User's Manual

CONFIDENTIAL

Initial Program Loader for CR7

User's Manual: Software

R-Car H3/M3/M3N/E3 Series

Renesas Electronics
www.renesas.com

Rev.1.0.14 Apr. 2022

CONFIDENTIAL

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 processor.

- **[Purpose]**
  This manual is intended to give users an understanding of the functions of the R-Car H3/M3/M3N/E3 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
  → See the R-Car H3/M3/M3N/E3 User's Manual.
  — To know the electrical specifications of the multimedia processor for R-Car H3/M3/M3N/E3
  → See the R-Car H3/M3/M3N/E3 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 document explains about R-Car Series, 3rd Generation Initial Program Loader for CR7 (RCG3BIR7N0101ZDO) (hereinafter called ‘Loader’). Loader read image of RTOS and CA57/53 Loader (hereinafter called ‘CA5x Loader’) and certificate from HyperFlash/QSPI Flash. These images are certificated by Loader after these data are loaded into System RAM or LPDDR4-SDRAM/DDR3L-SDRAM (hereinafter called ‘SDRAM’). CR7 switch the process to RTOS after CPU0 of main CPU (CA57/53) executes CA5x Loader. Detail on implementation function is shown in Chapter 7.
This document applies only to start from CR7, does not apply to the main CPU (CA57 / CA53).

1.2 Function

Figure 1-1 shows software components in scope of the Initial Program Loader (hereinafter called ‘IPL’). Cortex-A57/53 and Cortex-R7 are in R-Car H3/M3/M3N/E3. These CPU are started to use Initial Program Loader each of Loader.

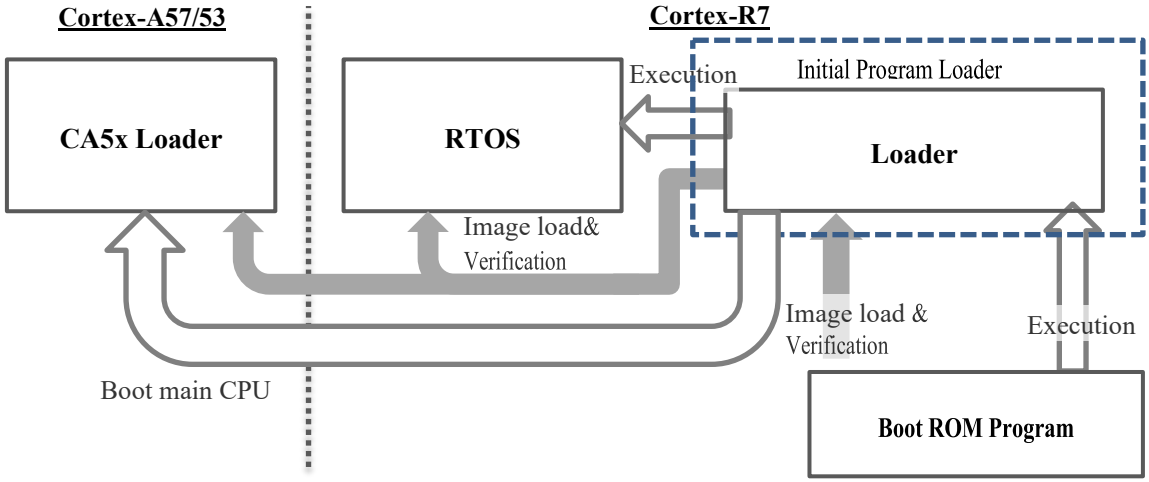


Figure 1-1 Scope of the Internal Program Loader

IPL starts RTOS and loads CA5x Loader and starts it.
Table 1-1 shows explanation of IPL functions.

Table 1-1 Explanation of IPL functions

Functions	Explanation
Hardware initialization	Initialize the configuration and the hardware used by the system.
Display the starting message	Output the message (software version, board name, etc.) to SCIF.
Decide the starting mode.	Decide starting mode from LCS, eFuse setting and MD pin.
Loading the image	Load the image (RTOS, CA5x Loader) and certificate / header from HyperFlash or QSPI Flash.
Authenticate the image using a certificate	If the boot mode is Secure mode, the images that you want to load are certificated by themselves certificate. If the certification fails, Loader output the error message after execution is stopped.
Releasing used resources	Release the resource used by Loader. This function initializes the hardware configuration changed by Loader and invalids the Instruction Cache.
Starting CA5x Loader	Start CPU0 of CA57/53 and execute CA5x Loader.
Starting RTOS	Start the RTOS after Loader is finish.

1.2.1 Hardware initialization

Loader executes following hardware initialization. Detail on these functions is shown in section 3.3.

- CPU initialization
- ECC initialization
- SWDT initialization
- PFC / GPIO initialization
- Module stop initialization
- SDRAM/PHY initialization
- RPC initialization
- DMA initialization
- Security and Safety setting
- QoS setting

1.2.2 Display the starting message

This function displays the starting message.
For example, Figure 1-2 shows the starting message of Salvator-X/XS.

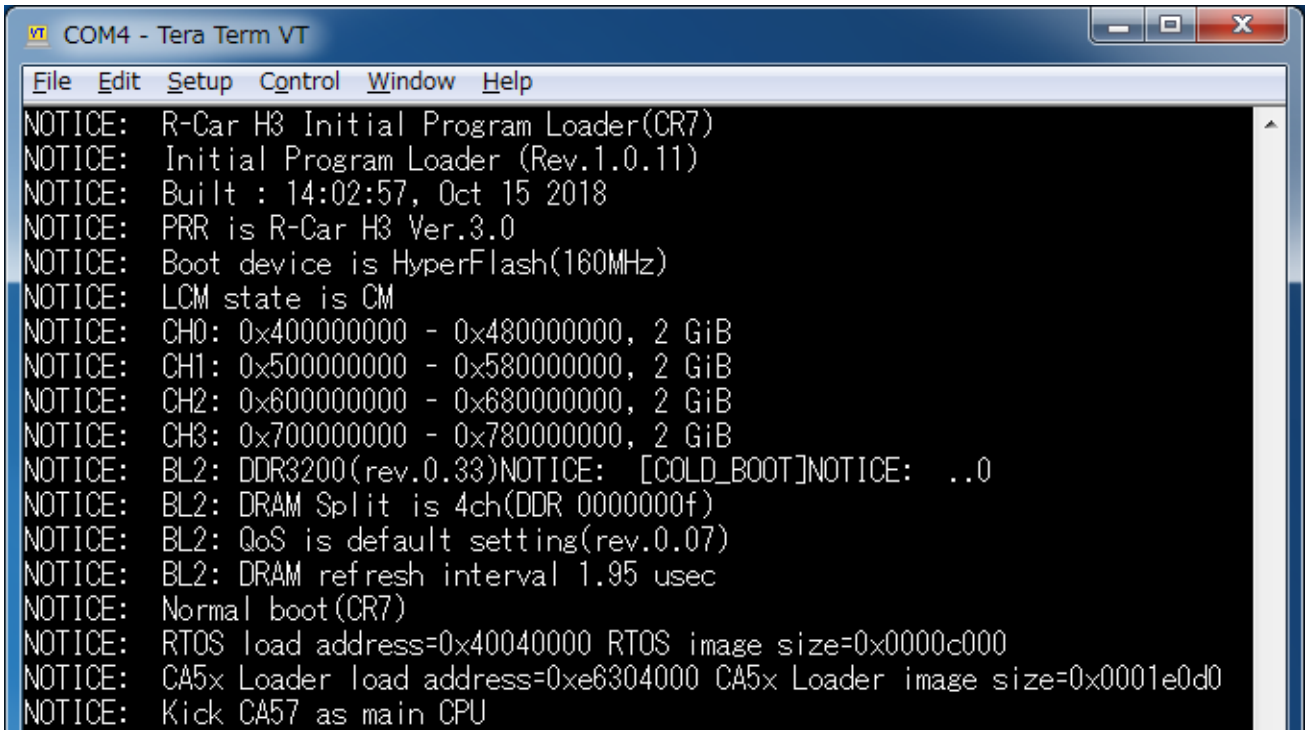


Figure 1-2 Starting message of Salvator X/XS

Table 1-2 shows item displayed as the starting message.

Table 1-2 Meaning of starting message

Item	Explanation
Loader name	R-Car <LSI name> Initial Program Loader (CR7)
Loader version	Initial Program Loader (Rev.2.0.x)
Built time	Built time and date.
LSI version (PRR)	The type and version of R-Car. ex: PRR is R-Car H3 Ver.3.0
SSCG Clock state	PLL1 SSCG or non SSCG Clock. Output in R-Car E3 only.
LCM state	Setting of LCS(CM / DM / SD / SE / FA / unknown)
DRAM configuration	Capacity and Split of DRAM. Output in R-Car H3 Ver.3.0 and R-Car E3 only.
LPDDR4-SDRAM / DDR3L-SDRAM clock speed	DDR4-1600/2400/2800/3200 and revision or DDR3L-1584/1856 and revision.
DRAM Split	Split function of SDRAM(4ch / 2ch / OFF)
QoS	Setting of QoS (Default setting / No setting) and revision.
DRAM refresh interval	DRAM refresh interval is 1.95 usec, 3.90 usec, or 7.80 usec.
DDR-PHY REG check	Result of internal bus interface check of DDR-PHY register.(for H3/M3/M3N)
Periodic Write DQ Training	Output only if Periodic Write DQ Training is enabled.
Dual Channel ECC / Single Channel ECC	Output Dual or Single Channel ECC setting. No output when Dual or single Channel ECC is disabled.
Boot mode	Whether to authorize (Secure boot / Normal boot).
Loading RTOS	Show destination address and image size of RTOS.
Loading CA5x Loader	Show destination address and image size of CA5x Loader. No output when CA53/CA57 is not kicked.
Whether CA53/CA57 is kicked	Which CPU is kicked by the Loader. No output when CA53/CA57 is not kicked.

Skipping LifeC setting for testing	Skip LifeC settings with build option RCAR_LIFEC_SETTING. LifeC settings are disabled by default.

SCIF2 outputs a log to a PC using the DEBUG SERIAL-0 on Salvator-X/XS or DEBUG SERIAL on Ebisu/Ebisu-4D.

1.2.3 Decide the starting mode.

Loader selects the starting mode. One is Normal mode to start without authentication, and the other is Secure mode with authentication.
Figure 1-3 shows flow of mode selection.

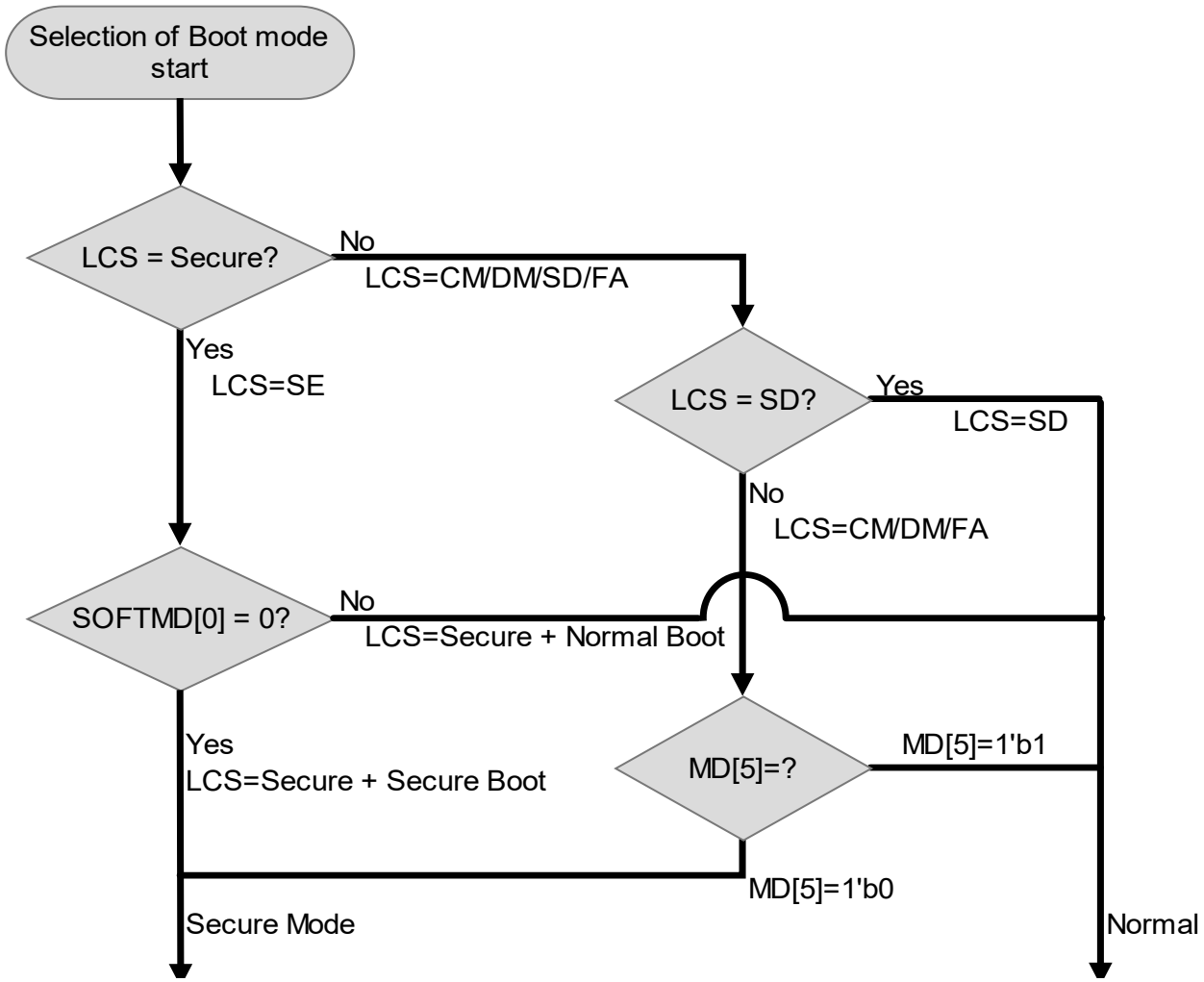


Figure 1-3 Flow of boot mode

Note) SOFTMD value read MFIS register in eFuse.

1.2.4 Loading the image

Loader loads images from HyperFlash/QSPI Flash to System RAM/SDRAM. About loading images refer to Table 4-1. These images are HyperFlash/QSPI Flash before loading.
If the boot mode is Secure mode, images are loaded refer to the load destination address and the image size that is included in the certificate after loading certification. If the boot mode is Normal mode, images are loaded refer to the load destination address and the image size that is included in the headers after loading headers. RTOS max image size is 1MB, CA5x Loader max image size is 170KB for R-Car H3/M3/M3N. CA5x Loader max image size is 80KB for R-Car E3.
Figure 1-4 shows the structure of a certificate and Figure 1-5 shows the structure of a header.

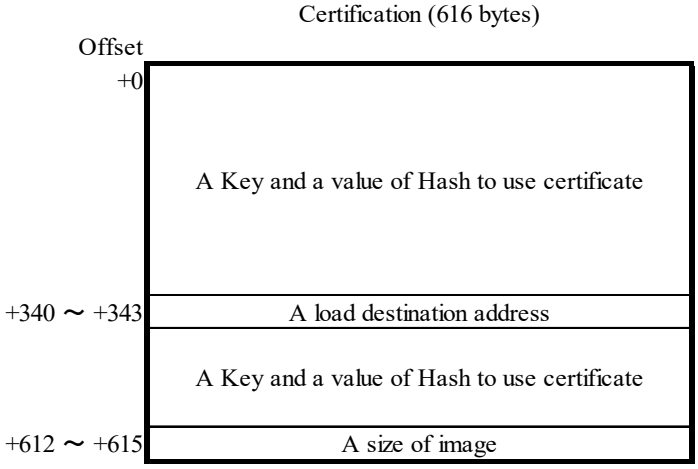


Figure 1-4 Structure of a certificate

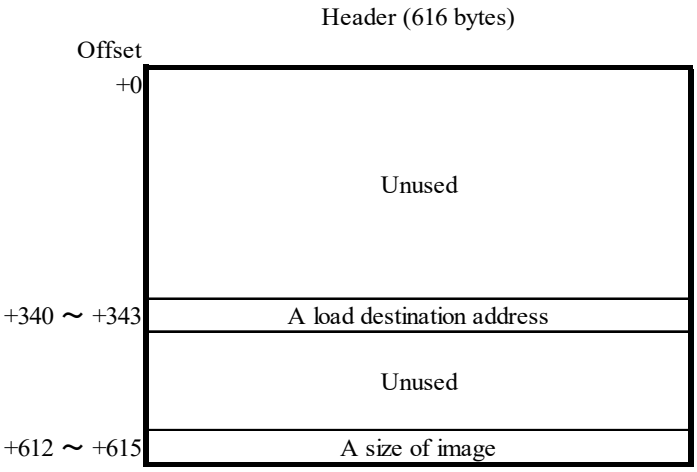


Figure 1-5 Structure of a header

A certificate includes information such as a value of a hash and authentication key for use to authenticate an image. A certificate and a header include a destination address for an image and a size of an image in the same offset and the same sector.
Figure 1-6 shows loading mechanism of images. As mentioned above, a destination address of an image and a size of an image get from a certificate or a header. Source address of an image has a fixed address in case of image of a CA5x Loader and a RTOS.

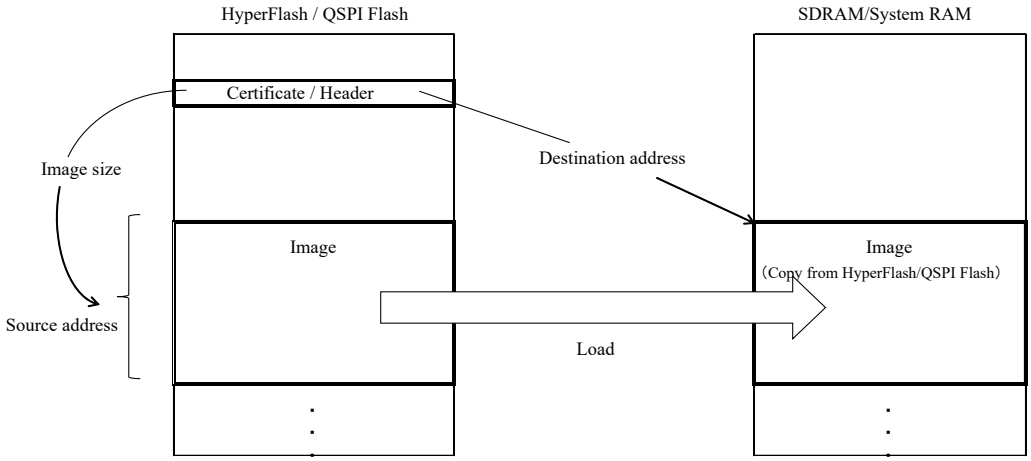


Figure 1-6 Loading mechanism

1.2.5 Authenticate the image using a certificate

The image is authenticated by the certificate if the boot mode is Secure mode. CA5x Loader and RTOS are started if the authentication is successful. And if it fails, Loader process is stopped after the display of the error messages.
Figure 1-7 shows flow of authentication in Secure mode.
When performing authentication, authentication of the image by the certificate is carried out by checking the return value of the Secure Boot function you pass the address of the certificate of authentication key and image. Secure Boot function is provided from the boot ROM.

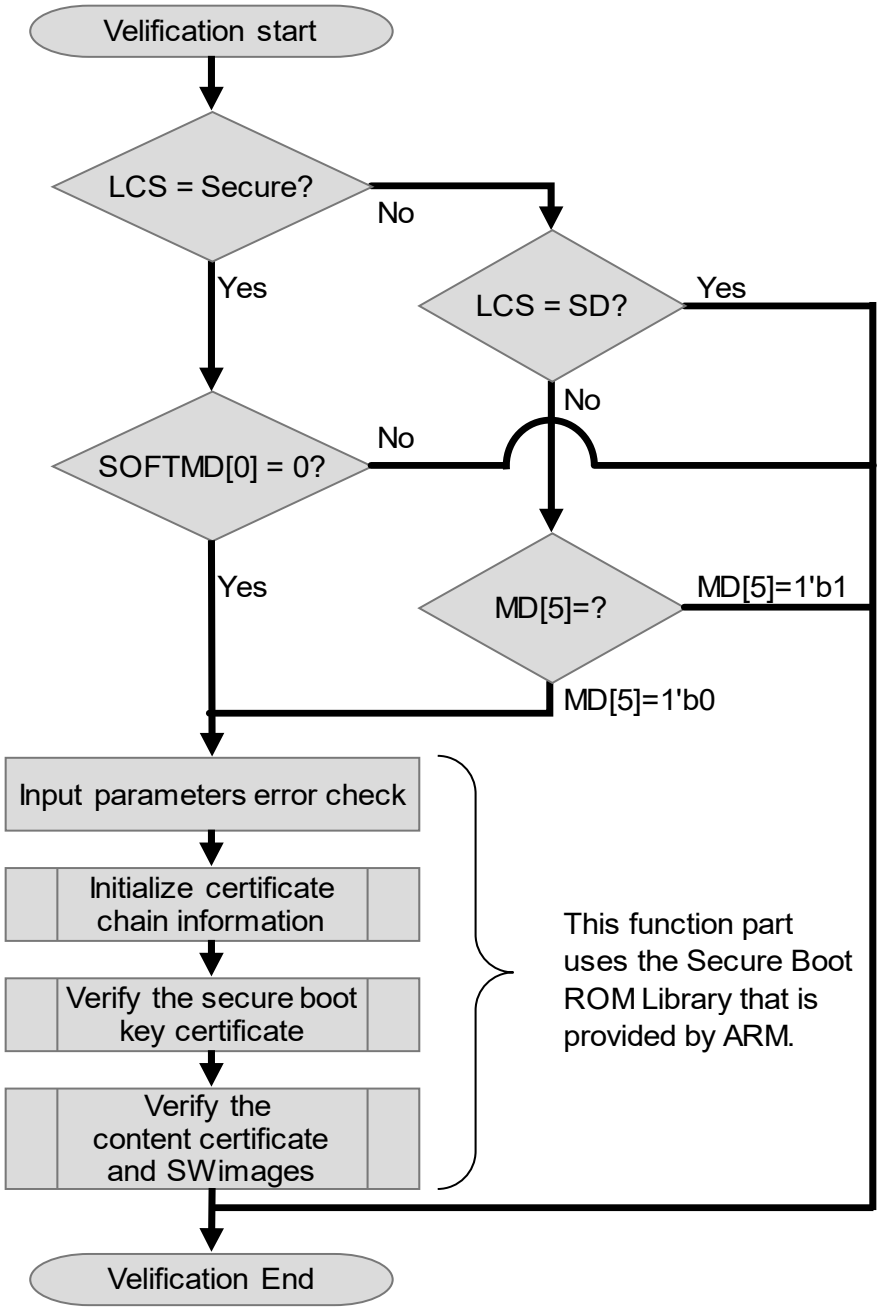


Figure 1-7 Flow of authentication in Secure mode

1.2.6 Releasing used resources

Release the resource used by Loader. This function initializes the hardware configuration changed by Loader and invalids the Instruction Cache.
This function executes the following process.

