

# RH850/U2B Group

Flash Memory

User's Manual: Hardware

Renesas microcontroller

RH850 Family

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

## 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. RENASAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENASAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENASAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENASAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENASAS 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](http://www.renesas.com)

## Trademarks

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

## Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:  
[www.renesas.com/contact/](http://www.renesas.com/contact/)

## General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

### 1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

### 2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

### 3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

### 4. Handling of unused pins

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to power supply or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

### 5. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

### 6. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

### 7. Power ON/OFF sequence

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

# Table of Contents

|           |                                                                         |    |
|-----------|-------------------------------------------------------------------------|----|
| Section 1 | Features.....                                                           | 6  |
| Section 2 | Module Structure.....                                                   | 7  |
| Section 3 | Address Map.....                                                        | 8  |
| Section 4 | Registers.....                                                          | 9  |
| 4.1       | Register Base Address .....                                             | 9  |
| 4.2       | Registers Related to Programming/Erase of Flash Memory .....            | 10 |
| 4.3       | FLMDCNT —FLMD Control Register .....                                    | 12 |
| 4.4       | FLMDFPLOCK —FLMD FACL LOCK Register.....                                | 13 |
| 4.5       | FHVE3FPn — FHVE3 Control Register for FACLn (n = 0, 2) .....            | 14 |
| 4.6       | FHVE15FPn — FHVE15 Control Register for FACLn (n = 0, 2) .....          | 15 |
| 4.7       | FPMON_n — Flash Pin Monitor Register (n = 0, 2) .....                   | 16 |
| 4.8       | FASTAT_n — Flash Access Status Register (n = 0, 2).....                 | 17 |
| 4.9       | FAEINT_n — Flash Access Error Interrupt Enable Register (n = 0, 2)..... | 19 |
| 4.10      | FAREASELC — Code Flash Memory Area Select Register .....                | 20 |
| 4.11      | FSADDR_n — Flash Command Start Address Register (n = 0, 2) .....        | 21 |
| 4.12      | FEADDR_n — Flash Command End Address Register (n = 0, 2).....           | 23 |
| 4.13      | FCVAPROT_n — Code Flash Valid Area Protection Register (n = 0, 2).....  | 24 |
| 4.14      | FSTATR_n — Flash Status Register (n = 0, 2) .....                       | 25 |
| 4.15      | FENTRYR_n — Flash Programming/Erase Mode Entry Register (n = 0, 2)..... | 31 |
| 4.16      | FSUINTR_n — Flash Sequencer Set-up Initialize Register (n = 0, 2).....  | 33 |
| 4.17      | FCMDR_n — FACL Command Register (n = 0, 2) .....                        | 34 |
| 4.18      | FCMDMON_n — Flash Sequencer Command Monitor Register (n = 0, 2).....    | 35 |
| 4.19      | FSWASTAT_0 — Switch Area Status Register .....                          | 37 |
| 4.20      | FPESTAT_n — Flash Programming/Erase Status Register (n = 0, 2).....     | 38 |
| 4.21      | FBCCNT_n — Data Flash Blank Check Control Register (n = 0, 2).....      | 39 |
| 4.22      | FBCSTAT_n — Data Flash Blank Check Status Register (n = 0, 2) .....     | 40 |
| 4.23      | FPSADDR_n — Programmed Area Start Address Register (n = 0, 2).....      | 41 |
| 4.24      | FECCEMON_n — Flash ECC Encoder Monitor Register (n=0, 2).....           | 42 |
| 4.25      | FECCTMD_n — Flash ECC Test Mode Register (n = 0, 2) .....               | 43 |
| 4.26      | FDMYECC_n — Flash Dummy ECC Register (n = 0, 2).....                    | 44 |
| 4.27      | SPIDINn — Serial Programmer ID Input Register (n = 0 to 7) .....        | 45 |
| 4.28      | DFIDINn — Data Flash ID Input Register (n = 0 to 7) .....               | 46 |
| 4.29      | OCDIDINn — OCD ID Input Register (n = 0 to 7) .....                     | 47 |
| 4.30      | CUSTIDAINn — Customer ID A Input Register (n = 0 to 7) .....            | 48 |
| 4.31      | CUSTIDBINn — Customer ID B Input Register (n = 0 to 7) .....            | 49 |
| 4.32      | CUSTIDCINn — Customer ID C Input Register (n = 0 to 7).....             | 50 |
| 4.33      | IDST —ID Authentication Status Register .....                           | 51 |
| 4.34      | RHSIFIDINn — RHSIF ID Input Register (n = 0 to 7).....                  | 53 |
| 4.35      | IDST2 —ID Authentication Status Register 2 .....                        | 54 |

|            |                                                                          |     |
|------------|--------------------------------------------------------------------------|-----|
| Section 5  | Flash Sequencer Operating Modes .....                                    | 55  |
| Section 6  | FACI Command .....                                                       | 57  |
| 6.1        | List of FACI Commands.....                                               | 57  |
| 6.2        | Relationship between Flash Sequencer Status and FACI Commands .....      | 59  |
| 6.3        | FACI Command .....                                                       | 62  |
| 6.3.1      | Overview of the Command Usage in Code Flash Programming/Erase Mode ..... | 62  |
| 6.3.2      | Overview of the Command Usage in Data Flash Programming/Erase Mode ..... | 63  |
| 6.3.3      | Shift to Code Flash Programming/Erase Mode .....                         | 64  |
| 6.3.4      | Shift to Data Flash Programming/Erase Mode .....                         | 65  |
| 6.3.5      | Shift to Read Mode .....                                                 | 66  |
| 6.3.6      | ID Authentication .....                                                  | 67  |
| 6.3.7      | Return from Command Lock State .....                                     | 68  |
| 6.3.8      | Issuing of Programming Command .....                                     | 69  |
| 6.3.9      | Issuing of Multi Programming Command .....                               | 72  |
| 6.3.10     | DMA Programming Command .....                                            | 74  |
| 6.3.11     | Block Erase Command.....                                                 | 75  |
| 6.3.12     | Area Erase Command.....                                                  | 76  |
| 6.3.13     | Programming/Erase Suspension Command .....                               | 77  |
| 6.3.14     | Programming/Erase Resumption Command.....                                | 81  |
| 6.3.15     | Status Clearing Command .....                                            | 82  |
| 6.3.16     | Forced Stop Command .....                                                | 83  |
| 6.3.17     | Blank Checking Command .....                                             | 84  |
| 6.3.18     | Property Programming Command.....                                        | 86  |
| 6.3.19     | Property Erase Command.....                                              | 88  |
| 6.3.20     | Switching of Hardware Property Area .....                                | 89  |
| 6.3.21     | Recovering Switch Area Status .....                                      | 101 |
| 6.3.22     | Injecting ECC Errors for the Flash Memory .....                          | 102 |
| Section 7  | Security Function .....                                                  | 105 |
| Section 8  | Protection Function .....                                                | 106 |
| 8.1        | Software Protection .....                                                | 106 |
| 8.1.1      | Protection by FENTRYR.....                                               | 106 |
| 8.1.2      | Code Flash Valid area protection .....                                   | 106 |
| 8.2        | Error Protection.....                                                    | 106 |
| 8.3        | Boot Program Protection .....                                            | 109 |
| 8.3.1      | User Boot Protection .....                                               | 109 |
| Section 9  | Usage Notes .....                                                        | 110 |
| Section 10 | Electrical Characteristics.....                                          | 111 |

## Section 1 Features

The features of the flash memory are described below. See the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory* for information on the capacity, block configuration, and addresses of the flash memory in a given product.

### Flash Memory Programming/Erasure

A dedicated sequencer for the flash memory (flash sequencer) executes programming and erasure via Peripheral Bus. The flash sequencer also supports the programming/erasure processing suspension/resumption and BGO (Background Operation)/Multi FPSYS Operation.

For details of Background Operation and Multi FPSYS Operation, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

### Security Functions

The flash memory incorporates hardware functions to prevent illicit tampering.

### Protection Functions

The flash memory incorporates hardware functions to prevent erroneous writing.

### Interrupts

The flash memory supports an interrupt to indicate completion of processing by the flash sequencer and an error interrupt to indicate erroneous operations.

### DMA

The data flash memory can be programmed using the DMA.

## Section 2 Module Structure

This product contains 2 Flash Programming System (FPSYS0, FPSYS2).

For details, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

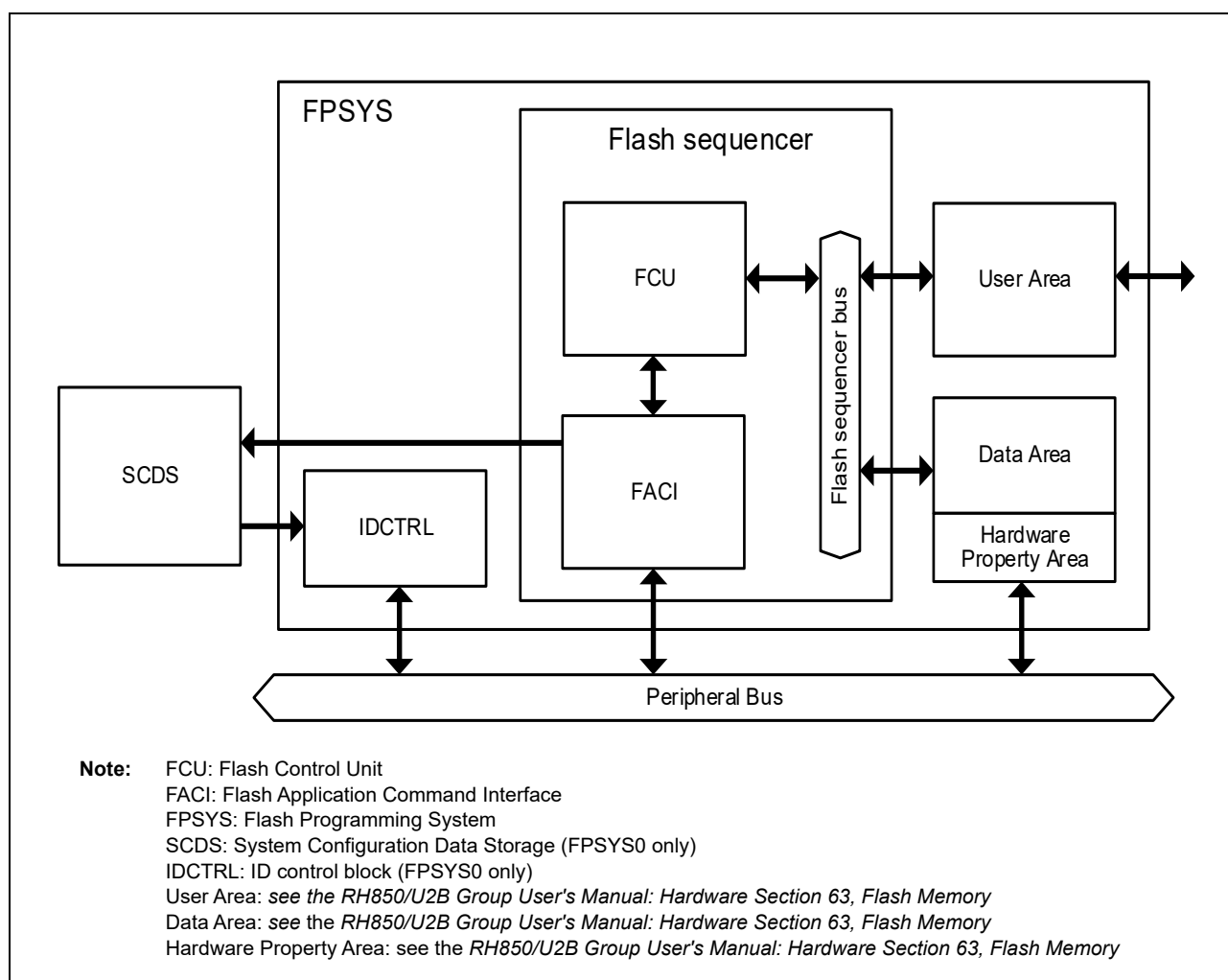
Module structure of each FPSYS is shown in **Figure 2.1**.

The flash sequencer contains Flash Control Unit (FCU) and Flash Application Command Interface (FACI). The FCU executes basic control of programming/erasure of the flash memory.

The FACI receives FACI commands via Peripheral Bus and controls FCU operations accordingly.

System Configuration Data Storage (SCDS) stores the data (configuration settings, security settings) which is transferred from flash memory by FACI at reset (FACI reset transfer).

ID control block (IDCTRL) compares the ID stored in the SCDS with the value input via Peripheral Bus for authentication.



**Figure 2.1** Structure of Flash Programming System

## Section 3 Address Map

**Table 3.1** gives information on all of these areas.

**Table 3.1 Information on the Hardware Interface Area**

| Area                                                  | Information                                                                     |
|-------------------------------------------------------|---------------------------------------------------------------------------------|
| Area containing the various registers of the hardware | See <b>Section 4, Registers</b> .                                               |
| FACI0 command-issuing area                            | See <b>Section 4, Registers</b> .                                               |
| FACI2 command-issuing area                            | See <b>Section 4, Registers</b> .                                               |
| User Area                                             | See the <i>RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory</i> |
| Data Area                                             | See the <i>RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory</i> |
| Extended Data Area <sup>*1</sup>                      | See the <i>RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory</i> |
| Configuration Setting Area <sup>*1</sup>              | See the <i>RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory</i> |
| Security Setting Area <sup>*1</sup>                   | See the <i>RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory</i> |
| Block Protection Setting Area <sup>*1</sup>           | See the <i>RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory</i> |
| Erase Counter <sup>*1</sup>                           | See the <i>RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory</i> |
| Switch Area <sup>*1</sup>                             | See the <i>RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory</i> |
| TAG Area <sup>*1</sup>                                | See the <i>RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory</i> |

Note 1. These areas are generically called "Hardware Property Area".



## Section 4 Registers

This section gives information on the registers.

Registers are reset only by the condition in **Table 4.4**, if no other condition is specified in each register description. For information of the option bytes, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*. For information of the registers to control from ICUMHB, see the *RH850/U2B Group Security User's Manual: Hardware*.

Register addresses of command-issuing area are given as follows.

**Table 4.1 Register Address of command-issuing area**

| Register Name              | Register Address       | Bus Group     |
|----------------------------|------------------------|---------------|
| FACI0 command-issuing area | FFA2 0000 <sub>H</sub> | P-Bus Group 6 |
| FACI2 command-issuing area | FFA4 0000 <sub>H</sub> | P-Bus Group 6 |

### 4.1 Register Base Address

Register addresses other than command-issuing area are given as offsets from the base addresses.

**Table 4.2 Register Base Address (other than command-issuing area)**

| Base Address Name | Base Address           | Bus Group      |
|-------------------|------------------------|----------------|
| <FACI0_base>      | FFA1 0000 <sub>H</sub> | P-Bus Group 6  |
| <FACI2_base>      | FFA1 8000 <sub>H</sub> | P-Bus Group 6  |
| <IDCTRL_base>     | FFA0 8000 <sub>H</sub> | P-Bus Group 6  |
| <SCDS_base>       | FFCD 0000 <sub>H</sub> | P-Bus Group 6  |
| <FLMD_base>       | FFA0 0000 <sub>H</sub> | P-Bus Group 6  |
| <FHVE_base>       | FF98 4800 <sub>H</sub> | P-Bus Group 2L |

## 4.2 Registers Related to Programming/Erase of Flash Memory

Table 4.3 shows the list of registers related to programming/erase of flash memory.

