in Figure 5.10.

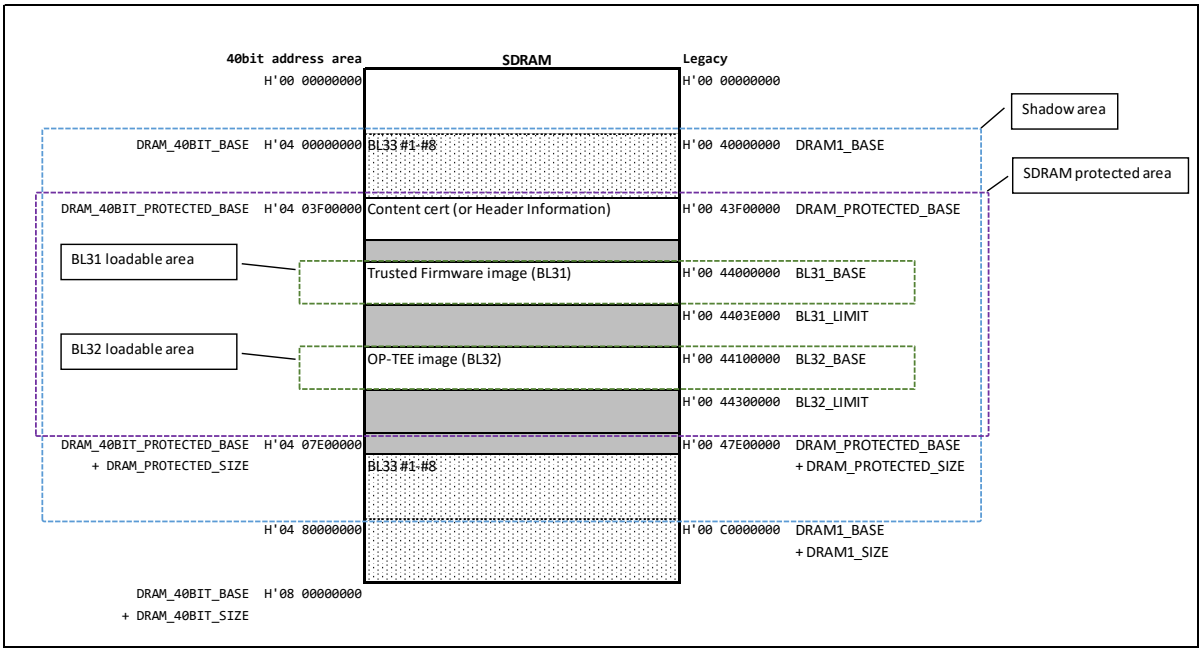


Figure 5.10 Range of valid address to load of BL31, BL32 and BL33 #1-#8

The range check of BL31 and BL32 is executed with the following function.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/platform/renesas]

```
rcar
 |    bl2_plat_setup.c
```

```
static uint64_t check_secure_load_area(uintptr_t base, uint32_t size,
                        uintptr_t dest, uint32_t len)
{
        :
        :
        return 1;    /* if error detected */
        :
        return 0;    /* if no error detected */
}
```

The range check of BL33 #1-#8 is executed with the following function.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/drivers/renesas/common]

```
io
 |   io_rcar.c
```

```
static int32_t check_load_area(uintptr_t dst, uintptr_t len)
{
                    :
                    :
        return result;
}
```

Note) When customizing range check, be aware of destination address and size integer overflow.

The boundary definition of the area shown in Figure 5.10 is define in the following source file.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/plat/renesas/common]

```
include
 |   rcar_def.h
 |   platform_def.h
```

rcar_def.h

```
#define     DRAM1_BASE                  U(0x40000000)
#define     DRAM1_SIZE                  U(0x80000000)

#define     DRAM_40BIT_BASE             ULL(0x0400000000)
#define     DRAM_40BIT_SIZE             ULL(0x0400000000)

#define     DRAM_PROTECTED_BASE         ULL(0x43F00000)
#define     DRAM_40BIT_PROTECTED_BASE   ULL(0x0403F00000)
#define     DRAM_PROTECTED_SIZE         ULL(0x03F00000)

#define     RCAR_TRUSTED_SRAM_BASE        0x44000000
#define     RCAR_TRUSTED_SRAM_SIZE        0x0003E000
```

Note) Even if you change this setting, it does not affect the setting of the SDRAM protected area (refer to clause 5.4.3.6).

platform_def.h

```
#define BL31_BASE               (RCAR_TRUSTED_SRAM_BASE)
#define BL31_LIMIT              (RCAR_TRUSTED_SRAM_BASE + \
                                  RCAR_TRUSTED_SRAM_SIZE)

#define BL32_BASE               U(0x44100000)
#define BL32_LIMIT              (BL32_BASE + U(0x200000))
```

Note) Please notice that change these addresses, because the execution address in BL31 and BL32 are hard-coded.

According to the definition in Figure 5.10, the loading range of BL31 and BL32 to be checked is shown in the following Table 5.27.

Table 5.27 Check the loading range of BL31 and BL32

	Top address	Bottom address
BL31	BL31_BASE (0x44000000)	BL31_LIMIT (0x4403DFFF)
BL32	BL32_BASE (0x44100000)	BL32_LIMIT (0x442FFFFF)

According to the definition in Figure 5.10, the loading range of BL33 #1-# 8 to be checked is shown in the following Table 5.28.

Table 5.28 Check the loading range of BL33 #1-#8

	Top address	Bottom address
Legacy area	DRAM1_BASE	DRAM_PROTECTED_BASE - 1
	DRAM_PROTECTED_BASE + DRAM_PROTECTED_SIZE	DRAM1_BASE + DRAM1_SIZE - 1
40bit address area	DRAM_40BIT_BASE	DRAM_40BIT_PROTECTED_BASE - 1
	DRAM_40BIT_PROTECTED_BASE + DRAM_PROTECTED_SIZE	DRAM_40BIT_BASE + DRAM_40BIT_SIZE - 1

5.4.3.8 Image loading (eMMC)

In this clause, describes the image load from eMMC shown in the following Figure 5.11, it is described only the functions different from Clause5.4.3.7.

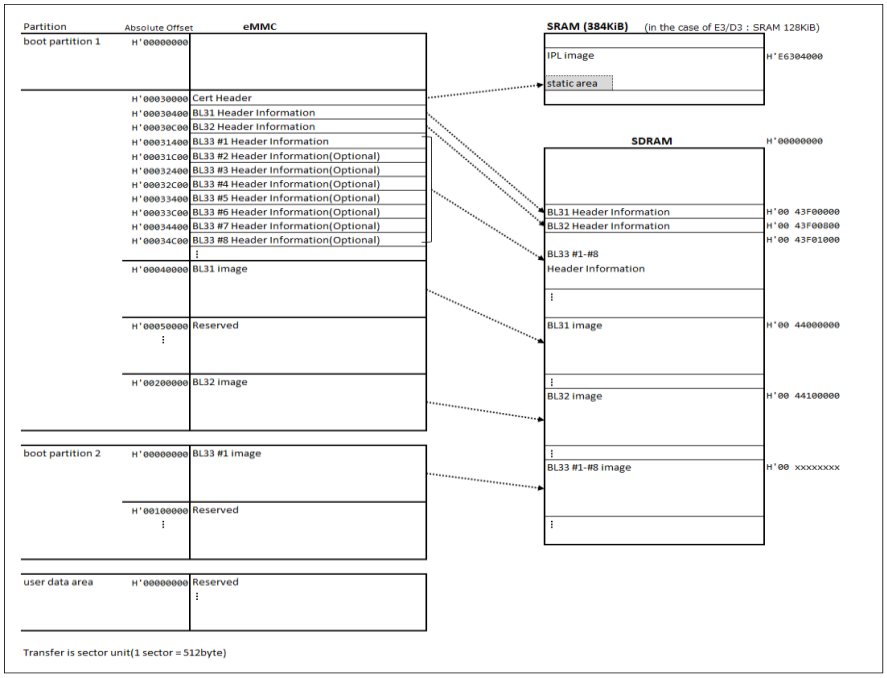


Figure 5.11 Memory Map for eMMC

IPL supports the following features.

- High-speed Single Data Rate mode 25MHz and 50MHz : Select to setting of high frequency by eMMC supports. Read the DEVICE_TYPE[196] field in the EXT_CSD register of eMMC(refer to "1.3.1 Related Document No.3") to validate whether the IPL supports 25MHz or 50Mhz.
- Only 8-bits bus width available.
- Use the DMA transfer : DMA transfer by setting the DMAC that have been implemented in the SD Host Interface(refer to "1.3.1 Related Document No.2").
- Available access partition : boot partition 1 / boot partition 2 / user data area
- Unit of transfer size is 512byte.
- Address of the image is 512byte-aligned: eMMC is accessed by the transfer size unit(512byte).
- Range check of the source address: When the eMMC is requested for an out of range address with the block read command, ADDRESS_OUT_OF_RANGE response is returned. IPL uses this as the range check of the source address.
- Loadable image size is up to (4GiB – 512 byte).

The storage location on the eMMC of BL31 and BL32, BL33(BL33 #1 - #8) refers to “Offset of address” and "Partition Number" in Cert Header shown in the following Figure 5.12.

Each field of the Cert Header is 64-bit little endian.