- Change the RT-DMAC to HW initial value.
- Change the SWDT to HW initial value.
- Change the SCIF2 to HW initial after the log has finished output.
- Disable the Instruction Cache.

1.2.7 Starting CA5x Loader

Start SCU and CPU0 of main CPU (CA57/53), and execute CA5x Loader. This function provides a build option for the selection of the main CPU.

1.2.8 Starting RTOS

Provide the function to jump to the RTOS.

1.2.9 Exception handling

When an exception is detected program is output error message.
For details about the error message is the followings.

- Undefined Instruction
  Occurred address

```
ERROR:
ERROR: CR7: Undefined Instruction occurred.
ERROR:        Occurred address : 0xeb000024
ERROR:
```

- Prefetch Abort
  IFSR, IFAR

```
ERROR:
ERROR: CR7: prefetch abort.
ERROR:        IFSR : 0x0000000d IFAR : 0xffff0000
ERROR:
```

- Data Abort
  Occurred address, DFSR, DFAR

```
ERROR:
ERROR: CR7: data abort.
ERROR:        data abort Occurred address : 0xe6303c80
ERROR:        DFSR : 0x00000001 DFAR : 0xeb027fe9
ERROR:
```

- IRQ interrupt
  IRQ error, occurred address

```
ERROR:
ERROR: CR7: System WDT overflow.
ERROR:        Occurred address : 0xe6303bf4
```

1.3 References

1.3.1 Related Document

The following table shows the document related to this function.

Table 1-3 Related Document

Number	Issue	Title	Edition
1	Renesas Electronics	R-Car Series, 3rd Generation User's Manual: Hardware	#0
2	Renesas Electronics	Linux Interface Specification Yocto recipe Start-Up Guide	#0
3	Renesas Electronics	R-CarH3-SiP System Evaluation Board Salvator-X Hardware Manual	#0
4	Renesas Electronics	R-CarM3-SiP System Evaluation Board Salvator-X Hardware Manual	#0
5	Renesas Electronics	R-CarH3-SiP System Evaluation Board Salvator-XS Hardware Manual	#0
6	Renesas Electronics	R-CarM3-SiP System Evaluation Board Salvator-XS Hardware Manual	#0
7	Renesas Electronics	R-CarM3N-SiP System Evaluation Board Salvator-XS Hardware Manual	#0
8	Renesas Electronics	R-CarE3 System Evaluation Board Ebisu Hardware Manual	#0
9	Renesas Electronics	R-CarE3 System Evaluation Board Ebisu-4D Hardware Manual	#0
10	Renesas Electronics	R-Car Series, 3rd Generation Safety Application Note	#0
11	Renesas Electronics	R-Car Series, 3rd Generation Hardware Description For Functional Safety	#0
12	Renesas Electronics	Initial Program Loader User's Manual: Software R-Car H3/M3/M3N/E3 Series	#0

#0 : This manual refers to the latest edition.

1.4 Restrictions

There are no restrictions in Loader.

2. Terminology

The following table shows the terminology to this function.

Table 2-1 Terminology

Terms	Explanation
API	Application Programming Interface
AXI	Advanced eXtensible Interface
CA53	Cortex-A53
CA57	Cortex-A57
CPG	Clock Pulse Generator
CR7	Cortex-R7
DBSC	DRAM Bus State Controller
DDR4	Double-Data-Rate4
DMAC	Direct Memory Access Controller
DRAM	Dynamic RAM
DTCM	Data TCM
ECC	Error Correction Code
FuSa	Functional Safety
GIC	Generic Interrupt Controller
GPIO	General-Purpose Input/Output ports
IPMMU	Intellectual Property MMU
IRQ	Interrupt ReQuest
ITCM	Instruction TCM
LCS	Life Cycle State
LPDDR4-SDRAM	Low Power DDR4 SDRAM
DDR3L-SDRAM	Double Data Rate 3 Low voltage SDRAM
LifeC	Life cycle Count
RT-DMAC	Real Time Direct Memory Access Controller
boot ROM	boot ROM program
PFC	Pin Function Controller
PRR	Product Register
QoS	Quality of Service
QSPI	Quad Serial Peripheral Interface
RAM	Random Access Memory
ROM	Read Only Memory
RPC	Reduced Pin Count
SCIF	Serial Communication Interface with FIFO
SDRAM	Synchronous DRAM
SoC	System on Chip
SPI	Serial Peripheral Interface
SWDT	System Watchdog Timer
TCM	Tightly Coupled 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 Series, 3rd Generation)

Name	Explanation
Evaluation Board	R-Car H3/M3/M3N-SiP System Evaluation Board (Salvator-X/XS) R-Car E3 System Evaluation Board (Ebisu/Ebisu-4D) Renesas Electronics
Host PC (Linux)	It is used CR7Loader build and debugging environment. Terminal software is executed.

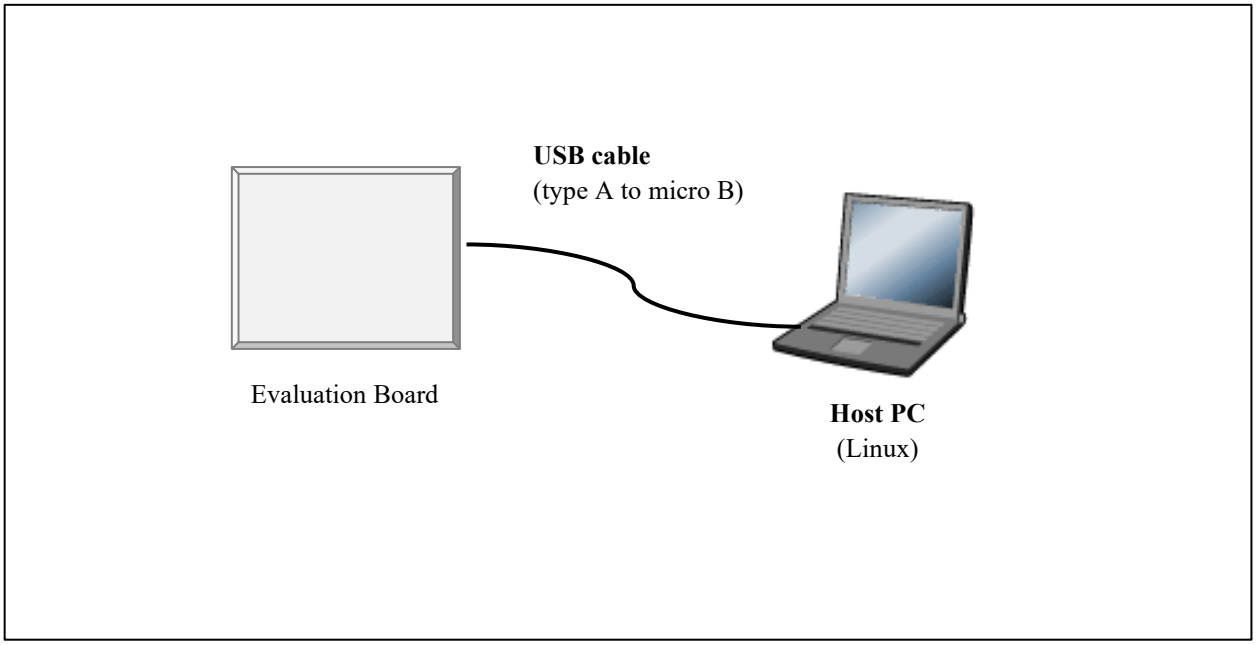


Figure 3-1 Recommended Environment

3.2 Module Configuration

Figure 3-2 shows module configuration of Salvator-X/XS and Ebisu/Ebisu-4D. Table 3-2 shows explanation of Hardware resource using in Figure 3-2.

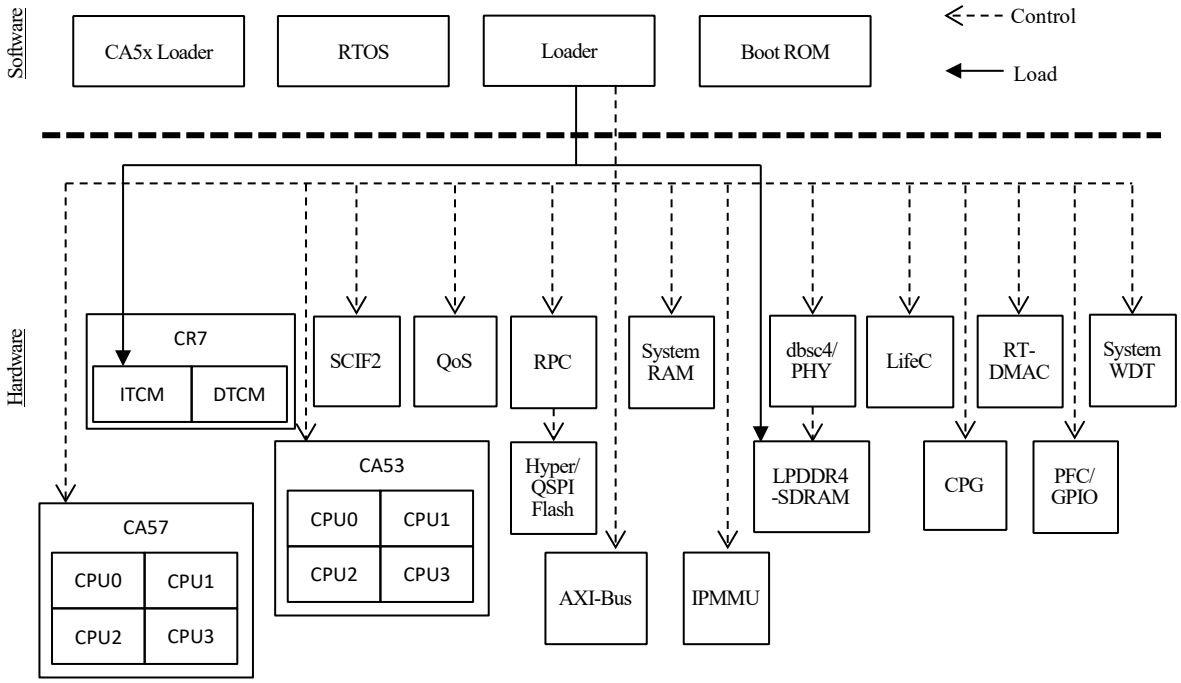


Figure 3-2 Module configuration

Note: Figure 3-2 is for R-Car H3.
In case of R-Car M3, there are not CPU2/CPU3 in CA57.
In case of R-Car M3N, there are not CA53 and CPU2/CPU3 in CA57.
In case of R-Car E3, there are not CA57 and CPU2/CPU3 in CA53.

Table 3-2 Hardware resource

Component	Uses
RPC/QSPI	Reading the HyperFlash/QSPI Flash.
PFC/GPIO	Setting the External pin.
LifeC	Setting the Secure protection of IP on SoC.
CPG	Setting the access protection of reset/stop module in CPG.
System RAM	Executing the Loader.
ITCM	Executing the Loader.
DTCM	Working memory and stack for Loader.
DBSC4/PHY	Initialization for access to the LPDDR4-SDRAM.
RAlloc/STATQ (1)	Setting the priority of the bus.
AXI-Bus	Enabling the ECC of System RAM and SDRAM.
IPMMU	Setting the System RAM area as continuous address
SWDT	Detecting the freeze of Loader by the interrupt.
RT-DMAC	Accelerating software loading with the DMA.
SCIF2	Displaying starting message and log message.
HyperFlash	Loading a source data from this device (External device).
LPDDR4-SDRAM	Loading a destination data from this device for R-Car H3/M3/M3N (External device).
DDR3L-SDRAM	Loading a destination data from this device for R-Car E3 (External device).
CA57/CA53	Starting the CPU when CA5x Loader is executed.

(1) RAlloc stands for Resource Alloc.
STATQ is the general name for M-STATQ and L-STATQ and T-STATQ.

3.3 Processing Flow Diagram

Table 3-3 shows the explanation of Loader's function.

Table 3-3 Explanation of Loader's functions

Function	Explanation
CPU initialization	Initialize registers and enable the lock step for CPU.
ECC initialization	Enable ECC to the error correction.
SWDT initialization	Output an error message detected by SWDT interruption.
PFC / GPIO initialization	Initialize PFC and GPIO to use peripherals.
Module stop initialization	Initialize module stop control to unused modules.
SCIF2 initialization	Initialize SCIF2 to output log.
GIC initialization	Initialize interrupt setting used by the Loader.
SDRAM/PHY initialization	Initialize SDRAM/PHY setting to store images.
RPC initialization	Initialize RPC to read data.
DMA initialization	Initialize RT-DMAC to transfer data at high speed.
Security and Safety setting	Perform Security and Safety setting to access control between IP.
QoS setting	Set the priority of the data that passes through the bus.
Release HW resource	Release SCIF2 and SWDT.

Figure 3-3 shows the flow of Loader in System RAM and Figure 3-4 shows the flow of Loader in TCM. The flow of Figure 3-4 is executed after the flow of Figure 3-3 is finished.
In case of cold boot, Loader executes the flow of following figure.
The difference of the CPU and ECC initialization refer to Table 6-21.

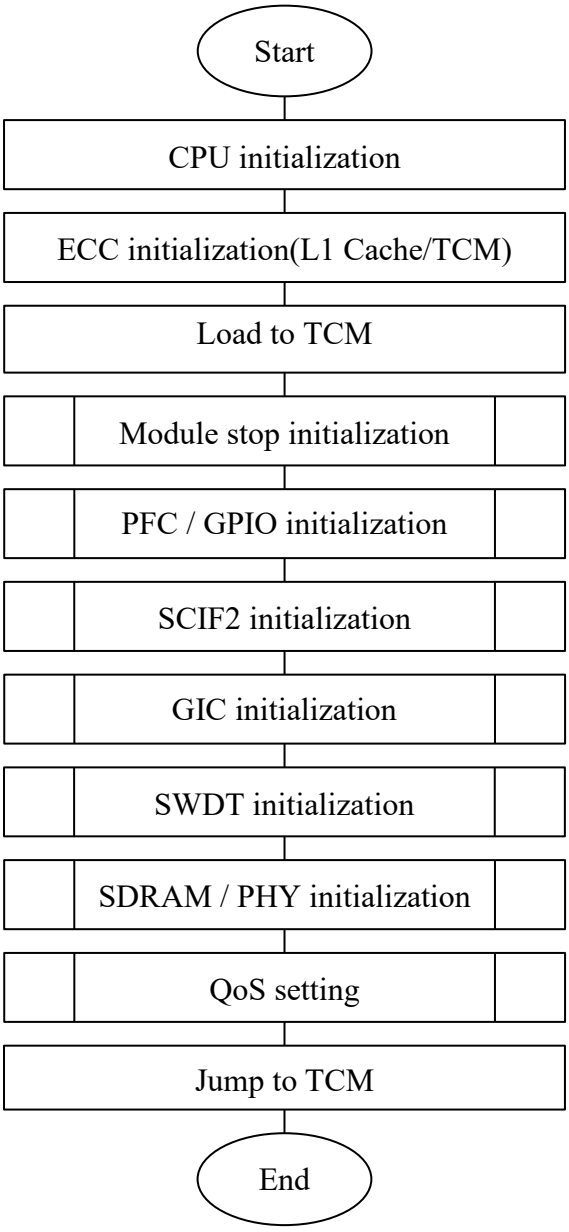


Figure 3-3 Flow of Loader processing

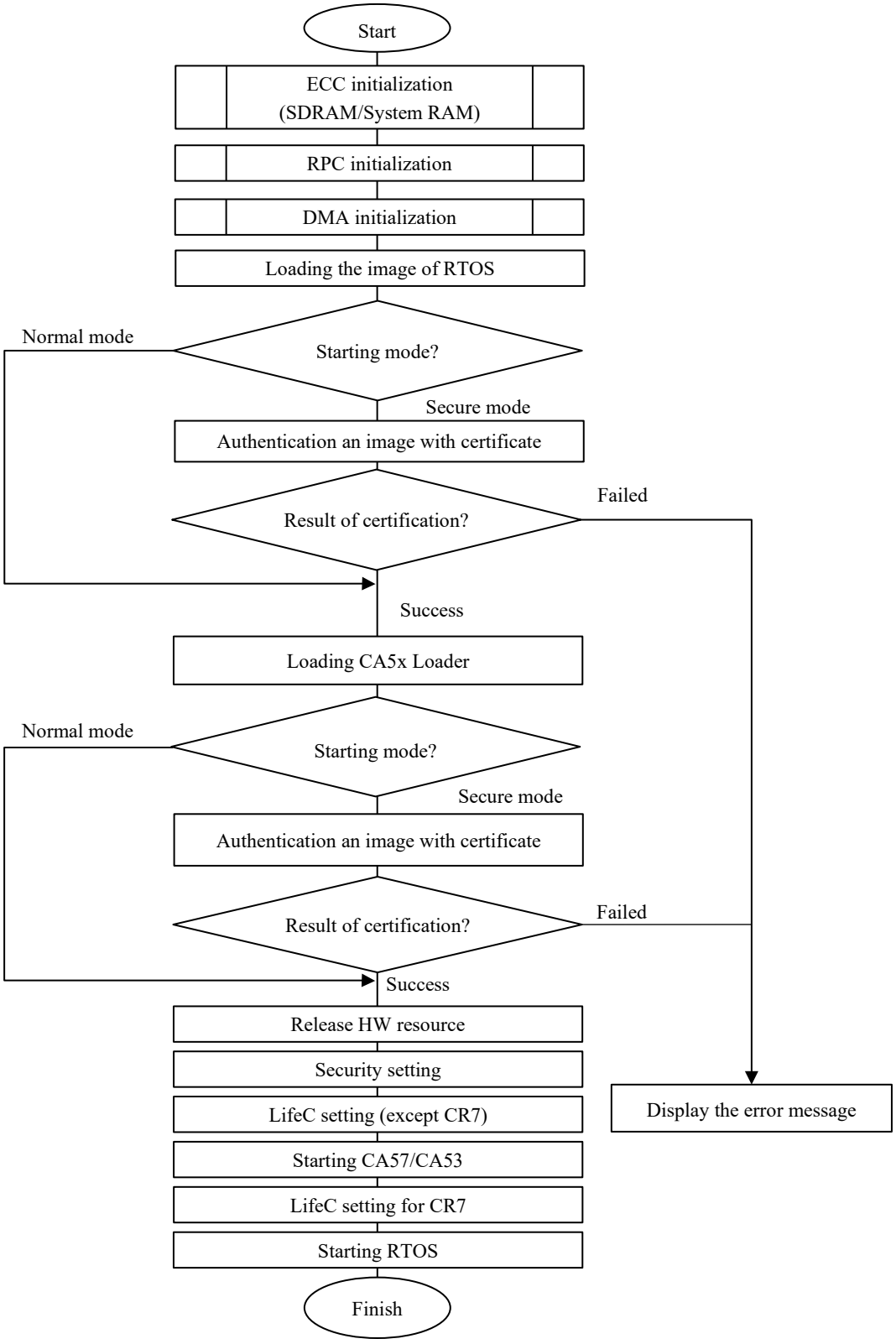


Figure 3-4 Flow of Loader processing in TCM

3.3.1 CPU initialization

Table 3-4 shows CPU initializing process and its explanation.

Table 3-4 CPU initializing process

Process	Explanation
Initialization of the general-purpose register	R0-R12 in the general-purpose register is cleared to zero.
Stack configuration	Initializing of Stack pointer (SP) of ABT/UND/FIQ/IRQ/SVC.
Global Timer configuration	Starting Global Timer. Decrease counter in 200 MHz.
TCM configuration	Enabling ITCM and DTCM.
Enable Lock step	Enabling Dual core lock step in CR7.
Enable the Instruction Cache	Enabling Instruction Cache and branch predictor after disabling the Instruction Cache. MPU keep the set value of Mask ROM. Data Cache is invalidated by Loader.
Clear bss section	Clear bss section with 0.
Copy data section	Copy data section from System RAM to Data TCM.
Load TCM Loader	Load TCM Loader from System RAM to Instruction TCM.