Table 4.3 Registers Related to Programming/Erase of Flash Memory (1/2)

| Unit Name | Register Name                          | Symbol     | Address                          | Access Size | Access Protection |       |
|-----------|----------------------------------------|------------|----------------------------------|-------------|-------------------|-------|
|           |                                        |            |                                  |             | PBG               | Other |
| FACI0     | Flash Pin Monitor 0                    | FPMON_0    | <FACI0_base> + 0000 <sub>H</sub> | 8           | PBG62#1           | —     |
| FACI0     | Flash Access Status 0                  | FASTAT_0   | <FACI0_base> + 0010 <sub>H</sub> | 8           | PBG62#1           | —     |
| FACI0     | Flash Access Error Interrupt Enable 0  | FAEINT_0   | <FACI0_base> + 0014 <sub>H</sub> | 8           | PBG62#1           | —     |
| FACI0     | Code Flash Memory Area Select Register | FAREASELC  | <FACI0_base> + 0020 <sub>H</sub> | 16          | PBG62#1           | —     |
| FACI0     | Flash Command Start Address 0          | FSADDR_0   | <FACI0_base> + 0030 <sub>H</sub> | 32          | PBG62#1           | —     |
| FACI0     | Flash Command End Address 0            | FEADDR_0   | <FACI0_base> + 0034 <sub>H</sub> | 32          | PBG62#1           | —     |
| FACI0     | Code Flash Valid Area Protection 0     | FCVAPROT_0 | <FACI0_base> + 0040 <sub>H</sub> | 16          | PBG62#1           | —     |
| FACI0     | Flash Status 0                         | FSTATR_0   | <FACI0_base> + 0080 <sub>H</sub> | 32          | PBG62#1           | —     |
| FACI0     | Flash Programming/Erase Mode Entry 0   | FENTRYR_0  | <FACI0_base> + 0084 <sub>H</sub> | 16          | PBG62#1           | —     |
| FACI0     | Flash Sequencer Set-up Initialize 0    | FSUINITR_0 | <FACI0_base> + 008C <sub>H</sub> | 16          | PBG62#1           | —     |
| FACI0     | FACI Command Register 0                | FCMDR_0    | <FACI0_base> + 00A0 <sub>H</sub> | 16          | PBG62#1           | —     |
| FACI0     | Flash Sequencer Command Monitor 0      | FCMDMON_0  | <FACI0_base> + 00A4 <sub>H</sub> | 32          | PBG62#1           | —     |
| FACI0     | Switch Area Status                     | FSWASTAT_0 | <FACI0_base> + 00A8 <sub>H</sub> | 8           | PBG62#1           | —     |
| FACI0     | Flash Programming/Erase Status 0       | FPESTAT_0  | <FACI0_base> + 00C0 <sub>H</sub> | 16          | PBG62#1           | —     |
| FACI0     | Data Flash Blank Check Control 0       | FBCCNT_0   | <FACI0_base> + 00D0 <sub>H</sub> | 8           | PBG62#1           | —     |
| FACI0     | Data Flash Blank Check Status 0        | FBCSTAT_0  | <FACI0_base> + 00D4 <sub>H</sub> | 8           | PBG62#1           | —     |
| FACI0     | Programmed Area Start Address 0        | FPSADDR_0  | <FACI0_base> + 00D8 <sub>H</sub> | 32          | PBG62#1           | —     |
| FACI0     | Flash ECC Encoder Monitor 0            | FECCEMON_0 | <FACI0_base> + 0100 <sub>H</sub> | 16          | PBG62#1           | —     |
| FACI0     | Flash ECC Test Mode 0                  | FECCTMD_0  | <FACI0_base> + 0104 <sub>H</sub> | 16          | PBG62#1           | —     |
| FACI0     | Flash Dummy ECC 0                      | FDMYECC_0  | <FACI0_base> + 0108 <sub>H</sub> | 16          | PBG62#1           | —     |
| FACI2     | Flash Pin Monitor 2                    | FPMON_2    | <FACI2_base> + 0000 <sub>H</sub> | 8           | PBG62#3           | —     |
| FACI2     | Flash Access Status 2                  | FASTAT_2   | <FACI2_base> + 0010 <sub>H</sub> | 8           | PBG62#3           | —     |
| FACI2     | Flash Access Error Interrupt Enable 2  | FAEINT_2   | <FACI2_base> + 0014 <sub>H</sub> | 8           | PBG62#3           | —     |
| FACI2     | Flash Command Start Address 2          | FSADDR_2   | <FACI2_base> + 0030 <sub>H</sub> | 32          | PBG62#3           | —     |
| FACI2     | Flash Command End Address 2            | FEADDR_2   | <FACI2_base> + 0034 <sub>H</sub> | 32          | PBG62#3           | —     |
| FACI2     | Code Flash Valid Area Protection 2     | FCVAPROT_2 | <FACI2_base> + 0040 <sub>H</sub> | 16          | PBG62#3           | —     |
| FACI2     | Flash Status 2                         | FSTATR_2   | <FACI2_base> + 0080 <sub>H</sub> | 32          | PBG62#3           | —     |
| FACI2     | Flash Programming/Erase Mode Entry 2   | FENTRYR_2  | <FACI2_base> + 0084 <sub>H</sub> | 16          | PBG62#3           | —     |
| FACI2     | Flash Sequencer Set-up Initialize 2    | FSUINITR_2 | <FACI2_base> + 008C <sub>H</sub> | 16          | PBG62#3           | —     |
| FACI2     | FACI Command Register 2                | FCMDR_2    | <FACI2_base> + 00A0 <sub>H</sub> | 16          | PBG62#3           | —     |
| FACI2     | Flash Sequencer Command Monitor 2      | FCMDMON_2  | <FACI2_base> + 00A4 <sub>H</sub> | 32          | PBG62#3           | —     |
| FACI2     | Flash Programming/Erase Error Status 2 | FPESTAT_2  | <FACI2_base> + 00C0 <sub>H</sub> | 16          | PBG62#3           | —     |
| FACI2     | Data Flash Blank Check Control 2       | FBCCNT_2   | <FACI2_base> + 00D0 <sub>H</sub> | 8           | PBG62#3           | —     |
| FACI2     | Data Flash Blank Check Status 2        | FBCSTAT_2  | <FACI2_base> + 00D4 <sub>H</sub> | 8           | PBG62#3           | —     |

Table 4.3 Registers Related to Programming/Erase of Flash Memory (2/2)

| Unit Name  | Register Name                           | Symbol                | Address                                                | Access Size                            | Access Protection |       |
|------------|-----------------------------------------|-----------------------|--------------------------------------------------------|----------------------------------------|-------------------|-------|
|            |                                         |                       |                                                        |                                        | PBG               | Other |
| FACI2      | Programmed Area Start Address 2         | FPSADDR_2             | <FACI2_base> + 00D8 <sub>H</sub>                       | 32                                     | PBG62#3           | —     |
| FACI2      | Flash ECC Encoder Monitor 2             | FECCEMON_2            | <FACI2_base> + 0100 <sub>H</sub>                       | 16                                     | PBG62#3           | —     |
| FACI2      | Flash ECC Test Mode 2                   | FECCTMD_2             | <FACI2_base> + 0104 <sub>H</sub>                       | 16                                     | PBG62#3           | —     |
| FACI2      | Flash Dummy ECC 2                       | FDMYECC_2             | <FACI2_base> + 0108 <sub>H</sub>                       | 16                                     | PBG62#3           | —     |
| IDCTR<br>L | Serial Programmer ID input n (n=0 to 7) | SPIDINn (n=0 to 7)    | <IDCTRL_base> + 0000 <sub>H</sub> to 001C <sub>H</sub> | 32                                     | PBG60#4           | —     |
| IDCTR<br>L | Data Flash ID input n (n=0 to 7)        | DFIDINn (n=0 to 7)    | <IDCTRL_base> + 0020 <sub>H</sub> to 003C <sub>H</sub> | 32                                     | PBG60#4           | —     |
| IDCTR<br>L | OCD ID input n (n=0 to 7)               | OCDIDINn (n=0 to 7)   | <IDCTRL_base> + 0040 <sub>H</sub> to 005C <sub>H</sub> | 32                                     | PBG60#4           | —     |
| IDCTR<br>L | Customer ID A input n (n=0 to 7)        | CUSTIDAINn (n=0 to 7) | <IDCTRL_base> + 0080 <sub>H</sub> to 009C <sub>H</sub> | 32                                     | PBG60#4           | —     |
| IDCTR<br>L | Customer ID B input n (n=0 to 7)        | CUSTIDBINn (n=0 to 7) | <IDCTRL_base> + 00A0 <sub>H</sub> to 00BC <sub>H</sub> | 32                                     | PBG60#4           | —     |
| IDCTR<br>L | Customer ID C input n (n=0 to 7)        | CUSTIDCINn (n=0 to 7) | <IDCTRL_base> + 00C0 <sub>H</sub> to 00DC <sub>H</sub> | 32                                     | PBG60#4           | —     |
| IDCTR<br>L | ID Authentication Status Register       | IDST                  | <IDCTRL_base> + 01FC <sub>H</sub>                      | 32                                     | PBG60#4           | —     |
| IDCTR<br>L | RHSIF ID input n (n=0 to 7)             | RHSIFIDINn (n=0 to 7) | <IDCTRL_base> + 0200 <sub>H</sub> to 021C <sub>H</sub> | 32                                     | PBG60#4           | —     |
| IDCTR<br>L | ID Authentication Status Register 2     | IDST2                 | <IDCTRL_base> + 03FC <sub>H</sub>                      | 32                                     | PBG60#4           | —     |
| FLMD       | FLMD Control Register                   | FLMDCNT               | <FLMD_base> + 0000 <sub>H</sub>                        | 32                                     | PBG60#5           | —     |
| FLMD       | FLMD FACI Lock Register                 | FLMDFPLOCK            | <FLMD_base> + 000C <sub>H</sub>                        | 32                                     | PBG60#5           | —     |
| FHVE       | FHVE3 Control Register for FACI0        | FHVE3FP0              | <FHVE_base> + 0000 <sub>H</sub>                        | 32                                     | PBG20#2           | —     |
| FHVE       | FHVE15 Control Register for FACI0       | FHVE15FP0             | <FHVE_base> + 0004 <sub>H</sub>                        | 32                                     | PBG20#2           | —     |
| FHVE       | FHVE3 Control Register for FACI2        | FHVE3FP2              | <FHVE_base> + 0020 <sub>H</sub>                        | 32                                     | PBG20#2           | —     |
| FHVE       | FHVE15 Control Register for FACI2       | FHVE15FP2             | <FHVE_base> + 0024 <sub>H</sub>                        | 32                                     | PBG20#2           | —     |
| FACI0      | FACI0 command-issuing area              | FCMDAREA_0            | See Table 4.1                                          | 32, 16 <sup>*1</sup> , 8 <sup>*1</sup> | PBG62#1           | —     |
| FACI2      | FACI2 command-issuing area              | FCMDAREA_2            | See Table 4.1                                          | 32, 16 <sup>*1</sup> , 8 <sup>*1</sup> | PBG62#3           | —     |

Note 1. By 16bit access, only an address of lower 2 bytes is accessible.  
By 8bit access, only an address of lower 1 byte is accessible.

Table 4.4 Register Reset Conditions

| Unit Name | Reset Category |                |                |                   |                |              |            |
|-----------|----------------|----------------|----------------|-------------------|----------------|--------------|------------|
|           | Power On Reset | System Reset 1 | System Reset 2 | Application Reset | DeepSTOP Reset | Module Reset | JTAG Reset |
| FACI0     | ✓              | ✓              | ✓              | —                 | ✓              | —            | —          |
| FACI2     | ✓              | ✓              | ✓              | —                 | ✓              | —            | —          |
| IDCTRL    | ✓              | ✓              | ✓              | —                 | ✓              | —            | —          |
| FLMD      | ✓              | ✓              | ✓              | —                 | ✓              | —            | —          |
| FHVE      | ✓              | ✓              | ✓              | ✓                 | ✓              | —            | —          |

### 4.3 FLMDCNT —FLMD Control Register

FLMDCNT is a readable / writable register for software protection of code flash P/E mode entry. Set SFWE bit of FLMDCNT register in the programmable and erasable state (0000 0001<sub>H</sub>) before shifting the flash sequencer to code flash P/E mode. If SFWE bit of FLMDCNT register is not set, Flash sequencer cannot shift to code flash P/E mode.

**Access:** This register can be read/written in 32-bit units.

**Address:** <FLMD\_base> + 0000<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

|                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit               | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|                   | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W               | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

|                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| Bit               | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
|                   | —  | —  | —  | —  | —  | —  | — | — | — | — | — | — | — | — | — | SFWE |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    |
| R/W               | R  | R  | R  | R  | R  | R  | R | R | R | R | R | R | R | R | R | R/W  |

**Table 4.5 FLMDCNT Register Contents**

| Bit Position | Bit Name | Function                                                                                                          |
|--------------|----------|-------------------------------------------------------------------------------------------------------------------|
| 31 to 1      | Reserved | These bits are read as 0. The write value should be 0.                                                            |
| 0            | SFWE     | 0: Flash sequencer cannot shift to code flash P/E mode.<br>1: Flash sequencer can shift to code flash P/E mode*1. |

Note 1. Setting this bit is one of necessary conditions to enable shifting to code flash P/E mode. It is also necessary to unlock security protection controlled by ICUMHB if ICUMHB function is active. For details, see the *RH850/U2B Group Security User's Manual: Hardware*.

## 4.4 FLMDFPLOCK —FLMD FACI LOCK Register

FLMDFPLOCK is a readable / writable register for software protection of FACI0 access from ICUMHB.

Set FP0LOCK bit of FLMDFPLOCK register to prohibit ICUMHB to access FACI0.

Response error will occur if ICUMHB accesses to the FACI which is protected by this register.

**Access:** This register can be read/written in 32-bit units.

**Address:** <FLMD\_base> + 000C<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

|                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit               | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|                   | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W               | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

|                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |         |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---------|
| Bit               | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
|                   | —  | —  | —  | —  | —  | —  | — | — | — | — | — | — | — | — | — | FP0LOCK |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |
| R/W               | R  | R  | R  | R  | R  | R  | R | R | R | R | R | R | R | R | R | R/W     |

**Table 4.6 FLMDFPLOCK Register Contents**

| Bit Position | Bit Name | Function                                                                                      |
|--------------|----------|-----------------------------------------------------------------------------------------------|
| 31 to 1      | Reserved | These bits are read as 0. The write value should be 0.                                        |
| 0            | FP0LOCK  | 0: Access from ICUMHB to FACI0 is available.<br>1: Access from ICUMHB to FACI0 is prohibited. |

## 4.5 FHVE3FPn — FHVE3 Control Register for FACIn (n = 0, 2)

FHVE3FPn is a readable / writable register for software protection of flash memory against programming and erasure. Set both FHVE15FPn and FHVE3FPn in the programmable and erasable state (0000 0001<sub>H</sub>) to program or erase flash memory.

If these registers are set to 0000 0000<sub>H</sub> that does not allow programming and erasure of the flash memory, the following commands cannot be executed. Issuing any of the following commands leads to setting of the FHVEERR bit in the FSTATR\_n register to 1.

Programming / Multi Programming / DMA Programming  
 Block Erasure / Area Erasure / Blank Checking  
 Programming/Erase Suspension / Programming/Erase Resumption  
 Property Erasure / Property Programming  
 Switch Erasure / Switch Programming  
 TAG Erasure / TAG Update

**Access:** This register can be read/written in 32-bit units.

**Address:** <FHVE\_base> + 0000<sub>H</sub> + 0010<sub>H</sub> × n

**Value after reset:** 0000 0000<sub>H</sub>

|                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit               | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|                   | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W               | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

|                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |              |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|--------------|
| Bit               | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0            |
|                   | —  | —  | —  | —  | —  | —  | — | — | — | — | — | — | — | — | — | FHVE3<br>CNT |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0            |
| R/W               | R  | R  | R  | R  | R  | R  | R | R | R | R | R | R | R | R | R | R/W          |

**Table 4.7 FHVE3FPn Register Contents**

| Bit Position | Bit Name | Function                                                                                                                  |
|--------------|----------|---------------------------------------------------------------------------------------------------------------------------|
| 31 to 1      | Reserved | These bits are read as 0. The write value should be 0.                                                                    |
| 0            | FHVE3CNT | 0: Programming/erasure/DataFlashBlankChecking are disabled.<br>1: Programming/erasure/DataFlashBlankChecking are enabled. |

## 4.6 FHVE15FPn — FHVE15 Control Register for FACIn (n = 0, 2)

FHVE15FPn is a readable / writable register for software protection of flash memory against programming and erasure. Set both the FHVE15FPn and FHVE3FPn registers in the programmable and erasable state (0000 0001<sub>H</sub>) to program or erase flash memory.

If these registers are set to 0000 0000<sub>H</sub> that does not allow programming and erasure of the flash memory, the following commands cannot be executed.

Programming / Multi Programming / DMA Programming  
 Block Erasure / Area Erasure / Blank Checking  
 Programming/Erasure Suspension / Programming/Erasure Resumption  
 Property Erasure / Property Programming  
 Switch Erasure / Switch Programming  
 TAG Erasure / TAG Update

**Access:** This register can be read/written in 32-bit units.

**Address:** <FHVE\_base> + 0004<sub>H</sub> + 0010<sub>H</sub> × n

**Value after reset:** 0000 0000<sub>H</sub>

|                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit               | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|                   | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W               | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

|                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |               |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---------------|
| Bit               | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0             |
|                   | —  | —  | —  | —  | —  | —  | — | — | — | — | — | — | — | — | — | FHVE<br>15CNT |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0             |
| R/W               | R  | R  | R  | R  | R  | R  | R | R | R | R | R | R | R | R | R | R/W           |

**Table 4.8 FHVE15FPn Register Contents**

| Bit Position | Bit Name  | Function                                                                                                                  |
|--------------|-----------|---------------------------------------------------------------------------------------------------------------------------|
| 31 to 1      | Reserved  | These bits are read as 0. The write value should be 0.                                                                    |
| 0            | FHVE15CNT | 0: Programming/erasure/DataFlashBlankChecking are disabled.<br>1: Programming/erasure/DataFlashBlankChecking are enabled. |

## 4.7 FPMON\_n — Flash Pin Monitor Register (n = 0, 2)

FPMON\_n register indicates code flash P/E protection status.