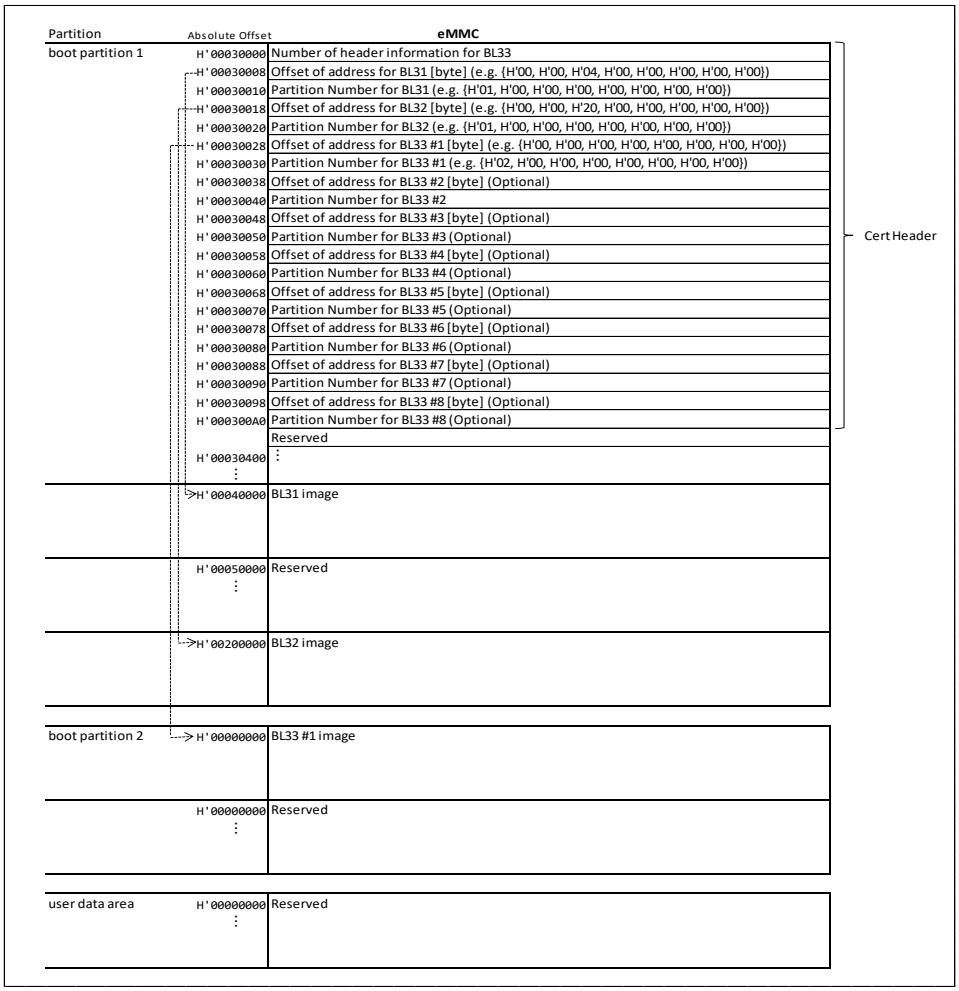


Figure 5.12 Cert Header for eMMC

“Offset of address” is setting the offset from the beginning of the partition in eMMC.

"Partition Number" set the value in below Table 5.29.

Table 5.29 Value of Partition Number

Value	Partition
0	user data area
1	boot partition 1
2	boot partition 2
3 (Other : same as ‘3’)	boot partition 1 or 2 Boot from the same boot partition as the boot partition that Boot ROM used for booting. (refer to chapter 19.2.5 of the Table 1.2 Related Document[2])

Header information is the same as the image loading of HyperFlash / QSPI Flash. Refer to Figure 5.8 and its description.

The Cert Header and the Header information of eMMC boot is created by the layout_create (see clause 5.4.3.9).

5.4.3.9 layout_create setting

The layout_create has been prepared to create the information for image loading.

The following shows file and directory structure of the layout_create configuration.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/tools/]

```
tools
└──renesas
    └──rcar_layout_create
        │   makefile
        │   sa0.c
        │   sa0.ld.S
        │   sa6.c
        │   sa6.ld.S
```

The following files are created by building the layout_create.

- bootparam_sa0.srec
- cert_header_sa6.srec

bootparam_sa0.srec includes the following information. (refer to chapter 19 of the Table 1.2 Related Document[2])

- Boot ROM Parameters
- Load destination address of IPL
- IPL image size[word]

The structure on the flash memory of bootparam_sa0.srec for HyperFlash / QSPI Flash boot is shown in the following Figure 5.13.

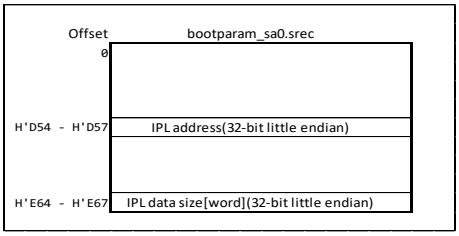


Figure 5.13 Structure of bootparam_sa0.srec for HyperFlash /QSPI Flash

The structure on the flash memory of bootparam_sa0.srec for eMMC boot is shown in the following Figure 5.14.

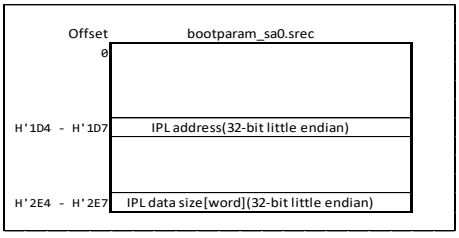


Figure 5.14 Structure of bootparam_sa0.srec for eMMC

To change the information of bootparam_sa0.srec will edit the sa0.c and sa0.ld.S.

sa0.c sets the values included in bootparam_sa0.srec to each variable. Each section is defined to the variables.

sa0.ld.S sets the offset address according to the implementation of MaskROM based on the section defined in sa0.c.

Show the code of sa0.c below.

```
/* SA0 */
/* 0x00000000 */
const unsigned int __attribute__ ((section (".sa0_bootrom"))) bootrom_paramA = 0x00000100;
/* 0x00000080 */
/* 0x000001D4 */
const unsigned int __attribute__ ((section (".sa0_bl2dst_addr3 "))) bl2dst_addr3 = BL2_ADDRESS;
/* 0x000002E4 */
const unsigned int __attribute__ ((section (".sa0_bl2dst_size3 "))) bl2dst_size3 = BL2_SIZE;
/* 0x00000C00 */
/* 0x00000D54 */
const unsigned int __attribute__ ((section (".sa0_bl2dst_addr1 "))) bl2dst_addr1 = BL2_ADDRESS;
/* 0x00000E64 */
const unsigned int __attribute__ ((section (".sa0_bl2dst_size1 "))) bl2dst_size1 = BL2_SIZE;
```

IPL image size is switched by build option of RCAR_SA0_SIZE. Refer to Section 5.3.

Table 5.30 Description of variable in BL2_ADDRESS and BL2_SIZE

Product ID	BL2_ADDRESS	BL2_SIZE	Description
H3,M3,M3N	0xE6304000	0x0000AA00	BL2 size is 170KB(170*1024/4)
E3,D3	0xE6304000	0x00005000	BL2 size is 80KB(80*1024/4)

Description of implemented variable in the sa0.c shown in the following Table 5.31.

Table 5.31 Description of variable in sa0.c

sa0.c			
Variable	**Default value**	**Section**	**Description**
bootrom_paramA	0x00000100	.sa0_bootrom	Boot ROM Parameters
bl2dst_addr3	0xE6304000	.sa0_bl2dst_addr3	Load destination address of IPL for eMMC boot
bl2dst_size3	0x0000AA00 [R-CarH3/M3/M3N] 0x00005000 [R-CarE3/D3]	.sa0_bl2dst_size3	IPL image size[word] for eMMC boot
bl2dst_addr1	0xE6304000	.sa0_bl2dst_addr1	Load destination address of IPL for HyperFlash / QSPI Flash boot
bl2dst_size1	0x0000AA00 [R-CarH3/M3/M3N] 0x00005000 [R-CarE3/D3]	.sa0_bl2dst_size1	IPL image size[word] for HyperFlash / QSPI Flash boot

Note) Image size is set in a “word” unit. (e.g. image size is : 0x0000AA00[word] = 0x0002A800[byte])

Note) The address of bl2dst_addr3 and bl2dst_addr1 needs to match the address defined in BL2_BASE.

Note) Do not to change the loaded destination because can be specified only System RAM.

Show the code of sa0.ld.S below.

```
SECTIONS
{
        . = 0x00000000;
        .rodata : {
                KEEP(*(.sa0_bootrom))
                /* Map Type 3 for eMMC Boot */
                /* A-side IPL content cert "Start Address" */
                . = 0x000001D4;         /* H'00000080 + H'00000154 */
                KEEP(*(.sa0_bl2dst_addr3))
                /* A-side IPL content cert "Size" */
                . = 0x000002E4;         /* H'00000080 + H'00000264 */
                KEEP(*(.sa0_bl2dst_size3))
                /* Map Type 1 for HyperFlash/QSPI Flash Boot */
                /* A-side IPL content cert "Start Address" */
                . = 0x00000D54;         /* H'00000C00 + H'00000154 */
                KEEP(*(.sa0_bl2dst_addr1))
                /* A-side IPL content cert "Size" */
                . = 0x00000E64;         /* H'00000C00 + H'00000264 */
                KEEP(*(.sa0_bl2dst_size1))
        }

}
```

Description of implemented section in the sa0.ld.S shown in the following Table 5.32.

Table 5.32 Description of section in sa0.ld.S

sa0.ld.S		
Section	**offset address**	**Description**
.sa0_bootrom	0x00000000	Boot ROM Parameters
.sa0_bl2dst_addr3	0x000001D4	Load destination address of IPL for eMMC boot
.sa0_bl2dst_size3	0x000002E4	IPL image size[word] for eMMC boot
.sa0_bl2dst_addr1	0x00000D54	Load destination address of IPL for HyperFlash / QSPI Flash boot
.sa0_bl2dst_size1	0x00000E64	IPL image size[word] for HyperFlash / QSPI Flash boot

The cert_header_sa6.srec contains information in the written data to the load image information (Cert Header and Header Information) shown below.

- Cert Header : Number of header information for BL33
- Cert Header : Offset of address for BL31
- Cert Header : Offset of address for BL32
- Cert Header : Offset of address for BL33 #1-#8
- Header Information(BL31) : Load address
- Header Information(BL31) : Image size[word]
- Header Information(BL32) : Load address
- Header Information(BL32) : Image size[word]
- Header Information(BL33 #1-#8) : Load address
- Header Information(BL33 #1-#8) : Image size[word]

To change the information of cert_header_sa6.srec will edit sa6.c and sa6.ld.S.