3.3.2 ECC initialization

Table 3-5 shows ECC initializing process and its explanation.

Table 3-5 ECC initializing process

Process	Explanation
Enable ECC of ITCM / DTCM / I-Cache	Enabling ECC of TCM and Cache is executed before the code that runs on System RAM and runs on TCM.
ECC setting of SDRAM and System RAM	This process is executed before loading images to SDRAM or System RAM.

3.3.3 SWDT initialization

This function enables SWDT. And it stops Loader after this function output an error message detected by SWDT interruption if Loader is not completed within 5 seconds.
SWDT must not be occur source conflict to be used by main CPU. So it resets SWDT registers to a hardware initial value before starting main CPU (CA57/53) and loading images.

3.3.4 PFC / GPIO initialization

Setting PFC and GPIO to peripherals used by Loader. The following is a set value required by Hardware Initialization for Salvator-X/XS and Ebisu/Ebisu-4D. Detail of the set value refers to 'R-Car Series, 3rd Generation User's Manual: Hardware: clause 6.1.4, 6D.1.4, and 7.1.4'.

Table 3-6 PFC setting for H3 Ver.2.0 / Ver.3.0

Register Name	Address	Setting value	Hardware Initialization
GPSR0	0xE6060100	0x0000FF00	0x00000000
GPSR1	0xE6060104	0x180FF0FF	0x10000000
GPSR2	0xE6060108	0x00007BBF	0x00000200
GPSR3	0xE606010C	0x00003F3F	0x00000000
GPSR4	0xE6060110	0x00019FFF	0x00000000
GPSR5	0xE6060114	0x002DFFF1	0x00000000
GPSR6	0xE6060118	0x3FDEC70F	0x00000000
GPSR7	0xE606011C	0x00000003	0x00000003
IPSR0	0xE6060200	0x00000000	0x00000000
IPSR1	0xE6060204	0x60003333	0x00000000
IPSR2	0xE6060208	0x06666666	0x00000000
IPSR3	0xE606020C	0x66666000	0x00000000
IPSR4	0xE6060210	0x00000666	0x00000000
IPSR5	0xE6060214	0x00000600	0x00000000
IPSR6	0xE6060218	0x66666000	0x00000000
IPSR7	0xE606021C	0x00000666	0x00000000
IPSR8	0xE6060220	0x11110000	0x00000000
IPSR9	0xE6060224	0x00000000	0x00000000
IPSR10	0xE6060228	0x10000000	0x00000000
IPSR11	0xE606022C	0x04000001	0x00000000
IPSR12	0xE6060230	0x00000400	0x00000000
IPSR13	0xE6060234	0x80000300	0x00000000
IPSR14	0xE6060238	0x00000038	0x00000000
IPSR15	0xE606023C	0x00000000	0x00000000
IPSR16	0xE6060240	0x00000000	0x00000000
IPSR17	0xE6060244	0x00000010	0x00000000
IPSR18	0xE6060248	0x00000000	0x00000000
DRVCTRL0	0xE6060300	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL1	0xE6060304	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL2	0xE6060308	0xFFFFFBBB	0xFFFFFFFF
DRVCTRL3	0xE606030C	0xBBBFFFFF	0xFFFFFFFF
DRVCTRL4	0xE6060310	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL5	0xE6060314	0xFFFFBBBB	0xFFFFFFFF
DRVCTRL6	0xE6060318	0xBBBBFFFF	0xFFFFFFFF
DRVCTRL7	0xE606031C	0xBBBBBBBB	0xFFFFFFFF
DRVCTRL8	0xE6060320	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL9	0xE6060324	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL10	0xE6060328	0xFFBBBBBB	0xFFFFFFFF
DRVCTRL11	0xE606032C	0xBBFFFFFF	0xFFFFFFFF

DRVCTRL12	0xE6060330	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL13	0xE6060334	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL14	0xE6060338	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL15	0xE606033C	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL16	0xE6060340	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL17	0xE6060344	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL18	0xE6060348	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL19	0xE606034C	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL20	0xE6060350	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL21	0xE6060354	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL22	0xE6060358	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL23	0xE606035C	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL24	0xE6060360	0xFFFFFFFF	0xFFFFFFFF
POCCTRL0	0xE6060380	0x3FF8003F	0x3FFFFFFF
PUEN0	0xE6060400	0x00000FFF	0xCFFF9000
PUEN1	0xE6060404	0x00100234	0x00B24FFF
PUEN2	0xE6060408	0x000004C4	0xFF0005E6
PUEN3	0xE606040C	0x00000200	0x000002F7
PUEN4	0xE6060410	0x3E000000	0xFFFFFF00
PUEN5	0xE6060414	0x1F000805	0xFF7FFF87
PUEN6	0xE6060418	0x00000006	0x0000007F
PUD0	0xE6060440	0x00005FBF	0x30007FFF
PUD1	0xE6060444	0x00300FFE	0xFF7FBFFE
PUD2	0xE6060448	0x330001E6	0x33FFFBFF
PUD3	0xE606044C	0x000002E0	0xFFFFFEEC
PUD4	0xE6060450	0xFFFFFF00	0xFFFFFFFF
PUD5	0xE6060454	0x7F5FFF87	0x7FDFFFFF
PUD6	0xE6060458	0x00000055	0x00000055
MOD_SEL0	0xE6060500	0x00000000	0x00000000
MOD_SEL1	0xE6060504	0x00000000	0x00000000
MOD_SEL2	0xE6060508	0x00000000	0x00000000

Table 3-7 GPIO setting for H3 Ver.2.0 / Ver.3.0

Register Name	Address	Setting value	Hardware Initialization
GPIO_IOINTSEL0	0xE6050000	0x00000000	0x00000000
GPIO_IOINTSEL1	0xE6051000	0x00000000	0x00000000
GPIO_IOINTSEL2	0xE6052000	0x00000000	0x00000000
GPIO_IOINTSEL3	0xE6053000	0x00000000	0x00000000
GPIO_IOINTSEL4	0xE6054000	0x00000000	0x00000000
GPIO_IOINTSEL5	0xE6055000	0x00000000	0x00000000
GPIO_IOINTSEL6	0xE6055400	0x00000000	0x00000000
GPIO_IOINTSEL7	0xE6055800	0x00000000	0x00000000
GPIO_INOUTSEL0	0xE6050004	0x00000000	0x00000000
GPIO_INOUTSEL1	0xE6051004	0x01000A00	0x00000000
GPIO_INOUTSEL2	0xE6052004	0x00000400	0x00000000
GPIO_INOUTSEL3	0xE6053004	0x0000C000	0x00000000
GPIO_INOUTSEL4	0xE6054004	0x00000000	0x00000000
GPIO_INOUTSEL5	0xE6055004	0x0000020E	0x00000000
GPIO_INOUTSEL6	0xE6055404	0x00013880	0x00000000
GPIO_INOUTSEL7	0xE6055804	0x00000000	0x00000000
GPIO_OUTDT1	0xE6051008	0x00000000	0x00000000
GPIO_OUTDT2	0xE6052008	0x00000400	0x00000000
GPIO_OUTDT3	0xE6053008	0x0000C000	0x00000000
GPIO_OUTDT5	0xE6055008	0x00000006	0x00000000
GPIO_OUTDT6	0xE6055408	0x00003880	0x00000000
GPIO_POSNEG0	0xE6050020	0x00000000	0x00000000
GPIO_POSNEG1	0xE6051020	0x00000000	0x00000000
GPIO_POSNEG2	0xE6052020	0x00000000	0x00000000
GPIO_POSNEG3	0xE6053020	0x00000000	0x00000000
GPIO_POSNEG4	0xE6054020	0x00000000	0x00000000
GPIO_POSNEG5	0xE6055020	0x00000000	0x00000000
GPIO_POSNEG6	0xE6055420	0x00000000	0x00000000
GPIO_POSNEG7	0xE6055820	0x00000000	0x00000000

Table 3-8 PFC setting for M3 Ver.1.1 / Ver.1.2 / Ver.1.3 / Ver.3.0

Register Name	Address	Setting value	Hardware Initialization
GPSR0	0xE6060100	0x0000FF00	0x00000000
GPSR1	0xE6060104	0x180FF0FF	0x10000000
GPSR2	0xE6060108	0x00007BBF	0x00000200
GPSR3	0xE606010C	0x00018F5F	0x00000000
GPSR4	0xE6060110	0x00019FFF	0x00000000
GPSR5	0xE6060114	0x002DFFF1	0x00000000
GPSR6	0xE6060118	0x3FDEC70F	0x00000000
GPSR7	0xE606011C	0x00000003	0x00000003
IPSR0	0xE6060200	0x00000000	0x00000000
IPSR1	0xE6060204	0x60003333	0x00000000
IPSR2	0xE6060208	0x06666666	0x00000000
IPSR3	0xE606020C	0x66666000	0x00000000
IPSR4	0xE6060210	0x00000666	0x00000000
IPSR5	0xE6060214	0x00000600	0x00000000
IPSR6	0xE6060218	0x66666000	0x00000000
IPSR7	0xE606021C	0x00000666	0x00000000
IPSR8	0xE6060220	0x11110000	0x00000000
IPSR9	0xE6060224	0x00000000	0x00000000
IPSR10	0xE6060228	0x10000000	0x00000000
IPSR11	0xE606022C	0x04000001	0x00000000
IPSR12	0xE6060230	0x00000400	0x00000000
IPSR13	0xE6060234	0x80000300	0x00000000
IPSR14	0xE6060238	0x00000038	0x00000000
IPSR15	0xE606023C	0x00000000	0x00000000
IPSR16	0xE6060240	0x00000000	0x00000000
IPSR17	0xE6060244	0x00000010	0x00000000
IPSR18	0xE6060248	0x00000000	0x00000000
DRVCTRL0	0xE6060300	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL1	0xE6060304	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL2	0xE6060308	0xFFFFFBBB	0xFFFFFFFF
DRVCTRL3	0xE606030C	0xBBBFFFFF	0xFFFFFFFF
DRVCTRL4	0xE6060310	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL5	0xE6060314	0xFFFFBBBB	0xFFFFFFFF
DRVCTRL6	0xE6060318	0xBBBBFFFF	0xFFFFFFFF
DRVCTRL7	0xE606031C	0xBBBBBBBB	0xFFFFFFFF
DRVCTRL8	0xE6060320	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL9	0xE6060324	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL10	0xE6060328	0xFFBBBBBB	0xFFFFFFFF
DRVCTRL11	0xE606032C	0xBBFFFFFF	0xFFFFFFFF
DRVCTRL12	0xE6060330	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL13	0xE6060334	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL14	0xE6060338	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL15	0xE606033C	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL16	0xE6060340	0xFFFFFFFF	0xFFFFFFFF

DRVCTRL17	0xE6060344	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL18	0xE6060348	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL19	0xE606034C	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL20	0xE6060350	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL21	0xE6060354	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL22	0xE6060358	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL23	0xE606035C	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL24	0xE6060360	0xFFFFFFFF	0xFFFFFFFF
POCCTRL0	0xE6060380	0x3FF8003F	0x3FFFFFFF
PUEN0	0xE6060400	0x00000FFF	0xCFFF9000
PUEN1	0xE6060404	0x00100234	0x00B24FFF
PUEN2	0xE6060408	0x000004C4	0xFF0005E6
PUEN3	0xE606040C	0x00000200	0x000002F5
PUEN4	0xE6060410	0x3E000000	0xFFFFFF00
PUEN5	0xE6060414	0x1F000805	0xFFFFFF87
PUEN6	0xE6060418	0x00000006	0x000000FF
PUD0	0xE6060440	0x00005FBF	0x30007FFF
PUD1	0xE6060444	0x00300FFE	0xFF7FBFFE
PUD2	0xE6060448	0x330001E6	0x33FFFBFF
PUD3	0xE606044C	0x000002E0	0xFFFFFEEC
PUD4	0xE6060450	0xFFFFFF00	0xFFFFFFFF
PUD5	0xE6060454	0x7F5FFF87	0x7F5FFFFF
PUD6	0xE6060458	0x00000055	0x00000055
MOD_SEL0	0xE6060500	0x00000000	0x00000000
MOD_SEL1	0xE6060504	0x00000000	0x00000000
MOD_SEL2	0xE6060508	0x00000000	0x00000000

Table 3-9 GPIO setting for M3 Ver.1.1 / Ver.1.2 / Ver.1.3 / Ver.3.0

Register Name	Address	Setting value	Hardware Initialization
GPIO_IOINTSEL0	0xE6050000	0x00000000	0x00000000
GPIO_IOINTSEL1	0xE6051000	0x00000000	0x00000000
GPIO_IOINTSEL2	0xE6052000	0x00000000	0x00000000
GPIO_IOINTSEL3	0xE6053000	0x00000000	0x00000000
GPIO_IOINTSEL4	0xE6054000	0x00000000	0x00000000
GPIO_IOINTSEL5	0xE6055000	0x00000000	0x00000000
GPIO_IOINTSEL6	0xE6055400	0x00000000	0x00000000
GPIO_IOINTSEL7	0xE6055800	0x00000000	0x00000000
GPIO_INOUTSEL0	0xE6050004	0x00000000	0x00000000
GPIO_INOUTSEL1	0xE6051004	0x01000A00	0x00000000
GPIO_INOUTSEL2	0xE6052004	0x00000400	0x00000000
GPIO_INOUTSEL3	0xE6053004	0x0000C000	0x00000000
GPIO_INOUTSEL4	0xE6054004	0x00000000	0x00000000
GPIO_INOUTSEL5	0xE6055004	0x0000020E	0x00000000
GPIO_INOUTSEL6	0xE6055404	0x00013880	0x00000000
GPIO_INOUTSEL7	0xE6055804	0x00000000	0x00000000
GPIO_OUTDT1	0xE6051008	0x00000000	0x00000000
GPIO_OUTDT2	0xE6052008	0x00000400	0x00000000
GPIO_OUTDT3	0xE6053008	0x0000C000	0x00000000
GPIO_OUTDT5	0xE6055008	0x00000006	0x00000000
GPIO_OUTDT6	0xE6055408	0x00003880	0x00000000
GPIO_POSNEG0	0xE6050020	0x00000000	0x00000000
GPIO_POSNEG1	0xE6051020	0x00000000	0x00000000
GPIO_POSNEG2	0xE6052020	0x00000000	0x00000000
GPIO_POSNEG3	0xE6053020	0x00000000	0x00000000
GPIO_POSNEG4	0xE6054020	0x00000000	0x00000000
GPIO_POSNEG5	0xE6055020	0x00000000	0x00000000
GPIO_POSNEG6	0xE6055420	0x00000000	0x00000000
GPIO_POSNEG7	0xE6055820	0x00000000	0x00000000

Table 3-10 PFC setting for M3N Ver.1.1

Register Name	Address	Setting value	Hardware Initialization
GPSR0	0xE6060100	0x0000FF00	0x00000000
GPSR1	0xE6060104	0x180FF0FF	0x10000000
GPSR2	0xE6060108	0x00007BBF	0x00000200
GPSR3	0xE606010C	0x00018F5F	0x00000000
GPSR4	0xE6060110	0x00019FFF	0x00000000
GPSR5	0xE6060114	0x002DFFF1	0x00000000
GPSR6	0xE6060118	0x3FDEC70F	0x00000000
GPSR7	0xE606011C	0x00000003	0x00000003
IPSR0	0xE6060200	0x00000000	0x00000000
IPSR1	0xE6060204	0x60003333	0x00000000
IPSR2	0xE6060208	0x06666666	0x00000000
IPSR3	0xE606020C	0x66666000	0x00000000
IPSR4	0xE6060210	0x00000666	0x00000000
IPSR5	0xE6060214	0x00000600	0x00000000
IPSR6	0xE6060218	0x66666000	0x00000000
IPSR7	0xE606021C	0x00000666	0x00000000
IPSR8	0xE6060220	0x11110000	0x00000000
IPSR9	0xE6060224	0x00000000	0x00000000
IPSR10	0xE6060228	0x10000000	0x00000000
IPSR11	0xE606022C	0x04000001	0x00000000
IPSR12	0xE6060230	0x00000400	0x00000000
IPSR13	0xE6060234	0x80000300	0x00000000
IPSR14	0xE6060238	0x00000038	0x00000000
IPSR15	0xE606023C	0x00000000	0x00000000
IPSR16	0xE6060240	0x00000000	0x00000000
IPSR17	0xE6060244	0x00000010	0x00000000
IPSR18	0xE6060248	0x00000000	0x00000000
DRVCTRL0	0xE6060300	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL1	0xE6060304	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL2	0xE6060308	0xFFFFFBBB	0xFFFFFFFF
DRVCTRL3	0xE606030C	0xBBBFFFFF	0xFFFFFFFF
DRVCTRL4	0xE6060310	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL5	0xE6060314	0xFFFFBBBB	0xFFFFFFFF
DRVCTRL6	0xE6060318	0xBBBBFFFF	0xFFFFFFFF
DRVCTRL7	0xE606031C	0xBBBBBBBB	0xFFFFFFFF
DRVCTRL8	0xE6060320	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL9	0xE6060324	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL10	0xE6060328	0xFFBBBBBB	0xFFFFFFFF
DRVCTRL11	0xE606032C	0xBBFFFFFF	0xFFFFFFFF
DRVCTRL12	0xE6060330	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL13	0xE6060334	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL14	0xE6060338	0xFFDDDDDD	0xFFFFFFFF
DRVCTRL15	0xE606033C	0xDDDDDFFF	0xFFFFFFFF
DRVCTRL16	0xE6060340	0xFFFFFFFF	0xFFFFFFFF

DRVCTRL17	0xE6060344	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL18	0xE6060348	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL19	0xE606034C	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL20	0xE6060350	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL21	0xE6060354	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL22	0xE6060358	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL23	0xE606035C	0xFFFFFFFF	0xFFFFFFFF
DRVCTRL24	0xE6060360	0xFFFFFFFF	0xFFFFFFFF
POCCTRL0	0xE6060380	0x3FF8003F	0x3FFFFFFF
PUEN0	0xE6060400	0x00000FFF	0xCFFF9000
PUEN1	0xE6060404	0x00100234	0x00B24FFF
PUEN2	0xE6060408	0x000004C4	0xFF0005E6
PUEN3	0xE606040C	0x00000200	0x000002F5
PUEN4	0xE6060410	0x3E000000	0xFFFFFF00
PUEN5	0xE6060414	0x1F000805	0xFF7FFF87
PUEN6	0xE6060418	0x00000006	0x0000007F
PUD0	0xE6060440	0x00005FBF	0x30007FFF
PUD1	0xE6060444	0x00300FFE	0xFF7FBFFE
PUD2	0xE6060448	0x330001E6	0x33FFFBFF
PUD3	0xE606044C	0x000002E0	0xFFFFFEEC
PUD4	0xE6060450	0xFFFFFF00	0xFFFFFFFF
PUD5	0xE6060454	0x7F5FFF87	0x7FDFFFFF
PUD6	0xE6060458	0x00000055	0x00000055
MOD_SEL0	0xE6060500	0x00000000	0x00000000
MOD_SEL1	0xE6060504	0x00000000	0x00000000
MOD_SEL2	0xE6060508	0x00000000	0x00000000

Table 3-11 GPIO setting for M3N Ver.1.1

Register Name	Address	Setting value	Hardware Initialization
GPIO_IOINTSEL0	0xE6050000	0x00000000	0x00000000
GPIO_IOINTSEL1	0xE6051000	0x00000000	0x00000000
GPIO_IOINTSEL2	0xE6052000	0x00000000	0x00000000
GPIO_IOINTSEL3	0xE6053000	0x00000000	0x00000000
GPIO_IOINTSEL4	0xE6054000	0x00000000	0x00000000
GPIO_IOINTSEL5	0xE6055000	0x00000000	0x00000000
GPIO_IOINTSEL6	0xE6055400	0x00000000	0x00000000
GPIO_IOINTSEL7	0xE6055800	0x00000000	0x00000000
GPIO_INOUTSEL0	0xE6050004	0x00000000	0x00000000
GPIO_INOUTSEL1	0xE6051004	0x01000A00	0x00000000
GPIO_INOUTSEL2	0xE6052004	0x00000400	0x00000000
GPIO_INOUTSEL3	0xE6053004	0x0000C000	0x00000000
GPIO_INOUTSEL4	0xE6054004	0x00000000	0x00000000
GPIO_INOUTSEL5	0xE6055004	0x0000020E	0x00000000
GPIO_INOUTSEL6	0xE6055404	0x00013880	0x00000000
GPIO_INOUTSEL7	0xE6055804	0x00000000	0x00000000
GPIO_OUTDT1	0xE6051008	0x00000000	0x00000000
GPIO_OUTDT2	0xE6052008	0x00000400	0x00000000
GPIO_OUTDT3	0xE6053008	0x0000C000	0x00000000
GPIO_OUTDT5	0xE6055008	0x00000006	0x00000000
GPIO_OUTDT6	0xE6055408	0x00003880	0x00000000
GPIO_POSNEG0	0xE6050020	0x00000000	0x00000000
GPIO_POSNEG1	0xE6051020	0x00000000	0x00000000
GPIO_POSNEG2	0xE6052020	0x00000000	0x00000000
GPIO_POSNEG3	0xE6053020	0x00000000	0x00000000
GPIO_POSNEG4	0xE6054020	0x00000000	0x00000000
GPIO_POSNEG5	0xE6055020	0x00000000	0x00000000
GPIO_POSNEG6	0xE6055420	0x00000000	0x00000000
GPIO_POSNEG7	0xE6055820	0x00000000	0x00000000