**Access:** This register can be read in 8-bit units.

**Address:** <FACIn\_base> + 0000<sub>H</sub> (n = 0, 2)

**Value after reset:** 00<sub>H</sub>

| Bit               | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------|---|---|---|---|---|---|---|
|                   | SFWE | — | — | — | — | — | — | — |
| Value after reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W               | R    | R | R | R | R | R | R | R |

**Table 4.9 FPMON\_n Register Contents**

| Bit Position | Bit Name | Function                                                                                                                                                                                      |
|--------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7            | SFWE     | Flash Write Enable<br>Monitors the status of SFWE bit in FLMDCNT register.<br>0: Flash sequencer cannot shift to code flash P/E mode.<br>1: Flash sequencer can shift to code flash P/E mode. |
| 6 to 0       | Reserved | These bits are always read as "0".                                                                                                                                                            |



## 4.8 FASTAT\_n — Flash Access Status Register (n = 0, 2)

FASTAT\_n register indicates access error status for code/data flash. If either of CFAE/CMDLK/DFAE bits in FASTAT\_n is set to 1, flash sequencer enters the command lock state (see **Section 8.2, Error Protection**). To cancel the command lock state, issue a Status Clearing or Forced Stop Command to each FACI.

**Access:** This register can be read / written in 8-bit units.

**Address:** <FACIn\_base> + 0010<sub>H</sub> (n = 0, 2)

**Value after reset:** 00<sub>H</sub>

| Bit               | 7                 | 6 | 5 | 4     | 3                 | 2 | 1 | 0     |
|-------------------|-------------------|---|---|-------|-------------------|---|---|-------|
|                   | CFAE              | — | — | CMDLK | DFAE              | — | — | ECRCT |
| Value after reset | 0                 | 0 | 0 | 0     | 0                 | 0 | 0 | 0     |
| R/W               | R/W <sup>*1</sup> | R | R | R     | R/W <sup>*1</sup> | R | R | R     |

Note 1. Only 0 can be written to clear flag after 1 is read.

**Table 4.10 FASTAT\_n Register Contents (1/2)**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7            | CFAE     | Code Flash Access Error<br>Indicates whether or not code flash access error has been generated. If this bit becomes 1, ILGLERR bit in FSTATR_n is set to 1 and flash sequencer enters the command lock state.<br>0: No code flash access error has occurred.<br>1: Code flash access error has occurred.<br>[Setting Condition]<br>Flash sequencer command has been issued to wrong address <sup>*1</sup> in code flash programming mode.<br>[Clearing Condition]<br>(1) 0 is written after reading 1 from this bit.<br>(2) The flash sequencer starts the Status Clearing or Forced Stop Command processing. |
| 6, 5         | Reserved | These bits are always read as “0”. Write value should always be “0”.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 4            | CMDLK    | Command Lock<br>Indicates whether flash sequencer is in the command lock state.<br>0: Flash sequencer is not in the command lock state.<br>1: Flash sequencer is in the command lock state.<br>[Setting Condition]<br>FACI detects error and enters the command lock state.<br>[Clearing Condition]<br>The flash sequencer starts the Status Clearing or Forced Stop Command processing                                                                                                                                                                                                                       |

Table 4.10 FASTAT\_n Register Contents (2/2)

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3            | DFAE     | <p>Data Flash Access Error</p> <p>Indicates whether or not data flash access error has been generated. If this bit becomes 1, ILGLERR bit in FSTATR_n is set to 1 and flash sequencer enters the command lock state.</p> <p>0: No data flash access error has occurred.</p> <p>1: Data flash access error has occurred.</p> <p>[Setting Conditions]</p> <p>Flash sequencer command has been issued to wrong address*1 in data flash programming mode.</p> <p>[Clearing Condition]</p> <p>(1) 0 is written after reading 1 from this bit.</p> <p>(2) The flash sequencer starts the Status Clearing or Forced Stop Command processing.</p> |
| 2, 1         | Reserved | These bits are always read as "0". Write value should always be "0".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 0            | ECRCT    | <p>Error Correction</p> <p>Indicates that a 1-bit error has been corrected when the flash sequencer reads the flash memory (Configuration Setting Area, Block Protection Area, Security Setting Area, Erase Counter, Switch Area, TAG Area or the P/E parameter table)</p> <p>0: 1-bit error has not been corrected.</p> <p>1: 1-bit error has been corrected.</p> <p>[Clearing conditions]</p> <p>The flash sequencer starts the Status Clearing or Forced Stop Command processing</p>                                                                                                                                                   |

Note 1. FSADDR\_n register is not used in Status Clearing, Forced Stop, Programming/Erase Suspension and Programming/Erase Resumption. FSADDR\_n value does not influence on these command operations, and CFAE or DFAE is not set for these commands even if FSADDR\_n is wrong address in code flash programming mode or data flash programming mode.

Capacities of the code flash memory and data flash memory vary from product to product. See the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

## 4.9 FAEINT\_n — Flash Access Error Interrupt Enable Register (n = 0, 2)

FAEINT\_n register enables or disables output of flash access error (FLERR) interrupt.

See the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory* for connection destination of interrupt.

**Access:** This register can be read/written in 8-bit units.

**Address:** <FACIn\_base> + 0014<sub>H</sub> (n = 0, 2)

**Value after reset:** 99<sub>H</sub>

| Bit               | 7      | 6 | 5 | 4       | 3      | 2 | 1 | 0       |
|-------------------|--------|---|---|---------|--------|---|---|---------|
|                   | CFAEIE | — | — | CMDLKIE | DFAEIE | — | — | ECRCTIE |
| Value after reset | 1      | 0 | 0 | 1       | 1      | 0 | 0 | 1       |
| R/W               | R/W    | R | R | R/W     | R/W    | R | R | R/W     |

**Table 4.11 FAEINT\_n Register Contents**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7            | CFAEIE   | Code Flash Access Error Interrupt Enable<br>Enables or disables the FLERR interrupt request when code flash access error occurs and CFAE bit in FASTAT_n becomes 1.<br>0: Does not generate the FLERR interrupt request when FASTAT_n.CFAE = 1.<br>1: Generates the FLERR interrupt request when FASTAT_n.CFAE = 1.                                                                                                                                                                                 |
| 6, 5         | Reserved | These bits are always read as "0". Write value should always be "0".                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 4            | CMDLKIE  | Command Lock Interrupt Enable<br>Enables or disables the FLERR interrupt request when flash sequencer enters the command lock state and CMDLK bit in FASTAT_n becomes 1.<br>0: Does not generate the FLERR interrupt request when FASTAT_n.CMDLK = 1.<br>1: Generates the FLERR interrupt request when FASTAT_n.CMDLK = 1.                                                                                                                                                                          |
| 3            | DFAEIE   | Data Flash Access Error Interrupt Enable<br>Enables or disables the FLERR interrupt request when data flash access error occurs and DFAE bit in FASTAT_n becomes 1.<br>0: Does not generate the FLERR interrupt request when FASTAT_n.DFAE = 1.<br>1: Generates the FLERR interrupt request when FASTAT_n.DFAE = 1.                                                                                                                                                                                 |
| 2, 1         | Reserved | These bits are always read as "0". Write value should always be "0".                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 0            | ECRCTIE  | Error Correction Interrupt Enable<br>Enables or disables the FLERR interrupt request when a 1-bit error has been corrected and the ECRCT bit in FASTAT_n has been set to 1 on the flash memory read (Configuration Setting Area, Block Protection Area, Security Setting Area, Erase Counter, Switch Area, TAG Area, or the P/E parameter table).<br>0: Does not generate the FLERR interrupt request when FASTAT_n.ECRCT = 1.<br>1: Generates the FLERR interrupt request when FASTAT_n.ECRCT = 1. |

## 4.10 FAREASELC — Code Flash Memory Area Select Register

This register is used to specify the User Boot Area as the target area for FACL command processing.  
This register is reset when SUINIT bit in the FSUINITR\_0 register is set to 1.

This register is also reset as **Table 4.4**.

**Access:** This register can be read/written in 16-bit units.

**Address:** <FACIn\_base> + 0020<sub>H</sub> (n = 0)

**Value after reset:** 0000<sub>H</sub>

| Bit               | 15              | 14              | 13              | 12              | 11              | 10              | 9               | 8               | 7 | 6 | 5 | 4 | 3 | 2 | 1                       | 0 |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|---|---|---|---|---|---|-------------------------|---|
|                   | KEY             |                 |                 |                 |                 |                 |                 |                 | — | — | — | — | — | — | CFAS                    | — |
| Value after reset | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0 | 0 | 0 | 0 | 0 | 0 | 0                       | 0 |
| R/W               | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | R | R | R | R | R | R | R/W <sup>*2,*3,*4</sup> | R |

Note 1. Written data is not stored in this bit. The value read is always 00<sub>H</sub>.

Note 2. Writing to this bit is enabled only when 3B<sub>H</sub> is written to the KEY bits.

Note 3. This bit can be written in serial programming mode, and break status of On-chip debug.

Note 4. Writing to this bit in periods from the first write access of FACL command to FRDY bit in FSTATR\_0 register returns to "1" from "0" is ignored. (Except for FRDY = "1" in "Command Lock" state)

**Table 4.12 FAREASELC Register Contents**

| Bit Position | Bit Name | Function                                                                                         |
|--------------|----------|--------------------------------------------------------------------------------------------------|
| 15 to 8      | KEY      | Key Code<br>These bits enable or disable CFAS bit modification                                   |
| 7 to 2       | Reserved | These bits are always read as "0". Write value should always be "0".                             |
| 1            | CFAS     | User Boot Area Selection<br>0: User Boot Area is not selected.<br>1: User Boot Area is selected. |
| 0            | Reserved | These bits are always read as "0". Write value should always be "0".                             |

### CAUTION

- In the case of CFAS = 0: When FACL command has been issued to User Boot Area, it becomes the Code flash access error.
- In the case of CFAS = 1: When FACL command has been issued to User Area, it becomes the Code flash access error.

## 4.11 FSADDR\_n — Flash Command Start Address Register (n = 0, 2)

FSADDR\_n register specifies the start address of the target area for command processing when an FACL command (Programming, Multi Programming, DMA Programming, Block Erasure, Area Erasure, Blank Checking, Property Programming, Property Erasure, Switch Programming, Switch Erasure, TAG Update, TAG Erasure) is issued.

This register is reset when SUINIT bit in FSUINITR\_n register is set to 1.

This register is also reset as **Table 4.4**.

**Access:** This register can be read / written in 32-bit units.

**Address:** <FACIn\_base> + 0030<sub>H</sub> (n = 0, 2)

**Value after reset:** 0000 0000<sub>H</sub>

| Bit               | 31                | 30                | 29                | 28                | 27                | 26                | 25                | 24                | 23                | 22                | 21                | 20                | 19                | 18                | 17                | 16                |
|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
|                   | FSADDR[31:16]     |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |
| Value after reset | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 |
| R/W               | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> |
| Bit               | 15                | 14                | 13                | 12                | 11                | 10                | 9                 | 8                 | 7                 | 6                 | 5                 | 4                 | 3                 | 2                 | 1                 | 0                 |
|                   | FSADDR[15:0]      |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |
| Value after reset | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 | 0                 |
| R/W               | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R/W <sup>*1</sup> | R                 | R                 |

Note 1. Writing to this bit in periods from the first write access of FACL command to FRDY bit in FSTATR\_n register returns to "1" from "0" is ignored. (Except for FRDY = "1" in "Command Lock" state)

Table 4.13 FSADDR\_n Register Contents

| Bit Position                                                  | Bit Name               | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
|---------------------------------------------------------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|------------------|--------------------------|-----------|--------------------------|---------|-----------------------------------|---------|----------------------------------------------------|---------|-----------------------------------------------------|----------|-----------------------------------------------------|----------|-----------------------------------------------------|----------|------------------------------------------------------|-----------|-------------------------------------------------------------|---------|--------------------------------------------------------------|----------|--------------------------------------------------------------|----------|--------------------------------------------------------------|----------|---------------------------------------------------------------|-----------|------------------|---------|----------------------------|------------------------|----------------------------|----------|-------------------------------------|----------|---------------------------|----------|------------------------------------|----------|-----------------|---------|----------------------|----------|------------------|----------|-------------------|----------|----------------|----------|------------|---------|
| 31 to 0                                                       | FSADDR[31:0]           | Start Address of FACL Command Processing<br>These bits specify the start address of the FACL command processing.<br>Bits 31 to 28 are ignored in the FACL command processing for the code flash memory.<br>Bits 31 to 21 are ignored in the FACL command processing for the data flash memory.<br>Lower address bits for smaller address than boundary below are also ignored*1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
|                                                               |                        | <table><tr><th>Command</th><th>Address Boundary</th></tr><tr><td>Programming (User Area):</td><td>512 bytes</td></tr><tr><td>Programming (Data Area):</td><td>4 bytes</td></tr><tr><td>Programming (Extended Data Area):</td><td>4 bytes</td></tr><tr><td>Multi Programming (Data Area): 8-byte programming:</td><td>8 bytes</td></tr><tr><td>Multi Programming (Data Area): 16-byte programming:</td><td>16 bytes</td></tr><tr><td>Multi Programming (Data Area): 32-byte programming:</td><td>32 bytes</td></tr><tr><td>Multi Programming (Data Area): 64-byte programming:</td><td>64 bytes</td></tr><tr><td>Multi Programming (Data Area): 128-byte programming:</td><td>128 bytes</td></tr><tr><td>Multi Programming (Extended Data Area): 8-byte programming:</td><td>8 bytes</td></tr><tr><td>Multi Programming (Extended Data Area): 16-byte programming:</td><td>16 bytes</td></tr><tr><td>Multi Programming (Extended Data Area): 32-byte programming:</td><td>32 bytes</td></tr><tr><td>Multi Programming (Extended Data Area): 64-byte programming:</td><td>64 bytes</td></tr><tr><td>Multi Programming (Extended Data Area): 128-byte programming:</td><td>128 bytes</td></tr><tr><td>DMA Programming:</td><td>4 bytes</td></tr><tr><td>Block Erasure (User Area):</td><td>16 Kbytes or 64 Kbytes</td></tr><tr><td>Block Erasure (Data Area):</td><td>4 Kbytes</td></tr><tr><td>Block Erasure (Extended Data Area):</td><td>2 Kbytes</td></tr><tr><td>Area Erasure (Data Area):</td><td>4 Kbytes</td></tr><tr><td>Area Erasure (Extended Data Area):</td><td>2 Kbytes</td></tr><tr><td>Blank Checking:</td><td>4 bytes</td></tr><tr><td>Property Programming</td><td>32 bytes</td></tr><tr><td>Property Erasure</td><td>2 Kbytes</td></tr><tr><td>SwitchProgramming</td><td>32 bytes</td></tr><tr><td>Switch Erasure</td><td>2 Kbytes</td></tr><tr><td>TAG Update</td><td>4 bytes</td></tr><tr><td>TAG Erasure</td><td>2 Kbytes</td></tr></table> | Command | Address Boundary | Programming (User Area): | 512 bytes | Programming (Data Area): | 4 bytes | Programming (Extended Data Area): | 4 bytes | Multi Programming (Data Area): 8-byte programming: | 8 bytes | Multi Programming (Data Area): 16-byte programming: | 16 bytes | Multi Programming (Data Area): 32-byte programming: | 32 bytes | Multi Programming (Data Area): 64-byte programming: | 64 bytes | Multi Programming (Data Area): 128-byte programming: | 128 bytes | Multi Programming (Extended Data Area): 8-byte programming: | 8 bytes | Multi Programming (Extended Data Area): 16-byte programming: | 16 bytes | Multi Programming (Extended Data Area): 32-byte programming: | 32 bytes | Multi Programming (Extended Data Area): 64-byte programming: | 64 bytes | Multi Programming (Extended Data Area): 128-byte programming: | 128 bytes | DMA Programming: | 4 bytes | Block Erasure (User Area): | 16 Kbytes or 64 Kbytes | Block Erasure (Data Area): | 4 Kbytes | Block Erasure (Extended Data Area): | 2 Kbytes | Area Erasure (Data Area): | 4 Kbytes | Area Erasure (Extended Data Area): | 2 Kbytes | Blank Checking: | 4 bytes | Property Programming | 32 bytes | Property Erasure | 2 Kbytes | SwitchProgramming | 32 bytes | Switch Erasure | 2 Kbytes | TAG Update | 4 bytes |
| Command                                                       | Address Boundary       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Programming (User Area):                                      | 512 bytes              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Programming (Data Area):                                      | 4 bytes                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Programming (Extended Data Area):                             | 4 bytes                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Multi Programming (Data Area): 8-byte programming:            | 8 bytes                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Multi Programming (Data Area): 16-byte programming:           | 16 bytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Multi Programming (Data Area): 32-byte programming:           | 32 bytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Multi Programming (Data Area): 64-byte programming:           | 64 bytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Multi Programming (Data Area): 128-byte programming:          | 128 bytes              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Multi Programming (Extended Data Area): 8-byte programming:   | 8 bytes                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Multi Programming (Extended Data Area): 16-byte programming:  | 16 bytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Multi Programming (Extended Data Area): 32-byte programming:  | 32 bytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Multi Programming (Extended Data Area): 64-byte programming:  | 64 bytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Multi Programming (Extended Data Area): 128-byte programming: | 128 bytes              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| DMA Programming:                                              | 4 bytes                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Block Erasure (User Area):                                    | 16 Kbytes or 64 Kbytes |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Block Erasure (Data Area):                                    | 4 Kbytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Block Erasure (Extended Data Area):                           | 2 Kbytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Area Erasure (Data Area):                                     | 4 Kbytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Area Erasure (Extended Data Area):                            | 2 Kbytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Blank Checking:                                               | 4 bytes                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Property Programming                                          | 32 bytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Property Erasure                                              | 2 Kbytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| SwitchProgramming                                             | 32 bytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| Switch Erasure                                                | 2 Kbytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| TAG Update                                                    | 4 bytes                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |
| TAG Erasure                                                   | 2 Kbytes               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                  |                          |           |                          |         |                                   |         |                                                    |         |                                                     |          |                                                     |          |                                                     |          |                                                      |           |                                                             |         |                                                              |          |                                                              |          |                                                              |          |                                                               |           |                  |         |                            |                        |                            |          |                                     |          |                           |          |                                    |          |                 |         |                      |          |                  |          |                   |          |                |          |            |         |

Note 1. Bits 20 to 0 are used in the inconsistency check between FSADDR and FEADDR.