The value of the Cert Header and the Header Information have been implemented in sa6.c. Each value is defined, and each value of defined is assigned to a section.

sa6.ld.S is setting an offset address to the section defined by the sa6.c in accordance with the memory map on the load image information. (see Figure 5.6, Figure 5.7, Figure 5.8)

sa6.c is shared by the Hyper Flash / QPSI Flash and eMMC. Creating cert_header_sa6.srec is switched by build option of RCAR_SA6_TYPE. Refer to Section 5.3.

Description of defined value in the sa6.c are :

For Hyper Flash / QPSI Flash, shown in the following Table 5.33.

For eMMC, shown in the following Table 5.34.

Table 5.33 Defined value in sa6.c for Hyper Flash / QPSI Flash

sa6.c		
define	**Default value**	**Description**
RCAR_IMAGE_NUM	0x00000001	Cert Header : Number of header information for BL33
RCAR_BL31SRC_ADDRESS	0x001C0000	Cert Header : Offset of address for BL31
RCAR_BL31_PARTITION	0x00000000	Reserved
RCAR_BL32SRC_ADDRESS	0x00200000	Cert Header : Offset of address for BL32
RCAR_BL32_PARTITION	0x00000000	Reserved
RCAR_BL33SRC_ADDRESS	0x00640000	Cert Header : Offset of address for BL33 #1
RCAR_BL33_PARTITION	0x00000000	Reserved
RCAR_BL332SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #2
RCAR_BL332_PARTITION	0x00000000	Reserved
RCAR_BL333SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #3
RCAR_BL333_PARTITION	0x00000000	Reserved
RCAR_BL334SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #4
RCAR_BL334_PARTITION	0x00000000	Reserved
RCAR_BL335SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #5
RCAR_BL335_PARTITION	0x00000000	Reserved
RCAR_BL336SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #6
RCAR_BL336_PARTITION	0x00000000	Reserved
RCAR_BL337SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #7
RCAR_BL337_PARTITION	0x00000000	Reserved
RCAR_BL338SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #8

RCAR_BL338_PARTITION	0x00000000	Reserved
RCAR_BL31DST_ADDRESS	0x44000000	Header Information(BL31) : Load address(Low)
RCAR_BL31DST_ADDRESSH	0x00000000	Header Information(BL31) : Load address(High)
RCAR_BL31DST_SIZE	0x00004000	Header Information(BL31) : Image size[word]
RCAR_BL32DST_ADDRESS	0x44100000	Header Information(BL32) : Load address(Low)
RCAR_BL32DST_ADDRESSH	0x00000000	Header Information(BL32) : Load address(High)
RCAR_BL32DST_SIZE	0x00080000	Header Information(BL32) : Image size[word]
RCAR_BL33DST_ADDRESS	0x50000000	Header Information(BL33 #1) : Load address(Low)
RCAR_BL33DST_ADDRESSH	0x00000000	Header Information(BL33 #1) : Load address(High)
RCAR_BL33DST_SIZE	0x00040000	Header Information(BL31 #1) : Image size[word]
RCAR_BL332DST_ADDRESS	0x00000000	Header Information(BL33 #2) : Load address(Low)
RCAR_BL332DST_ADDRESSH	0x00000000	Header Information(BL33 #2) : Load address(High)
RCAR_BL332DST_SIZE	0x00000000	Header Information(BL33 #2) : Image size[word]
RCAR_BL333DST_ADDRESS	0x00000000	Header Information(BL33 #3) : Load address(Low)
RCAR_BL333DST_ADDRESSH	0x00000000	Header Information(BL33 #3) : Load address(High)
RCAR_BL333DST_SIZE	0x00000000	Header Information(BL33 #3) : Image size[word]
RCAR_BL334DST_ADDRESS	0x00000000	Header Information(BL33 #4) : Load address(Low)
RCAR_BL334DST_ADDRESSH	0x00000000	Header Information(BL33 #4) : Load address(High)
RCAR_BL334DST_SIZE	0x00000000	Header Information(BL33 #4) : Image size[word]
RCAR_BL335DST_ADDRESS	0x00000000	Header Information(BL33 #5) : Load address(Low)
RCAR_BL335DST_ADDRESSH	0x00000000	Header Information(BL33 #5) : Load address(High)
RCAR_BL335DST_SIZE	0x00000000	Header Information(BL33 #5) : Image size[word]
RCAR_BL336DST_ADDRESS	0x00000000	Header Information(BL33 #6) : Load address(Low)
RCAR_BL336DST_ADDRESSH	0x00000000	Header Information(BL33 #6) : Load address(High)
RCAR_BL336DST_SIZE	0x00000000	Header Information(BL33 #6) : Image size[word]
RCAR_BL337DST_ADDRESS	0x00000000	Header Information(BL33 #7) : Load address(Low)
RCAR_BL337DST_ADDRESSH	0x00000000	Header Information(BL33 #7) : Load address(High)
RCAR_BL337DST_SIZE	0x00000000	Header Information(BL33 #7) : Image size[word]
RCAR_BL338DST_ADDRESS	0x00000000	Header Information(BL33 #8) : Load address(Low)
RCAR_BL338DST_ADDRESSH	0x00000000	Header Information(BL33 #8) : Load address(High)
RCAR_BL338DST_SIZE	0x00000000	Header Information(BL33 #8) : Image size[word]

Note) Image size is set in a “word” unit. (e.g. the image size of the BL33 #1 is : 0x00040000[word] = 0x00100000[byte])

Note) Please notice that change the destination address(bl31cert_addr and bl32cert_addr) of BL31 and BL32, because the execution address in BL31 and BL32 side are hard-coded.

Note) Load address of Header Information is represented by 64-bit and set to divide in the high word(bit[63:32]) and the low word(bit[31:0]).

Note) “RCAR_BLxxx_PARTITION” is reserved in the case of Hyper Flash / QPSI Flash load.

Table 5.34 Description of defined value in sa6.c for eMMC

sa6.c		
define	**Default value**	**Description**
RCAR_IMAGE_NUM	0x00000001	Cert Header : Number of header information for BL33
RCAR_BL31SRC_ADDRESS	0x00040000	Cert Header : Offset of address for BL31
RCAR_BL31_PARTITION	0x00000001	Cert Header : Partition number for BL31
RCAR_BL32SRC_ADDRESS	0x00200000	Cert Header : Offset of address for BL32
RCAR_BL32_PARTITION	0x00000001	Cert Header : Partition number for BL32
RCAR_BL33SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #1
RCAR_BL33_PARTITION	0x00000002	Cert Header : Partition number for BL33 #1
RCAR_BL332SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #2
RCAR_BL332_PARTITION	0x00000000	Cert Header : Partition number for BL33 #2
RCAR_BL333SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #3
RCAR_BL333_PARTITION	0x00000000	Cert Header : Partition number for BL33 #3
RCAR_BL334SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #4
RCAR_BL334_PARTITION	0x00000000	Cert Header : Partition number for BL33 #4
RCAR_BL335SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #5
RCAR_BL335_PARTITION	0x00000000	Cert Header : Partition number for BL33 #5
RCAR_BL336SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #6
RCAR_BL336_PARTITION	0x00000000	Cert Header : Partition number for BL33 #6
RCAR_BL337SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #7
RCAR_BL337_PARTITION	0x00000000	Cert Header : Partition number for BL33 #7
RCAR_BL338SRC_ADDRESS	0x00000000	Cert Header : Offset of address for BL33 #8
RCAR_BL338_PARTITION	0x00000000	Cert Header : Partition number for BL33 #8
RCAR_BL31DST_ADDRESS	0x44000000	Header Information(BL31) : Load address(Low)
RCAR_BL31DST_ADDRESSH	0x00000000	Header Information(BL31) : Load address(High)
RCAR_BL31DST_SIZE	0x00004000	Header Information(BL31) : Image size[word]
RCAR_BL32DST_ADDRESS	0x44100000	Header Information(BL32) : Load address(Low)
RCAR_BL32DST_ADDRESSH	0x00000000	Header Information(BL32) : Load address(High)
RCAR_BL32DST_SIZE	0x00040000	Header Information(BL32) : Image size[word]
RCAR_BL33DST_ADDRESS	0x50000000	Header Information(BL33 #1) : Load address(Low)
RCAR_BL33DST_ADDRESSH	0x00000000	Header Information(BL33 #1) : Load address(High)
RCAR_BL33DST_SIZE	0x00040000	Header Information(BL31 #1) : Image size[word]
RCAR_BL332DST_ADDRESS	0x00000000	Header Information(BL33 #2) : Load address(Low)
RCAR_BL332DST_ADDRESSH	0x00000000	Header Information(BL33 #2) : Load address(High)
RCAR_BL332DST_SIZE	0x00000000	Header Information(BL33 #2) : Image size[word]
RCAR_BL333DST_ADDRESS	0x00000000	Header Information(BL33 #3) : Load address(Low)
RCAR_BL333DST_ADDRESSH	0x00000000	Header Information(BL33 #3) : Load address(High)
RCAR_BL333DST_SIZE	0x00000000	Header Information(BL33 #3) : Image size[word]
RCAR_BL334DST_ADDRESS	0x00000000	Header Information(BL33 #4) : Load address(Low)
RCAR_BL334DST_ADDRESSH	0x00000000	Header Information(BL33 #4) : Load address(High)
RCAR_BL334DST_SIZE	0x00000000	Header Information(BL33 #4) : Image size[word]
RCAR_BL335DST_ADDRESS	0x00000000	Header Information(BL33 #5) : Load address(Low)
RCAR_BL335DST_ADDRESSH	0x00000000	Header Information(BL33 #5) : Load address(High)
RCAR_BL335DST_SIZE	0x00000000	Header Information(BL33 #5) : Image size[word]
RCAR_BL336DST_ADDRESS	0x00000000	Header Information(BL33 #6) : Load address(Low)
RCAR_BL336DST_ADDRESSH	0x00000000	Header Information(BL33 #6) : Load address(High)
RCAR_BL336DST_SIZE	0x00000000	Header Information(BL33 #6) : Image size[word]
RCAR_BL337DST_ADDRESS	0x00000000	Header Information(BL33 #7) : Load address(Low)
RCAR_BL337DST_ADDRESSH	0x00000000	Header Information(BL33 #7) : Load address(High)
RCAR_BL337DST_SIZE	0x00000000	Header Information(BL33 #7) : Image size[word]
RCAR_BL338DST_ADDRESS	0x00000000	Header Information(BL33 #8) : Load address(Low)
RCAR_BL338DST_ADDRESSH	0x00000000	Header Information(BL33 #8) : Load address(High)

RCAR_BL338DST_SIZE	0x00000000	Header Information(BL33 #8) : Image size[word]

Note) Image size is set in a “word” unit. (e.g. the image size of the BL33 #1 is : 0x00040000[word] = 0x00100000[byte])

Note) Please notice that change the destination address(bl31cert_addr and bl32cert_addr) of BL31 and BL32, because the execution address in BL31 and BL32 side are hard-coded.