Table 3-12 PFC setting for E3 Ver.1.0 / Ver.1.1

Register Name	Address	Setting value	Hardware Initialization
GPSR0	0xE6060100	0x00018FEF	0x00000000
GPSR1	0xE6060104	0x006FFF1F	0x10000000
GPSR2	0xE6060108	0x0CBFFFFF	0x00000200
GPSR3	0xE606010C	0x0000FFFF	0x00000000
GPSR4	0xE6060110	0x000007FF	0x00000000
GPSR5	0xE6060114	0x0001C398	0x00000000
GPSR6	0xE6060118	0x00039FEF	0x00000000
IPSR0	0xE6060200	0x00000000	0x00000000
IPSR1	0xE6060204	0x00000000	0x00000000
IPSR2	0xE6060208	0x10020000	0x00000000
IPSR3	0xE606020C	0x50005551	0x00000000
IPSR4	0xE6060210	0x55555555	0x00000000
IPSR5	0xE6060214	0x55555555	0x00000000
IPSR6	0xE6060218	0x55525505	0x00000000
IPSR7	0xE606021C	0x00550005	0x00000000
IPSR8	0xE6060220	0x00000000	0x00000000
IPSR9	0xE6060224	0x00000000	0x00000000
IPSR10	0xE6060228	0x00000000	0x00000000
IPSR11	0xE606022C	0x00220000	0x00000000
IPSR12	0xE6060230	0x00000020	0x00000000
IPSR13	0xE6060234	0x00000211	0x00000000
IPSR14	0xE6060238	0x30000000	0x00000000
IPSR15	0xE606023C	0x00011003	0x00000000
IOCTRL30	0xE6060380	0x0007FFFF	0xFFFFFFFF
IOCTRL32	0xE6060388	0xFFFFFFFE	0xFFFFFFFF
IOCTRL40	0xE60603C0	0x00000000	0x00000000
PUEN0	0xE6060400	0xFFF00000	0xCFFF9000
PUEN1	0xE6060404	0x00000000	0x00B24FFF
PUEN2	0xE6060408	0x00000004	0xFF0005E6
PUEN3	0xE606040C	0x00000000	0x000002F5
PUEN4	0xE6060410	0x07800010	0xFFFFFF00
PUEN5	0xE6060414	0x00000000	0xFF7FFF87
PUD0	0xE6060440	0xFDF80000	0x30007FFF
PUD1	0xE6060444	0xCE298464	0xFF7FBFFE
PUD2	0xE6060448	0xA4C380F4	0x33FFFBFF
PUD3	0xE606044C	0x0000079F	0xFFFFFEEC
PUD4	0xE6060450	0xFFF0FFFF	0xFFFFFFFF
PUD5	0xE6060454	0x40000000	0x7FDFFFFF
MOD_SEL0	0xE6060500	0x00000000	0x00000000
MOD_SEL1	0xE6060504	0x10000000	0x00000000

Table 3-13 GPIO setting for E3 Ver.1.0 / Ver.1.1

Register Name	Address	Setting value	Hardware Initialization
GPIO_IOINTSEL0	0xE6050000	0x00020000	0x00000000
GPIO_IOINTSEL1	0xE6051000	0x00000000	0x00000000
GPIO_IOINTSEL2	0xE6052000	0x00000000	0x00000000
GPIO_IOINTSEL3	0xE6053000	0x00000000	0x00000000
GPIO_IOINTSEL4	0xE6054000	0x00000000	0x00000000
GPIO_IOINTSEL5	0xE6055000	0x00000000	0x00000000
GPIO_IOINTSEL6	0xE6055400	0x00000000	0x00000000
GPIO_INOUTSEL0	0xE6050004	0x00000000	0x00000000
GPIO_INOUTSEL1	0xE6051004	0x00100020	0x00000000
GPIO_INOUTSEL2	0xE6052004	0x03000000	0x00000000
GPIO_INOUTSEL3	0xE6053004	0x00008000	0x00000000
GPIO_INOUTSEL4	0xE6054004	0x00000000	0x00000000
GPIO_INOUTSEL5	0xE6055004	0x00060000	0x00000000
GPIO_INOUTSEL6	0xE6055404	0x00004000	0x00000000
GPIO_OUTDT0	0xE6050008	0x00000010	0x00000000
GPIO_OUTDT1	0xE6051008	0x00100000	0x00000000
GPIO_OUTDT2	0xE6052008	0x00000000	0x00000000
GPIO_OUTDT3	0xE6053008	0x00008000	0x00000000
GPIO_OUTDT5	0xE6055008	0x00060000	0x00000000
GPIO_OUTDT6	0xE6055408	0x00000000	0x00000000
GPIO_POSNEG0	0xE6050020	0x00000000	0x00000000
GPIO_POSNEG1	0xE6051020	0x00000000	0x00000000
GPIO_POSNEG2	0xE6052020	0x00000000	0x00000000
GPIO_POSNEG3	0xE6053020	0x00000000	0x00000000
GPIO_POSNEG4	0xE6054020	0x00000000	0x00000000
GPIO_POSNEG5	0xE6055020	0x00000000	0x00000000
GPIO_POSNEG6	0xE6055420	0x00000000	0x00000000

3.3.5 Module stop initialization

Stop the clock of the unused modules. The following is a set value required by Hardware Initialization for Salvator-X/XS and Ebisu/Ebisu-4D. Detail of the set value refers to 'R-Car Series, 3rd Generation User's Manual: Hardware: clause 8A.1.3'.

Table 3-14 Module stop setting for H3 Ver.2.0 / Ver.3.0

Register Name	Address	Setting value	Hardware Initialization
RMSTPCR0	0xE6150110	0x00010000	0x00000000
RMSTPCR1	0xE6150114	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR2	0xE6150118	0x040E0FDC	0x00000000
RMSTPCR3	0xE615011C	0xFFFFFFDF	0xFFFFFFDF
RMSTPCR4	0xE6150120	0x80000004	0x00000000
RMSTPCR5	0xE6150124	0xC3FFFFFF	0x83BFFFFF
RMSTPCR6	0xE6150128	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR7	0xE615012C	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR8	0xE6150980	0x01F1FFF7	0x00F1FFF7
RMSTPCR9	0xE6150984	0xFFFFFFFF	0x03F1E017
RMSTPCR10	0xE6150988	0xFFFEFFE0	0xFFFEFFE0
RMSTPCR11	0xE615098C	0x000000B7	0x000000B7

Table 3-15 Module stop setting for M3 Ver.1.1 / Ver.1.2 / Ver.1.3 / Ver.3.0

Register Name	Address	Setting value	Hardware Initialization
RMSTPCR0	0xE6150110	0x00000000	0x00000000
RMSTPCR1	0xE6150114	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR2	0xE6150118	0x040E0FDC	0x00000000
RMSTPCR3	0xE615011C	0xFFFFFFDF	0xFFFFFFDF
RMSTPCR4	0xE6150120	0x80000004	0x00000000
RMSTPCR5	0xE6150124	0xC3FFFFFF	0x83BFFFFF
RMSTPCR6	0xE6150128	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR7	0xE615012C	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR8	0xE6150980	0x01F1FFF7	0x00F1FFF7
RMSTPCR9	0xE6150984	0xFFFFFFFF	0x03F1E017
RMSTPCR10	0xE6150988	0xFFFEFFE0	0xFFFEFFE0
RMSTPCR11	0xE615098C	0x000000B7	0x000000B7

Table 3-16 Module stop setting for M3N Ver.1.1

Register Name	Address	Setting value	Hardware Initialization
RMSTPCR0	0xE6150110	0x00010000	0x00000000
RMSTPCR1	0xE6150114	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR2	0xE6150118	0x040E0FDC	0x00000000
RMSTPCR3	0xE615011C	0xFFFFFFDF	0xFFFFFFDF
RMSTPCR4	0xE6150120	0x80000004	0x00000000
RMSTPCR5	0xE6150124	0xC3FFFFFF	0x83BFFFFF
RMSTPCR6	0xE6150128	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR7	0xE615012C	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR8	0xE6150980	0x00F1FFF7	0x00F1FFF7
RMSTPCR9	0xE6150984	0xFFFFFFFF	0x03F1E017
RMSTPCR10	0xE6150988	0xFFFEFFE0	0xFFFFFFE0
RMSTPCR11	0xE615098C	0x000000B7	0x000000B7

Table 3-17 Module stop setting for E3 Ver.1.0 / Ver.1.1

Register Name	Address	Setting value	Hardware Initialization
RMSTPCR0	0xE6150110	0x00010000	0x00000000
RMSTPCR1	0xE6150114	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR2	0xE6150118	0x000E0FDC	0x00000000
RMSTPCR3	0xE615011C	0xFFFFFFDF	0xFFFFFFDF
RMSTPCR4	0xE6150120	0x80000004	0x00000000
RMSTPCR5	0xE6150124	0xC3FFFFFF	0x83BFFFFF
RMSTPCR6	0xE6150128	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR7	0xE615012C	0xFFFFFFFF	0xFFFFFFFF
RMSTPCR8	0xE6150980	0x00F1FFF7	0x00F1FFF7
RMSTPCR9	0xE6150984	0xFFFFFFDF	0x03F1E017
RMSTPCR10	0xE6150988	0xFFFFFFE8	0xFFFFFFE0
RMSTPCR11	0xE615098C	0x000000B7	0x000000B7

3.3.6 SCIF2 initialization

Initialize SCIF2 to output log.

Shows a pin name using SCIF2 setting and shows SCIF2 setting to output log to Host PC.

Table 3-18 SCIF2 pin name

Pin name	in/out	Function
TX2_A	out	Send data

Table 3-19 SCIF2 setting

Item	Setting
Baud rate	115200bps
Data size	8bit
Parity	None
Stop bit	1bit
Flow control	None

3.3.7 GIC initialization

Initialize interrupt setting used by the Loader. For example, the Loader uses SWDT.

3.3.8 SDRAM/PHY initialization

Initialize SDRAM/PHY setting to store images loaded from HyperFlash/QSPI Flash. The following shows setting to use SDRAM.

- DBSC4 setting
- PHY setting
- SDRAM mode registers setting
- Internal bus interface check of DDR-PHY register (For R-Car H3 Ver.3.0, M3 Ver.1.2/Ver.1.3/Ver.3.0, and M3N)

Note) About “Internal bus interface check of DDR-PHY register”, refer to section 6.6 of 1.3.1 Related Document No.10. It can select by build option “RCAR_DDR_REG_CHECK” whether execute the check. For more detail, refer to “RCAR_DDR_REG_CHECK” in clause 6.3.3.

3.3.9 RPC initialization

Initialize RPC to read data from HyperFlash/QSPI Flash to System RAM/SDRAM.

3.3.10 DMA initialization

Initialize RT-DMAC to transfer data at high speed between HyperFlash/QSPI Flash and SDRAM.

3.3.11 Security and Safety setting

Perform Security and Safety setting to access control between IP. The following shows applied functions.

- LifeC Protection Setting

  Function to protect the access of the internal bus between the Master/Slave by LifeC.
  The default of the Security access protection setting to LifeC is shown below.

Table 3-20 Register setting value of the LifeC

Module	Register Name	Setting value
LifeC	SEC_SEL0	0xFFFFFFFF
	SEC_SEL1	0xFFFFFFFF
	SEC_SEL2	0xFFFFFFFF
	SEC_SEL3	0xFFF7FDFF
	SEC_SEL4	0xFFFFFFFF
	SEC_SEL5	0xFFFFFFBF
	SEC_SEL6	R-Car H3/M3/M3N : 0xFFFFCBFF R-Car E3 : 0xFFFFFBFF
	SEC_SEL7	0xFFFFFFFF
	SEC_SEL8	0xFFFFFFFF
	SEC_SEL9	0xFFFFFFFF
	SEC_SEL10	0xFFFFFFFF
	SEC_SEL11	0xFFFFFFFF
	SEC_SEL12	0xFFFFFFFF
	SEC_SEL13	0xFFBFFFFF
	SEC_SEL14	0xF3FFFFFF
	SEC_SEL15	0xFFFFFF3F
	SEC_GRP0CR0	0x00000000
	SEC_GRP1CR0	0x00000000
	SEC_GRP0CR1	0x00000000
	SEC_GRP1CR1	0x00000000
	SEC_GRP0CR2	0x00020000
	SEC_GRP1CR2	0x00020000
	SEC_GRP0CR3	0x00000000
	SEC_GRP1CR3	0x00000000
	SEC_GRP0COND0	0x00000000
	SEC_GRP1COND0	0x00000000
	SEC_GRP0COND1	0x00000000
	SEC_GRP1COND1	0x00000000
	SEC_GRP0COND2	0x00000000
	SEC_GRP1COND2	0x00000000
	SEC_GRP0COND3	0x00080200
	SEC_GRP1COND3	0x00080200
	SEC_GRP0COND4	0x00000000
	SEC_GRP1COND4	0x00000000
	SEC_GRP0COND5	0x00000040
	SEC_GRP1COND5	0x00000040
	SEC_GRP0COND6	R-Car H3/M3/M3N : 0x00000400 R-Car E3 : 0x00003400
	SEC_GRP1COND6	R-Car H3/M3/M3N : 0x00000400 R-Car E3 : 0x00003400
	SEC_GRP0COND7	0x00000000
	SEC_GRP1COND7	0x00000000

Module	Register Name	Setting value
	SEC_GRP0COND8	0x00000000
	SEC_GRP1COND8	0x00000000
	SEC_GRP0COND9	0x00000000
	SEC_GRP1COND9	0x00000000
	SEC_GRP0COND10	0x00000000
	SEC_GRP1COND10	0x00000000
	SEC_GRP0COND11	0x00000000
	SEC_GRP1COND11	0x00000000
	SEC_GRP0COND12	0x00000000
	SEC_GRP1COND12	0x00000000
	SEC_GRP0COND13	0x00400000
	SEC_GRP1COND13	0x00400000
	SEC_GRP0COND14	0x0C000000
	SEC_GRP1COND14	0x0C000000
	SEC_GRP0COND15	0x000000C0
	SEC_GRP1COND15	0x000000C0
	SEC_READONLY0	0x00000000
	SEC_READONLY1	0x00000000
	SEC_READONLY2	0x00000000
	SEC_READONLY3	0x00000000
	SEC_READONLY4	0x00000000
	SEC_READONLY5	0x00000000
	SEC_READONLY6	0x00000000
	SEC_READONLY7	0x00000000
	SEC_READONLY8	0x00000000
	SEC_READONLY9	0x00000000
	SEC_READONLY10	0x00000000
	SEC_READONLY11	0x00000000
	SEC_READONLY12	0x00000000
	SEC_READONLY13	0x00000000
	SEC_READONLY14	0x00000000
	SEC_READONLY15	0x00000000
	SAFE_GRP0CR0	0x00000000
	SAFE_GRP1CR0	0x00000000
	SAFE_GRP0CR1	0x00000000
	SAFE_GRP1CR1	0x00000000
	SAFE_GRP0CR2	0x00000000
	SAFE_GRP1CR2	0x00000000
	SAFE_GRP0CR3	0x00000000
	SAFE_GRP1CR3	0x00000000
	SAFE_GRP0COND0	0x00000000
	SAFE_GRP1COND0	0x00000000
	SAFE_GRP0COND1	0x00000000
	SAFE_GRP1COND1	0x00000000
	SAFE_GRP0COND2	0x00000000
	SAFE_GRP1COND2	0x00000000
	SAFE_GRP0COND3	0x00000000
	SAFE_GRP1COND3	0x00000000
	SAFE_GRP0COND4	0x00000000
	SAFE_GRP1COND4	0x00000000
	SAFE_GRP0COND5	0x00000000
	SAFE_GRP1COND5	0x00000000
	SAFE_GRP0COND6	0x00000000
	SAFE_GRP1COND6	0x00000000
	SAFE_GRP0COND7	0x00000000
	SAFE_GRP1COND7	0x00000000
	SAFE_GRP0COND8	0x00000000
	SAFE_GRP1COND8	0x00000000
	SAFE_GRP0COND9	0x00000000
	SAFE_GRP1COND9	0x00000000

Module	Register Name	Setting value
	SAFE_GRP0COND10	0x00000000
	SAFE_GRP1COND10	0x00000000
	SAFE_GRP0COND11	0x00000000
	SAFE_GRP1COND11	0x00000000
	SAFE_GRP0COND12	0x00000000
	SAFE_GRP1COND12	0x00000000
	SAFE_GRP0COND13	0x00000000
	SAFE_GRP1COND13	0x00000000
	SAFE_GRP0COND14	0x00000000
	SAFE_GRP1COND14	0x00000000
	SAFE_GRP0COND15	0x00000000
	SAFE_GRP1COND15	0x00000000
	SAFE_READONLY0	0x00000000
	SAFE_READONLY1	0x00000000
	SAFE_READONLY2	0x00000000
	SAFE_READONLY3	0x00000000
	SAFE_READONLY4	0x00000000
	SAFE_READONLY5	0x00000000
	SAFE_READONLY6	0x00000000
	SAFE_READONLY7	0x00000000
	SAFE_READONLY8	0x00000000
	SAFE_READONLY9	0x00000000
	SAFE_READONLY10	0x00000000
	SAFE_READONLY11	0x00000000
	SAFE_READONLY12	0x00000000
	SAFE_READONLY13	0x00000000
	SAFE_READONLY14	0x00000000
	SAFE_READONLY15	0x00000000

If want to change the Security and Safety access protection setting, please refer to 1.3.1 Related document No.5' R-Car Series, 3rd Generation User's Manual: Hardware, chapter 68' and 6.3.8.5 Security and Safety access protection setting.

- SDRAM protection settings

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

Table 3-21 List of the Security access protection setting for SDRAM

Module	Register Name	Setting value	Comment
AXI-bus	DPTDIVCR0	0x0E0403F0	SDRAM Protected Area Division #0 Protection area division physical address is H'04_03F00000
	DPTDIVCR1	0x0E0407E0	SDRAM Protected Area Division #1 Protection area division physical address is H'04_07E00000
	DPTDIVCR2	0x0E080000	SDRAM Protected Area Division #2 Protection area division physical address is H'08_00000000
	DPTDIVCR3	0x0E080000	SDRAM Protected Area Division #3 Protection area division physical address is H'08_00000000
	DPTDIVCR4	0x0E080000	SDRAM Protected Area Division #4 Protection area division physical address is H'08_00000000
	DPTDIVCR5	0x0E080000	SDRAM Protected Area Division #5 Protection area division physical address is H'08_00000000
	DPTDIVCR6	0x0E080000	SDRAM Protected Area Division #6 Protection area division physical address is H'08_00000000
	DPTDIVCR7	0x0E080000	SDRAM Protected Area Division #7 Protection area division physical address is H'08_00000000
	DPTDIVCR8	0x0E080000	SDRAM Protected Area Division #8

Module	Register Name	Setting value	Comment
			Protection area division physical address is H'08_00000000
	DPTDIVCR9	0x0E080000	SDRAM Protected Area Division #9 Protection area division physical address is H'08_00000000
	DPTDIVCR10	0x0E080000	SDRAM Protected Area Division #10 Protection area division physical address is H'08_00000000
	DPTDIVCR11	0x0E080000	SDRAM Protected Area Division #11 Protection area division physical address is H'08_00000000
	DPTDIVCR12	0x0E080000	SDRAM Protected Area Division #12 Protection area division physical address is H'08_00000000
	DPTDIVCR13	0x0E080000	SDRAM Protected Area Division #13 Protection area division physical address is H'08_00000000
	DPTDIVCR14	0x0E080000	SDRAM Protected Area Division #14 Protection area division physical address is H'08_00000000
	DPTCR0	0x0E000000	SDRAM Protected Area Setting #0 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR1	0x0E000E0E	SDRAM Protected Area Setting #1 The Security Group 3 has the privilege to write the relevant DRAM area.
	DPTCR2	0x0E000000	SDRAM Protected Area Setting #2 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR3	0x0E000000	SDRAM Protected Area Setting #3 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR4	0x0E000000	SDRAM Protected Area Setting #4 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR5	0x0E000000	SDRAM Protected Area Setting #5 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR6	0x0E000000	SDRAM Protected Area Setting #6 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR7	0x0E000000	SDRAM Protected Area Setting #7 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR8	0x0E000000	SDRAM Protected Area Setting #8 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR9	0x0E000000	SDRAM Protected Area Setting #9 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR10	0x0E000000	SDRAM Protected Area Setting #10 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR11	0x0E000000	SDRAM Protected Area Setting #11 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR12	0x0E000000	SDRAM Protected Area Setting #12 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR13	0x0E000000	SDRAM Protected Area Setting #13 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR14	0x0E000000	SDRAM Protected Area Setting #14 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.
	DPTCR15	0x0E000000	SDRAM Protected Area Setting #15 All of the Security Group (0-3) has the privilege to write the relevant DRAM area.

■ System RAM protection settings

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

Table 3-22 List of the Security access protection setting for System RAM

Module	Register Name	Setting value	Comment
AXI-bus	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 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.

This function initializes the access protection setting for these IP.

3.3.12 QoS setting

Set the priority of the data that passes through the bus.

3.3.13 Release HW resource

Release HW resource used by the Loader. Target resources are following.

- SCIF2
- SWDT

4. Memory

This chapter explains the memory construction and the memory layout.

4.1 Memory Constitution

Table 4-1 shows memory constitution. Loader use System RAM, SDRAM and TCM.