## 4.12 FEADDR\_n — Flash Command End Address Register (n = 0, 2)

FEADDR\_n register specifies the end address in the target area in Area Erasure or Blank Checking Command processing.

When Area Erasure Command is issued, address specified in FSADDR\_n should be equal to or smaller than address in FEADDR\_n. If setting of FSADDR\_n and FEADDR\_n are inconsistent, FACL detects error and flash sequencer enters command lock state.

When blank check addressing mode is set to incremental mode (i.e. FBCCNT\_n.BCDIR = 0), address specified in FSADDR\_n should be equal to or smaller than address in FEADDR\_n. Conversely, address in FSADDR\_n should be equal to or larger than address in FEADDR\_n when blank check addressing mode is set to decremental mode (i.e. FBCCNT\_n.BCDIR = 1). If setting of BCDIR\_n, FSADDR\_n, and FEADDR\_n are inconsistent, FACL detects error and flash sequencer enters the command lock state. (See **Section 8.2, Error Protection.**)

In addition, target area should be within one Data Area\*<sup>1</sup>. If Area Erasure or Blank Checking Command is processed crossing over different Data Area boundary, FACL detects error and flash sequencer enters command lock state.

This register is reset when SUINIT bit in FSUINITR\_n register is set to 1.

This register is also reset as **Table 4.4**.

**Note 1.** For each Data Area size, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

**Access:** This register can be read/written in 32-bit units.

**Address:** <FACIn\_base> + 0034<sub>H</sub> (n = 0, 2)

**Value after reset:** 0000 0000<sub>H</sub>

| Bit               | 31            | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   |
|-------------------|---------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
|                   | FEADDR[31:16] |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| Value after reset | 0             | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| R/W               | R/W*          | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |
| Bit               | 15            | 14   | 13   | 12   | 11   | 10   | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|                   | FEADDR[15:0]  |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| Value after reset | 0             | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| R/W               | R/W*          | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R    | R    |

**Note 1.** Writing to this bit in periods from the first write access of FACL command to FRDY bit in FSTATR\_n register returns to 1 from 0 is ignored. (Except for FRDY = "1" in "Command Lock" state)

**Table 4.14 FEADDR\_n Register Contents**

| Bit Position | Bit Name     | Function                                                                                                                                                                                                                                                                |
|--------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31 to 0      | FEADDR[31:0] | End Address of FACL Command Target Area<br>Specifies end address of target area in the Area Erasure or Blank Checking Command.<br>Bits 31 to 21 are ignored in the FACL command processing. Bits 20 to 0 are used in the inconsistency check between FSADDR and FEADDR. |

### 4.13 FCVAPROT\_n — Code Flash Valid Area Protection Register (n = 0, 2)

This register is used to specify the Code Flash Valid Area Protection for FACL command processing.

This register is reset when SUINIT bit in FSUINITR\_n register is set to 1.

This register is also reset as **Table 4.4**.

**Access:** This register can be read/written in 16-bit units.

**Address:** <FACIn\_base> + 0040<sub>H</sub> (n = 0, 2)

**Value after reset:** 0001<sub>H</sub>

| Bit               | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0         |
|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|-----------|
|                   | KEY |     |     |     |     |     |     |     | — | — | — | — | — | — | — | CVAPROT   |
| Value after reset | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1         |
|                   | R/W | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | R | R | R | R | R | R | R | R/W*2, *3 |

Note 1. Written data is not stored in this bit. The value read is always 00<sub>H</sub>.

Note 2. Writing to this bit is enabled only when FE<sub>H</sub> is written to the KEY bits.

Note 3. Writing to this bit in periods from the first write access of FACL command to FRDY bit in FSTATR\_n register returns to "1" from "0" is ignored. (Except for FRDY = "1" in "Command Lock" state).

**Table 4.15 FCVAPROT\_n Register Contents**

| Bit Position | Bit Name  | Function                                                                                                                                                                                                                                                                |
|--------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15 to 8      | KEY       | Key Code<br>These bits enable or disable CVAPROT bit modification                                                                                                                                                                                                       |
| 7 to 1       | Reserved  | These bits are always read as "0". Write value should always be "0".                                                                                                                                                                                                    |
| 0            | CVAPROT*1 | Code Flash Valid Area Protection<br>Valid area of code flash is protected from FACL command (Programming, Block Erasure, Programming/Erasure Resumption) in double map mode<br>0: Valid Area of User Area is not protected.<br>1: Valid Area of User Area is protected. |

Note 1. This bit is no function when User Area does not exist.



## 4.14 FSTATR\_n — Flash Status Register (n = 0, 2)

FSTATR\_n register indicates flash sequencer status.

**Access:** This register can be read in 32-bit units.

**Address:** <FACIn\_base> + 0080<sub>H</sub> (n = 0, 2)

**Value after reset:** 0000 8000<sub>H</sub>

| Bit               | 31 | 30 | 29            | 28            | 27            | 26            | 25            | 24            | 23            | 22           | 21         | 20    | 19 | 18         | 17            | 16            |
|-------------------|----|----|---------------|---------------|---------------|---------------|---------------|---------------|---------------|--------------|------------|-------|----|------------|---------------|---------------|
|                   | —  | —  | ERCDT<br>CT*2 | ERCCR<br>CT*2 | SWTDT<br>CT*1 | SWTCR<br>CT*1 | SECDT<br>CT*1 | SECCR<br>CT*1 | ILGCO<br>MERR | FESET<br>ERR | SECER<br>R | OTERR | —  | EBFUL<br>L | BPLDT<br>CT*2 | BPLCR<br>CT*2 |
| Value after reset | 0  | 0  | 0             | 0             | 0             | 0             | 0             | 0             | 0             | 0            | 0          | 0     | 0  | 0          | 0             | 0             |
| R/W               | R  | R  | R             | R             | R             | R             | R             | R             | R             | R            | R          | R     | R  | R          | R             | R             |

| Bit               | 15   | 14          | 13         | 12         | 11         | 10         | 9          | 8          | 7 | 6           | 5             | 4             | 3           | 2           | 1 | 0 |
|-------------------|------|-------------|------------|------------|------------|------------|------------|------------|---|-------------|---------------|---------------|-------------|-------------|---|---|
|                   | FRDY | ILGLER<br>R | ERSER<br>R | PRGER<br>R | SUSRD<br>Y | DBFUL<br>L | ERSSP<br>D | PRGSP<br>D | — | FHVE<br>ERR | CFGDT<br>CT*1 | CFGCR<br>CT*1 | TBLDT<br>CT | TBLCR<br>CT | — | — |
| Value after reset | 1    | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0 | 0           | 0             | 0             | 0           | 0           | 0 | 0 |
| R/W               | R    | R           | R          | R          | R          | R          | R          | R          | R | R           | R             | R             | R           | R           | R | R |

Note 1. [For FPSYS other than FPSYS0] It is reserved bit and read data is always "0".

Note 2. [For FPSYS which does not have Code Flash Memory] It is reserved bit and read data is always "0".

**Table 4.16 FSTATR\_n Register Contents (1/6)**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31 to 30     | Reserved | These bits are always read as "0".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 29           | ERCDTCT  | Erase counter ECC 2-Bit Error Detection Monitoring Bit<br>Indicates that 2-bit error has been detected when FACL is reading Erase counter. FACL reads Erase counter in Block Erasure (User Area), Erasure Resumption (User Area) Command processing. When this bit is "1", flash sequencer is in "Command Lock" state.<br>0: No error has been detected.<br>1: An error has been detected.<br>[Clearing condition]<br>Status Clearing or Forced Stop Command processing is started.                                            |
| 28           | ERCCRCT  | Erase counter 1-Bit Error Correction Monitoring Bit<br>Indicates that 1-bit error has been corrected when FACL is reading Erase counter. FACL reads Erase counter in Block Erasure (User Area), Erasure Resumption (User Area) Command processing. When this bit is "1", flash sequencer continues command processing, and does not enter "Command Lock" state.<br>0: No error has been corrected.<br>1: An error has been corrected.<br>[Clearing condition]<br>Status Clearing or Forced Stop Command processing is started. |
| 27           | SWTDTCT  | Switch Area and TAG Area ECC 2-Bit Error Detection Monitoring Bit<br>Indicates that 2-bit error has been detected when FACL is reading Switch Area and TAG Area. FACL reads Switch Area and TAG Area in Switch Erasure, TAG Update, and TAG Erasure Command processing. When this bit is "1", flash sequencer is in "Command Lock" state.<br>0: No error has been detected.<br>1: An error has been detected.<br>[Clearing condition]<br>Status Clearing or Forced Stop Command processing is started.                         |

Table 4.16 FSTATR\_n Register Contents (2/6)

| Bit Position | Bit Name      | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26           | SWTCRCT       | Switch Area and TAG Area ECC 1-Bit Error Correction Monitoring Bit<br>Indicates that 1-bit error has been detected when FACI is reading Switch Area and TAG Area. FACI reads Switch Area and TAG Area in Switch Erasure, TAG Update, and TAG Erasure Command processing. When this bit is "1", flash sequencer continues command processing, and does not enter "Command Lock" state.<br>0: No error has been corrected.<br>1: An error has been corrected.<br>[Clearing condition]<br>Status Clearing or Forced Stop Command processing is started.               |
| 25           | SECDTCT       | Security Setting Area ECC 2-Bit Error Detection Monitoring Bit<br>Indicates that 2-bit error has been detected when flash sequencer is reading Security Setting Area. Flash sequencer reads Security Setting Area in Property Programming, TAG Update, TAG Erasure Command processing. When this bit is "1", flash sequencer is in command lock state.<br>0: No error has been detected.<br>1: An error has been detected.<br>[Clearing condition]<br>Status Clearing or Forced Stop Command processing is started.                                                |
| 24           | SECCRCT       | Security Setting Area ECC 1-Bit Error Correction Monitoring Bit<br>Indicates that 1-bit error has been corrected when flash sequencer is reading Security Setting Area. Flash sequencer reads Security Setting Area in Property Programming, TAG Update, TAG Erasure Command processing. When this bit is "1", flash sequencer continues command processing, and does not enter command lock state.<br>0: No error has been corrected.<br>1: An error has been corrected.<br>[Clearing condition]<br>Status Clearing or Forced Stop Command processing is started. |
| 23           | ILGCOMER<br>R | Illegal Command Error<br>See <b>Table 8.1</b> . When this bit is "1", flash sequencer enters command lock state.<br>[Setting conditions]<br>An error has been detected.<br>[Clearing condition]<br>Status Clearing or Forced Stop Command processing is started.                                                                                                                                                                                                                                                                                                   |
| 22           | FESETERR      | FENTRY Setting Error<br>See <b>Table 8.1</b> . When this bit is "1", flash sequencer enters command lock state.<br>[Setting conditions]<br>An error has been detected.<br>[Clearing condition]<br>FENTRYR register is set to 0000 <sub>H</sub> .                                                                                                                                                                                                                                                                                                                   |
| 21           | SECERR        | Security Error<br>See <b>Table 8.1</b> . When this bit is "1", flash sequencer enters command lock state.<br>[Setting conditions]<br>An error has been detected.<br>[Clearing condition]<br>Status Clearing or Forced Stop Command processing is started.                                                                                                                                                                                                                                                                                                          |
| 20           | OTERR         | Other Error<br>See <b>Table 8.1</b> . When this bit is "1", flash sequencer enters command lock state.<br>[Setting conditions]<br>An error has been detected.<br>[Clearing condition]<br>Status Clearing or Forced Stop Command processing is started.                                                                                                                                                                                                                                                                                                             |
| 19           | Reserved      | These bits are always read as "0".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

Table 4.16 FSTATR\_n Register Contents (3/6)

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 18           | EBFULL   | <p>FDMYECC_n Buffer Full</p> <p>Indicates the FDMYECC_n buffer status when issuing the Programming Command. Flash sequencer incorporates a buffer for FDMYECC_n register (ECC buffer).</p> <p>It is possible to use FDMYECC_n register as the ECC buffer by setting the ECCDISE bit in the FECCTMD_n register to 1. When FDMYECC_n is written to while EBFULL bit is 1, flash sequencer inserts a wait in Peripheral Bus.</p> <p>0: The ECC buffer is not full.<br/>1: The ECC buffer is full.</p> <p>[Setting condition]<br/>The ECC buffer becomes full while issuing the Programming Command.</p> <p>[Clearing condition]<br/>The ECC buffer becomes not full.</p>                                                                                                                                                                                                                                                                                                                                                                             |
| 17           | BPLDTCT  | <p>Block Protection Area ECC 2-Bit Error Detection Monitoring Bit</p> <p>Indicates that 2-bit error has been detected when flash sequencer is reading Block Protection Area. Flash sequencer reads Block Protection Area in Programming, Block Erasure for the code flash memory or Programming, Multi Programming, Block Erasure, Area Erasure for the Extended Data Area or Property Programming, TAG Update, TAG Erasure Command processing.</p> <p>When this bit is "1", flash sequencer is in command lock state.</p> <p>0: No error has been detected.<br/>1: An error has been detected.</p> <p>[Clearing condition]<br/>Status Clearing or Forced Stop Command processing is started.</p>                                                                                                                                                                                                                                                                                                                                                 |
| 16           | BPLCRCT  | <p>Block Protection Area ECC 1-Bit Error Correction Monitoring Bit</p> <p>Indicates that 1-bit error has been corrected when flash sequencer is reading Block Protection Area. Flash sequencer reads Block Protection Area in Programming, Block Erasure for the code flash memory or Programming, Multi Programming, Block Erasure, Area Erasure for the Extended Data Area or Property Programming, TAG Update, TAG Erasure Command processing.</p> <p>When this bit is "1", flash sequencer continues command processing, and does not enter command lock state.</p> <p>0: No error has been corrected.<br/>1: An error has been corrected.</p> <p>[Clearing condition]<br/>Status Clearing or Forced Stop Command processing is started.</p>                                                                                                                                                                                                                                                                                                  |
| 15           | FRDY     | <p>Flash Ready</p> <p>Indicates the processing state in flash sequencer.</p> <p>0: Processing of the command (Programming, Multi Programming, DMA Programming, Block Erasure, Area Erasure, Programming/Erasure Suspension, Programming/Erasure Resumption, Forced Stop, Blank Checking, Property Programming, Property Erasure, Switch Programming, Switch Erasure, TAG Update, TAG Erasure) is in progress.<br/>1: None of the above is in progress.</p> <p>[Setting Conditions]<br/>Flash sequencer completes processing.<br/>Flash sequencer suspends processing by a Programming/Erasure Suspension Command.<br/>Flash sequencer terminates processing by a Forced Stop Command.</p> <p>[Clearing Conditions]<br/>When the flash sequencer accepts the FACL command<br/>For a Programming, Multi Programming, DMA Programming, Property Programming, or Switch Programming Command, after the first write access to the FACL command-issuing area.<br/>For other commands, after the last write access to the FACL command-issuing area.</p> |

Table 4.16 FSTATR\_n Register Contents (4/6)