Note) Load address of Header Information is represented by 64-bit and set to divide in the high word(bit[63:32]) and the low word(bit[31:0]).

Note) In the case of eMMC image loading is to set the value of the “RCAR_BLxxx_PARTITION” to access partition.

Example 1: How to change the source address of image loading.

The following shows an example of case in changing the source address of the “BL33 #1” to SA30(0x00C00000) from SA25(0x00640000).

Blue string indicates the changed or added.

- Step1: Edit the define value in code of sa6.c.

```
#if (RCAR_SA6_TYPE == RCAR_SA6_TYPE_HYPERFLASH)
        :
/* Source address on flash for BL33 */
#define RCAR_BL33SRC_ADDRESS                    (0x00C00000U)
        :
#else /* RCAR_SA6_TYPE == RCAR_SA6_TYPE_HYPERFLASH */
```

- Step2: Import the edited code refer to Clause 5.4.2.

- Step3: How to writing data to Hyper Flash / QSPI Flash, refer to "1.3.1 Related Document No.1" Linux Interface Specification Yocto recipe Start-Up Guide documentation. The created the Writing data by this example is shown in the following Table 5.35.

Table 5.35 Writing data of Example 1

filename	Program Top Address	Flash Save Address	description
bootparam_sa0.srec	H’E6320000	H’000000	Loader(Boot parameter)
bl2-<board_name><extension>	H’E6304000	H’040000	Loader
cert_header_sa6.srec	H’E6320000	H’180000	Loader(Certification)
bl31-<board_name>.srec	H’44000000	H’1C0000	Trusted Firmware-A
tee-<board_name>.srec	H’44100000	H’200000	OP-Tee
u-boot-elf-<board_name>.srec	H’50000000	**H’C00000**	U-boot

Note) <extension>: .srec, .-4x2g.srec, .-4d.srec.

Example 2: How to change the destination address of image loading.

The following shows an example of case in changing the destination address of the “BL33 #1” to 0x51000000 from 0x50000000.

Blue string indicates the changed or added.

- Step1: Edit the define value in code of sa6.c.

```
#if (RCAR_SA6_TYPE == RCAR_SA6_TYPE_HYPERFLASH)
            :
/* Destination address for BL33 */
#define RCAR_BL33DST_ADDRESS                (0x51000000U)
#define RCAR_BL33DST_ADDRESSH               (0x00000000U)
            :
#else /* RCAR_SA6_TYPE == RCAR_SA6_TYPE_HYPERFLASH */
```

- Step2: import the edited code refer to Clause 5.4.2.

- Step3: How to writing data to Hyper Flash / QSPI Flash, refer to "1.3.1 Related Document No.1" Linux Interface Specification Yocto recipe Start-Up Guide documentation. The created the Writing data by this example is shown in the following Table 5.36.

Table 5.36 Writing data of Example 2

filename	Program Top Address	Flash Save Address	description
bootparam_sa0.srec	H’E6320000	H’000000	Loader(Boot parameter)
bl2-<board_name><extension>	H’E6304000	H’040000	Loader
cert_header_sa6.srec	H’E6320000	H’180000	Loader(Certification)
bl31-<board_name>.srec	H’44000000	H’1C0000	Trusted Firmware-A
tee-<board_name>.srec	H’44100000	H’200000	OP-Tee
u-boot-elf-<board_name>.srec	**H’51000000**	H’640000	U-boot

Note) If you change the destination address of image loading, please rebuild to the loaded image, because it is necessary to be changed to match the execution address.

Note) <extension>: .srec, .-4x2g.srec, .-4d.srec.

Example 3: How to add the loading image.

The following shows an example of the case the loading images (BL33 #2-#4) is added.

Table 5.37 Added the loading images of Example 3

filename	Program Top Address	Flash Save Address	Image size[word]
<bl33 #2 image name>.srec	H'50160000	H'7C0000	H'40000
<bl33 #3 image name>.srec	H'502C0000	H'940000	H'40000
<bl33 #4 image name>.srec	H'50420000	H'AC0000	H'40000

Blue string indicates the changed or added.

- Step1: Edit the define value in code of sa6.c.

```
#if (RCAR_SA6_TYPE == RCAR_SA6_TYPE_HYPERFLASH)

/* Number of content cert for Non-secure Target Program(BL33x) */
#define RCAR_IMAGE_NUM                          (0x00000004U)
            :
#define RCAR_BL332SRC_ADDRESS                   (0x007C0000U)
/* Reserved */
#define RCAR_BL332_PARTITION                    (0x00000000U)
#define RCAR_BL333SRC_ADDRESS                   (0x00940000U)
/* Reserved */
#define RCAR_BL333_PARTITION                    (0x00000000U)
#define RCAR_BL334SRC_ADDRESS                   (0x00AC0000U)
/* Reserved */
#define RCAR_BL334_PARTITION                    (0x00000000U)
            :
/* Reserved */
#define RCAR_BL332DST_ADDRESS                   (0x50160000U)
#define RCAR_BL332DST_ADDRESSH                  (0x00000000U)
#define RCAR_BL332DST_SIZE                      (0x00040000U)
/* Reserved */
#define RCAR_BL333DST_ADDRESS                   (0x502C0000U)
#define RCAR_BL333DST_ADDRESSH                  (0x00000000U)
#define RCAR_BL333DST_SIZE                      (0x00040000U)
/* Reserved */
#define RCAR_BL334DST_ADDRESS                   (0x50420000U)
#define RCAR_BL334DST_ADDRESSH                  (0x00000000U)
#define RCAR_BL334DST_SIZE                      (0x00040000U)
            :
#else /* RCAR_SA6_TYPE == RCAR_SA6_TYPE_HYPERFLASH */
```

- Step2: import the edited code refer to Clause 5.4.2.

- Step3: How to writing data to Hyper Flash / QSPI Flash, refer to "1.3.1 Related Document No.1" Linux Interface Specification Yocto recipe Start-Up Guide documentation. The created the Writing data by this example is shown in the following Table 5.38.

Table 5.38 Writing data of Example 3

filename	Program Top Address	Flash Save Address	description
bootparam_sa0.srec	H'E6320000	H'000000	Loader(Boot parameter)
bl2-<board_name><extension>	H'E6304000	H'040000	Loader
cert_header_sa6.srec	H'E6320000	H'180000	Loader(Certification)
bl31-<board_name>.srec	H'44000000	H'1C0000	Trusted Firmware-A
tee-<board_name>.srec	H'44100000	H'200000	OP-Tee
u-boot-elf-<board_name>.srec	H'50000000	H'640000	U-boot
<bl33 #2 image name>.srec	**H'50160000**	**H'7C0000**	**bl33 #2 image**
<bl33 #3 image name>.srec	**H'502C0000**	**H'940000**	**bl33 #3 image**
<bl33 #4 image name>.srec	**H'50420000**	**H'AC0000**	**bl33 #4 image**

Note) <extension>: .srec, .-4x2g.srec, .-4d.srec.

Example 4: How to eMMC boot by default setting.

- Step1: Do the following operations after Step4 of chapter3 of the "1.3.1 Related Document No.1".
- Step2: Add a build option “RCAR_SA6_TYPE=1” refer to section 5.3 and clause 5.4.1.
- Step3: To complete the build, perform the Step5 and later operations, of chapter 3 of the "1.3.1 Related Document No.1".
- Step4: How to writing data to eMMC, refer to "1.3.2 Related Site for original software No.5" Flash Writer Application Note documentation. The created the Writing data by this example is shown in the following Table 5.39.

Table 5.39 Writing data for eMMC boot

filename	Program Top Address	eMMC Save Partition	eMMC Save Address	description
bootparam_sa0.srec	H’E6320000	boot partition1	H’000000	Loader(Boot parameter)
bl2-<board_name><extension>	H’E6304000	boot partition1	H’003C00	Loader
cert_header_sa6.srec	H’E6320000	boot partition1	H’030000	Loader(Certification)
bl31-<board_name>.srec	H’44000000	boot partition1	H’040000	Trusted Firmware-A
tee-<board_name>.srec	H’44100000	boot partition1	H’200000	OP-Tee
u-boot-elf-<board_name>.srec	H’50000000	boot partition2	H’000000	U-boot

Note) <extension>: .srec, .-4x2g.srec, .-4d.srec.

- Step5: Set the EXT_CSD register. For setting of EXT_CSD register, refer to "1.3.1 Related Document No.2" R-Car Series, 3rd Generation User's Manual: Hardware:19.2.5 eMMC Using DMA Boot Sequence.

Example 5: How to add the loading image for eMMC boot.

The following shows an example of the case the loading images (BL33 #2-#4) is added.