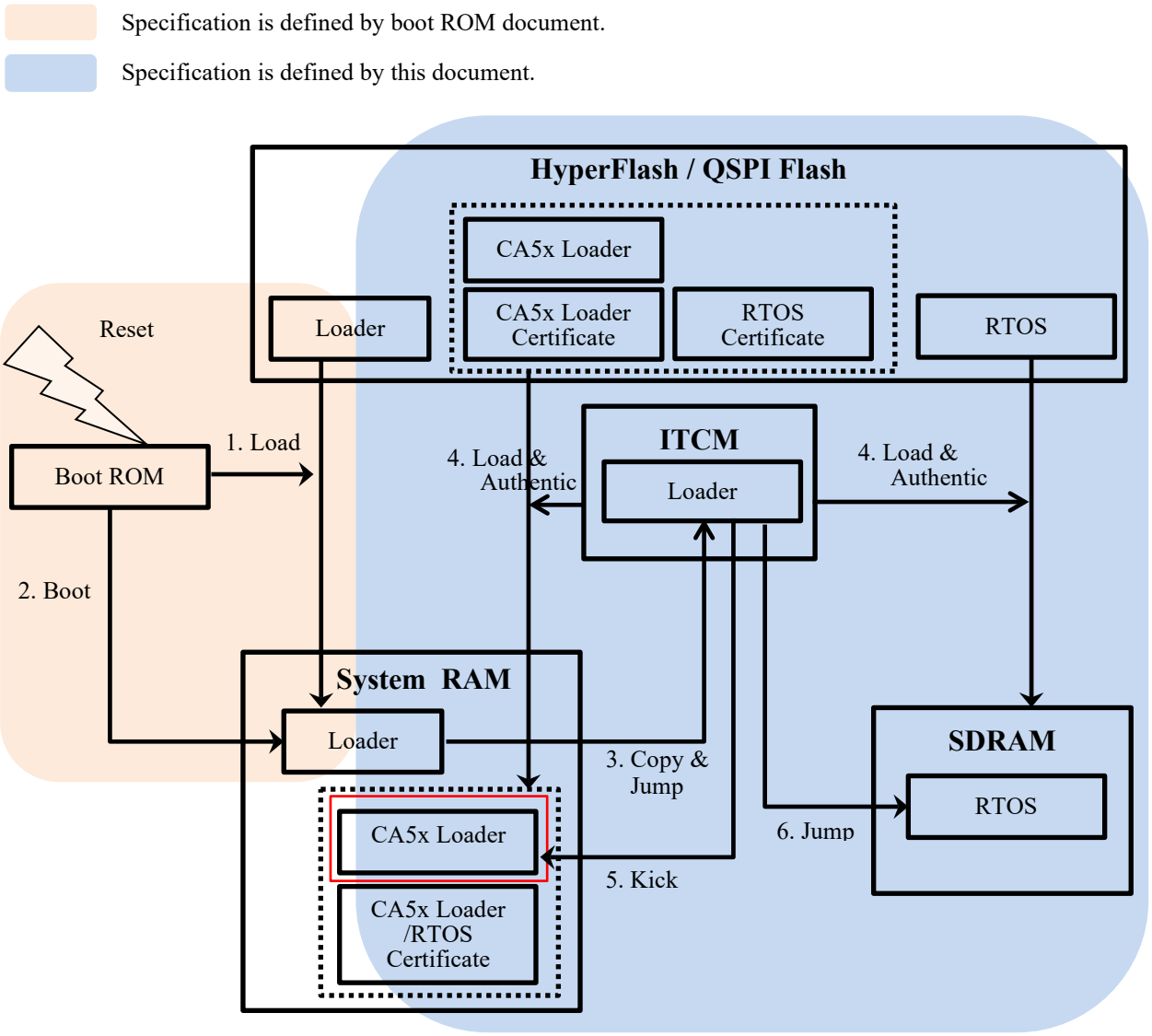
Table 4-1 Memory Constitution

Memory	Image
System RAM	Loader
	CA5x Loader image
	CA5x Loader certificate
	RTOS certificate
ITCM	Loader
DTCM	-
SDRAM	RTOS image

4.2 Memory Layout

Figure 4-1 shows the memory layout used by Loader. Red frame in Figure 4-1 represents that run in CA57/53.

Figure 4-1 Memory layout



The following shows the data flow refer to Figure 4-1 . Specification of process from No.3 to No.6 is defined by this document.

1. Boot ROM load Loader from HyperFlash/QSPI Flash to System RAM.
2. Loader is started by Boot ROM.
3. Loader copies a part of code to ITCM and jump to its code.
4. When boot mode is Normal mode, Loader loads using an image and a header from HyperFlash/QSPI Flash to System RAM or SDRAM without the authentication process.
   When boot mode is Secure mode, execute authentication process.
5. CA5x Loader turn on SCU and CPU0 of CA57/53 and kick.
6. Jump to RTOS.

4.3 Release image

Overview of release image is shown in Table 4-2.

Table 4-2 release image list

Filename	Description
bootparam_sa0.srec	Loader (Boot parameter)
cr7_loader.srec	Loader
cert_header_sa3.srec	Loader (Certificate)
dummy_rtos.srec	Dummy RTOS (Instead of CR7 main OS)

Build option of each release images is following. (Except Dummy RTOS)
Refer to '6.3.3 Build option' about build option.

- H3
  LSI=H3 RCAR_DRAM_SPLIT=1 RCAR_KICK_MAIN_CPU=1
  RCAR_LIFEC_NON_SECURE_MASTER=0
- M3
  LSI=M3 RCAR_DRAM_SPLIT=2 RCAR_KICK_MAIN_CPU=1
  RCAR_LIFEC_NON_SECURE_MASTER=0
- M3N
  LSI=M3N RCAR_DRAM_SPLIT=0 RCAR_KICK_MAIN_CPU=1
  RCAR_LIFEC_NON_SECURE_MASTER=0
- E3(Ebisu Board)
  LSI=E3 RCAR_DRAM_SPLIT=0 RCAR_KICK_MAIN_CPU=2
  RCAR_LIFEC_NON_SECURE_MASTER=0 RCAR_SA0_SIZE=0
  RCAR_DRAM_DDR3L_MEMCONF=0 RCAR_DRAM_DDR3L_MEMDUAL=0
- E3(Ebisu-4D Board)
  LSI=E3 RCAR_DRAM_SPLIT=0 RCAR_KICK_MAIN_CPU=2
  RCAR_LIFEC_NON_SECURE_MASTER=0 RCAR_SA0_SIZE=0

4.4 Example of writing data

The information needed when writing data to Flash is shown in Table 4-3, Table 4-4, and Table 4-5.
Program Top Address and Flash Save Address are default value.
Refer to '6.3.8.9 Loading images' how to modify Program Top Address. (Except Linux BSP images.)

4.4.1 Example 1 (CR7 boot only)

Example of CR7 boot only. (When the build option RCAR_KICK_MAIN_CPU=0 is specified.)

Table 4-3 Information of Example 1

Filename	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	H'E6320000 *1	H'000000	Loader (Boot parameter)
cr7_loader.srec	H'E6304000	H'040000	Loader
cert_header_sa3.srec	H'E6320000 *2	H'0C0000	Loader (Certificate)
dummy_rtos.srec	H'40040000	H'740000	Dummy RTOS (Instead of CR7 main OS)

Note *1) BootROM loads Boot parameter from Flash to H'E6300400. Loader refers it.

Note *2) Loader loads 'Dummy RTOS content certificate' from Flash to H'E6302000.

4.4.2 Example 2 (Boot CA5x image from CR7)

Example of booting CA5x image from CR7. (When the build option RCAR_KICK_MAIN_CPU=1 or 2 is specified.)

Table 4-4 Information of Example 2

Filename	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	H'E6320000 *1	H'000000	Loader (Boot parameter)
cr7_loader.srec	H'E6304000	H'040000	Loader
cert_header_sa3.srec	H'E6320000 *2	H'0C0000	Loader (Certificate)
dummy_rtos.srec	H'40040000	H'740000	Dummy RTOS (Instead of CR7 main OS)
CA5x image *3	H'E6304000	H'140000	Image executed by CA5x.

Note *1) BootROM loads Boot parameter from Flash to H'E6300400. Loader refers it.

Note *2) Loader loads 'CA5x IPL content certificate' and 'Dummy RTOS content certificate' from Flash to H'E6302000.

Note *3) This is not included in release image. Prepare by user.

4.4.3 Example 3 (Boot Linux BSP from CR7)

Example of booting arm-trusted-firmware and Linux BSP from CR7. (When the build option RCAR_KICK_MAIN_CPU=1 or 2 is specified.)

Table 4-5 Information of Example 3

Filename	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	H'E6320000	H'000000	Loader (Boot parameter)
cr7_loader.srec	H'E6304000	H'040000	CR7 Loader
cert_header_sa3.srec	H'E6320000	H'0C0000	Loader (Certificate)
dummy_rtos.srec	H'40040000	H'740000	Dummy RTOS (Instead of CR7 main OS)
bl2-<board_name>.srec *1	H'E6304000	H'140000	CA5x Loader
cert_header_sa6.srec *1	H'E6320000	H'180000	CA5x Loader (Certificate)
bl31-<board_name>.srec *1	H'44000000	H'1C0000	ARM Trusted Firmware
tee-<board_name>.srec *1	H'44100000	H'200000	OP-Tee
u-boot-elf-<board_name>.srec *1	H'50000000	H'640000	U-boot

Note *1) Refer to 1.3.1 Related Document[2] 'Linux Interface Specification Yocto recipe Start-Up Guide' and [12] 'Initial Program Loader User's Manual: Software R-Car H3/M3/M3N/E3 Series' about these images.

5. External Interface

There is no external interface for this module.

6. Integration

6.1 Directory configuration

Figure 6-1 shows the directory configuration that extract ‘CortexR7_Loader_<date>.tar.gz’.

```
CortexR7_Loader
    ├──── common/                 # Common files placed
    ├──── ddr/                    # DDR setting directory
    │     ├──── ddr_a/            # Setting for R-Car E3
    │     ├──── ddr_b/            # Setting for R-Car H3/M3/M3N
    │     └──── ddr_b_regcheck/   # Internal bus interface check
    ├──── include/                # Include directory
    │     ├──── machine/          # Include file for hardware
    │     └──── sys/              # Include file for system
    ├──── loader/                 # Loader's code on System RAM.
    ├──── Makefile                # Makefile
    ├──── pfc/                    # Pin Function setting directory
    │     ├──── E3/               # Setting for E3
    │     ├──── H3/               # Setting for H3
    │     ├──── M3/               # Setting for M3
    │     └──── M3N/              # Setting for M3N
    ├──── qos/                    # QoS setting directory
    │     ├──── E3/               # Setting for E3
    │     ├──── H3/               # Setting for H3
    │     ├──── M3/               # Setting for M3
    │     └──── M3N/              # Setting for M3N
    ├──── tcm_loader/             # Loader's code on TCM.
    └──── tools/                  # Dummy certificate
          └──── dummy_create/     # Dummy create tools
```

Figure 6-1 Directories for Loader of R-Car H3/M3/M3N/E3

6.2 Integration Procedure

There is no external interface for this module.

6.3 How to

This section shows how to build Loader.

6.3.1 Components

Table 6-1 shows necessary components to build.

Table 6-1 Components to build

Item	File name
compiler	gcc-linaro-7.3.1-2018.05-x86_64_arm-eabi.tar.xz
Loader	CortexR7_Loader_<date>.tar.gz
Dummy RTOS	Dummy_RTOS_<date>.tar.gz

6.3.2 Environment

Build IPL on Ubuntu 16.04LTS (64bit) or later. A compiler to build Loader is shown in Table 6-1.

6.3.3 Build option

The IPL support the following build options. Options that are not described in this section have default value in building.

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

• LOG_LEVEL

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

Table 6-2 Association table for the LOG_LEVEL value and valid logging functions

LOG_LEVEL	Valid logging function
0	No functions output logs
10	ERROR()
20	NOTICE(), ERROR() [default]
30	NOTICE(), ERROR(), WARN()
40	NOTICE(), ERROR(), WARN(), INFO()
50	NOTICE(), ERROR(), WARN(), INFO(), VERBOSE()

• RCAR_SECURE_BOOT

Set value is 0 or 1. If this option is not set, the value is set 1 internally.

Table 6-3 Association table for the RCAR_SECURE_BOOT and valid boot setting

RCAR_SECURE_BOOT	Boot setting
0	Starting mode is always Normal boot.
1	Execute the processing flow of Figure 1-3 to decide the starting mode. [default]

• RCAR_KICK_MAIN_CPU

Set value is 0 or 1 or 2. These values decide CPU to kick.
If this option is not set, the value is set 0 internally.

Table 6-4 Association table for the RCAR_KICK_MAIN_CPU and valid CPU setting

RCAR_KICK_MAIN_CPU	Boot setting
0	Not kick the CA57 and CA53. [default]
1	Kick CA57 as main CPU.
2	Kick CA53 as main CPU.

Note) 'Kick CA57 as main CPU' option is available with R-Car H3/M3/M3N.
Note) 'Kick CA53 as main CPU' option is available with R-Car H3/M3/E3.

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

Table 6-5 Association table for the DEBUG value and valid logging functions

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

• RCAR_QOS_TYPE

AXI-bus has the QoS arbitration to control latency and bandwidth.
Set value is 0 or 3. If this option is not set, the value is set 0 internally.

Table 6-6 Association table for the RCAR_QOS_TYPE value and QoS setting

RCAR_QOS_TYPE	QoS setting
0	Enable the QoS arbitration setting [default]
3	Disable the QoS arbitration setting

• RCAR_DRAM_SPLIT

DRAM split setting in the SDRAM setting.
Set value is 0 to 2. If this option is not set, the value is set 0 internally.

Table 6-7 Association table for the RCAR_DRAM_SPLIT value and DRAM split setting

RCAR_DRAM_SPLIT	DRAM split setting
0	Linear (No split) [default]
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

Note) '4 channel split' option is available only with R-Car H3.
Note) '2 channel split' option is available with R-Car H3/M3.

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

Table 6-8 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)
	AUTO	Linear (No split)	*1	*2	*3

*1 R-Car H3 is set 4 channel split. In other cases, error log is output and stop running when boot the Loader.
*2 R-Car H3/M3 is set 2 channel split. In other cases, error log is output and stop running when boot the Loader.
*3 R-Car H3 is set 4 channel split, R-Car M3 is set 2 channel split, and R-Car M3N is set linear.
R-Car E3 is not supported LSI=AUTO. (Refer to Table 6-11.)

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)"
```

• RCAR_ECC_ENABLE

Dual Channel ECC or Single Channel ECC setting on System RAM and SDRAM.
Set value is 0 to 2. If this option is not set, the value is set 0 internally.

Table 6-9 Association table for the RCAR_ECC_ENABLE value and ECC setting

RCAR_ECC_ENABLE	ECC Setting
0	Single Channel/Dual Channel ECC is disable [default]
1	Single Channel ECC is enable
2	Dual Channel ECC on SDRAM is enable

Note) 'Dual Channel ECC on SDRAM is enable' setting is available only with R-Car H3 or R-Car M3.

The following table shows the relationship between the build options LSI, LSI_CUT and RCAR_ECC_ENABLE.

Table 6-10 Association table for the RCAR_ECC_ENABLE and LSI

build option			RCAR_ECC_ENABLE		
	value		0	1	2
LSI	H3	LSI_CUT	ECC Disable		Dual Channel ECC
		10, 11		build error	
		20, 30		Single Channel ECC	
		-		Ver.1.x : error log output Others : Single Channel ECC	
	M3	10, 11, 13	ECC Disable	build error	Dual Channel ECC
		30		Single Channel ECC	
		-		Ver.1.x : error log output Ver.3.0 : Single Channel ECC	
	M3N		ECC Disable	Single Channel ECC	build error
	E3		ECC Disable	Single Channel ECC	build error
	AUTO		ECC Disable	H3 Ver.1.x, M3 Ver.1.x : error log output Others: Single Channel ECC	H3, M3: Dual Channel ECC M3N: error log output

Note) 'error log output' means that don't occur build error but error log is output when boot the Loader. And then stop running.

• LSI

Product ID number setting of IPL program for the target LSI.
Internal configuration changes are made to operate at the set product.
This is mandatory option. If it is not set, build error occurs.

Table 6-11 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.
AUTO	R-Car H3, R-Car M3 and R-Car M3N. (not supported E3.)

• LSI_CUT

R-Car CUT number setting of IPL program.
Internal configuration changes are made to operate at the set product.
Set value 10, 11, 13, 20, or 30.
If this option is not set, the value is reflected the CUT number field value of product register internally.

Table 6-12 Association table for the LSI_CUT value and R-Car CUT number setting.

LSI_CUT	R-Car H3's CUT number setting of IPL program.	R-Car M3's CUT number setting of IPL program.	R-Car M3N's CUT number setting of IPL program.	R-Car E3's CUT number setting of IPL program.
10	-	-	-	Ver.1.0
11	-	Ver.1.1	Ver.1.1	Ver.1.1
13	-	Ver.1.3	-	-
20	Ver.2.0	-	-	-
30	Ver.3.0	Ver.3.0	-	-

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.

• RCAR_LIFEC_NON_SECURE_MASTER

LifeC Security setting.
Set value is 0 to 1. If this option is not set, the value is set 1 internally.

Table 6-13 Association table for the RCAR_LIFEC_NON_SECURE_MASTER setting

RCAR_LIFEC_NON_SECURE_MASTER	LifeC Security Setting
0	LifeC Security of CR7 is Secure.
1	LifeC Security of CR7 is Non-secure. [default]

• RCAR_REF_INT

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

Table 6-14 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 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.

• RCAR_REWT_TRAINING

Select ‘periodic write DQ training’ mode. If this option is not set, the value is set 0 internally.
‘periodic write DQ training’ adjusts write signal timings for LPDDR4 skew correction.
For details, refer to the 8.1 Periodic write DQ training.

Table 6-15 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]

Note) This 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.

• RCAR_SA0_SIZE

Switch the IPL size of dummy certificate.
Set value is 0 or 1.
If this option is not set, the value is set 1 internally.

Table 6-16 Association table for the RCAR_SA0_SIZE value and the IPL size information

RCAR_SA0_SIZE	IPL size information
0	For R-Car E3, IPL size is 80KB.
1	For R-Car H3/M3/M3N, IPL size is 170KB. [default]

Note) RCAR_SA0_SIZE is the build option for the dummy_create.

• RCAR_DRAM_DDR3L_MEMCONF

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

Table 6-17 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]

Note) This option is available only with R-Car E3.
Note) Except for the above chips, the value is not available.
Note) In case of Ebisu board, DRAM memory size is 1G Byte.

• 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 6-18 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]

Note) The option is available only with R-Car E3.
Note) Except for the above chips, the value is not available.
Note) In case of Ebisu board, implemented SDRAM is 2pieces.

• RCAR_DRAM_LPDDR4_MEMCONF

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

Table 6-19 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 x4 channel : 8GB [default]
Others	Prohibited

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

• RCAR_DDR_REG_CHECK

It can select whether execute internal bus interface check of DDR-PHY register.
About “internal bus interface check of DDR-PHY register”, refer to section 6.6 of 1.3.1 Related Document No.10.
By default, it is defined to '0' (Do not check DDR-PHY register).

Table 6-6-20 Association table for the RCAR_DDR_REG_CHECK setting

RCAR_DDR_REG_CHECK	Whether check DDR-PHY register
0	Do not execute internal bus interface check of DDR-PHY register [default]
1	Execute internal bus interface check of DDR-PHY register
Others	Prohibited

Note) This option is available for R-Car H3 Ver.3.0, M3 Ver.1.2/Ver.1.3/Ver.3.0, and M3N, so must not set to 1 when using other products.
Note) As a result of DDR-PHY register check, IPL keep running even if fault is detected.

• RCAR_LIFEC_SETTING

Select enable / disable LifeC settings. The default value for RCAR_LIFEC_SETTING is "0" and the LifeC setting is disabled.If the LifeC setting is disabled, the startup message will say "Skipping LifeC setting for testing"
If LifeC setting is required, change the setting value of RCAR_LIFEC_SETTING to "1".

Table 6-21 LifeC setting

RCAR_LIFEC_SETTING	LifeC setting
0	LifeC setting disabled [default]
1	LifeC setting enabled

6.3.4 Prepare the compiler

You locate the components shown Table 6-1 to $WORK and execute following steps.

```
$ cd $WORK
$ wget https://releases.linaro.org/components/toolchain/binaries/7.3-
  2018.05/arm-eabi/gcc-linaro-7.3.1-2018.05-x86_64_arm-eabi.tar.xz
$ tar xvf gcc-linaro-7.3.1-2018.05-x86_64_arm-eabi.tar.xz
```

6.3.5 Prepare the source code

You execute the following steps after clause 6.3.4.

```
$ cd $WORK
$ tar xvf CortexR7_Loader_<date>.tar.gz
$ tar xvf Dummy_RTOS_<date>.tar.gz
```

6.3.6 Build the Loader

You build Loader using following steps.

```
$ cd $WORK/CortexR7_Loader
$ make clean LSI=H3
$ CROSS_COMPILE=~/gcc-linaro-7.3.1-2018.05-x86_64_arm-eabi/bin/arm-
  eabi- make LSI=H3 RCAR_DRAM_SPLIT=1 RCAR_KICK_MAIN_CPU=1
```

Output the following images.

- ./bootparam_sa0.srec
- ./cert_header_sa3.srec
- ./cr7_loader.srec

6.3.7 Build the Dummy RTOS

You build Dummy RTOS using following steps.