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14           | ILGLERR  | <p>Illegal Error</p> <p>Indicates that flash sequencer has detected an illegal command or illegal flash memory access. When this bit is 1, flash sequencer is in the command lock state.</p> <p>0: Flash sequencer has not detected any illegal command or illegal flash memory access.</p> <p>1: Flash sequencer has detected an illegal command or illegal flash memory access</p> <p>[Setting conditions] (see <b>Section 8.2, Error Protection</b>)</p> <p>Flash sequencer has detected an illegal command.</p> <p>Flash sequencer has detected an illegal flash memory access.</p> <p>FENTRYR_n setting is illegal.</p> <p>[Clearing condition]</p> <p>Status Clearing or Forced Stop Command processing is started.</p>                                                                                                                                   |
| 13           | ERSERR   | <p>Erase Error</p> <p>Indicates result of code or data flash erasure by flash sequencer. When this bit is 1, flash sequencer is in the command lock state.</p> <p>0: Erasure processing has been completed successfully</p> <p>1: An error has occurred during erasure</p> <p>[Setting conditions]</p> <p>An error has occurred during erasure.</p> <p>[Clearing condition]</p> <p>Status Clearing or Forced Stop Command processing is started.</p>                                                                                                                                                                                                                                                                                                                                                                                                            |
| 12           | PRGERR   | <p>Programming Error</p> <p>Indicates the result of code or data flash programming by flash sequencer. When this bit is 1, flash sequencer is in the command lock state.</p> <p>0: Programming has been completed successfully</p> <p>1: An error has occurred during programming</p> <p>[Setting conditions]</p> <p>An error has occurred during programming.</p> <p>[Clearing condition]</p> <p>Status Clearing or Forced Stop Command processing is started.</p>                                                                                                                                                                                                                                                                                                                                                                                             |
| 11           | SUSRDY   | <p>Suspend Ready</p> <p>Indicates whether flash sequencer is ready to accept the Programming/Erase Suspension Command.</p> <p>0: Flash sequencer cannot accept a Programming/Erase Suspension Command.</p> <p>1: Flash sequencer can accept a Programming/Erase Suspension Command.</p> <p>[Setting condition]</p> <p>After initiating programming or erasure, Flash sequencer entered a state where it is ready to accept a Programming/Erase Suspension Command.</p> <p>[Clearing conditions]</p> <p>Flash sequencer has accepted a Programming/Erase Suspension or Forced Stop Command.</p> <p>(after the write access to the FACI command-issuing area is completed)</p> <p>Flash sequencer has entered the command lock state during programming or erasure. Programming/Multi Programming/Block Erasure/Area Erasure Command processing is completed.</p> |
| 10           | DBFULL   | <p>Data Buffer Full</p> <p>This bit is always read as "0".</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 9            | ERSSPD   | <p>Erase-Suspended Status</p> <p>Indicates that flash sequencer has entered the erasure command suspension process or erasure-suspended status.</p> <p>0: Flash sequencer is in status other than the below mentioned.</p> <p>1: Flash sequencer is in erasure suspension process or erasure-suspended status.</p> <p>[Setting condition]</p> <p>Flash sequencer has initiated a Programming/Erase Suspension Command during Block Erasure or Area Erasure Command processing.</p> <p>[Clearing conditions]</p> <p>Flash sequencer has accepted a Programming/Erase Resumption Command.</p> <p>(after the write access to the FACI command-issuing area is completed)</p> <p>Forced Stop Command processing is started.</p> <p>FHVEERR bit is "1".</p>                                                                                                          |

Table 4.16 FSTATR\_n Register Contents (5/6)

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8            | PRGSPD   | <p>Programming Suspension Status</p> <p>Indicates that flash sequencer has entered the Programming Command Suspension process or Programming Suspension status.</p> <p>0: Flash sequencer is in status other than the below mentioned.</p> <p>1: Flash sequencer is in Programming Suspension process or Programming Suspension status.</p> <p>[Setting condition]</p> <p>Flash sequencer has initiated a Programming/Erasure Suspension Command during Programming or Multi Programming Command processing.</p> <p>[Clearing condition]</p> <p>Flash sequencer has accepted a Programming/Erasure Resumption Command. (after the write access to the FACI command-issuing area is completed)</p> <p>Forced Stop Command processing is started.</p> <p>FHVEERR bit is "1".</p> |
| 7            | Reserved | These bits are always read as "0".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 6            | FHVEERR  | <p>FHVE Setting Error</p> <p>Indicates the violation of the flash memory programming/erasure protection in the FHVE3FPn register. When FHVEERR bit is 1, the flash sequencer is in the command lock state.</p> <p>0: No error has occurred.</p> <p>1: An error has occurred.</p> <p>[Clearing condition]</p> <p>Status Clearing or Forced Stop Command processing is started.</p>                                                                                                                                                                                                                                                                                                                                                                                              |
| 5            | CFGDTCT  | <p>Configuration Setting Area ECC 2-Bit Error Detection Monitoring Bit</p> <p>Indicates that 2-bit error has been detected when flash sequencer is reading Configuration Setting Area. Flash sequencer reads Configuration Setting Area in Property Programming, TAG Update, TAG Erasure Command processing. When this bit is "1", flash sequencer is in command lock state.</p> <p>0: No error has been detected.</p> <p>1: An error has been detected.</p> <p>[Clearing condition]</p> <p>Status Clearing or Forced Stop command processing is started.</p>                                                                                                                                                                                                                  |
| 4            | CFGCRCT  | <p>Configuration Setting Area ECC 1-Bit Error Correction Monitoring Bit</p> <p>Indicates that 1-bit error has been corrected when flash sequencer is reading Configuration Setting Area. Flash sequencer reads Configuration Setting Area in Property Programming, TAG Update, TAG Erasure. When this bit is "1", flash sequencer continues command processing, and does not enter command lock state.</p> <p>0: No error has been corrected.</p> <p>1: An error has been corrected.</p> <p>[Clearing condition]</p> <p>Status Clearing or Forced Stop Command processing is started.</p>                                                                                                                                                                                      |
| 3            | TBLDTCT  | <p>2-Bit Error Detection Monitor (P/E Parameter Table *<sup>1</sup>)</p> <p>Indicates that a 2-bit error has been detected on reading the P/E parameter table. Flash sequencer reads the P/E parameter table in Programming, Multi Programming, DMA Programming, Block Erasure, Area Erasure, Blank Checking, Property Programming, Property Erasure, Switch Programming, Switch Erasure, TAG Update, TAG Erasure Command processing. When this bit is 1, the flash sequencer is in the command lock state.</p> <p>0: No error has been detected.</p> <p>1: An error has been detected.</p> <p>[Clearing condition]</p> <p>Status Clearing or Forced Stop Command processing is started.</p>                                                                                   |

Table 4.16 FSTATR\_n Register Contents (6/6)

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2            | TBLCRCT  | <p>1-Bit Error Correction Monitor (P/E Parameter Table *<sup>1</sup>)</p> <p>Indicates that a 1-bit error has been corrected on reading the P/E parameter table. Flash sequencer reads the P/E parameter table in Programming, Multi Programming, DMA Programming, Block Erasure, Area Erasure, Blank Checking, Property Programming, Property Erasure, Switch Programming, Switch Erasure, TAG Update, TAG Erasure Command processing. When this bit is 1, the flash sequencer continues command processing, and does not enter command lock state.</p> <p>0: No error has been corrected.<br/>1: An error has been corrected.<br/>[Clearing condition]<br/>Status Clearing or Forced Stop Command processing is started.</p> |
| 1, 0         | Reserved | These bits are always read as "0".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

Note 1. P/E parameter table is data in flash memory area which cannot be accessed by user.

## 4.15 FENTRYR\_n — Flash Programming/Erase Mode Entry Register (n = 0, 2)

FENTRYR\_n register specifies programming/erase mode for code flash or data flash. To specify programming/erase mode for code flash or data flash so that flash sequencer can accept FACL commands, set either of FENTRYD or FENTRYC bit to 1.

Note that if this register is set to a value other than 0000<sub>H</sub>, 0001<sub>H</sub> and 0080<sub>H</sub>, ILGLERR bit in the FSTATR\_n register will be set and flash sequencer will enter the command lock state.

This register is reset when SUINIT bit in FSUINITR\_n register is set to 1.

This register is also reset as **Table 4.4**.

**Access:** This register can be read/written in 16-bit units.

**Address:** <FACIn\_base> + 0084<sub>H</sub> (n = 0, 2)

**Value after reset:** 0000<sub>H</sub>

| Bit               | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0              |
|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|----------|---|---|---|---|---|---|----------------|
|                   | KEY |     |     |     |     |     |     |     | FENTRYD  | — | — | — | — | — | — | FENTRYC        |
| Value after reset | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0              |
|                   | R/W | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | R/W*2,*3 | R | R | R | R | R | R | R/W*2,*3,*4,*5 |

Note 1. Written data is not stored in this bit. The value read is always 00<sub>H</sub>.

Note 2. Writing to this bit is enabled only when AA<sub>H</sub> is written to the KEY bits.

Note 3. Writing to this bit in periods from the first write access of FACL command to FRDY bit in FSTATR\_n register returns to 1 from 0 is ignored. (Except for FRDY = "1" in "Command Lock" state)

Note 4. Writing to this bit is enabled only when FPMON\_n.SFWE="1".

Note 5. [For FPSYS which has Code Flash Memory] ICUMHB is enabled: Writing to this bit is enabled only when It's permitted by ICUMHB. (See the *RH850/U2B Group Security User's Manual: Hardware*)

**Table 4.17 FENTRYR\_n Register Contents (1/2)**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15 to 8      | KEY      | Key Code<br>These bits enable or disable FENTRYD and FENTRYC bits modification.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 7            | FENTRYD  | Data Flash Programming/Erase Mode Entry<br>This bit specifies programming/erase mode for data flash.<br>0: Data flash is in read mode<br>1: Data flash is in programming/erase mode<br>[Setting condition]<br>1 is written to FENTRYD while write enabling conditions are satisfied and FENTRYR_n is 0000 <sub>H</sub> .<br>[Clearing conditions]<br>A value other than AA <sub>H</sub> is written to KEY in FENTRYR_n while FRDY bit is 1. 0 is written to FENTRYD while the write enabling conditions are satisfied.<br>FENTRYR_n is written to while FENTRYR_n is not 0000 <sub>H</sub> and the write enabling conditions are satisfied. |
| 6 to 1       | Reserved | These bits are always read as "0". Write value should always be "0".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

Table 4.17 FENTRYR\_n Register Contents (2/2)

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0            | FENTRYC  | <p>Code Flash Programming/Erase Mode Entry</p> <p>This bit specifies programming/erase mode for code flash.</p> <p>0: Code flash is in read mode</p> <p>1: Code flash is in programming/erase mode</p> <p>[Setting condition]</p> <p>1 is written to FENTRYC while write enabling conditions are satisfied and FENTRYR_n is 0000<sub>H</sub>.</p> <p>[Clearing conditions]</p> <p>A value other than AA<sub>H</sub> is written to KEY in FENTRYR_n while FRDY bit is 1. SFWE bit in FPMON_n is set to "0" while FRDY bit is "1".</p> <p>0 is written to FENTRYC while the write enabling conditions are satisfied.</p> <p>FENTRYR_n is written to while FENTRYR_n is not 0000<sub>H</sub> and the write enabling conditions are satisfied.</p> <p>[For FPSYS which has Code Flash Memory] ICUMHB is enabled: ICUMHB prohibits Code Flash Programming/Erase Mode Entry while FRDY bit is 1.</p> |



## 4.16 FSUINITR\_n — Flash Sequencer Set-up Initialize Register (n = 0, 2)

FSUINITR\_n register is used for initialization of flash sequencer set-up.

**Access:** This register can be read/written in 16-bit units.

**Address:** <FACIn\_base> + 008C<sub>H</sub> (n = 0, 2)

**Value after reset:** 0000<sub>H</sub>

| Bit               | 15              | 14              | 13              | 12              | 11              | 10              | 9               | 8               | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0                    |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|---|---|---|---|---|---|---|----------------------|
|                   | KEY             |                 |                 |                 |                 |                 |                 |                 | — | — | — | — | — | — | — | SUINIT               |
| Value after reset | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0                    |
| R/W               | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | R | R | R | R | R | R | R | R/W <sup>*2,*3</sup> |

Note 1. Written data is not stored in this bit. The value read is always 00<sub>H</sub>.

Note 2. Writing to this bit is enabled only when 2D<sub>H</sub> is written to the KEY bits.

Note 3. Writing to this bit in periods from the first write access of FACI command to FRDY bit in FSTATR\_n register returns to 1 from 0 is ignored. (Except for FRDY = "1" in "Command Lock" state)

**Table 4.18 FSUINITR\_n Register Contents**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                            |
|--------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15 to 8      | KEY      | Key Code<br>These bits enable or disable SUINIT bits modification.                                                                                                                                                                                                                  |
| 7 to 1       | Reserved | These bits are always read as "0". Write value should always be "0".                                                                                                                                                                                                                |
| 0            | SUINIT   | Set-up Initialization<br>Initializes following flash sequencer set-up registers.<br>FSADDR_n<br>FEADDR_n<br>FENTRYR_n<br>FBCCNT_n<br>FCVAPROT_n<br>0: The above flash sequencer set-up registers keep its' value.<br>1: The above flash sequencer set-up registers are initialized. |

## 4.17 FCMDR\_n — FACL Command Register (n = 0, 2)

FCMDR\_n register stores commands that Flash sequencer has accepted.

**Access:** This register can be read in 16-bit units.

**Address:** <FACIn\_base> + 00A0<sub>H</sub> (n = 0, 2)

**Value after reset:** FFFF<sub>H</sub>

| Bit               | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|
|                   | CMDR |    |    |    |    |    |   |   | PCMDR |   |   |   |   |   |   |   |
| Value after reset | 1    | 1  | 1  | 1  | 1  | 1  | 1 | 1 | 1     | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W               | R    | R  | R  | R  | R  | R  | R | R | R     | R | R | R | R | R | R | R |

**Table 4.19 FCMDR\_n Register Contents**

| Bit Position | Bit Name | Function                                                                                |
|--------------|----------|-----------------------------------------------------------------------------------------|
| 15 to 8      | CMDR     | Command<br>These bits store the latest command accepted by each Flash sequencer.        |
| 7 to 0       | PCMDR    | Previous Command<br>These bits store previous command accepted by each Flash sequencer. |

**Table 4.20 States of FCMDR\_n after Acceptance of the Various Commands**

| FACL Command                   | CMDR            | PCMDR            |
|--------------------------------|-----------------|------------------|
| Programming                    | E8 <sub>H</sub> | Previous Command |
| Multi Programming              | ED <sub>H</sub> | Previous Command |
| DMA Programming                | EA <sub>H</sub> | Previous Command |
| Block Erasure                  | D0 <sub>H</sub> | 20 <sub>H</sub>  |
| Area Erasure                   | D0 <sub>H</sub> | 23 <sub>H</sub>  |
| Programming/Erasure Suspension | B0 <sub>H</sub> | Previous Command |
| Programming/Erasure Resumption | D0 <sub>H</sub> | Previous Command |
| Status Clearing                | 50 <sub>H</sub> | Previous Command |
| Forced Stop                    | B3 <sub>H</sub> | Previous Command |
| Blank Checking                 | D0 <sub>H</sub> | 71 <sub>H</sub>  |
| Property Programming           | 45 <sub>H</sub> | Previous Command |
| Property Erasure               | D0 <sub>H</sub> | 47 <sub>H</sub>  |
| Switch Programming             | 85 <sub>H</sub> | Previous Command |
| Switch Erasure                 | D0 <sub>H</sub> | 87 <sub>H</sub>  |
| TAG Update                     | D0 <sub>H</sub> | 83 <sub>H</sub>  |
| TAG Erasure                    | D0 <sub>H</sub> | 89 <sub>H</sub>  |

## 4.18 FCMDMON\_n — Flash Sequencer Command Monitor Register (n = 0, 2)

FCMDMON\_n register stores commands that Flash sequencer has accepted.