Table 5.40 Added the loading images for eMMC of Example 5

filename	Program Top Address	eMMC Save Partition	eMMC Save Address	Image size[word]
<bl33 #2 image name>.srec	H'50160000	boot partition2	H'00100000	H'40000
<bl33 #3 image name>.srec	H'450C0000	user data area	H'00000000	H'40000
<bl33 #4 image name>.srec	H'50420000	user data area	H'00200000	H'40000

Blue string indicates the changed or added.

- Step1: Edit the define value in code of sa6.c.

```
        :
#else /* RCAR_SA6_TYPE == RCAR_SA6_TYPE_HYPERFLASH */

/* Number of content cert for Non-secure Target Program(BL33x) */
#define RCAR_IMAGE_NUM                  (0x00000004U)
        :
#define RCAR_BL332SRC_ADDRESS           (0x00100000U)
/* Reserved */
#define RCAR_BL332_PARTITION            (0x00000002U)
#define RCAR_BL333SRC_ADDRESS           (0x00000000U)
/* Reserved */
#define RCAR_BL333_PARTITION            (0x00000000U)
#define RCAR_BL334SRC_ADDRESS           (0x00200000U)
/* Reserved */
#define RCAR_BL334_PARTITION            (0x00000000U)
        :
/* Reserved */
#define RCAR_BL332DST_ADDRESS           (0x50160000U)
#define RCAR_BL332DST_ADDRESSH          (0x00000000U)
#define RCAR_BL332DST_SIZE              (0x00040000U)
/* Reserved */
#define RCAR_BL333DST_ADDRESS           (0x502C0000U)
#define RCAR_BL333DST_ADDRESSH          (0x00000000U)
#define RCAR_BL333DST_SIZE              (0x00040000U)
/* Reserved */
#define RCAR_BL334DST_ADDRESS           (0x50420000U)
#define RCAR_BL334DST_ADDRESSH          (0x00000000U)
#define RCAR_BL334DST_SIZE              (0x00040000U)
        :
#endif /* RCAR_SA6_TYPE == RCAR_SA6_TYPE_HYPERFLASH */
```

- Step2: Add a build option “RCAR_SA6_TYPE=1” refer to section 5.3 and clause 5.4.1.
- Step3: Import the edited code refer to clause 5.4.2.
- Step4: How to writing data to eMMC, refer to "1.3.2 Related Site for original software No.5" Flash Writer Application Note documentation. The created the Writing data by this example is shown in the following Table 5.41.

Table 5.41 Writing data for eMMC of Example 5

filename	Program Top Address	eMMC Save Partition	eMMC Save Address	description
bootparam_sa0.srec	H’E6320000	boot partition1	H’000000	Loader(Boot parameter)
bl2-<board_name><extension>	H’E6304000	boot partition1	H’003C00	Loader
cert_header_sa6.srec	H’E6320000	boot partition1	H’030000	Loader(Certification)
bl31-<board_name>.srec	H’44000000	boot partition1	H’040000	Trusted Firmware-A
tee-<board_name>.srec	H’44100000	boot partition1	H’200000	OP-Tee
u-boot-elf-<board_name>.srec	H’50000000	boot partition2	H’000000	U-boot
<bl33 #2 image name>.srec	**H’50160000**	**boot partition2**	**H’100000**	**bl33 #2 image**
<bl33 #3 image name>.srec	**H’502C0000**	**user area partition**	**H’000000**	**bl33 #3 image**
<bl33 #4 image name>.srec	**H’50420000**	**user area partition**	**H’200000**	**bl33 #4 image**

Note) <extension>: .srec, .-4x2g.srec, .-4d.srec.

5.4.3.10 Lossy register setting

It is possible to change or add the CMA region which supports Near Lossless data compression (FCNL) for media playback.

The following shows file and directory structure of Lossy register setting.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/plat/renesas/rcar]

```
rcar
└  bl2_plat_setup.c
```

The default setting of each Lossy entry is shown in the below Table 5.42.

Table 5.42 Default settings of Lossy entries

Entry No.	Format	Physical Range (Size)	Enable or Disable
0	YUV Planar	0x5400_0000 – 0x5700_0000 (48 MB)	Enable
1	ARGB8888	Undefined	Disable
2	YUV 422 Interleave	Undefined	Disable

There are two steps to change the Lossy register setting.

- Step1: Modify the start address and size of CMA region for Lossy comp in Device tree [$WORK/build_<Building case>/tmp/work-shared/salvator-x/kernel-source/arch/arm64/boot/dts/renesas/xxxx.dts] (e.g. H3: xxxx.dts means r8a7795-salvator-x.dts)

```
lossy_decompress: linux,lossy_decompress {
        no-map;
        reg = <0x00000000 0xXXXXXXXX 0x0 0xYYYYYYYY>;
};

0xXXXXXXXX is start address of CMA region.
0xYYYYYYYY is size of CMA region.
```

- Step2: Modify macros of the entry defined in bl2_plat_setup.c.

```
/* Settings of Entry X */              (X=0/1/2)
#define  LOSSY_ST_ADDRX
#define  LOSSY_END_ADDRX
#define  LOSSY_FMTX
#define  LOSSY_ENA_DISX
```

Example 1: How to change the CMA region for YUV Planar.

< Default > YUV Planar: 0x5400_0000 – 0x5700_0000 (48 MB)

< After > YUV Planar: 0x5300_0000 – 0x5500_0000 (32 MB)

- Step1: Modify the start address and size of CMA region for Lossy comp in Device tree

```
lossy_decompress: linux,lossy_decompress {
        no-map;
        reg = <0x00000000 0x53000000 0x0 0x02000000>;
};
```

- Step2: Modify macros of the entry defined in bl2_plat_setup.c.

```
/* Settings of Entry 0 */
#define   LOSSY_ST_ADDR0        (0x53000000U)
#define   LOSSY_END_ADDR0       (0x55000000U)
#define   LOSSY_FMT0            LOSSY_FMT_YUVPLANAR
#define   LOSSY_ENA_DIS0        LOSSY_ENABLE
```

Example 2: How to change the CMA region for YUV Planar and add another one for ARGB8888

< Default > YUV Planar: 0x5400_0000 – 0x5700_0000 (48 MB)

< After > YUV Planar: 0x5300_0000 – 0x5500_0000 (32 MB)

ARGB8888: 0x5500_0000 – 0x5700_0000 (32 MB)

- Step1: Modify the start address and total size of CMA region for Lossy comp in Device tree

```
lossy_decompress: linux,lossy_decompress {
        no-map;
        reg = <0x00000000 0x53000000 0x0 0x04000000>;
};
```

- Step2: Modify macros of the entries defined in bl2_plat_setup.c.

```
/* Settings of Entry 0 */
#define   LOSSY_ST_ADDR0        (0x53000000U)
#define   LOSSY_END_ADDR0       (0x55000000U)
#define   LOSSY_FMT0            LOSSY_FMT_YUVPLANAR
#define   LOSSY_ENA_DIS0        LOSSY_ENABLE

/* Settings of Entry 1 */
#define   LOSSY_ST_ADDR1        (0x55000000U)
#define   LOSSY_END_ADDR1       (0x57000000U)
#define   LOSSY_FMT1            LOSSY_FMT_ARGB8888
#define   LOSSY_ENA_DIS1        LOSSY_ENABLE
```

Example 3: How to disable the CMA region for YUV Planar

< Default > YUV Planar: 0x5400_0000 – 0x5700_0000 (48 MB)

< After > YUV Planar: -

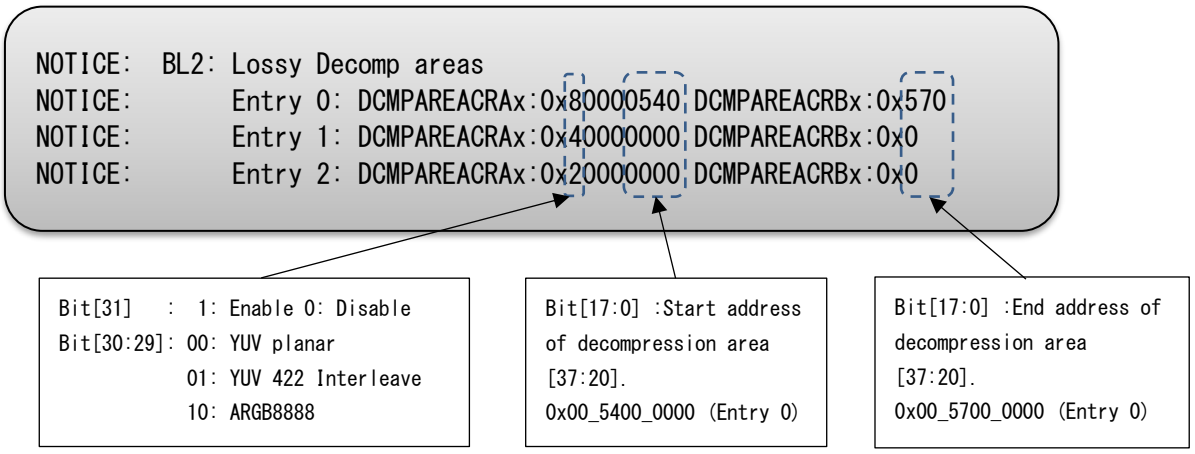
- Step1: Remove the following description of CMA region for Lossy comp in Device tree

```
lossy_decompress: linux,lossy_decompress {
        no-map;
        reg = <0x00000000 0x54000000 0x0 0x03000000>;
};
```

- Step2: Modify macros of the entry defined in bl2_plat_setup.c.

```
/* Settings of Entry 0 */
#define  LOSSY_ST_ADDR0        0x0
#define  LOSSY_END_ADDR0       0x0
#define  LOSSY_FMT0            LOSSY_FMT_YUVPLANAR
#define  LOSSY_ENA_DIS0        LOSSY_DISABLE
```

In the boot message, the register setting information of Lossy entries is outputted.



5.4.3.11 MMU mapping setting

The following shows the MMU mapping setting when D-Cache is enabled by build option.

It is possible to change or add the MMU mapping for d-cache.

The following shows file and directory structure of the MMU mapping setting.