```
$ cd $WORK /Dummy_RTOS
$ make clean
$ CROSS_COMPILE=~/gcc-linaro-7.3.1-2018.05-x86_64_arm-eabi/bin/arm-
  eabi-
```

Output the following images.

- ./ dummy_rtos.srec

6.3.8 How to customize

The following shows methods of customize. Table 6-22 shows a location of a code implemented functions.

Table 6-22 Location of a code implemented functions

Function		Path	Function's name
CPU initialization	Initialization of the general-purpose register	loader/loader.S	Startup: line 64
	Stack configuration		Startup: line 82
	Start Global Timer		Startup: line 102
	TCM configuration		Startup: line 108
	Enable Lock step		Startup: line 119
	Enable the Instruction Cache		Startup: line 142
ECC initialization	Enable ECC of ITCM / DTCM / I-Cache	loader/loader.S	Startup: line 125
ECC initialization (TCM loader)	Enable ECC of SDRAM and System RAM	Refer to 6.3.8.1 about Dual Channel ECC Refer to 6.3.8.2 about Single Channel ECC	
SWDT initialization		common/swdt.c	swdt_init()
PFC / GPIO initialization		Refer to 6.3.8.3	
SCIF2 initialization		common/scif.S	console_init()
SDRAM/PHY initialization		Refer to 6.3.8.4	
RPC initialization		common/rpc_driver.c	initRPC()
DMA initialization		common/dma_driver.c	initDMA()
Security and Safety setting		Refer to 6.3.8.6	
QoS setting		Refer to 6.3.8.5	

6.3.8.1 Dual Channel ECC setting

Dual Channel ECC is only available on SDRAM.
It is possible to select Enable or Disable by build option RCAR_ECC_ENABLE.
Dual Channel ECC and Single Channel ECC cannot be used together.
Dual Channel ECC setting is executed by dram_dual_ecc_init() function implemented in dram_dual_ecc.c.
dram_dual_ecc_init() function is called by ecc_init() function implemented in ecc_init.c.
dram_dual_ecc_init() function judges SDRAM split mode and calls dual_ecc_setting_4ch(), dual_ecc_setting_2ch() or dual_ecc_setting_linear(). Checking of restrictions and setting to registers are executed in each function.
In this clause describes how to change the setting.
Only Area0-0 is set FuSa Area (i.e. Data area protected by ECC) as default setting.
Note) Dual Channel ECC setting is available with only R-Car H3, R-Car M3.

```
CortexR7_Loader
|----common
|        dram_dual_ecc.c
|        ecc_init.c
```

Setting value of FuSa Area size is defined in dram_ecc.c and shown in Table 6-23.
Area size is divided SDRAM size per channel by 8. Setting value is exponent that converted the Area size to a power of 2.
Extra Area size is divided Area size by 8. Setting value is exponent that converted the Extra Area size to a power of 2.

Table 6-23 Setting value list of Area size and Extra Area size

define name	Default value	Explanation
ADSPLCR0_AREA_1GB	0x1B	When size of SDRAM per channel is 1GB, Area size is 128MB (2^27)
ADSPLCR3_AREA_1GB	0x18	When Area size is 128MB, Extra Area size is 16MB (2^24)
ADSPLCR0_AREA_2GB	0x1C	When size of SDRAM per channel is 2GB, Area size is 256MB (2^28)
ADSPLCR3_AREA_2GB	0x19	When Area size is 256MB, Extra Area size is 32MB (2^25)
ADSPLCR0_AREA_4GB	0x1D	When size of SDRAM per channel is 4GB, Area size is 512MB (2^29)
ADSPLCR3_AREA_4GB	0x1A	When Area size is 512MB, Extra Area size is 64MB (2^26)

Area and Extra Area setting is executed by dram_dual_ecc_init() function implemented in dram_dual_ecc.c.
The default setting value of FuSa Area is shown in the Table 6-24.
Area and Extra Area can't be set the same area number. If there are any duplication settings, output error log 'DRAM ECC area setting error' and stop booting.

Table 6-24 Setting value list of ECC Area setting

Array name	Default value	Explanation
area_ecc_enable_bit[]	{0,0,0,0,0,0,0,0}	Setting of ECC for Area. 1 = ECC enable / 0 = ECC disable
extra_area_ecc_enable_bit[]	{0,0,0,0,0,0,0,1}	Setting of ECC for Extra Area. 1 = ECC enable / 0 = ECC disable

Note) The elements of the arrays are in descending order.
ex.) area_ecc_enable_bit[0] is setting of Area7, and area_ecc_enable_bit[7] is setting of Area0.

Extra Split setting is executed by dram_ecc_init() function implemented in dram_ecc.c.
The default setting value of ECC Extra Split is shown in the Table 6-25.
Only Area that Extra Split is valid enable to set Extra Area.
Extra Split setting can set only one area. If multiple setting is made, only the area with the smallest number is valid.

Table 6-25 Setting value of ECC Extra split setting

Array name	Default value	Explanation
extra_split_enable_bit[]	{0,0,0,0,0,0,0,1}	Setting of Extra Split for Area. 1 = Extra split enable / 0 = Extra split disable

Note) The elements of the array are in descending order.
ex.) extra_split_enable_bit[0] is setting of Area7, and extra_split_enable_bit [7] is setting of Area0.

SDRAM split setting for Extra Area is executed by dram_dual_ecc_init() function implemented in dram_dual_ecc.c.
The default setting value of SDRAM split setting for Extra Area is shown in the Table 6-26.
Extra Area can't be set SDRAM 2ch split mode. If 2ch split mode is specified for Extra Area, SDRAM split mode is changed to Linear mode by dual_ecc_setting_2ch() function.

Table 6-26 Setting value of SDRAM split setting for Extra Area

Array name	DRAM split	Default value	Explanation
extra_area[]	4ch / Linear	{1,1,1,1,1,1,1,1}	SDRAM split setting for Extra Area 1 = 4ch / 0 = Linear

Note) The elements of the array are in descending order.
ex.) extra_split_enable_bit[0] is setting of Area7, and extra_split_enable_bit [7] is setting of Area0.

It is able to select whether fill FuSa Area with zero to generate ECC code by dram_dual_ecc_init() function.
The default setting value is shown in Table 6-27.
This setting is possible to set for each Area or Extra Area.
This setting is only available with the Area that ECC is valid.

Table 6-27 Setting value of whether fill FuSa Area with zero

Array name	Default value	Explanation
area_fill_enable_bit []	{1,1,1,1,1,1,1,1}	Setting whether fill Area with zero. 1 = fill with zero / 0 = no fill
extra_area_fill_enable_bit[]	{1,1,1,1,1,1,1,1}	Setting whether fill Extra Area with zero. 1 = fill with zero / 0 = no fill

Note) The elements of the arrays are in descending order.
ex.) area_fill_enable_bit[0] is setting of Area7, and area_fill_enable_bit[7] is setting of Area0.

More details about Dual Channel ECC, see the documents of Table 1-3 No10/11.

Show example setting below.

ex.1) If it is required to set Area 0/4/6 are FuSa Areas, set as follow:
area_ecc_enable_bit[] = {0,1,0,1,0,0,0,1}

ex.2) If it is required to set Extra Area 2-1/2-3/2-5/2-7 are FuSa Areas, set as follow:
extra_split_enable_bit[] = {0,0,0,0,0,1,0,0}
extra_area_ecc_enable_bit[] = {1,0,1,0,1,0,1,0}

ex.3) If it is required to set SDRAM split to be valid for Area 3-3/3-6, set as follow:
extra_split_enable_bit[] = {0,0,0,0,1,0,0,0}
extra_area_ecc_enable_bit[] = {0,1,0,0,1,0,0,0}
extra_area[] = {0,1,0,0,1,0,0,0}

ex.4) If filling Area 1/5 with zero is required, set as follow:
area_ecc_enable_bit[] = {0,0,1,0,0,0,1,0}
area_fill_enable_bit[] = {0,0,1,0,0,0,1,0}

ex.5) If filling Extra Area 4-2/4-4 with zero is required, set as follow:
extra_split_enable_bit[] = {0,0,0,1,0,0,0,0}
extra_area_ecc_enable_bit[] = {0,0,0,1,0,1,0,0}
extra_area_fill_enable_bit[] = {0,0,0,1,0,1,0,0}

6.3.8.2 Single Channel ECC setting

Single Channel ECC is available on SDRAM and System RAM.
Single Channel ECC setting is possible to select Enable or Disable by build option 'RCAR_ECC_ENABLE'.
Single Channel ECC function and Dual Channel ECC function cannot be used together.
Single Channel ECC setting is executed by single_ecc_init() function implemented in single_ecc.c.
single_ecc_init() function is called by ecc_init() function implemented in ecc_init.c.
In this clause describes how to change the setting.
If modification is required from default setting, have only to change the values in single_ecc_conf.h.
Only Area from 0x04_00000000 to 0x04_00200000 on SDRAM is set to FuSa Area as default setting.

```
CortexR7_Loader
|----common
|        ecc_init.c
|        single_ecc.c
|----include
|        single_ecc_conf.h
```

When using Single Channel ECC function, be sure to read following 'Notes'.
<Notes>

- Single Channel ECC setting is available with only R-Car H3 Ver.2.0/Ver.3.0, M3 Ver.3.0, R-Car M3N, and R-Car E3.
- Either SDRAM Split function or Single Channel ECC function are available.
- If set build option 'RCAR_ECC_ENABLE=1', one or more Areas must be set FuSa Area.
- Overlapping each Areas is not checked by the Loader. Set each FuSa Area and ECC Area no overlap. Set FuSa Area (ECC Area) each other does not overlap too.
- The array fill_data[] that is used when fill FuSa Area with zero must be placed on other than ECC Area. fill_data[] is declared in dma_driver.c and indicated to place to '.test_data' section in loader.ld.S.
- If set the area from 0xE6300000 to 0xE6303FFF on System RAM to FuSa Area, must set to 8bit data / 5bit ECC mode.
- Single Channel ECC on SDRAM and on System RAM can be set together.
- When using 64 bit/8 bit mode, ECC Area size is one-eighth of FuSa Area size.

There are macros DFUSAn_EN(n=0-15) in single_ecc_conf.h.
The default setting value of DFUSAn_EN is shown in the Table 6-28.
If set Area 'n' on SDRAM to FuSa Area, set DFUSAn_EN to SINGLE_ECC_ENABLE.
If don't set Area 'n' on SDRAM to FuSa Area, set DFUSAn_EN to SINGLE_ECC_DISABLE.

Table 6-28 Setting value list of DFUSAn_EN(n=0-15) in single_ecc_conf.h

Array name	Default value	Explanation
DFUSA0_EN	SINGLE_ECC_ENABLE	Whether set Area0 on SDRAM to FuSa Area.
DFUSA1_EN	SINGLE_ECC_DISABLE	Whether set Area1 on SDRAM to FuSa Area.
DFUSA2_EN	SINGLE_ECC_DISABLE	Whether set Area2 on SDRAM to FuSa Area.
DFUSA3_EN	SINGLE_ECC_DISABLE	Whether set Area3 on SDRAM to FuSa Area.
DFUSA4_EN	SINGLE_ECC_DISABLE	Whether set Area4 on SDRAM to FuSa Area.
DFUSA5_EN	SINGLE_ECC_DISABLE	Whether set Area5 on SDRAM to FuSa Area.
DFUSA6_EN	SINGLE_ECC_DISABLE	Whether set Area6 on SDRAM to FuSa Area.
DFUSA7_EN	SINGLE_ECC_DISABLE	Whether set Area7 on SDRAM to FuSa Area.
DFUSA8_EN	SINGLE_ECC_DISABLE	Whether set Area8 on SDRAM to FuSa Area.
DFUSA9_EN	SINGLE_ECC_DISABLE	Whether set Area9 on SDRAM to FuSa Area.
DFUSA10_EN	SINGLE_ECC_DISABLE	Whether set Area10 on SDRAM to FuSa Area.
DFUSA11_EN	SINGLE_ECC_DISABLE	Whether set Area11 on SDRAM to FuSa Area.
DFUSA12_EN	SINGLE_ECC_DISABLE	Whether set Area12 on SDRAM to FuSa Area.
DFUSA13_EN	SINGLE_ECC_DISABLE	Whether set Area13 on SDRAM to FuSa Area.
DFUSA14_EN	SINGLE_ECC_DISABLE	Whether set Area14 on SDRAM to FuSa Area.
DFUSA15_EN	SINGLE_ECC_DISABLE	Whether set Area15 on SDRAM to FuSa Area.

There are macros DFUSAn_SADDR (n=0-15) in single_ecc_conf.h.
The default setting value of DFUSAn_SADDR is shown in the Table 6-29.
Set start address of FuSa Area ‘n’ on SDRAM to DFUSAn_SADDR.
Set the value in 64 bit.
Note) Setting value is only available 40 bit address.

Table 6-29 Setting value list of DFUSAn_SADDR(n=0-15) in single_ecc_conf.h

Array name	Default value	Explanation
DFUSA0_SADDR	0x0400000000U	Start address of FuSa Area0 on SDRAM.
DFUSA1_SADDR	0x0U	Start address of FuSa Area1 on SDRAM.
DFUSA2_SADDR	0x0U	Start address of FuSa Area2 on SDRAM.
DFUSA3_SADDR	0x0U	Start address of FuSa Area3 on SDRAM.
DFUSA4_SADDR	0x0U	Start address of FuSa Area4 on SDRAM.
DFUSA5_SADDR	0x0U	Start address of FuSa Area5 on SDRAM.
DFUSA6_SADDR	0x0U	Start address of FuSa Area6 on SDRAM.
DFUSA7_SADDR	0x0U	Start address of FuSa Area7 on SDRAM.
DFUSA8_SADDR	0x0U	Start address of FuSa Area8 on SDRAM.
DFUSA9_SADDR	0x0U	Start address of FuSa Area9 on SDRAM.
DFUSA10_SADDR	0x0U	Start address of FuSa Area10 on SDRAM.
DFUSA11_SADDR	0x0U	Start address of FuSa Area11 on SDRAM.
DFUSA12_SADDR	0x0U	Start address of FuSa Area12 on SDRAM.
DFUSA13_SADDR	0x0U	Start address of FuSa Area13 on SDRAM.
DFUSA14_SADDR	0x0U	Start address of FuSa Area14 on SDRAM.
DFUSA15_SADDR	0x0U	Start address of FuSa Area15 on SDRAM.

There are macros DFUSAn_SIZE (n=0-15) in single_ecc_conf.h.
The default setting value of DFUSAn_SIZE is shown in the Table 6-30.
Set size of FuSa Area ‘n’ on SDRAM to DFUSAn_SIZE.
Following setting value is available.
DFUSA_SIZE_xMB (x=1/2/4/8/16/32/64/128/256/512)

Table 6-30 Setting value list of DFUSAn_SIZE (n=0-15) in single_ecc_conf.h

Array name	Default value	Explanation
DFUSA0_SIZE	DFUSA_SIZE_2MB	Size of FuSa Area0 on SDRAM.
DFUSA1_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area1 on SDRAM.
DFUSA2_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area2 on SDRAM.
DFUSA3_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area3 on SDRAM.
DFUSA4_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area4 on SDRAM.
DFUSA5_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area5 on SDRAM.
DFUSA6_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area6 on SDRAM.
DFUSA7_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area7 on SDRAM.
DFUSA8_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area8 on SDRAM.
DFUSA9_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area9 on SDRAM.
DFUSA10_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area10 on SDRAM.
DFUSA11_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area11 on SDRAM.
DFUSA12_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area12 on SDRAM.
DFUSA13_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area13 on SDRAM.
DFUSA14_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area14 on SDRAM.
DFUSA15_SIZE	DFUSA_SIZE_1MB	Size of FuSa Area15 on SDRAM.

There are macros DECCn_MODE (n=0-15) in single_ecc_conf.h.
The default setting value of DECCn_MODE is shown in the Table 6-31.
Set ECC mode of FuSa Area ‘n’ on SDRAM to DECCn_MODE.
Following setting value is available.
MODE_8_5 or MODE_64_8.
MODE_8_5 is 8 bit data / 5 bit ECC mode.
MODE_64_8 is 64 bit data / 8 bit ECC mode.

Table 6-31 Setting value list of DECCn_MODE (n=0-15) in single_ecc_conf.h

Array name	Default value	Explanation
DECC0_MODE	MODE_8_5	ECC mode of FuSa Area0 on SDRAM.
DECC1_MODE	MODE_8_5	ECC mode of FuSa Area1 on SDRAM.
DECC2_MODE	MODE_8_5	ECC mode of FuSa Area2 on SDRAM.
DECC3_MODE	MODE_8_5	ECC mode of FuSa Area3 on SDRAM.
DECC4_MODE	MODE_8_5	ECC mode of FuSa Area4 on SDRAM.
DECC5_MODE	MODE_8_5	ECC mode of FuSa Area5 on SDRAM.
DECC6_MODE	MODE_8_5	ECC mode of FuSa Area6 on SDRAM.
DECC7_MODE	MODE_8_5	ECC mode of FuSa Area7 on SDRAM.
DECC8_MODE	MODE_8_5	ECC mode of FuSa Area8 on SDRAM.
DECC9_MODE	MODE_8_5	ECC mode of FuSa Area9 on SDRAM.
DECC10_MODE	MODE_8_5	ECC mode of FuSa Area10 on SDRAM.
DECC11_MODE	MODE_8_5	ECC mode of FuSa Area11 on SDRAM.
DECC12_MODE	MODE_8_5	ECC mode of FuSa Area12 on SDRAM.
DECC13_MODE	MODE_8_5	ECC mode of FuSa Area13 on SDRAM.
DECC14_MODE	MODE_8_5	ECC mode of FuSa Area14 on SDRAM.
DECC15_MODE	MODE_8_5	ECC mode of FuSa Area15 on SDRAM.

There are macros DECCn_SADDR (n=0-15) in single_ecc_conf.h.
The default setting value of DECCn_SADDR is shown in the Table 6-32.
Set start address of ECC Area ‘n’ on SDRAM to DECCn_SADDR.
Set the value in 64 bit.
Note) Setting value is only available 40 bit address.

Table 6-32 Setting value list of DECCn_SADDR (n=0-15) in single_ecc_conf.h

Array name	Default value	Explanation
DECC0_SADDR	0x043FE00000U	Start address of ECC Area0 on SDRAM.
DECC1_SADDR	0x0U	Start address of ECC Area1 on SDRAM.
DECC2_SADDR	0x0U	Start address of ECC Area2 on SDRAM.
DECC3_SADDR	0x0U	Start address of ECC Area3 on SDRAM.
DECC4_SADDR	0x0U	Start address of ECC Area4 on SDRAM.
DECC5_SADDR	0x0U	Start address of ECC Area5 on SDRAM.
DECC6_SADDR	0x0U	Start address of ECC Area6 on SDRAM.
DECC7_SADDR	0x0U	Start address of ECC Area7 on SDRAM.
DECC8_SADDR	0x0U	Start address of ECC Area8 on SDRAM.
DECC9_SADDR	0x0U	Start address of ECC Area9 on SDRAM.
DECC10_SADDR	0x0U	Start address of ECC Area10 on SDRAM.
DECC11_SADDR	0x0U	Start address of ECC Area11 on SDRAM.
DECC12_SADDR	0x0U	Start address of ECC Area12 on SDRAM.
DECC13_SADDR	0x0U	Start address of ECC Area13 on SDRAM.
DECC14_SADDR	0x0U	Start address of ECC Area14 on SDRAM.
DECC15_SADDR	0x0U	Start address of ECC Area15 on SDRAM.

There are macros DFUSAn_FILL (n=0-15) in single_ecc_conf.h.
The default setting value of DFUSAn_FILL is shown in the Table 6-33.
It is available to select whether fill FuSa Area ‘n’ on SDRAM with zero to generate ECC code.
Following setting is available.
FUSA_0FILL_ENABLE or FUSA_0FILL_DISABLE.
FUSA_0FILL_ENABLE means fill FuSa Area with zero.
FUSA_0FILL_DISABLE means not to fill FuSa Area.
This setting is only available with the Area that ECC is valid.

Table 6-33 Setting value list of DFUSAn_FILL (n=0-15) in single_ecc_conf.h

Array name	Default value	Explanation
DFUSA0_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 0 with zero on SDRAM.
DFUSA1_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 1 with zero on SDRAM.
DFUSA2_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 2 with zero on SDRAM.
DFUSA3_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 3 with zero on SDRAM.
DFUSA4_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 4 with zero on SDRAM.
DFUSA5_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 5 with zero on SDRAM.
DFUSA6_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 6 with zero on SDRAM.
DFUSA7_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 7 with zero on SDRAM.
DFUSA8_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 8 with zero on SDRAM.
DFUSA9_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 9 with zero on SDRAM.
DFUSA10_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 10 with zero on SDRAM.
DFUSA11_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 11 with zero on SDRAM.
DFUSA12_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 12 with zero on SDRAM.
DFUSA13_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 13 with zero on SDRAM.
DFUSA14_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 14 with zero on SDRAM.
DFUSA15_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 15 with zero on SDRAM.

There are macros SFUSAn_EN(n=0-7) in single_ecc_conf.h.
The default setting value of SFUSAn_EN is shown in the Table 6-34.
If set Area ‘n’ on System RAM to FuSa Area, set SFUSAn_EN to SINGLE_ECC_ENABLE.
If not set Area ‘n’ on System RAM to FuSa Area, set SFUSAn_EN to SINGLE_ECC_DISABLE.

Table 6-34 Setting value list of SFUSAn_EN (n=0-7) in single_ecc_conf.h

Array name	Default value	Explanation
SFUSA0_EN	SINGLE_ECC_DISABLE	Whether set Area0 on System RAM to FuSa Area.
SFUSA1_EN	SINGLE_ECC_DISABLE	Whether set Area1 on System RAM to FuSa Area.
SFUSA2_EN	SINGLE_ECC_DISABLE	Whether set Area2 on System RAM to FuSa Area.
SFUSA3_EN	SINGLE_ECC_DISABLE	Whether set Area3 on System RAM to FuSa Area.
SFUSA4_EN	SINGLE_ECC_DISABLE	Whether set Area4 on System RAM to FuSa Area.
SFUSA5_EN	SINGLE_ECC_DISABLE	Whether set Area5 on System RAM to FuSa Area.
SFUSA6_EN	SINGLE_ECC_DISABLE	Whether set Area6 on System RAM to FuSa Area.
SFUSA7_EN	SINGLE_ECC_DISABLE	Whether set Area7 on System RAM to FuSa Area.

There are macros SFUSAn_SADDR (n=0-7) in single_ecc_conf.h.
The default setting value of SFUSAn_SADDR is shown in the Table 6-35.
Set start address of FuSa Area ‘n’ on System RAM to SFUSAn_SADDR.

Table 6-35 Setting value list of SFUSAn_SADDR (n=0-7) in single_ecc_conf.h

Array name	Default value	Explanation
SFUSA0_SADDR	0x0U	Start address of FuSa Area0 on System RAM.
SFUSA1_SADDR	0x0U	Start address of FuSa Area1 on System RAM.
SFUSA2_SADDR	0x0U	Start address of FuSa Area2 on System RAM.
SFUSA3_SADDR	0x0U	Start address of FuSa Area3 on System RAM.
SFUSA4_SADDR	0x0U	Start address of FuSa Area4 on System RAM.
SFUSA5_SADDR	0x0U	Start address of FuSa Area5 on System RAM.
SFUSA6_SADDR	0x0U	Start address of FuSa Area6 on System RAM.
SFUSA7_SADDR	0x0U	Start address of FuSa Area7 on System RAM.

There are macros SFUSAn_SIZE (n=0-7) in single_ecc_conf.h.
The default setting value of SFUSAn_SIZE is shown in the Table 6-36.
Set size of FuSa Area ‘n’ on System RAM to SFUSAn_SIZE.
Following setting value is available.
SFUSA_SIZE_xKB (x=4/8/16/32/64/128/256)

Table 6-36 Setting value list of SFUSAn_SIZE (n=0-7) in single_ecc_conf.h

Array name	Default value	Explanation
SFUSA0_SIZE	SFUSA_SIZE_4KB	Size of FuSa Area0 on System RAM.
SFUSA1_SIZE	SFUSA_SIZE_4KB	Size of FuSa Area1 on System RAM.
SFUSA2_SIZE	SFUSA_SIZE_4KB	Size of FuSa Area2 on System RAM.
SFUSA3_SIZE	SFUSA_SIZE_4KB	Size of FuSa Area3 on System RAM.
SFUSA4_SIZE	SFUSA_SIZE_4KB	Size of FuSa Area4 on System RAM.
SFUSA5_SIZE	SFUSA_SIZE_4KB	Size of FuSa Area5 on System RAM.
SFUSA6_SIZE	SFUSA_SIZE_4KB	Size of FuSa Area6 on System RAM.
SFUSA7_SIZE	SFUSA_SIZE_4KB	Size of FuSa Area7 on System RAM.

There are macros SECCn_MODE (n=0-7) in single_ecc_conf.h.
The default setting value of SECCn_MODE is shown in the Table 6-37.
Set ECC mode of FuSa Area ‘n’ on System RAM to SECCn_MODE.
Following setting value is available.
MODE_8_5 or MODE_64_8.
MODE_8_5 is 8 bit data / 5 bit ECC mode.
MODE_64_8 is 64 bit data / 8 bit ECC mode.

Table 6-37 Setting value list of SECCn_MODE (n=0-7) in single_ecc_conf.h

Array name	Default value	Explanation
SECC0_MODE	MODE_8_5	ECC mode of FuSa Area0 on System RAM.
SECC1_MODE	MODE_8_5	ECC mode of FuSa Area1 on System RAM.
SECC2_MODE	MODE_8_5	ECC mode of FuSa Area2 on System RAM.
SECC3_MODE	MODE_8_5	ECC mode of FuSa Area3 on System RAM.
SECC4_MODE	MODE_8_5	ECC mode of FuSa Area4 on System RAM.
SECC5_MODE	MODE_8_5	ECC mode of FuSa Area5 on System RAM.
SECC6_MODE	MODE_8_5	ECC mode of FuSa Area6 on System RAM.
SECC7_MODE	MODE_8_5	ECC mode of FuSa Area7 on System RAM.

There are macros SECCn_SADDR (n=0-7) in single_ecc_conf.h.
The default setting value of SECCn_SADDR is shown in the Table 6-38.
Set start address of ECC Area ‘n’ on System RAM to SECCn_SADDR.

Table 6-38 Setting value list of SECCn_SADDR (n=0-7) in single_ecc_conf.h

Array name	Default value	Explanation
SECC0_SADDR	0x0U	Start address of ECC Area0 on System RAM.
SECC1_SADDR	0x0U	Start address of ECC Area1 on System RAM.
SECC2_SADDR	0x0U	Start address of ECC Area2 on System RAM.
SECC3_SADDR	0x0U	Start address of ECC Area3 on System RAM.
SECC4_SADDR	0x0U	Start address of ECC Area4 on System RAM.
SECC5_SADDR	0x0U	Start address of ECC Area5 on System RAM.
SECC6_SADDR	0x0U	Start address of ECC Area6 on System RAM.
SECC7_SADDR	0x0U	Start address of ECC Area7 on System RAM.

There are macros SFUSAn_FILL (n=0-7) in single_ecc_conf.h.
The default setting value of SFUSAn_FILL is shown in the Table 6-39.
It is available to select whether fill FuSa Area 'n' on System RAM with zero to generate ECC code.
Following setting is available.
FUSA_0FILL_ENABLE or FUSA_0FILL_DISABLE.
FUSA_0FILL_ENABLE means fill FuSa Area with zero.
FUSA_0FILL_DISABLE means not to fill FuSa Area with zero.
This setting is only available with the Area that ECC is valid.

Table 6-39 Setting value list of SECCn_SADDR (n=0-7) in single_ecc_conf.h

Array name	Default value	Explanation
SFUSA0_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 0 with zero on System RAM.
SFUSA1_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 1 with zero on System RAM.
SFUSA2_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 2 with zero on System RAM.
SFUSA3_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 3 with zero on System RAM.
SFUSA4_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 4 with zero on System RAM.
SFUSA5_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 5 with zero on System RAM.
SFUSA6_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 6 with zero on System RAM.
SFUSA7_FILL	FUSA_0FILL_ENABLE	Whether filling FuSa Area 7 with zero on System RAM.

More details about Single Channel ECC, refer to section 5.5 of 1.3.1 Related Document No.10 and No.11.

Show example setting below.

ex.1) If it is required to set FuSa Area0 on SDRAM from 0x0400000000 to 0x040FFFFFFF,
ECC Area0 from 0x0470000000 to 0x0471FFFFFF, 64 bit data/8 bit ECC mode and fill with zero,
set as follow:

```
#define DFUSA0_EN       (SINGLE_ECC_ENABLE)
#define DFUSA0_SADDR    (0x0400000000U)
#define DFUSA0_SIZE     (DFUSA_SIZE_256MB)
#define DECC0_MODE      (MODE_64_8)
#define DECC0_SADDR     (0x0470000000U)
#define DFUSA0_FILL     (FUSA_0FILL_ENABLE)
```

ex.2) If it is required to set FuSa Area0 on System RAM from 0xE6320000 to 0xE632FFFF,
ECC Area0 from 0xE6350000 to 0xE635FFFF, 8 bit data/5 bit ECC mode and no fill FuSa Area0,
set as follow:

```
#define SFUSA0_EN       (SINGLE_ECC_ENABLE)
#define SFUSA0_SADDR    (0xE6320000U)
#define SFUSA0_SIZE     (SFUSA_SIZE_64KB)
#define SECC0_MODE      (MODE_8_5)
#define SECC0_SADDR     (0xE6350000U)
#define SFUSA0_FILL     (FUSA_0FILL_DISABLE)
```

6.3.8.3 PFC/GPIO setting

Execute the PFC/GPIO setting for the Salvator-X/XS and Ebisu/Ebisu-4D.
The following shows file and directory structure of The PFC/GPIO setting.

```
CortexR7_Loader
 ├──include
 │        pfc_init.h
 └──pfc
      │   pfc.mk
      │   pfc_init.c
      │
      ├──E3
      │        pfc_init_e3.c
      │        pfc_init_e3.h
      │
      ├──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
```

If customization is required, modify the files and function listed in Table 6-40.

Table 6-40 File and Function list of PFC/GPIO setting

Product ID	Cut Number	Filename	Function
H3	Ver.3.0 Ver.2.0	pfc_init_h3_v2.c	pfc_init_h3_v2()
M3	Ver.3.0 Ver.1.3 Ver.1.2 Ver.1.1	pfc_init_m3.c	pfc_init_m3()
M3N	Ver.1.1	pfc_init_m3n.c	pfc_init_m3n()
E3	Ver.1.1 Ver.1.0	pfc_init_e3.c	pfc_init_e3()

6.3.8.4 SDRAM setting

Set for access to the LPDDR4-SDRAM implemented in the Salvator-X/XS and the DDR3L-SDRAM implemented in the Ebisu/Ebisu-4D.
The following shows file and directory structure of the SDRAM setting.

```
CortexR7_Loader
\-----ddr
      |   boot_init_dram.h
      |   ddr.mk
      |   dram_sub_func.c
      |   dram_sub_func.h
      |
      |----ddr_a
      |         boot_init_dram_regdef_e3.h
      |         ddr_a.mk
      |         ddr_init_e3.c
      |         ddr_init_e3.h
      |
      |----ddr_b
      |         boot_init_dram.c
      |         boot_init_dram_config.c
      |         boot_init_dram_regdef.h
      |         ddr_b.mk
      |         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_regcheck
                boot_init_dram_regcheck.c
                ddr_b.mk
                init_dram_tbl_chk.h
```

SDRAM settings is executed by InitDram() function implemented in ddr_init_e3.c or boot_init_dram.c.
boot_init_dram_config.c is used to configure the board.

Internal bus interface check of DDR-PHY register is executed by InitDram_regcheck() function implemented in boot_init_dram_regcheck.c. It can select whether execute InitDram_regcheck() function by specifying build option “RCAR_DDR_REG_CHECK”. See 6.3.3 Build option about “RCAR_DDR_REG_CHECK”.
Note) IPL does NOT check the return value of InitDram_regcheck(). So users need to implement what IPL should do if InitDram_regcheck() is failed.

Set SDRAM bus clock setting corresponding to product id.
SDRAM bus clock is switched by MD pin.
Table 6-41 shows SDRAM bus clock setting.

Table 6-41 SDRAM bus clock

Product ID	Cut Number	MD19	MD17	SDRAM bus clock
H3	Ver.3.0 Ver.2.0	0	0	1600MHz(LPDDR4-3200)
M3	Ver.3.0 Ver.1.3 Ver.1.2 Ver.1.1	0	0	1600MHz(LPDDR4-3200)
M3N	Ver.1.1	0	0	1600MHz(LPDDR4-3200)
E3	Ver.1.1 Ver.1.0	0	-	792MHz(DDR3L-1600) *1
		1	-	928MHz(DDR3L-1856)

*1 1584Mbps is recommended for DDR3L-1600.

If customization is required, please base on the above.

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

Table 6-42 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)	-	-
	Ver.2.0			-	-	-
R-Car M3	Ver.3.0	8GB (4GB x2)	2 or 3 *1 (2 channel split)	-	-	-
	Ver.1.3 Ver.1.2 Ver.1.1	4GB (2GB x2)				
R-Car M3N	Ver.1.1	2GB (2GB x1)	0 or 3 *1 (No split)	-	-	-
R-Car E3	Ver.1.1	2GB (512MB x4)		-	1 (2G Byte)	1 (4 SDRAM)
	Ver.1.0	1GB (512MB x2)		-	0 (1G Byte)	0 (2 SDRAM)

*1 Refer to value 3 of RCAR_DRAM_SPLIT in Table 6-8.

Supported memory configurations are determined by a combination of build options.
The combinations that are not explicitly listed above are not supported.

6.3.8.5 QoS arbitration setting

QoS arbitration setting is possible to select Enable or Disable 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.

```
CortexR7_Loader
└──qos
    │   qos.mk
    │   qos_common.h
    │   qos_init.c
    │   qos_init.h
    │   qos_reg.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
    │       qos_init_m3_v30_mstat195.h
    │       qos_init_m3_v30_mstat390.h
    │       qos_init_m3_v30_qoswt195.h
    │       qos_init_m3_v30_qoswt390.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
```

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 below.

Table 6-43 File and Function list of QoS arbitration setting

Product ID	Cut Number	Filename	Function
H3	Ver.3.0	qos_init_h3_v30.c	qos_init_h3_v30()
	Ver.2.0	qos_init_h3_v20.c	qos_init_h3_v20()
M3	Ver.3.0	qos_init_m3_v30.c	qos_init_m3_v30()
	Ver.1.3 Ver.1.2 Ver.1.1	qos_init_m3_v11.c	qos_init_m3_v11()
M3N	Ver.1.1	qos_init_m3n_v10.c	qos_init_m3n_v10()
E3	Ver.1.1 Ver.1.0	qos_init_e3_v10.c	qos_init_e3_v10()

The QoS data tables shown in the Figure 6-2 in respectively file of the Table 6-43. 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 6-2 QoS data tables (H3 Ver.3.0/Ver.2.0, M3 Ver.3.0/Ver.1.3/Ver.1.2/Ver.1.1, M3N and E3)

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.

6.3.8.6 Security and Safety access protection setting

Execute the Security and Safety access protection setting for the Salvator-X/XS and the Ebisu/Ebisu-4D.
The following shows file and directory structure of The Security and Safety access protection setting.

```
CortexR7_Loader
└──common
    ├      lifec_init.c
    └      protection_setting.c
```

To change the Security and Safety access protection setting parameter will edit lifec_init.c.
lifec_init.c sets the values of the Security and Safety access protection setting register.
Show the code of lifec_init.c below.

```
void lifec_init(void)
{
        uint32_t loop;
        const uint32_t lifec_reg_tbl[][2] = {
                {       SEC_GRP0CR2  ,          0x00020000U     },
                {       SEC_GRP1CR2  ,          0x00020000U     },
                {       SAFE_GRP0CR2,           0x00000000U     },
                {       SAFE_GRP1CR2,           0x00000000U     },
                                :
```

If want to change the Security and Safety access protection setting, please refer to 1.3.1 Related document No.5 ‘R-Car Series, 3rdGeneration User’s Manual: Hardware, chapter 68 Life Cycle’ and please edit lifec_init.c.

To change the System RAM/SDRAM security access protection setting parameter will edit protection_setting.c.

```
/* AXI settings */
static const struct {
        uint32_t adr;
        uint32_t val;
} axi_reg_setting[] = {
         /* AXI dram protected area division */
        {AXI_DPTDIVCR0,                 0x0E0403F0U},
        {AXI_DPTDIVCR1,                 0x0E0407E0U},
                :
        /* AXI dram protected area setting */
        {AXI_DPTCR0,            0x0E000000U},
        {AXI_DPTCR1,            0x0E000E0EU},
                :
```

If you change to Security and Safety access protection setting, change the value of AXI_DPTCRn.
The following is an example of Security and Safety access protection setting.

```
        /* AXI dram protected area setting */
        {AXI_DPTCR0,            0x0E000E0EU}, /* Security */
        {AXI_DPTCR1,            0xE000E0E0U}, /* Safety */
                :
```

If want to change the Security and Safety access protection setting, please refer to 1.3.1 Related document No.5 ‘R-Car Series, 3rd Generation User’s Manual: Hardware, chapter 15 AXI-bus’ and please edit protection_setting.c.

If CR7 is Non-secure and uses interrupt after IPL, following setting must be done before CR7 is set to Non-secure.

```
void gic_security_setting(void)
{
        /* When use function of interruption after IPL,                    */
        /* set the value to GICD_IGROUPRn registers and GICC_PMR register  */
        /* while CR7 is Secure (before calling lifec_cr7_setting function). */
        gicc_write_pmr(GICC_BASE, GIC_PRI_MASK);
}
```

6.3.8.7 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.

```
CortexR7_Loader
 ├──common
 │        swdt.c
 └──include
          swdt.h
```

Process timeout detection setting is performed by swdt_init() function implemented in swdt.c.

Setting value of wait time is Figure 6-3 defined in swdt.c. The settable range is 1 to 10 seconds.

```
#define SWDT_COUNT_SEC                (5U)                /* set param(1--10sec) */
```

Figure 6-3 setting value of wait time

Note) If SWDT_COUNT_SEC is in invalid range, it occurs build error.

6.3.8.8 Image load area check setting

Execute the image load area check setting for the Salvator-X/XS and the Ebisu/Ebisu-4D.
The following shows file and directory structure of the image load area check setting.
The range check is executed by check_load_area function implemented in tcm_loader_main.c.
To execute the range check, set the Flash Memory capacity to FLASH_SIZE defined in tcm_loader_main.c.

```
CortexR7_Loader
 └──tcm_loader
          tcm_loader_main.c
```

tcm_loader_main.c

```
static void check_load_area(uint32_t dst, uint32_t src,
                                uint32_t len, uint8_t load_check)
{
          static const struct address_range add_list[] = {
                    :
}
```

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

tcm_loader_main.c