**Access:** This register can be read in 32-bit units.

**Address:** <FACIn\_base> + 00A4<sub>H</sub> (n = 0, 2)

**Value after reset:** 0000 0000<sub>H</sub>

|                   |       |    |    |    |    |    |          |          |      |    |    |    |    |    |    |    |
|-------------------|-------|----|----|----|----|----|----------|----------|------|----|----|----|----|----|----|----|
| Bit               | 31    | 30 | 29 | 28 | 27 | 26 | 25       | 24       | 23   | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|                   | MSTCM | —  | —  | —  | —  | —  | FENTRYDM | FENTRYCM | CMDM |    |    |    |    |    |    |    |
| Value after reset | 0     | 0  | 0  | 0  | 0  | 0  | 0        | 0        | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W               | R     | R  | R  | R  | R  | R  | R        | R        | R    | R  | R  | R  | R  | R  | R  | R  |

|                   |        |    |    |    |    |    |           |           |       |   |   |   |   |   |   |   |
|-------------------|--------|----|----|----|----|----|-----------|-----------|-------|---|---|---|---|---|---|---|
| Bit               | 15     | 14 | 13 | 12 | 11 | 10 | 9         | 8         | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|                   | PMSTCM | —  | —  | —  | —  | —  | PFENTRYDM | PFENTRYCM | PCMDM |   |   |   |   |   |   |   |
| Value after reset | 0      | 0  | 0  | 0  | 0  | 0  | 0         | 0         | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W               | R      | R  | R  | R  | R  | R  | R         | R         | R     | R | R | R | R | R | R | R |

**Table 4.21 FCMDMON\_n Register Contents**

| Bit Position | Bit Name  | Function                                                                                                                                                                             |
|--------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31           | MSTCM     | Latest Command Master Monitor<br>The master which issued Latest command.<br>0: Master is not ICUMHB<br>1: Master is ICUMHB                                                           |
| 30 to 26     | Reserved  | These bits are always read as "0".                                                                                                                                                   |
| 25           | FENTRYDM  | FENTRYD status Monitor of Latest command<br>0: FENTRYD bit of FENTRY_n register is not set at latest command.<br>1: FENTRYD bit of FENTRY_n register is set at latest command.       |
| 24           | FENTRYCM  | FENTRYC status Monitor of Latest command<br>0: FENTRYC bit of FENTRY_n register is not set at latest command.<br>1: FENTRYC bit of FENTRY_n register is set at latest command.       |
| 23 to 16     | CMDM      | Command<br>These bits store the latest command accepted by each Flash sequencer.                                                                                                     |
| 15           | PMSTCM    | Previous Command Master Monitor<br>The master which issued Previous command.<br>0: Master is not ICUMHB<br>1: Master is ICUMHB                                                       |
| 14 to 10     | Reserved  | These bits are always read as "0".                                                                                                                                                   |
| 9            | PFENTRYDM | FENTRYD status Monitor of previous command<br>0: FENTRYD bit of FENTRY_n register is not set at previous command.<br>1: FENTRYD bit of FENTRY_n register is set at previous command. |
| 8            | PFENTRYCM | FENTRYC status Monitor of previous command<br>0: FENTRYC bit of FENTRY_n register is not set at previous command.<br>1: FENTRYC bit of FENTRY_n register is set at previous command. |
| 7 to 0       | PCMDM     | Previous Command<br>These bits store previous command accepted by each Flash sequencer.                                                                                              |

**Table 4.22 States of FCMDMON\_n after Acceptance of the Various Commands**

| <b>FACI Command</b>            | <b>CMDM</b>     | <b>PCMDM</b>     |
|--------------------------------|-----------------|------------------|
| Programming                    | E8 <sub>H</sub> | Previous Command |
| Multi Programming              | ED <sub>H</sub> | Previous Command |
| DMA Programming                | EA <sub>H</sub> | Previous Command |
| Block Erasure                  | 20 <sub>H</sub> | Previous Command |
| Area Erasure                   | 23 <sub>H</sub> | Previous Command |
| Programming/Erasure Suspension | B0 <sub>H</sub> | Previous Command |
| Programming/Erasure Resumption | D0 <sub>H</sub> | Previous Command |
| Status Clearing                | 50 <sub>H</sub> | Previous Command |
| Forced Stop                    | B3 <sub>H</sub> | Previous Command |
| Blank Checking                 | 71 <sub>H</sub> | Previous Command |
| Property Programming           | 45 <sub>H</sub> | Previous Command |
| Property Erasure               | 47 <sub>H</sub> | Previous Command |
| Switch Programming             | 85 <sub>H</sub> | Previous Command |
| Switch Erasure                 | 87 <sub>H</sub> | Previous Command |
| TAG Update                     | 83 <sub>H</sub> | Previous Command |
| TAG Erasure                    | 89 <sub>H</sub> | Previous Command |

## 4.19 FSWASTAT\_0 — Switch Area Status Register

FSWASTAT\_0 register indicates valid area and status of switch area.

**Access:** This register can be read in 8-bit units.

**Address:** <FACIn\_base> + 00A8<sub>H</sub> (n = 0)

**Value after reset:** XX<sub>H</sub>

| Bit               | 7 | 6 | 5 | 4     | 3     | 2     | 1    | 0    |
|-------------------|---|---|---|-------|-------|-------|------|------|
|                   | — | — | — | BPVA0 | SECVA | CFGVA | SWAS | SWVA |
| Value after reset | 0 | 0 | 0 | 0/1   | 0/1   | 0/1   | 0/1  | 0/1  |
| R/W               | R | R | R | R     | R     | R     | R    | R    |

**Table 4.23 FSWASTAT\_0 Register Contents**

| Bit Position | Bit Name | Function                                                                                   |
|--------------|----------|--------------------------------------------------------------------------------------------|
| 7 to 5       | Reserved | These bits are always read as “0”.                                                         |
| 4            | BPVA0    | Valid Area in Block Protection Area for FPSYS0<br>0: Area 0 is valid<br>1: Area 1 is valid |
| 3            | SECVA    | Valid Area in Security Setting Area<br>0: Area 0 is valid<br>1: Area 1 is valid            |
| 2            | CFGVA    | Valid Area in Configuration Setting Area<br>0: Area 0 is valid<br>1: Area 1 is valid       |
| 1            | SWAS     | Switch Area Status<br>0: Valid<br>1: Dirty                                                 |
| 0            | SWVA     | Valid Area in Switch Area<br>0: Area 0 is valid<br>1: Area 1 is valid                      |

## 4.20 FPESTAT\_n — Flash Programming/Erase Status Register (n = 0, 2)

FPESTAT\_n register is used to indicate the result of programming or erasure of the flash memory.

**Access:** This register can be read in 16-bit units.

**Address:** <FACIn\_base> + 00C0<sub>H</sub> (n = 0, 2)

**Value after reset:** 0000<sub>H</sub>

| Bit               | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
|                   | —  | —  | —  | —  | —  | —  | — | — | PEERRST |   |   |   |   |   |   |   |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0       | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W               | R  | R  | R  | R  | R  | R  | R | R | R       | R | R | R | R | R | R | R |

**Table 4.24 FPESTAT\_n Register Contents**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15 to 8      | Reserved | These bits are always read as "0".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 7 to 0       | PEERRST  | Programming/Erase Error Status<br>Indicates the source of error that occurs during programming or erasure for code flash or data flash. This bit value is only valid if ERSERR or PRGERR bit value in FSTATR_n register is 1, while FRDY bit in FSTATR_n register is 1.<br>When ERSERR and PRGERR are 0, the PEERRST bit retains the value to indicate the source of error that previously occurred.<br>00 <sub>H</sub> : No error<br>02 <sub>H</sub> : A programming error<br>12 <sub>H</sub> : An erase error<br>Other than above: Reserved |

## 4.21 FBCCNT\_n — Data Flash Blank Check Control Register (n = 0, 2)

FBCCNT\_n register specifies addressing mode in Blank Checking Command processing.

This register is reset when SUNIT bit in FSUNITR\_n register is set to 1.

This register is also reset as **Table 4.4**.

**Access:** This register can be read/written in 8-bit units.

**Address:** <FACIn\_base> + 00D0<sub>H</sub> (n = 0, 2)

**Value after reset:** 00<sub>H</sub>

| Bit               | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
|-------------------|---|---|---|---|---|---|---|-------|
|                   | — | — | — | — | — | — | — | BCDIR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0     |
| R/W               | R | R | R | R | R | R | R | R/W   |

**Table 4.25 FBCCNT\_n Register Contents**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                      |
|--------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 to 1       | Reserved | These bits are always read as "0". Write value should always be "0".                                                                                                                                                                                                          |
| 0            | BCDIR    | Blank Check Direction<br>Specifies addressing mode in blank checking operation.<br>0: Blank checking is executed from smaller address to larger address.<br>(Incremental mode)<br>1: Blank checking is executed from larger address to smaller address.<br>(Decremental mode) |

## 4.22 FBCSTAT\_n — Data Flash Blank Check Status Register (n = 0, 2)

FBCSTAT\_n register stores check results by executing the Blank Checking Command.

**Access:** This register can be read in 8-bit units.

**Address:** <FACIn\_base> + 00D4<sub>H</sub> (n = 0, 2)

**Value after reset:** 00<sub>H</sub>

| Bit               | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
|-------------------|---|---|---|---|---|---|---|------|
|                   | — | — | — | — | — | — | — | BCST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    |
| R/W               | R | R | R | R | R | R | R | R    |

**Table 4.26 FBCSTAT\_n Register Contents**

| Bit Position | Bit Name | Function                                                                                                                                                                                     |
|--------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 to 1       | Reserved | These bits are always read as “0”.                                                                                                                                                           |
| 0            | BCST     | Blank Check Status<br>Indicates the result of the Blank Checking Command.<br>0: The target area is not written (no writing after erasure)<br>1: The target area is filled with 0s and/or 1s. |



## 4.23 FPSADDR\_n — Programmed Area Start Address Register (n = 0, 2)

FPSADDR\_n register indicates address of the first programmed data which is found in Blank Checking Command execution.

**Access:** This register can be read in 32-bit units.

**Address:** <FACIn\_base> + 00D8<sub>H</sub> (n = 0, 2)

**Value after reset:** 0000 0000<sub>H</sub>

| Bit               | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20           | 19 | 18 | 17 | 16 |
|-------------------|-------------|----|----|----|----|----|----|----|----|----|----|--------------|----|----|----|----|
|                   | —           | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | PSADR[20:16] |    |    |    |    |
| Value after reset | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0            | 0  | 0  | 0  | 0  |
| R/W               | R           | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R            | R  | R  | R  | R  |
| Bit               | 15          | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4            | 3  | 2  | 1  | 0  |
|                   | PSADR[15:0] |    |    |    |    |    |    |    |    |    |    |              |    |    |    |    |
| Value after reset | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0            | 0  | 0  | 0  | 0  |
| R/W               | R           | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R            | R  | R  | R  | R  |

**Table 4.27 FPSADDR\_n Register Contents**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31 to 21     | Reserved | These bits are always read as 0.                                                                                                                                                                                                                                                                                                                                                                                                 |
| 20 to 0      | PSADR    | Programmed Area Start Address<br>Indicates address of the first programmed data which is found in Blank Checking Command execution. These bits stores address offset from the top address in the data flash memory.<br>This register value is only valid if BCST bit value in FBCSTAT_n register is 1, while FRDY bit in FSTATR_n register is 1.<br>When BCST bit is 0, the PSADR bit holds the address that previously checked. |

## 4.24 FECCEMON\_n — Flash ECC Encoder Monitor Register (n=0, 2)

FECCEMON\_n register monitors the outputs from the address parity generator and ECC encoder.

**Access:** This register can be read in 16-bit units.

**Address:** <FACIn\_base> + 0100<sub>H</sub> (n = 0, 2)

**Value after reset:** FFFF<sub>H</sub>

| Bit               | 15 | 14 | 13 | 12 | 11 | 10     | 9          | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|--------|------------|---|---|---|---|---|---|---|---|---|
|                   | —  | —  | —  | —  | —  | FAPARM | FECCM[9:0] |   |   |   |   |   |   |   |   |   |
| Value after reset | 1  | 1  | 1  | 1  | 1  | 1      | 1          | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W               | R  | R  | R  | R  | R  | R      | R          | R | R | R | R | R | R | R | R | R |

**Table 4.28 FECCEMON\_n Register Contents**

| Bit Position | Bit Name   | Function                                                                                                                                                                                                                                                                                                                                                                 |
|--------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15 to 11     | Reserved   | These bits are always read as "0".                                                                                                                                                                                                                                                                                                                                       |
| 10           | FAPARM     | Address Parity Monitor<br>Indicates the output from the address parity generator.<br>In code flash programming/erasure mode<br>This bit indicates the output from the address parity generator.<br>In data flash programming/erasure mode<br>This bit is fixed to 1.                                                                                                     |
| 9 to 0       | FECCM[9:0] | ECC Monitor<br>Indicates the ECC encoder output.<br>In code flash programming/erasure mode<br>The FECCM[9] to FECCM[0] bits indicate the ECC encoder output for the code flash memory.<br>In data flash programming/erasure mode<br>The FECCM[9] to FECCM[7] bits are fixed to 1.<br>The FECCM[6] to FECCM[0] indicate the ECC encoder output for the data flash memory. |

## 4.25 FECCTMD\_n — Flash ECC Test Mode Register (n = 0, 2)

FECCTMD\_n register sets the ECC test function for the flash memory.

**Access:** This register can be read/written in 16-bit units.

**Address:** <FACIn\_base> + 0104<sub>H</sub> (n = 0, 2)

**Value after reset:** 0030<sub>H</sub>

| Bit               | 15              | 14              | 13              | 12              | 11              | 10              | 9               | 8               | 7 | 6 | 5                 | 4                 | 3 | 2 | 1 | 0                 |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|---|---|-------------------|-------------------|---|---|---|-------------------|
|                   | KEY             |                 |                 |                 |                 |                 |                 |                 | — | — | CECCV<br>E        | DECCV<br>E        | — | — | — | ECDDI<br>SE       |
| Value after reset | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0 | 0 | 1                 | 1                 | 0 | 0 | 0 | 0                 |
| R/W               | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | W <sup>*1</sup> | R | R | R/W <sup>*2</sup> | R/W <sup>*2</sup> | R | R | R | R/W <sup>*2</sup> |

Note 1. Written data is not stored in this bit. The value read is always 00<sub>H</sub>.

Note 2. Writing to this bit is enabled only when A6<sub>H</sub> is written to the KEY bits.

**Table 4.29 FECCTMD\_n Register Contents**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                                                                                                         |
|--------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15 to 8      | KEY      | Key Code<br>These bits enable or disable CECCVE, DECCVE, and ECDDISE bits modification.                                                                                                                                                                                                                                                          |
| 7,6          | Reserved | These bits are always read as "0". Write value should always be "0".                                                                                                                                                                                                                                                                             |
| 5            | CECCVE   | Code Flash Memory ECC Bits Verify Enable<br>Specifies the verify operation on programming/erasure the code flash memory.<br>0: Verifies the data bits only.<br>1: Verifies the data bits, the ECC bits, and address parity bit.                                                                                                                  |
| 4            | DECCVE   | Data Flash Memory ECC Bits Verify Enable<br>Specifies the verify operation on programming/erasure the data flash memory.<br>0: Verifies the data bits only.<br>1: Verifies the data bits and the ECC area.                                                                                                                                       |
| 3 to 1       | Reserved | These bits are always read as "0". Write value should always be "0".                                                                                                                                                                                                                                                                             |
| 0            | ECDDISE  | ECC Encoder Disable<br>Disables the address parity generator and the ECC encoder. If the address parity generator and the ECC encoder are disabled, the FDMYECC_n value is written to the flash memory.<br>0: The address parity generator and the ECC encoder are enabled.<br>1: The address parity generator and the ECC encoder are disabled. |

## 4.26 FDMYECC\_n — Flash Dummy ECC Register (n = 0, 2)

FDMYECC\_n register specifies the address parity and ECC value to be written into the flash memory when the ECCDISE bit in the FECCTMD\_n register is 1. The bit functions in code flash programming/erasure mode are different from those in data flash programming/erasure mode as shown below.

**Access:** This register can be read/written in 16-bit units.

**Address:** <FACIn\_base> + 0108<sub>H</sub> (n = 0, 2)

**Value after reset:** FFFF<sub>H</sub>

| Bit               | 15 | 14 | 13 | 12 | 11 | 10      | 9           | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|-------------------|----|----|----|----|----|---------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                   | —  | —  | —  | —  | —  | DMYAPAR | DMYECC[9:0] |     |     |     |     |     |     |     |     |     |
| Value after reset | 1  | 1  | 1  | 1  | 1  | 1       | 1           | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| R/W               | R  | R  | R  | R  | R  | R/W     | R/W         | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

**Table 4.30 FDMYECC\_n Register Contents (in Code Flash Programming/Erasure Mode)**

| Bit Position | Bit Name    | Function                                                                              |
|--------------|-------------|---------------------------------------------------------------------------------------|
| 15 to 11     | Reserved    | These bits are always read as 1. Write value should always be 1.                      |
| 10           | DMYAPAR     | Dummy Address Parity<br>Specifies the address parity value when the ECCDISE bit is 1. |
| 9 to 0       | DMYECC[9:0] | Dummy ECC<br>Specify the ECC value when the ECCDISE bit is 1.                         |

**Table 4.31 FDMYECC\_n Register Contents (in Data Flash Programming/Erasure Mode)**

| Bit Position | Bit Name    | Function                                                                  |
|--------------|-------------|---------------------------------------------------------------------------|
| 15 to 11     | Reserved    | These bits are always read as 1. Write value should always be 1.          |
| 10           | DMYAPAR     | Reserved<br>This bit is always read as 1. Write value should always be 1. |
| 9 to 7       | DMYECC[9:7] | Reserved<br>This bit is always read as 1. Write value should always be 1. |
| 6 to 0       | DMYECC[6:0] | Dummy ECC<br>Specify the ECC value when the ECCDISE bit is 1.             |

## 4.27 SPIDINn — Serial Programmer ID Input Register (n = 0 to 7)

SPIDINn registers are for the input of an ID for use in authentication at the time of serial-programming. The ID is authenticated by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the SPIDIN0 to SPIDIN7 registers. (see the *RH850/U2B Group User's Manual: Hardware Section 58, Basic Hardware Protection*) The ID which is stored in Security Setting Area can be set by the FACI command.(see **Section 6, FACI Command**)

SPIDINn have to be inputted in sequence from lowest register number.