[$WORK/build_<Building case>/tmp/work/salvator_x-poky-linux/arm-trusted-firmware/<Properties of yocto environment>/git/plat/renesas]

```
renesas
 ├common
 │  ├aarch64
 │  │    platform_common.c
 │  └include
 │       rcar_def.h
 └rcar
     bl2_plat_setup.c
```

The MMU mapping is set by the rcar_bl2_plat_arch_setup() function implemented in bl2_plat_setup.c and the rcar_mmap[] implemented in platform_common.c.

When changing the number of entries of rcar_mmap[], need to change the value of RCAR_MMAP_ENTRIES defined in rcar_def.h.

platform_common.c

```
#if IMAGE_BL2
const mmap_region_t rcar_mmap[] = {
        MAP_FLASH0,    /*    0x08000000 -    0x0BFFFFFF   RPC area                */
        MAP_DRAM0,     /*    0x40000000 -    0xBFFFFFFF   DRAM area(Legacy)     */
        MAP_REG0,      /*    0xE6000000 -    0xE62FFFFF   SoC register area     */
        MAP_RAM0,      /*    0xE6300000 -    0xE6303FFF   System RAM area       */
        MAP_REG1,      /*    0xE6400000 -    0xEAFFFFFF   SoC register area     */
        MAP_ROM,       /*    0xEB100000 -    0xEB127FFF   boot ROM area         */
        MAP_REG2,      /*    0xEC000000 -    0xFFFFFFFF   SoC register area     */
        MAP_DRAM1,     /* 0x0400000000 - 0x07FFFFFFFF   DRAM area(4GB over) */
        {          0}
};
#endif
```

rcar_def.h

```
#if IMAGE_BL2
#define RCAR_MMAP_ENTRIES               (9)
#endif
```

bl2_plat_setup.c

```
/*
 * Following symbols are only used during plat_arch_setup()
 */
static const uint64_t BL2_RO_BASE                    = BL_CODE_BASE;
static const uint64_t BL2_RO_LIMIT                   = BL_CODE_END;

#if USE_COHERENT_MEM
static const uint64_t BL2_COHERENT_RAM_BASE   = BL_COHERENT_RAM_BASE;
static const uint64_t BL2_COHERENT_RAM_LIMIT   = BL_COHERENT_RAM_END;
#endif
```

bl2_plat_setup.c

```
void bl2_el3_plat_arch_setup(void)
{
#if RCAR_BL2_DCACHE == 1
        NOTICE("BL2: D-Cache enable¥n");
#endif
        rcar_configure_mmu_el3(BL2_BASE,
                                BL2_END - BL2_BASE,
                                BL2_RO_BASE, BL2_RO_LIMIT
#if USE_COHERENT_MEM
                                , BL2_COHERENT_RAM_BASE,
BL2_COHERENT_RAM_LIMIT
#endif
                );
}
```

6. Appendix

6.1 Periodic write DQ training

6.1.1 Outline

The adjustment of write signal timing for LPDDR4 skew correction is needed as to amount of temperature change of R-Car chip.

Renesas proposes “periodic write DQ training” as a measure of the adjustment.

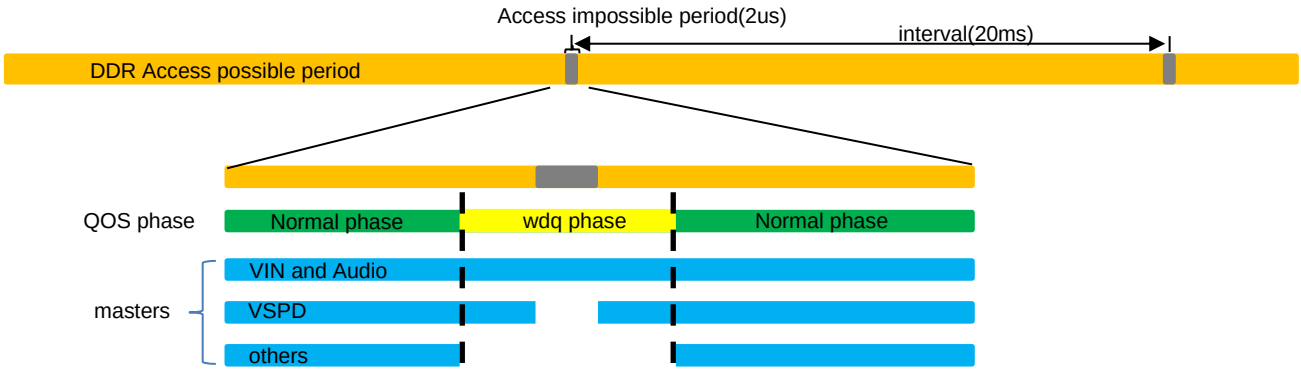
The interval of periodic write DQ training is 20ms and a write DQ training period is about 2us. In the period, each of read/write requests cannot access DDR. Therefore, sensitive latency master like VSPD or VIN may report error because of latency deterioration.

In order to resolve the issue, Renesas prepares another QoS control during periodic write DQ training to realize the adjustment of write signal timing without latency deterioration.

6.1.2 QoS control of periodic write DQ training

DBSC preferentially accepts access requests of VIN, Audio and VSPD to reduce the latency deterioration impact. VIN and Audio requests are always buffered not only in “Normal phase” but also in “wdq phase”. VSPD requests are buffered except for “Access impossible period” in wdq phase. The others are only buffered in “Normal phase”.

In case that interval of periodic write DQ training is set to 20ms, there is hardly each master bandwidth reduction. (0.02% or less)



REVISION HISTORY	Initial Program Loader User's Manual: Software

Rev.	Date	Description	
		Page	Summary
1.0.0	Mar. 25, 2016	—	New creation.
1.0.1	Apr. 27, 2016	—	R-Car M3 support.
		2	Update Table 1.1 (add the description of "Lossy register setting")
		3	Changed the edition of 'Linux Interface Specification Yocto recipe Start-Up Guide' 0.90
		5	Update Table 2.1 (add the description of "Lossy")
		7	Update Figure 3.2 (add "Lossy register setting")
		10	5.1 Directory configuration Change directory structure.
		11-12	5.3 Option setting Modified valid logging function of log level 10, NOTICE() to ERROR(). Changed the description in the order of ERROR(), NOTICE() . -LOG_LEVEL -DEBUG
		13	5.3 Option setting Append the description of the added options. -LSI -LSI_CUT
		17	5.4.3.1 PFC/GPIO setting Change directory structure. Delete the bl2_pfc_init. Add the pfc_init function in H3 / M3 correspondence. Add the internal functions that are called from pfc_init function.
		18-19	5.4.3.3 SDRAM setting Change directory structure. Add the M3 directory / function / SDRAM bus clock in M3 correspondence.
		19-20	5.4.3.4 SDRAM access arbitration setting Change directory structure. Add the M3 directory / function in M3 correspondence.
		23-26	Add 5.4.3.7 Lossy register setting
1.0.2	May 27, 2016	3	Changed the edition of 'Linux Interface Specification Yocto recipe Start-Up Guide'
		14	Modify Chapter 5.4.1 by corresponding the soc type name.
		20	Change to 1600MHz(LPDDR4-3200) from 800MHz(LPDDR4-1600) of the SDRAM bus clock for M3 in the Table 5.9.
1.0.3	Jun. 3, 2016	24-38	5.4.3.6 Image loading Add a description of how to change the specified loading of image.
1.0.4	Aug. 29, 2016	2	Update Table 1.1 - Change the Summary of "Image loading". - Add the "Suspend to RAM" function.
		3	Add the Related Document No.3 and No.4. - eMMC Writer Application Note - JEDEC STANDARD Embedded Multi-Media Card (e•MMC) Electrical Standard (5.0)
		4	1.4 Restrictions Removed all restrictions
		5	Change chapter 2 "Terminology" to "Terminology and Abbreviation" Add the "Suspend to RAM".
		7	Update Figure 3.2 - add "eMMC" and "SDHI2/MMC0" to "Image loading" - add "Suspend to RAM"
		10	Update Figure 5.1 (add "emmc" and "wait")

		12	5.3 Option setting Change RCAR_QOS_TYPE of Option setting.
		13-14	5.3 Option setting Append the description of the added options. - LIFEC_DBSC_PROTECT_ENABLE - RCAR_SA6_TYPE
		17-18	5.4.1 Build option Change the Sample of recipe file.
		22	5.4.3.4 Suspend to RAM Insert the “Suspend to RAM”.
		24-29	Add 5.4.3.6 Security access protection setting
		30-39	5.4.3.6 Image loading Change the format of memory map on SA6 area. Change the coding of dummy_create. Change the setting of source address for BL31 and BL32.
		40-46	Add 5.4.3.8 Image loading (eMMC)
		47-50	5.4.3.9 Lossy register setting Change the memory property of Lossy region (reusable -> no-map) Add a boot message description of register setting information of the Lossy entries.
1.0.5	Dec. 22, 2016	2, 7	Update the Table 1.1 and Figure 3.2 Change the Function name to “QoS arbitration setting” from “SDRAM access arbitration setting”
		10	Update the Figure 5.1 Delete under the DDR folder. Add the error folder, and rom folder.
		12-13	5.3 Option setting Add the SPD of Option setting. Change a description to RCAR_QOS_TYPE of Option setting. Add the setting value to RCAR_DRAM_SPLIT of Option setting. Add the setting string to LSI of Option setting.
		21	5.4.3.3 SDRAM setting Updated description of SDRAM setting.
		23-24	5.4.3.5 QoS arbitration setting Change the clause title to “QoS arbitration setting” from “SDRAM access arbitration setting” Update the QoS type description (Remove the older QoS type description) Figure 5.2 QoS data tables Remove RCAR_QOS_TYPE_GFX / RCAR_QOS_TYPE_HIGH_RESOLUTION, and Add RCAR_QOS_TYPE_DEFAULT
		27-29	5.4.3.6 Security access protection setting Table 5.13 List of the Security access protection setting for AXI-bus Modify the AXI_SPTDIVCR0 register setting. Figure 5.3 Table of the Security access protection setting Modify the AXI_SPTDIVCR0 register setting.
		30	5.4.3.7 Image loading Modify the typo of the SDRAM address in Figure 5.4 Memory Map.
		35-37	5.4.3.7 Image loading Table 5.16 Description of defined value in sa6.c Modify the Load address of BL33 #1. Table 5.17 Writing data of Example 1 Modify the Program Top Address of U-boot. Example 2: How to change the destination address of image loading. Update the destination address of the BL33#1. Table 5.18 Writing data of Example 2 Modify the Program Top Address of U-boot.
		38-39	5.4.3.7 Image loading Modify the typo of the address in Example 3. Table 5.20 Writing data of Example 3 Modify the Program Top Address of U-boot.