```
#define FLASH_SIZE        (0x04000000U)    /* 64MB */
```

6.3.8.9 Loading images

This clause shows the location that placed images. Table 6-44 shows the location of the top address to write IPL and the code in which the address is defined.

Table 6-44 Location of the top address to write IPL

Filename	Default Top Address	Path	Define
bootparam_sa0.srec	0xE6320000	Makefile	Line:415, 416
cr7_loader.srec	0xE6304000	loader/loader.ld.S	Line:13
		tools/dummy_create/sa0.c	Line:9
cert_header_sa3.srec	0xE6320000	Makefile	Line:424, 425
dummy_rtos.srec	0x70000000	tools/dummy_create/sa3.c	Line:9
bl2-<board_name>.srec	0xE6304000	tools/dummy_create/sa3.c	Line:11

In an environment that provided by R-Car H3/M3/M3N/E3, Loader load images from HyperFlash/QSPI Flash to SDRAM. Figure 6-4 and Figure 6-5 shows memory map to load the image from IPL image area to I-TCM. If you want to customize the memory map, refer to Table 6-44.

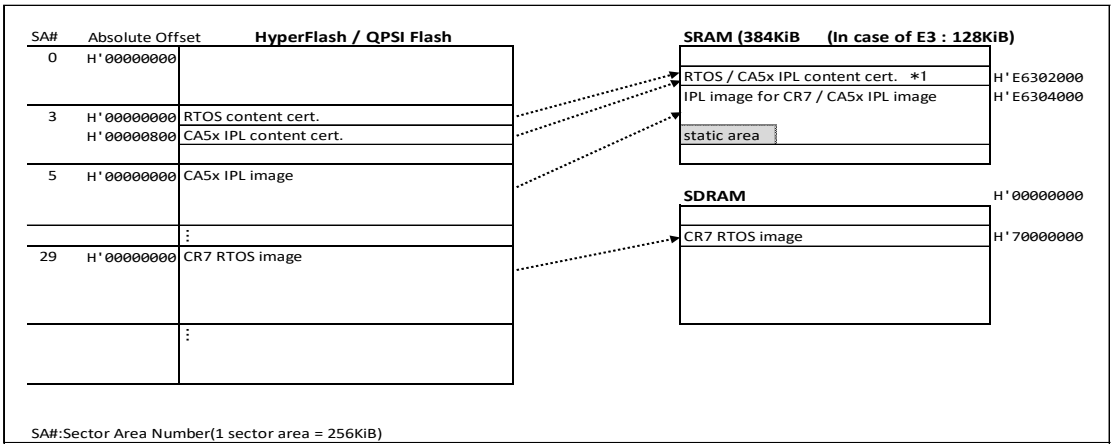


Figure 6-4 Memory Map

*1) After RTOS image is loaded and authenticated, CA5x IPL image is loaded and authenticated. Therefore, there is no problem to load each certificate to same address.

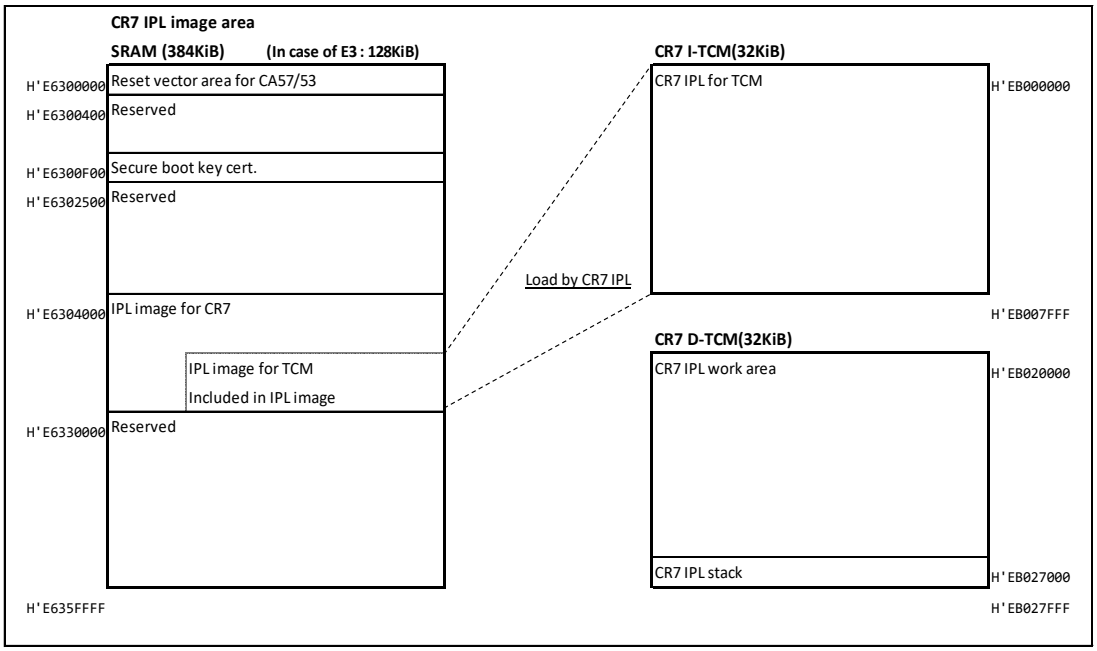


Figure 6-5 Memory map of TCM

7. Writing of IPL/Secure

Refer to document of Table 1-3[1].

8. Appendix

8.1 Periodic write DQ training

8.1.1 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.

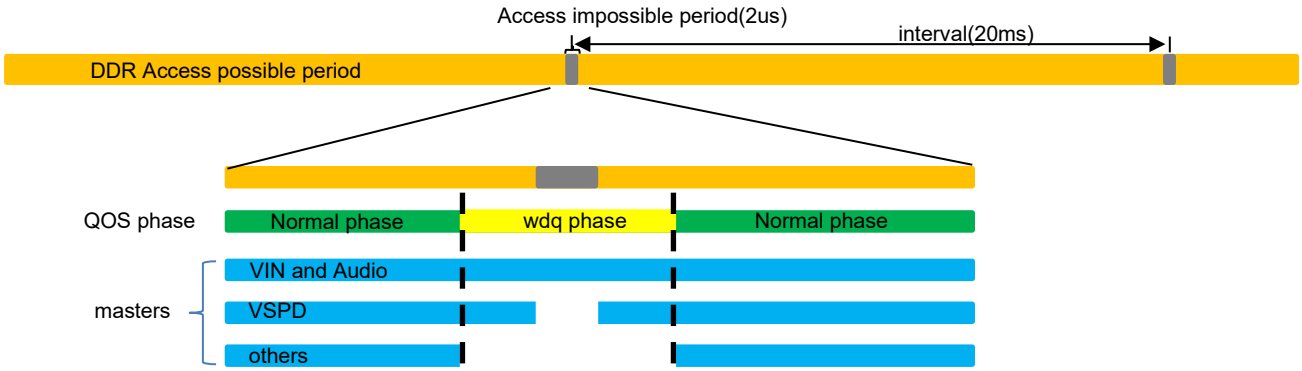
The interval of periodic write DQ training is 20ms and a write DQ training period is about 2us. In the period, each of read/write requests cannot access DDR. Therefore, sensitive latency master like VSPD or VIN may report error because of latency deterioration.

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.

8.1.2 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 for CR7: Software

Rev.	Date	Description	
		Page	Summary
1.0.0	May 31, 2017	—	New creation.
1.0.1	Jul. 7, 2017	—	Fixed the format of the document (trademark, etc.)
		1	Changed the name to "R-Car Series, 3rd Generation".
1.0.2	Nov.24, 2017	24,25	Changed RMSTPCR setting value and Hardware initialization.
		43	Changed Function's name.
1.0.3	Mar.9, 2018	—	Modified the format of the document (NOTICE, etc.)
		1	1.2 Function Added the M3N.
		3	1.2.2 Display the starting message Modified the Explanation of LSI version (PRR).
		9	1.3.1 Related Document Add related document of the M3N.
		12	3.1 Hardware Environment Added the M3N.
		13	3.2 Module Configuration Added the Notes.
		18	3.3.1 CPU initialization Update the Explanation of the Process "Enable the instruction cache".
		19,21, 23,24	3.3.4 PFC initialization Modified the Hardware Initialization.
		25-26	3.3.4 PFC initialization Added the PFC setting for the M3N.
		27	3.3.5 Module stop initialization Modified the Table number.
		28	3.3.5 Module stop initialization Modified the Table number. Added module stop setting for M3N.
		29	3.3.6 SCIF2 initialization Modified the Table number.
		29	3.3.7 SDRAM/PHY initialization Modified from "SARAM" to "SDRAM".
		30	3.3.10 Security and Safety setting Modified the Table number.
		32	3.3.10 Security and Safety setting Modified the Table number.
		34	3.3.10 Security and Safety setting Modified the Table number.
		36	4.1 Memory Constitution Modified from "certification" to "certificate". Removed the "Note" column.
		40	6.1 Directory configuration Added the M3N.
		42	6.3.3 Build option Added which default value is in Table 6-3.
		42	6.3.3 Build option Added the Note in RCAR_KICK_MAIN_CPU.
		43	6.3.3 Build option Added the Note in RCAR_DRAM_SPLIT and RCAR_DRAM_ECC.
		44	6.3.3 Build option Added the M3N in "Target LSI type setting" in table 6-9. Added the "R-Car M3N's CUT number setting of IPL program" column in Table 6-10. Added the Note in "LSI_CUT"option.
		46	6.3.8 How to customize Modified the Path of RPC initialization in Table 6-12.

		48	6.3.8.2 PFC/GPIO setting Added directory, files and PFC/GPIO setting of M3N.
		49	6.3.8.3 SDRAM setting Added a file of the M3N and information of SDRAM bus clock for M3N.
		50	6.3.8.4 QoS arbitration setting Added a directory and the files of the M3N in QoS arbitration setting.
		51	6.3.8.4 QoS arbitration setting Added the M3N in Table 6-19.
		54	6.3.8.8 Loading images Modified the “Line” number in Table 6-20.
1.0.4	Oct. 31, 2018	1	1.1 Overview Added “DDR3L-SDRAM”.
		3	1.2.2 Display the starting message Modified Figure 1-2.
		4	Table 1-2 Meaning of starting message Added following items. Loader version / SSCG Clock state / DRAM refresh interval / Whether CA5x is kicked.
		5	1.2.4 Loading the image Added CA5x Loader max image size for R-Car E3.
		10	Table 1-3 Related Document Added R-CarE3 System Evaluation Board “Ebisu”.
		12	Table 2-1 Terminology Added “DDR3/DDR3L-SDRAM”.
		13	Table 3-1 Hardware environment (R-Car Series, 3rd Generation) Added R-Car E3 System Evaluation Board (Ebisu).
		14	Figure 3-2 Module configuration Added Ebisu and R-Car E3.
		15	Table 3-2 Hardware resource Added DDR3L-SDRAM.
		20	Table 3-6 PFC setting for H3 Ver.1.x Modified Setting value of GPSR3.
		22	Table 3-7 PFC setting for H3 Ver.2.0 / Ver.3.0 Modified Setting value of GPSR3.
		28	Table 3-10 PFC setting for E3 Ver.1.0 Added Table for R-Car E3.
		29	Table 3-11 Module stop setting for H3 Ver.1.x Table 3-12 Module stop setting for H3 Ver.2.0 / Ver.3.0 Modified Setting value of RMSTPCR0 / RMSTPCR2.
		30	Table 3-13 Module stop setting for M3 Ver.1.x Table 3-14 Module stop setting for M3N Ver.1.x Modified Setting value of RMSTPCR0 / RMSTPCR2.
		31	Table 3-15 Module stop setting for E3 Ver.1.0 Added Table for R-Car E3.
		33	Table 3-18 Register setting value of the LifeC Added Setting value for R-Car E3.
		43	Figure 6-1 Directories for Loader of R-Car H3/M3/M3N Added R-Car E3.
		44	Table 6-1 Components to build Modified compiler version.
		47	Table 6-7 Association table for the RCAR_DRAM_SPLIT value and DRAM split setting Added case of RCAR_DRAM_SPLIT is 3. Table 6-8 Association table for the RCAR_DRAM_SPLIT and LSI Added this Table.
		48	6.3.3 Build option “RCAR_DRAM_ECC_ENABLE” Added Note. Table 6-10 Association table for the LSI string and Target LSI type setting Added R-Car E3.
		49	Table 6-11 Association table for the LSI_CUT value and R-Car CUT number setting.
		50	Added following build options. RCAR_REF_INT / RCAR_REWT_TRAINING / RCAR_SA0_SIZE.
		51	6.3.4 Prepare the compiler 6.3.6 Build the Loader

			6.3.7 Build the Dummy RTOS Modified compiler version.
		52	Table 6-16 Location of a code implemented functions Modified line number of Function's name.
		53	6.3.8.1 ECC setting of SDRAM Added Note.
		55	6.3.8.2 PFC/GPIO setting Added R-Car E3
		56	Table 6-21 File and Function list of PFC/GPIO setting Added R-Car E3.
		57	6.3.8.3 SDRAM setting Modified directory constitution. Added R-Car E3.
		58	Table 6-22 SDRAM bus clock Added R-Car E3.
		59	6.3.8.4 QoS arbitration setting Added R-Car E3 and H3 Ver.3.0.
		60	Table 6-23 File and Function list of QoS arbitration setting Added R-Car E3 and H3 Ver.3.0.
		61	Added R-Car E3.
		62	6.3.8.5 Security and Safety access protection setting Added Ebisu.
		64	6.3.8.6 Process timeout detection setting Added Note.
		65	6.3.8.7 Image load area check setting Added Ebisu. Table 6-24 Location of the top address to write IPL Modified Line number.
		66	Figure 6-5 Memory Map Modified address of the CA5x IPL Image.
1.0.5	Dec. 21, 2018	4	1.2.2 Display the starting message Added Ebisu-4D.
		10	1.3.1 Related Document Added Ebisu-4D to Table 1-3.
		13	3.1 Hardware Environment Added Ebisu-4D to Table 3-1.
		14	3.2 Module Configuration Added Ebisu-4D.
		20	3.3.4 PFC initialization Added Ebisu-4D.
		20-21	3.3.4 PFC initialization Removed Table 3-6.
		24-25	3.3.4 PFC initialization Table 3-8 Modified M3 Ver.1.x to M3 Ver.1.1.
		29	3.3.5 Module stop initialization Added Ebisu-4D.
		29	3.3.5 Module stop initialization Removed Table 3-11.
		30	3.3.5 Module stop initialization Table 3-13 Modified M3 Ver.1.x to M3 Ver.1.1.
		43	6.1 Directory configuration Modify directory construction in Figure 6-1.
		49	6.3.3 Build option Removed H3 Ver.1.0 / Ver.1.1 and M3 Ver.1.0.
		51	6.3.3 Build option Added build option "RCAR_DRAM_DDR3L_MEMDUAL" and Table 6-17.
		56	6.3.8.2 PFC/GPIO setting Added Ebisu-4D.
		57	6.3.8.2 PFC/GPIO setting Removed H3 Ver.1.0 / Ver.1.1 and M3 Ver.1.0.
		58	6.3.8.3 SDRAM setting Added Ebisu-4D.

		59	6.3.8.3 SDRAM setting Removed H3 Ver.1.0 / Ver.1.1 and M3 Ver.1.0.
		59	6.3.8.3 SDRAM setting Added Table 6-26.
		61	6.3.8.3 SDRAM setting Removed H3 Ver.1.0 / Ver.1.1 and M3 Ver.1.0. Removed Figure 6-2.
		63	6.3.8.5 Security and Safety access protection setting Added Ebisu-4D.
		66	6.3.8.7 Image load area check setting Added Ebisu-4D.
1.0.6	Mar. 29, 2019	2	Table 1-1 Added QSPI Flash.
		4	Table 1-2 Modified Loader version. Added R-Car E3 to DRAM configuration. Modified from DRAM ECC to Dual Channel ECC and Single Channel ECC.
		5	1.2.4 Loading the image Added QSPI Flash.
		10	Table 1-3 Related Document Added No.10 'R-Car Series, 3rd Generation Safety Application Note' and No.11 'R-Car Series, 3rd Generation Hardware Description For Functional Safety'.
		11	1.4 Restrictions Modified from DRAM ECC to Dual Channel ECC.
		12	2. Terminology Added FuSa to Table 2-1.
		17	Figure 3-3 Flow of Loader processing Added "(L1 Cache/TCM)" to ECC initialization.
		18	Figure 3-4 Flow of Loader processing in TCM Added "(SDRAM/System RAM)" to ECC initialization.
		19	Table 3-4 CPU initializing process Added process of Global Timer configuration. Added process of Clearing bss section. Added process of Copying data section. Added process of Loading TCM Loader.
		19	3.3.2 ECC initialization Modified "DRAM ECC" to "ECC setting of SDRAM and System RAM".
		27-28	3.3.5 Module stop initialization Modified the value of RMSTPCR4 and RMSTPCR9 in Table 3-10, Table 3-11, Table3-12, and Table 3-13. Modified the value of RMSTPCR8 in Table 3-12.
		29	3.3.7 SDRAM/PHY initialization Added QSPI Flash.
		29	3.3.8 RPC initialization Added QSPI Flash.
		29	3.3.9 DMA initialization Added QSPI Flash.
		36	4.1 Memory Constitution Moved RTOS certificate from SDRAM to System RAM.
		37	Figure 4-1 Memory layout Modified the place of RTOS Certificate.
		38	4.2 Memory Layout Added QSPI Flash.
		44	Table 6-8 Association table for the RCAR_DRAM_SPLIT and LSI Modified the Notes when LSI=AUTO is specified.
		45	Table 6-9 Association table for the RCAR_ECC_ENABLE value and ECC setting Added Single Channel ECC.
		45	Added "Table 6-10 Association table for the RCAR_ECC_ENABLE and LSI".
		47	Table 6-15 Association table for the RCAR_REWT_TRAINING value and Periodic write DQ training Modified default setting value and Notes.
		51	Table 6-20 Location of a code implemented functions

			Modified overall.
		52-54	Removed “DRAM ECC setting” and added “6.3.8.1 Dual Channel ECC setting”.
		55-61	Added “6.3.8.2 Single Channel ECC setting”.
		70	Table 6-42 Location of the top address to write IPL Modified column of Define.
		71	Figure 6-4 Memory Map Modified the place of RTOS content cert. Added size of System RAM.
		73	Added 8. Appendix and following chapters. 8.1 Periodic write DQ training 8.1.1 Outline 8.1.2 QoS control of periodic write DQ training
1.0.7	Jun. 28, 2019	2	1.2.1 Hardware initialization Added GPIO initialization
		4	Table 1-2 Meaning of starting message Added Periodic Write DQ Training. Added DDR-PHY REG check.
		12	Table 2-1 Terminology Added GPIO.
		16	Table 3-3 Explanation of Loader’s functions Added GPIO.
		20	3.3.4 PFC / GPIO initialization Added GPIO.
		20	Table 3-6 PFC setting for H3 Ver.2.0 / Ver.3.0 Modified Setting value of GPSR7.
		22	Added Table 3 7 GPIO setting for H3 Ver.2.0 / Ver.3.0.
		23	Table 3-8 PFC setting for M3 Ver.1.1 / Ver.1.2 / Ver.1.3 / Ver.3.0 Added M3 Ver.1.3/Ver.3.0. Modified Setting value of GPSR7.
		25	Added Table 3-9 GPIO setting for M3 Ver.1.1 / Ver.1.2 / Ver.1.3 / Ver.3.0
		26	Table 3-10 PFC setting for M3N Ver.1.1 Modified Setting value of GPSR7.
		28	Added Table 3-11 GPIO setting for M3N Ver.1.1
		30	Added Table 3-13 GPIO setting for E3 Ver.1.0 / Ver.1.1
		33	3.3.7 SDRAM/PHY initialization Added Internal bus interface check of DDR-PHY register. Added Note.
		45	Table 6-1 Components to build Updated compiler version.
		45	6.3.2 Environment Updated version of Ubuntu. (Because Ubuntu 14.04LTS is end of support.)
		49	Table 6-10 Association table for the RCAR_ECC_ENABLE and LSI Added LSI_CUT and M3 Ver.3.0.
		50	Table 6-12 Association table for the LSI_CUT value and R-Car CUT number setting. Added LSI_CUT=13 and M3 Ver.3.0.
		51	Table 6-14 Association table for the RCAR_REF_INT value and DRAM refresh interval Table 6-15 Association table for the RCAR_REWT_TRAINING value and Periodic write DQ training Added M3 Ver.3.0 / Ver.1.3 to Note.
		52	Table 6-17 Association table for the RCAR_DRAM_DDR3L_MEMCONF value and the DRAM memory size Removed a case of RCAR_DRAM_DDR3L_MEMCONF=2.
		54	6.3.4 Prepare the compiler 6.3.6 Build the Loader 6.3.7 Build the Dummy RTOS Updated compiler version.
		55	Table 6-20 Location of a code implemented functions Added GPIO initialization.
		56	Table 6-21 Setting value list of Area size and Extra Area size Added two definitions.
		59	6.3.8.2 Single Channel ECC setting Added M3 Ver.3.0 to Notes.

		60	Table 6-27 Setting value list of DFUSAn_SADDR(n=0-15) in single_ecc_conf.h Modified from UL to U.
		61	Table 6-30 Setting value list of DECCn_SADDR (n=0-15) in single_ecc_conf.h Modified from UL to U.
		63	Table 6-33 Setting value list of SFUSAn_SADDR (n=0-7) in single_ecc_conf.h Modified from UL to U.
		64	Table 6-36 Setting value list of SECCn_SADDR (n=0-7) in single_ecc_conf.h Modified from UL to U.
		65	6.3.8.2 Single Channel ECC setting Added reference document chapter.
		67	Table 6-38 File and Function list of PFC/GPIO setting Added M3 Ver.1.3/ Ver.3.0 and E3 Ver.1.1.
		68	6.3.8.4 SDRAM setting Added ddr_b_regcheck to directory structure. Added description about Internal bus interface check of DDR-PHY register. Added Note about Internal bus interface check of DDR-PHY register.
		69	Table 6-39 SDRAM bus clock Table 6-40 Combination of DRAM configuration related build options Added M3 Ver.1.3/ Ver.3.0 and E3 Ver.1.1.
		70	6.3.8.5 QoS arbitration setting Added files related with M3 Ver.3.0.
		71	Table 6-41 File and Function list of QoS arbitration setting Added M3 Ver.1.3/ Ver.3.0 and E3 Ver.1.1.
		75	6.3.8.9 Loading images Modified column of Define.
1.0.8	Jul. 5, 2019	33	3.3.7 SDRAM/PHY initialization Added description about build option “RCAR_DDR_REG_CHECK”.
		53	6.3.3 Build option Added build option “RCAR_DDR_REG_CHECK”.
		68	6.3.8.4 SDRAM setting Added description about build option “RCAR_DDR_REG_CHECK”.
1.0.9	Sep. 19, 2019	4	1.2.2 Display the starting message Added H3/M3 to the DDR-PHY REG check in Table 1-2.
		33	3.3.7 SDRAM/PHY initialization Added M3 Ver.1.2/Ver.1.3/Ver.3.0 to Internal bus interface check of DDR-PHY register.
		53	6.3.3 Build option Added M3 Ver.1.2/Ver.1.3/Ver.3.0 to the Note in Table 6-20.
1.0.10	Dec. 16, 2019	75	6.3.8.8 Image load area check setting Added description about check_load_area. Added note when customizing range check.
1.0.11	Mar. 1, 2021	12	2. Terminology Table 2-1 Terminology Added GIC.
		16	3.3 Processing Flow Diagram Table 3-3 Explanation of Loader’s functions Added GIC initialization. Added Release HW resource.
		17	3.3 Processing Flow Diagram Figure 3-3 Flow of Loader processing Added GIC initialization.
		18	3.3 Processing Flow Diagram Figure 3-4 Flow of Loader processing in TCM Added “Release HW resource”. Modified sequence of “Starting CA57/53” and “Security and Safety setting”.
		33	3.3 Processing Flow Diagram Added Chapter 3.3.7 GIC initialization.
		39	3.3 Processing Flow Diagram Added Chapter 3.3.13 Release HW resource.
		74	6.3.8.6 Security and Safety access protection setting Added description about gic_security_setting function.
1.0.12	Apr. 15, 2021	10	1.3.1 Related Document Added No.12 ‘Initial Program Loader User’s Manual: Software R-Car H3/M3/M3N/E3

			Series' in Table 1-3.
		42-44	4. Memory Added 4.3 Release image. Added 4.4 Example of writing data.
		57	6.3.8 How to customize Modified Function's name column in Table 6-21.
		77	6.3.8.9 Loading images Modified Define column in Table 6-43.
1.0.13	Jan. 21, 2022	5	1.2.2 Display the starting message Added Starting message in Table 1-2
		56	6.3.3 Build option Added build option in Table 6-21
		77	6.3.8.9 Loading images Modified the Default Top Address of dummy_RTOS in Table 6-44
		78	6.3.8.9 Loading images Modified the Default Top Address of dummy_RTOS in Figure 6-4
1.0.14	Apr. 25, 2022	11	1.4 Removed hardware restrictions

CONFIDENTIAL

Initial Program Loader for CR7
User's Manual: Software

Publication Date: Rev.1.0.0 May 31, 2017
Rev.1.0.14 Apr. 25, 2022

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

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/

Initial Program Loader for CR7
User's Manual

Renesas Electronics Corporation