**Access:** This register can be read/written in 32-bit units.

**Address:** <IDCTRL\_base> + 0000<sub>H</sub> + n × 4<sub>H</sub> (n = 0 to 7)

**Value after reset:** 0000 0000<sub>H</sub>

|                   |                              |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-------------------|------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit               | 31                           | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|                   | SPIDINn[31:16] <sup>*1</sup> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                            | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                          | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit               | 15                           | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                   | SPIDINn[15:0] <sup>*1</sup>  |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                            | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                          | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. n = 0 to 7

**Table 4.32 SPIDINn Register Contents**

| Bit Position | Bit Name      | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31 to 0      | SPIDINn[31:0] | <p>ID for Use in Authentication of Serial-Programming</p> <p>The ID for use in authentication at the time of Serial-programming is input to these bits.</p> <p>Authentication of the ID is executed by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the SPIDINn[31:0] bits.</p> <p>The correspondence of the 256-bit ID compared with SPIDINn[31:0] is as follows.</p> <p>ID[31:0]: SPIDIN0[31:0]<br/> ID[63:32]: SPIDIN1[31:0]<br/> ID[95:64]: SPIDIN2[31:0]<br/> ID[127:96]: SPIDIN3[31:0]<br/> ID[159:128]: SPIDIN4[31:0]<br/> ID[191:160]: SPIDIN5[31:0]<br/> ID[223:192]: SPIDIN6[31:0]<br/> ID[255:224]: SPIDIN7[31:0]</p> |

## 4.28 DFIDINn — Data Flash ID Input Register (n = 0 to 7)

DFIDINn are input registers for a Data Flash ID authentication by S/W. The ID is authenticated by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the DFIDIN0 to DFIDIN7 registers.(see the *RH850/U2B Group User's Manual: Hardware Section 58, Basic Hardware Protection*) The ID which is stored in Security Setting Area can be set by the FACI command.(see **Section 6, FACI Command**)

DFIDINn have to be inputted in sequence from lowest register number.

**Access:** This register can be read/written in 32-bit units.

**Address:** <IDCTRL\_base> + 0020<sub>H</sub> + n × 4<sub>H</sub> (n = 0 to 7)

**Value after reset:** 0000 0000<sub>H</sub>

|                   |                              |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-------------------|------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit               | 31                           | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|                   | DFIDINn[31:16] <sup>*1</sup> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                            | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                          | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit               | 15                           | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                   | DFIDINn[15:0] <sup>*1</sup>  |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                            | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                          | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. n = 0 to 7

**Table 4.33 DFIDINn Register Contents**

| Bit Position | Bit Name      | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31 to 0      | DFIDINn[31:0] | Input registers for Data Flash ID authentication by S/W.<br>Authentication of the ID is executed by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the DFIDINn[31:0] bits.<br>The correspondence of the 256-bit ID compared with DFIDINn[31:0] is as follows.<br>ID[31:0]: DFIDIN0[31:0]<br>ID[63:32]: DFIDIN1[31:0]<br>ID[95:64]: DFIDIN2[31:0]<br>ID[127:96]: DFIDIN3[31:0]<br>ID[159:128]: DFIDIN4[31:0]<br>ID[191:160]: DFIDIN5[31:0]<br>ID[223:192]: DFIDIN6[31:0]<br>ID[255:224]: DFIDIN7[31:0] |

## 4.29 OCDIDINn — OCD ID Input Register (n = 0 to 7)

OCDIDINn are input registers for OCD ID authentication by SW. The ID is authenticated by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the OCDIDIN0 to OCDIDIN7 registers. (see the *RH850/U2B Group User's Manual: Hardware Section 58, Basic Hardware Protection*) The ID which is stored in Security Setting Area of the flash memory can be set by the FACL command. (see **Section 6, FACL Command**)

OCDIDINn have to be inputted in sequence from lowest register number.

**Access:** This register can be read/written in 32-bit units.

**Address:** <IDCTRL\_base> + 0040<sub>H</sub> + n × 4<sub>H</sub> (n = 0 to 7)

**Value after reset:** 0000 0000<sub>H</sub>

|                   |                               |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-------------------|-------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit               | 31                            | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|                   | OCDIDINn[31:16] <sup>*1</sup> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                             | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                           | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit               | 15                            | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                   | OCDIDINn[15:0] <sup>*1</sup>  |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                             | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                           | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. n = 0 to 7

**Table 4.34** OCDIDINn Register Contents

| Bit Position | Bit Name       | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |           |                |            |                |            |                |             |                |              |                |              |                |              |                |              |                |
|--------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|----------------|------------|----------------|------------|----------------|-------------|----------------|--------------|----------------|--------------|----------------|--------------|----------------|--------------|----------------|
| 31 to 0      | OCDIDINn[31:0] | <p>Input registers for OCD ID authentication by S/W.</p> <p>Authentication of the ID is executed by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the OCDIDINn[31:0] bits.</p> <p>The correspondence of the 256-bit ID compared with OCDIDINn[31:0] is as follows.</p> <table><tr><td>ID[31:0]:</td><td>OCDIDIN0[31:0]</td></tr><tr><td>ID[63:32]:</td><td>OCDIDIN1[31:0]</td></tr><tr><td>ID[95:64]:</td><td>OCDIDIN2[31:0]</td></tr><tr><td>ID[127:96]:</td><td>OCDIDIN3[31:0]</td></tr><tr><td>ID[159:128]:</td><td>OCDIDIN4[31:0]</td></tr><tr><td>ID[191:160]:</td><td>OCDIDIN5[31:0]</td></tr><tr><td>ID[223:192]:</td><td>OCDIDIN6[31:0]</td></tr><tr><td>ID[255:224]:</td><td>OCDIDIN7[31:0]</td></tr></table> | ID[31:0]: | OCDIDIN0[31:0] | ID[63:32]: | OCDIDIN1[31:0] | ID[95:64]: | OCDIDIN2[31:0] | ID[127:96]: | OCDIDIN3[31:0] | ID[159:128]: | OCDIDIN4[31:0] | ID[191:160]: | OCDIDIN5[31:0] | ID[223:192]: | OCDIDIN6[31:0] | ID[255:224]: | OCDIDIN7[31:0] |
| ID[31:0]:    | OCDIDIN0[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                |            |                |            |                |             |                |              |                |              |                |              |                |              |                |
| ID[63:32]:   | OCDIDIN1[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                |            |                |            |                |             |                |              |                |              |                |              |                |              |                |
| ID[95:64]:   | OCDIDIN2[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                |            |                |            |                |             |                |              |                |              |                |              |                |              |                |
| ID[127:96]:  | OCDIDIN3[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                |            |                |            |                |             |                |              |                |              |                |              |                |              |                |
| ID[159:128]: | OCDIDIN4[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                |            |                |            |                |             |                |              |                |              |                |              |                |              |                |
| ID[191:160]: | OCDIDIN5[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                |            |                |            |                |             |                |              |                |              |                |              |                |              |                |
| ID[223:192]: | OCDIDIN6[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                |            |                |            |                |             |                |              |                |              |                |              |                |              |                |
| ID[255:224]: | OCDIDIN7[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                |            |                |            |                |             |                |              |                |              |                |              |                |              |                |

### 4.30 CUSTIDAINn — Customer ID A Input Register (n = 0 to 7)

CUSTIDAINn are input registers for Customer ID A authentication by S/W. The ID is authenticated by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the CUSTIDAIN0 to CUSTIDAIN7 registers.(see the *RH850/U2B Group User's Manual: Hardware Section 58, Basic Hardware Protection*) The ID which is stored in Security Setting Area of the flash memory can be set by the FACI command.(see **Section 6, FACI Command**)

CUSTIDAINn have to be inputted in sequence from lowest register number.

**Access:** This register can be read/written in 32-bit units.

**Address:** <IDCTRL\_base> + 0080<sub>H</sub> + n × 4<sub>H</sub> (n = 0 to 7)

**Value after reset:** 0000 0000<sub>H</sub>

|                   |                                 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-------------------|---------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit               | 31                              | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|                   | CUSTIDAINn[31:16] <sup>*1</sup> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit               | 15                              | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                   | CUSTIDAINn[15:0] <sup>*1</sup>  |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. n = 0 to 7

**Table 4.35 CUSTIDAINn Register Contents**

| Bit Position | Bit Name         | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31 to 0      | CUSTIDAINn[31:0] | <p>Input registers for Customer ID A authentication by S/W.<br/>Authentication of the ID is executed by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the CUSTIDAINn[31:0] bits.<br/>The correspondence of the 256-bit ID compared with CUSTIDAINn[31:0] is as follows.</p> <p>ID[31:0]: CUSTIDAIN0[31:0]<br/> ID[63:32]: CUSTIDAIN1[31:0]<br/> ID[95:64]: CUSTIDAIN2[31:0]<br/> ID[127:96]: CUSTIDAIN3[31:0]<br/> ID[159:128]: CUSTIDAIN4[31:0]<br/> ID[191:160]: CUSTIDAIN5[31:0]<br/> ID[223:192]: CUSTIDAIN6[31:0]<br/> ID[255:224]: CUSTIDAIN7[31:0]</p> |



### 4.31 CUSTIDBINn — Customer ID B Input Register (n = 0 to 7)

CUSTIDBINn are input registers for Customer ID B authentication by S/W. The ID is authenticated by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the CUSTIDBIN0 to CUSTIDBIN7 registers. (see the *RH850/U2B Group User's Manual: Hardware Section 58, Basic Hardware Protection*) The ID which is stored in Security Setting Area of the flash memory can be set by the FACI command.(see **Section 6, FACI Command**)

CUSTIDBINn have to be inputted in sequence from lowest register number.

**Access:** This register can be read/written in 32-bit units.

**Address:** <IDCTRL\_base> + 000A0<sub>H</sub> + n × 4<sub>H</sub> (n = 0 to 7)

**Value after reset:** 0000 0000<sub>H</sub>

|                   |                                 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-------------------|---------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit               | 31                              | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|                   | CUSTIDBINn[31:16] <sup>*1</sup> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit               | 15                              | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                   | CUSTIDBINn[15:0] <sup>*1</sup>  |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. n = 0 to 7

**Table 4.36 CUSTIDBINn Register Contents**

| Bit Position | Bit Name         | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
|--------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|------------------|------------|------------------|------------|------------------|-------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|
| 31 to 0      | CUSTIDBINn[31:0] | <p>Input registers for Customer ID B authentication by S/W.</p> <p>Authentication of the ID is executed by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the CUSTIDBINn[31:0] bits.</p> <p>The correspondence of the 256-bit ID compared with CUSTIDBINn[31:0] is as follows.</p> <table><tr><td>ID[31:0]:</td><td>CUSTIDBIN0[31:0]</td></tr><tr><td>ID[63:32]:</td><td>CUSTIDBIN1[31:0]</td></tr><tr><td>ID[95:64]:</td><td>CUSTIDBIN2[31:0]</td></tr><tr><td>ID[127:96]:</td><td>CUSTIDBIN3[31:0]</td></tr><tr><td>ID[159:128]:</td><td>CUSTIDBIN4[31:0]</td></tr><tr><td>ID[191:160]:</td><td>CUSTIDBIN5[31:0]</td></tr><tr><td>ID[223:192]:</td><td>CUSTIDBIN6[31:0]</td></tr><tr><td>ID[255:224]:</td><td>CUSTIDBIN7[31:0]</td></tr></table> | ID[31:0]: | CUSTIDBIN0[31:0] | ID[63:32]: | CUSTIDBIN1[31:0] | ID[95:64]: | CUSTIDBIN2[31:0] | ID[127:96]: | CUSTIDBIN3[31:0] | ID[159:128]: | CUSTIDBIN4[31:0] | ID[191:160]: | CUSTIDBIN5[31:0] | ID[223:192]: | CUSTIDBIN6[31:0] | ID[255:224]: | CUSTIDBIN7[31:0] |
| ID[31:0]:    | CUSTIDBIN0[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[63:32]:   | CUSTIDBIN1[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[95:64]:   | CUSTIDBIN2[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[127:96]:  | CUSTIDBIN3[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[159:128]: | CUSTIDBIN4[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[191:160]: | CUSTIDBIN5[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[223:192]: | CUSTIDBIN6[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[255:224]: | CUSTIDBIN7[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |

## 4.32 CUSTIDCINn — Customer ID C Input Register (n = 0 to 7)

CUSTIDCINn are input registers for Customer ID C authentication by S/W. The ID is authenticated by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the CUSTIDCIN0 to CUSTIDCIN7 registers.(see the *RH850/U2B Group User's Manual: Hardware Section 58, Basic Hardware Protection*) The ID which is stored in Security Setting Area of the flash memory can be set by the FACI command.(see **Section 6, FACI Command**)

CUSTIDCINn have to be inputted in sequence from lowest register number.

**Access:** This register can be read / written in 32-bit units.

**Address:** <IDCTRL\_base> + 000C0<sub>H</sub> + n × 4<sub>H</sub> (n = 0 to 7)

**Value after reset:** 0000 0000<sub>H</sub>

|                   |                                 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-------------------|---------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit               | 31                              | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|                   | CUSTIDCINn[31:16] <sup>*1</sup> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit               | 15                              | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                   | CUSTIDCINn[15:0] <sup>*1</sup>  |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                   | R/W                             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. n = 0 to 7

**Table 4.37 CUSTIDCINn Register Contents**

| Bit Position | Bit Name         | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
|--------------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|------------------|------------|------------------|------------|------------------|-------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|
| 31 to 0      | CUSTIDCINn[31:0] | <p>Input registers for Customer ID C authentication by S/W.<br/>Authentication of the ID is executed by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the CUSTIDCINn[31:0] bits.</p> <p>The correspondence of the 256-bit ID compared with CUSTIDCINn[31:0] is as follows.</p> <table><tr><td>ID[31:0]:</td><td>CUSTIDCIN0[31:0]</td></tr><tr><td>ID[63:32]:</td><td>CUSTIDCIN1[31:0]</td></tr><tr><td>ID[95:64]:</td><td>CUSTIDCIN2[31:0]</td></tr><tr><td>ID[127:96]:</td><td>CUSTIDCIN3[31:0]</td></tr><tr><td>ID[159:128]:</td><td>CUSTIDCIN4[31:0]</td></tr><tr><td>ID[191:160]:</td><td>CUSTIDCIN5[31:0]</td></tr><tr><td>ID[223:192]:</td><td>CUSTIDCIN6[31:0]</td></tr><tr><td>ID[255:224]:</td><td>CUSTIDCIN7[31:0]</td></tr></table> | ID[31:0]: | CUSTIDCIN0[31:0] | ID[63:32]: | CUSTIDCIN1[31:0] | ID[95:64]: | CUSTIDCIN2[31:0] | ID[127:96]: | CUSTIDCIN3[31:0] | ID[159:128]: | CUSTIDCIN4[31:0] | ID[191:160]: | CUSTIDCIN5[31:0] | ID[223:192]: | CUSTIDCIN6[31:0] | ID[255:224]: | CUSTIDCIN7[31:0] |
| ID[31:0]:    | CUSTIDCIN0[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[63:32]:   | CUSTIDCIN1[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[95:64]:   | CUSTIDCIN2[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[127:96]:  | CUSTIDCIN3[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[159:128]: | CUSTIDCIN4[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[191:160]: | CUSTIDCIN5[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[223:192]: | CUSTIDCIN6[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[255:224]: | CUSTIDCIN7[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |

### 4.33 IDST —ID Authentication Status Register

IDST indicates ID authentication result.

**Access:** This register can be read in 32-bit units.

**Address:** <IDCTRL\_base> + 001FC<sub>H</sub>

**Value after reset:** 0000 00XX<sub>H</sub>

| Bit               | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|                   | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W               | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

| Bit               | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6                 | 5                 | 4                 | 3                 | 2                 | 1                 | 0                 |
|-------------------|----|----|----|----|----|----|---|---|---|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
|                   | —  | —  | —  | —  | —  | —  | — | — | — | CUSTID<br>CR      | CUSTID<br>BR      | CUSTID<br>AR      | — <sup>*4</sup>   | OCDID<br>R        | DFIDR             | SPIDR             |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0/1 <sup>*7</sup> | 0/1 <sup>*6</sup> | 0/1 <sup>*5</sup> | 0/1 <sup>*4</sup> | 0/1 <sup>*3</sup> | 0/1 <sup>*2</sup> | 0/1 <sup>*1</sup> |
| R/W               | R  | R  | R  | R  | R  | R  | R | R | R | R                 | R                 | R                 | R                 | R                 | R                 | R                 |

- Note 1. When Serial Programming ID[255:0] = All-0, initial value is 0. Otherwise, initial value is 1.
- Note 2. When Data Flash ID[255:0] = All-1 or All-0, initial value is 0. Otherwise, initial value is 1.  
When Data Flash ID[255:0] = All-1, the value is always 0 because authentication is not required.
- Note 3. When OCD ID[255:0] = All-1 or All-0, initial value is 0. Otherwise, initial value is 1.  
When OCD ID[255:0] = All-1, the value is always 0 because authentication is not required.
- Note 4. For this reserved bit, see the *RH850/U2B Group Security User's Manual: Hardware*.
- Note 5. When Customer ID A[255:0] = All-1 or All-0, initial value is 0. Otherwise, initial value is 1.  
When Customer ID A[255:0] = All-1, the value is always 0 because authentication is not required.
- Note 6. When Customer ID B[255:0] = All-1 or All-0, initial value is 0. Otherwise, initial value is 1.  
When Customer ID B[255:0] = All-1, the value is always 0 because authentication is not required.
- Note 7. When Customer ID C[255:0] = All-1 or All-0, initial value is 0. Otherwise, initial value is 1.  
When Customer ID C[255:0] = All-1, the value is always 0 because authentication is not required.