		40	5.4.3.8 Image loading (eMMC) Modify the typo of the SDRAM address in Figure 5.7 Memory Map for eMMC.
		43	5.4.3.8 Image loading (eMMC) Table 5.22 Description of defined value in sa6.c for eMMC Modify the Load address of BL33 #1.
		45	5.4.3.8 Image loading (eMMC) Add a how to eMMC boot.
		46-47	5.4.3.8 Image loading (eMMC) Modify the typo of the address in Example 1. Table 5.25 Writing data for eMMC of Example 1 Modify the Program Top Address of U-boot.
1.0.6	Mar. 17, 2017	2	Update the Table 1.1. Added the related document on the Summary. ・PMIC initial voltage setting ・Suspend to RAM
		3	Update the Table 1.1. Add the “Start generic timer” function.
		4	Update the Table 1.2 Remove related document (Number 3 “eMMC Writer Application Note”) Added related document (Number 4,5,6,7,8)
		5	Update the Table 1.3 Added related site for original software document (Number 5)
		7	Update the Table 3.1 Added evaluation board to explanation
		11	Update the Figure 5.1 Modified the file path included in the qos folder
		13-14	5.3 Option setting Added the Note. If MD pins selection is other than MD7=0, MD6=0 or MD7=0, MD6=1, this option has been no effect. ・RCAR_QOS_TYPE ・RCAR_DRAM_SPLIT
		15	5.3 Option setting Added the Note. If MD pins selection is MD7 = 1, MD6 = 1, this option has been no effect. ・LIFEC_DBSC_PROTECT_ENABLE
		14-16	5.3 Option setting Added the Note. Notes on PMIC related functions when using other boards (not Salvator-X board). ・RCAR_AVS_SETTING_ENABLE ・PMIC_ON_BOARD
		15	Update the Table 5.7 Modified the CUT number display (WS->Ver) Add CUT number (Ver.2.0)
		16	5.3 Option setting Append the description of the added options. - RCAR_BL33_EXECUTION_EL
		18	5.4.1 Build option Deleted compile flag addition explanation from DEBUG option.
		21-27	5.4.3 How to customize Added the Note. If MD pins selection is other than MD7=0, MD6=0 or MD7=0, MD6=1,, following functions are not executed therefore no effect. ・PFC/GPIO setting ・SDRAM setting ・Suspend to RAM ・QoS arbitration setting ・Security access protection setting
		29-31	5.4.3 Security access protection setting Added the Note. If MD pins selection is MD7 = 1, MD6 = 1, register setting of LifeC is skipped. If MD pins selection is MD7 = 1, MD6 = 1, register setting of AXI-bus is skipped.

		22-23	5.4.3.2 Process timeout detection setting Added the Process timeout detection setting
		24-25	5.4.3.3 SDRAM setting Added the init_dram_tbl_h3ver2.h to file tree Added the H3 Ver.2.0 Modified the CUT number display (WS->Ver)
		26-27	5.4.3.5 QoS arbitration setting Modified directory structure and file name Added the H3 Ver.2.0 Modified the CUT number display (WS->Ver) Modified typo restriction of the Table 5.12 title string
		30-31	Update the Table 5.14 Modified the Setting value, Comment (SPTDIVCRn)
		32	Update the Figure 5.5 Modified the parameter (AXI_SPTDIVCR0)
		33-39	5.4.3.7 Image loading Added range check of source/destination image loading address Moved the dummy_create description to clause 5.4.3.9
		40-42	5.4.3.8 Image loading (eMMC) Added range check of source image loading address Moved the dummy_create description to clause 5.4.3.9
		43-56	5.4.3.9 dummy_create setting Insert the “dummy_create setting”.
		61-62	5.4.4 eMMC boot for R-Car M3 Ver.1.0 Added the clause “eMMC boot for R-Car M3 Ver.1.0”.
1.0.7	Apr. 12, 2017	6	Update Table 2.1. Added terms. (BL31, BL32, BL33, BL33x)
		8	Update Figure 3.2. Added relation line to the I2C for DVFS from the Suspend to RAM.
		25	5.4.3.4 Suspend to RAM Modified the description according to the implementation of the Suspend to RAM.
		29	5.4.3.6 Security access protection setting Fixed the end of SRAM protect range from 0xE6301FFF to 0xE6303FFF.
		40	5.4.3.8 Image loading (eMMC) Modify the explain of Partition in Table 5.18.
		63	6 Appendix Added the 6.1 explains the Timestamp.
1.0.8	Aug. 8, 2017	—	Fixed the format of the document (trademark, etc.)
		2	Table 1.1 PMIC initial voltage setting Added module number of PMIC to summary.
		3	Table 1.1 Evaluation board Identification Added Evaluation board identification processing.
		5	Table 1.3 Related Site for original software No.5 Changed eMMC Writer to Flash Writer
		12-15	5.3 Option setting Modified the typos, and updated to latest description.
		14,16	5.3 Option setting Fixed description of changed build options. RCAR_AVS_SETTING_ENABLE/PMIC_ROHM_DB9571/RCAR_SYSTEM_SUSPEND
		21-24, 38,62	5.4.3 How to customize Modified the typos.
		26-27	5.4.3.5 QoS arbitration setting Modified the typos, and updated to latest description.
		28	5.4.3.6 Security access protection setting Corrected of example sentence that sets bit 19 of SEC_SEL3 to 1.
		44-46	5.4.3.9 dummy_create setting Change the variable name and section name.
		62	5.4.4 eMMC boot for R-Car M3 Ver.1.0 Change the variable name and section name.
		63	5.4.4 eMMC boot for R-Car M3 Ver.1.0 Modify the eMMC Save Address of bl2-<board_name>.srec in Table 5.31.

1.0.9	Nov. 14, 2017	—	R-Car M3N support.
		11	5.1 Directory configuration Change directory structure. Added the M3N
		14-15	5.3 Option setting Table 5.5 Association table for the RCAR_DRAM_SPLIT value and DRAM split setting Table 5.6 Association table for the LSI string and Target LSI type setting Added the M3N
		21	5.4.3.1 PFC/GPIO setting Table 5.10 File and Function list of PFC/GPIO setting Modified directory structure and file name. Added the M3N
		26-27	5.4.3.5 QoS arbitration setting Table 5.13 File and Function list of QoS access arbitration setting Modified directory structure and file name. Added the M3N
1.0.10	Jan. 12, 2018	17-18	5.3 Option setting Append the description of the added options. - RCAR_REF_INT - RCAR_REWT_TRAINING - RCAR_BL2_DCACHE
		28-29	5.4.3.5 QoS arbitration setting Figure 5.5 QoS data tables (H3 Ver.2.0, M3 Ver.1.1 and M3N Ver.1.1) Modified directory structure and file name. Added the QoS data tables (H3 Ver.2.0, M3 Ver.1.1 and M3N Ver.1.1).
		46, 49-50	5.4.3.9 dummy_create setting Increased the loading size of BL2 in SA0. Increased the loading size of BL32 in SA6.
		63-64	5.4.3 How to customize Append the 5.4.3.11 MMU mapping setting.
		68	6 Appendix Append the 6.2 Periodic write DQ training.
1.0.11	Mar. 14, 2018	—	R-Car E3 support.
		3	Update the Table 1.1 Added the description of E3 to Start generic timer.
		4	Update the Table 1.2 Added related document (Number 10).
		7	Update the Table 3.1 Added Evaluation Board Ebisu for E3.
		8	Update Figure 3.2. Modified the module (SDHI).
		11	5.1 Directory configuration Change directory structure. Added the E3
		14-15, 17-19	5.3 Option setting - RCAR_DRAM_SPLIT - RCAR_AVS_SETTING_ENABLE - LSI - RCAR_REF_INT - RCAR_SA0_SIZE - RCAR_DRAM_DDR3L_MEMCONF Added the E3
		24-25	5.4.3.1 PFC/GPIO setting Table 5.10 File and Function list of PFC/GPIO setting Modified directory structure and file name. Added the E3

		28-29	5.4.3.3 SDRAM setting Modified directory structure. Table 5.18 SDRAM bus clock Added the M3N/E3
		30	5.4.3.4 Suspend to RAM Modified directory structure and file name.
		31-32	5.4.3.5 QoS arbitration setting Table 5.13 File and Function list of QoS access arbitration setting Modified directory structure and file name. Added the E3
		33-34	5.4.3.6 Security access protection setting Table 5.20 List of the Security access protection setting to the LifeC Added the E3
		43	5.4.3.7 Image loading Figure 5.11 Range of valid address to load of BL33 #1-#8 Modified the typo of the SDRAM address.
		50	5.4.3.9 dummy_create setting Table 5.25 Description of variable in BL2_ADDRESS and BL2_SIZE Table 5.26 Description of variable in sa0.c Added the E3
		68	5.4.3.11 MMU mapping setting Changed the bl2_plat_arch_setup() function in bl2_rcar_setup.c.
1.0.12	Apr. 11, 2018	14,16, 20	5.3 Option setting - LSI_CUT - RCAR_REWT_TRAINING Added the H3 Ver.3.0 and M3N - RCAR_DRAM_LPDDR4_MEMCONF Append the description of the added options.
		27	5.4.3.2 Process timeout detection setting Table 5.17 Setting value list of Process timeout detection setting Added the E3
		30	5.4.3.3 SDRAM setting Table 5.18 SDRAM bus clock Added the H3 Ver.3.0
		32-33	5.4.3.5 QoS arbitration setting Table 5.19 File and Function list of QoS access arbitration setting Modified directory structure and file name. Added the H3 Ver.3.0
		40	5.4.3.7 Image loading Changed the unit of transfer size of load image from 64byte to 256byte.
1.0.13	Jun. 11, 2018	—	Added the M3 Ver.1.2.
		3	Table 1.1 Function of the IPL Evaluation board identification Modified the board revision of log format.
		14	5.3 Option setting - RCAR_DRAM_SPLIT Added the table of relationship between the build options RCAR_DRAM_SPLIT and LSI.
		21	5.3 Option setting - RCAR_DRAM_LPDDR4_MEMCONF Added the description of the relation with volume of SDRAM.
		28	5.4.3.2 Process timeout detection setting Changed the definition name(WTCNT_COUNT_8p13k_H3VER10).
		71	6 Appendix 6.1 Timestamp Modified the example of boot log.

1.0.14	Sep. 3, 2018	17	5.3 Option setting - RCAR_SA6_TYPE Added the information of default setting.
		20	5.3 Option setting - RCAR_DRAM_DDR3L_MEMDUAL Append the description of the added options.
		62	5.4.3.9 dummy_create setting Added the step 5 to example 4. Added the reference destination of EXT_CSD register setting.
		62, 64, 72	5.4.3.9 dummy_create setting 5.4.4 eMMC boot for R-Car M3 Ver.1.0 Changed the eMMC Writer to Flash Writer.
1.0.15	Oct. 12, 2018	1	1.2 Function Removed the BL1 AP Trusted ROM in Figure 1.1. Removed the unnecessary explanation and added description that BL2 runs with EL3.
		4	Update the Table 1.2 Added related document (Number 11).
		5	Update the Table 1.3 Update related site for original software document (Number 2/3/4).
		7	Update the Table 3.1 Added Evaluation Board Ebisu-4D for E3.
		19	5.3 Option setting Table 5.14 Association table for the memory mapping when D-Cache is enabled - RCAR_BL2_DCACHE Modified the memory mapping of System RAM area (IPL data area).
		31	5.4.3.3 SDRAM setting Table 5.22 Combination of DRAM configuration related build options Added the Combination of DRAM configuration related build options.
		70	5.4.3.11 MMU mapping setting Modified the code of rcar_bl2_plat_arch_setup() function.
		73	6 Appendix 6.1 Timestamp Modified the example of boot log.
2.0.0	Nov. 26, 2018	—	Fixed the format of the document (Address List.)
		58, 59, 61, 62, 64	5.4.3.9 dummy_create setting Table 5.34 Writing data of Example 1 Table 5.35 Writing data of Example 2 Table 5.37 Writing data of Example 3 Table 5.38 Writing data for eMMC boot Table 5.40 Writing data for eMMC of Example 5 Changed file names of U-boot/IPL. Added note for extension. Removed the <extension> of file names other than bl2 and reverted to .srec.
		72	5.4.4 eMMC boot for R-Car M3 Ver.1.0 Table 5.42 Writing data of eMMC boot for R-Car M3 Ver.1.0 Changed u-boot file name.
2.0.1	Feb.15, 2019	16,19	5.3 Option setting Table 5.8 Association table for the LSI_CUT value and CUT number setting Added Ver.1.3 setting to LSI CUT. Table 5.11 Association table for the RCAR_REF_INT value and DRAM refresh interval Modified the Note that add M3 Ver.3.0/1.3 setting to RCAR_REF_INT option. Table 5.12 Association table for the RCAR_REWT_TRAINING value and Periodic write DQ training Modified the Note that add M3 Ver.3.0/1.3 setting to RCAR_REWT_TRAINING option.
		32	5.4.3.3 SDRAM setting Table 5.22 Combination of DRAM configuration related build options Added M3 Ver.3.0/1.3 and E3 Ver.1.1 setting to table.
		35,36	5.4.3.5 QoS arbitration setting Modified file name.

			Table 5.23 File and Function list of QoS access arbitration setting Modified file name. Added the M3 Ver.3.0/Ver.1.3.
2.0.2	Mar. 11, 2019	19	5.3 Option setting Table 5.12 Association table for the RCAR_REWT_TRAINING value and Periodic write DQ training Modified RCAR_REWT_TRAINING default setting. Modified the Note that add available conditions.
2.0.3	Mar. 22, 2019	-	No update.
2.0.4	Jul. 12, 2019	19,21	5.3 Option setting Modify the typo of the description of initial settings of RCAR_REWT_TRAINING. Update the Table 5.16. Remove 4G Byte configuration of RCAR_DRAM_DDR3L_MEMCONF setting.
2.0.5	Dec. 13, 2019	48,49	5.4.3.7 Image loading Added description about range check of BL33 #1-#8.
		51	5.4.3.8 Image loading (eMMC) Added description about loadable image size to feature.
2.0.6	Feb. 7, 2020	31	5.4.3.3 SDRAM setting Modified the Table 5.22
3.0.0	Dec. 9, 2020	-	Modified the software name from ARM Trusted Firmware to Trusted Firmware-A.
		-	Modified the tool name from dummy_create to layout_create.
		-	Modified source file names to match the v3.0.0 configuration.
		-	Modified function names and code examples to match the v3.0.0 code.
		-	Modified directory configurations to match a configuration of v3.0.0 code.
		-	Removed the description of older versions H3 Ver.1.0/1.1 and M3 Ver.1.0/1.1.
		5	1.3.1 Related Document Modified the URL of related site.
		11	5.1 Directory configuration Modified the directory configuration in Figure 5.1.
		19	5.3 Option setting Modified the description to show that MMU is enabled even if this option RCAR_BL2_DCACHE is 1.
		21	5.3 Option setting Added the description of the RCAR_DRAM_MEMRANK option.
		35	5.4.3.6 Security access protection setting Table 5.25 Update register information of SEC_SEL12, SECGRP0CR3, SECGRP1CR3
		45 46	Deleted a description about substantially invalidation of a range check process by is_mem_free()
		70	Deleted the section of 6.1 Timestamp because it is not exist function.
3.0.1	Apr. 6, 2021	—	R-Car D3 support.
		—	Changed ARM notation to Arm.
		5	1.4 Restrictions Added the restrictions of the Draak board.
		12	5.1 Directory configuration Figure 5.1 File tree for Loader of R-CarH3/M3/M3N/E3/D3 Deleted the V3M
		14	5.3 Option setting Table 5.6 Association table for the RCAR_DRAM_SPLIT and LSI Table 5.7 Association table for the LSI string and Target LSI type setting Table 5.11 Association table for the RCAR_REF_INT value and DRAM refresh interval Table 5.15 Association table for the RCAR_SA0_SIZE value and the IPL size information Added the D3
		26-27	5.4.3.1 PFC/GPIO setting Table 5.20 File and Function list of PFC/GPIO setting

			Modified directory structure and file name. Added the D3
		29	5.4.3.2 Process timeout detection setting Table 5.21 Setting value list of Process timeout detection setting Added the D3
		31	5.4.3.3 SDRAM setting Modified the Table 5.22/5.23 Added the D3
		33-34	5.4.3.5 QoS arbitration setting Modified file name. Table 5.24 File and Function list of QoS access arbitration setting Modified file name. Added the D3.
		37-42	5.4.3.6 Security access protection setting Table 5.25 List of the Security access protection setting to the LifeC Added the D3.
		43-48	5.4.3.7 Image loading Fugure 5.6 Memory Map Added the D3.
		49-51	5.4.3.8 Image loading(eMMC) Fugure 5.11 Memory Map for eMMC Added the D3.
		51-52	5.4.3.9 dummy_create setting Table 5.29 Description of variable in BL2_ADDRESS and BL2_SIZE Table 5.30 Description of variable in sa0.c Added the D3
3.0.2	Aug. 16, 2021	-	Add information of Gen3e.
		5	1.4 Restrictions Change the description to a table.
		12	5.1 Directory configuration Modified directory structure.
		22	5.3 Option setting Added description about M3N to RCAR_DRAM_LPDDR4_MEMCONF.
		28	5.4.3.1 PFC/GPIO setting Table 5.20 File and Function list of PFC/GPIO setting Modified Version for H3 to 3.0 from 2.0.
		29	5.4.3.2 Process timeout detection setting Modified directory structure. Figure 5.3 Calculation formula. Modified CUT number to 3.0 from 2.0.
		31	5.4.3.3 SDRAM setting Modified directory structure. Table 5.23 Combination of DRAM configuration related build options Added M3N 4GB
		34	5.4.3.5 QoS arbitration setting Modified directory structure. Removed Ver 2.0 for H3.
		36	5.4.3.6 Security access protection setting Modified directory structure.
		44	5.4.3.7 Image loading Modified directory structure.
		69	5.4.3.11 MMU mapping setting Modified directory structure.
3.0.3	Dec. 01, 2021	46-48	5.4.3.7 Image loading Added descriptions of loading address range check for BL31 and BL32 images.
		56, 58	5.4.3.9 layout_create setting Changed the words of note for changing the address of BL31 and BL32.

Initial Program Loader
User's Manual: Software

Publication Date: Rev.1.0.0 Mar. 25, 2016
Rev.3.0.3 Dec. 01, 2021

Published by: Renesas Electronics Corporation

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

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

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc. Milpitas Campus
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics America Inc. San Jose Campus
6024 Silver Creek Valley Road, San Jose, CA 95138, USA
Tel: +1-408-284-8200, Fax: +1-408-284-2775

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

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

Renesas Electronics (China) Co., Ltd.
Room 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

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

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

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

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

Renesas Electronics Malaysia Sdn.Bhd.
Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia
Tel: +60-3-5022-1288, Fax: +60-3-5022-1290

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：https://www.renesas.com/contact/

Colophon 3.1

Initial Program Loader
User's Manual

Renesas Electronics Corporation