**Table 4.38 IDST Register Contents (1/2)**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                      |
|--------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31 to 7      | Reserved | These bits are always read as 0.                                                                                                                                                                                              |
| 6            | CUSTIDCR | Customer ID C Authentication Status<br>This bit indicates the result of comparing the Customer ID C with the CUSTIDCINn registers.<br>0: The IDs match (protection unlocked).<br>1: The IDs do not match (protection locked). |
| 5            | CUSTIDBR | Customer ID B Authentication Status<br>This bit indicates the result of comparing the Customer ID B with the CUSTIDBINn registers.<br>0: The IDs match (protection unlocked).<br>1: The IDs do not match (protection locked). |
| 4            | CUSTIDAR | Customer ID A Authentication Status<br>This bit indicates the result of comparing the Customer ID A with the CUSTIDAINn registers.<br>0: The IDs match (protection unlocked).<br>1: The IDs do not match (protection locked). |
| 3            | Reserved | When read, the value returned is undefined.<br>For this reserved bit, see the <i>RH850/U2B Group Security User's Manual: Hardware</i> .                                                                                       |

**Table 4.38 IDST Register Contents (2/2)**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                   |
|--------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2            | OCDIDR   | OCD ID Authentication Status<br>This bit indicates the result of comparing the OCD ID with the OCDIDINn registers.<br>0: The IDs match (protection unlocked).<br>1: The IDs do not match (protection locked).                              |
| 1            | DFIDR    | Data-Flash ID Authentication Status<br>This bit indicates the result of comparing the Data Flash ID with the DFIDINn registers.<br>0: The IDs match (protection unlocked).<br>1: The IDs do not match (protection locked).                 |
| 0            | SPIDR    | Serial-Programming ID Authentication Status<br>This bit indicates the result of comparing the Serial Programming ID with the SPIDINn registers.<br>0: The IDs match (protection unlocked).<br>1: The IDs do not match (protection locked). |

### 4.34 RHSIFIDINn — RHSIF ID Input Register (n = 0 to 7)

RHSIFIDINn are input registers for RHSIF ID authentication by S/W. The ID is authenticated by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the RHSIFIDIN0 to RHSIFIDIN7 registers.(see the *RH850/U2B Group User's Manual: Hardware Section 58, Basic Hardware Protection*) The ID which is stored in Security Setting Area of the flash memory can be set by the FACL command.(see **Section 6, FACL Command**)

RHSIFIDINn have to be inputted in sequence from lowest register number.

[For U2B6] These registers are available but have no function.

**Access:** This register can be read/written in 32-bit units.

**Address:** <IDCTRL\_base> + 00200<sub>H</sub> + n × 4<sub>H</sub> (n = 0 to 7)

**Value after reset:** 0000 0000<sub>H</sub>

| Bit               | 31                              | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|-------------------|---------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                   | RHSIFIDINn[31:16] <sup>*1</sup> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W               | R/W                             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit               | 15                              | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                   | RHSIFIDINn[15:0] <sup>*1</sup>  |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Value after reset | 0                               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W               | R/W                             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. n = 0 to 7

**Table 4.39 RHSIFIDINn Register Contents**

| Bit Position | Bit Name         | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
|--------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|------------------|------------|------------------|------------|------------------|-------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|
| 31 to 0      | RHSIFIDINn[31:0] | <p>Input registers for RHSIF ID authentication by S/W.</p> <p>Authentication of the ID is executed by comparing the 256-bit ID that has been set in advance in Security Setting Area with the value in the RHSIFIDINn[31:0] bits.</p> <p>The correspondence of the 256-bit ID compared with RHSIFIDINn[31:0] is as follows.</p> <table><tr><td>ID[31:0]:</td><td>RHSIFIDIN0[31:0]</td></tr><tr><td>ID[63:32]:</td><td>RHSIFIDIN1[31:0]</td></tr><tr><td>ID[95:64]:</td><td>RHSIFIDIN2[31:0]</td></tr><tr><td>ID[127:96]:</td><td>RHSIFIDIN3[31:0]</td></tr><tr><td>ID[159:128]:</td><td>RHSIFIDIN4[31:0]</td></tr><tr><td>ID[191:160]:</td><td>RHSIFIDIN5[31:0]</td></tr><tr><td>ID[223:192]:</td><td>RHSIFIDIN6[31:0]</td></tr><tr><td>ID[255:224]:</td><td>RHSIFIDIN7[31:0]</td></tr></table> | ID[31:0]: | RHSIFIDIN0[31:0] | ID[63:32]: | RHSIFIDIN1[31:0] | ID[95:64]: | RHSIFIDIN2[31:0] | ID[127:96]: | RHSIFIDIN3[31:0] | ID[159:128]: | RHSIFIDIN4[31:0] | ID[191:160]: | RHSIFIDIN5[31:0] | ID[223:192]: | RHSIFIDIN6[31:0] | ID[255:224]: | RHSIFIDIN7[31:0] |
| ID[31:0]:    | RHSIFIDIN0[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[63:32]:   | RHSIFIDIN1[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[95:64]:   | RHSIFIDIN2[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[127:96]:  | RHSIFIDIN3[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[159:128]: | RHSIFIDIN4[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[191:160]: | RHSIFIDIN5[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[223:192]: | RHSIFIDIN6[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |
| ID[255:224]: | RHSIFIDIN7[31:0] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                  |            |                  |            |                  |             |                  |              |                  |              |                  |              |                  |              |                  |

## 4.35 IDST2 —ID Authentication Status Register 2

IDST2 indicates ID authentication result.

**Access:** This register can be read in 32-bit units.

**Address:** <IDCTRL\_base> + 003FC<sub>H</sub>

**Value after reset:** 0000 000X<sub>H</sub>

| Bit               | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|                   | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W               | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

| Bit               | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0        |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----------|
|                   | —  | —  | —  | —  | —  | —  | — | — | — | — | — | — | — | — | — | RHSIFIDR |
| Value after reset | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1**1   |
| R/W               | R  | R  | R  | R  | R  | R  | R | R | R | R | R | R | R | R | R | R        |

Note 1. When RHSIF ID[255:0] = All-1 or All-0, initial value is 0. Otherwise, initial value is 1.  
 When RHSIF ID[255:0] = All-1, the value is always 0 because authentication is not required.  
 [For U2B6] The value is always 0.

**Table 4.40 IDST2 Register Contents**

| Bit Position | Bit Name | Function                                                                                                                                                                                                                                                 |
|--------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31 to 1      | Reserved | These bits are always read as 0.                                                                                                                                                                                                                         |
| 0            | RHSIFIDR | RHSIF ID Authentication Status<br>This bit indicates the result of comparing the RHSIF ID with the RHSIFIDINn registers.<br>0: The IDs match (protection unlocked).<br>1: The IDs do not match (protection locked).<br>[For U2B6] The value is always 0. |

## Section 5 Flash Sequencer Operating Modes

This section gives information of operating modes of flash sequencer and flash memory of each FPSYS.

The flash sequencer has three operating modes as shown in **Figure 5.1**. The mode is shifted by the write to the FENTRYR\_0/2 register.

When the FENTRYR\_0/2 register is 0000<sub>H</sub>, the flash sequencer is in read mode. In this mode, it does not accept the FACI command except Status clear command. Code flash memory and data flash memory are both readable.

When the FENTRYR\_0 register is 0001<sub>H</sub>, the flash sequencer is in code flash programming/erasure mode where the code flash memory can be programmed/erased by the FACI command. In this mode, Flash sequencer controls a bank of code flash memory, and this bank of code flash memory is not readable while the flash sequencer is processing state (FRDY bit in FSTATR\_n register is “0”). In addition, the Hardware Property Area is not readable while a code flash block which is configured to enable erase counter is being erased (FRDY bit in FSTATR\_n register is “0”). The other bank of code flash memory and data flash memory are readable (i.e. BGO: background operation).

When the FENTRYR\_0/2 register is 0080<sub>H</sub>, the flash sequencer is in data flash programming/erasure mode where the data flash memory can be programmed/erased by the FACI command. In this mode, Flash sequencer controls a Data Area or Hardware Property Area of data flash memory, and this area of data flash memory is not readable while the flash sequencer is processing state (FRDY bit in FSTATR\_n register is “0”). Code flash memory, the other Data Area and Hardware Property Area are readable when Flash sequencer controls a Data Area. Code flash memory and Data Area are readable when Flash sequencer controls a Hardware Property Area (i.e. BGO).

In addition, it is independent to set FENTRYR\_0, FENTRYR\_1, and FENTRYR\_2 to 0001<sub>H</sub> or 0080<sub>H</sub>. Flash memory of a FPSYS and flash memory of the other FPSYS can be in a flash programming/erasure mode (i.e. Multi FPSYS Operation).

As for the condition to enable the BGO and Multi FPSYS Operation, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

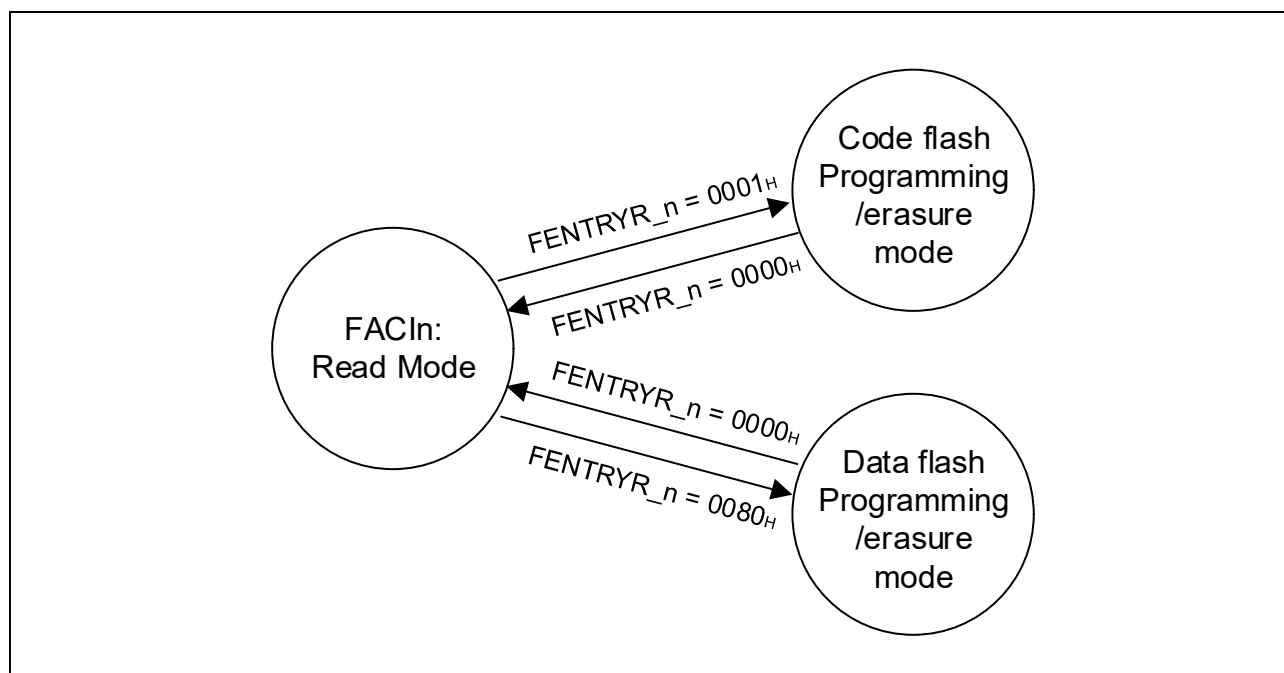


Figure 5.1 Flash Sequencer Modes



## Section 6 FACL Command

### 6.1 List of FACL Commands

Table 6.1 List of FACL Commands

| FACL Command                   | Function                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Programming                    | <ul style="list-style-type: none"> <li>User Area, Extended Data Area and Data Area can be programmed.</li> <li>Programming unit is 512 bytes for User Area and 4 bytes for Data Area and Extended Data Area</li> </ul>                                                                                                                                                                                           |
| Multi Programming              | <ul style="list-style-type: none"> <li>Data Area and Extended Data Area can be programmed.</li> <li>Program unit is 8, 16, 32, 64 or 128 bytes</li> </ul>                                                                                                                                                                                                                                                        |
| DMA Programming                | <ul style="list-style-type: none"> <li>Data Area can be programmed using the DMA controller.</li> <li>Programming unit is 4 bytes to "256 Kbytes - 4 bytes" (4 bytes step).</li> <li>Maximum program unit is limited within one Data Area*<sup>1</sup>.</li> <li>Address boundary between different Data Area cannot be crossed.</li> </ul>                                                                      |
| Block Erasure                  | <ul style="list-style-type: none"> <li>User Area, Extended Data Area and Data Area can be erased.</li> <li>Erasure unit is one block.</li> </ul>                                                                                                                                                                                                                                                                 |
| Area Erasure                   | <ul style="list-style-type: none"> <li>Data Area and Extended Data Area can be erased.</li> <li>Erasure unit are N blocks for Data Area. (4 Kbytes × N = 1, 2, 3 ....)</li> <li>Maximum erasure unit is limited within one Data Area*<sup>1</sup>.</li> <li>Address boundary between different Data Area cannot be crossed.</li> <li>Erasure unit is one block for Extended Data Area. (2 Kbytes × 1)</li> </ul> |
| Programming/Erasure Suspension | Programming/Multi Programming or Block Erasure/Area Erasure Command operation can be suspended.                                                                                                                                                                                                                                                                                                                  |
| Programming/Erasure Resumption | Suspended Programming/Multi Programming or Block Erasure/Area Erasure Command operation can be resumed.                                                                                                                                                                                                                                                                                                          |
| Status Clearing                | Error detection and correction flags in the FASTAT_n and FSTATR_n registers are cleared, and the flash sequencer is released from "Command Lock" state.                                                                                                                                                                                                                                                          |
| Forced Stop                    | Any command operation can be stopped forcibly, and the FASTAT_n and FSTATR_n registers are reset.                                                                                                                                                                                                                                                                                                                |
| Blank Checking                 | <ul style="list-style-type: none"> <li>Data Area can be checked.</li> <li>Blank checking unit is 4 to 256 Kbytes (4 bytes step).</li> <li>Maximum blank checking unit is limited within one Data Area*<sup>1</sup>.</li> <li>Address boundary between different Data Area cannot be crossed.</li> </ul>                                                                                                          |
| Property Programming           | <ul style="list-style-type: none"> <li>The back side (invalid) of Configuration Setting Area (Option bytes, Reset Vector, etc.), Block Protection Area (OTP settings, Block Protection Settings, etc.), and Security Setting Area (ID Codes, Debugger connection prohibited setting, etc.) can be programmed from the initial values for this product.</li> <li>Programming unit is 32 bytes.</li> </ul>         |
| Property Erasure               | <ul style="list-style-type: none"> <li>The back side (invalid) of Configuration Setting Area (Option bytes, Reset Vector, etc.), Block Protection Area (OTP settings, Block Protection Settings, etc.), and Security Setting Area (ID Codes, Debugger connection prohibited setting, etc.) can be erased from the initial values for this product.</li> <li>Erasure unit is 2 Kbytes.</li> </ul>                 |
| Switch Programming             | <ul style="list-style-type: none"> <li>The back side (invalid) of Switch Area can be programmed.</li> <li>Programming unit is 32 bytes.</li> </ul>                                                                                                                                                                                                                                                               |
| Switch Erasure                 | <ul style="list-style-type: none"> <li>The back side (invalid) of Switch Area can be erased.</li> <li>Erasure unit is 2 Kbytes.</li> </ul>                                                                                                                                                                                                                                                                       |
| TAG Update                     | TAG Area can be programmed, and Switch Area can be swapped back side (invalid) and front side (valid).                                                                                                                                                                                                                                                                                                           |
| TAG Erasure                    | <ul style="list-style-type: none"> <li>TAG Area can be erased.</li> <li>Erasure unit is 2 Kbytes.</li> </ul>                                                                                                                                                                                                                                                                                                     |

Note 1. For details of one Data Area, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

The FACL commands are issued by the write access to the FACL command-issuing area (see **Table 3.1**). When the write access as shown in **Table 6.2** is issued in the specified state, the flash sequencer executes the processing corresponding to each command (see **Section 6.2, Relationship between Flash Sequencer Status and FACL Commands**).

**Table 6.2 Flash Sequencer Command Format**

| FACL Command                                                                                                          | Number of Write Access | Write Data to FACL Command-issuing Area |                                                                                                                      |                                      |                    |
|-----------------------------------------------------------------------------------------------------------------------|------------------------|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------|--------------------------------------|--------------------|
|                                                                                                                       |                        | 1st Access*2                            | 2nd Access*1                                                                                                         | 3rd to (N + 2)th Access              | (N + 3)th Access*2 |
| Programming (User Area)<br>512-byte programming (N = 128)                                                             | 131                    | E8 <sub>H</sub>                         | 80 <sub>H</sub> (=N)                                                                                                 | WD <sub>1</sub> to WD <sub>128</sub> | D0 <sub>H</sub>    |
| Programming (Data Area)<br>4-byte programming (N = 1)                                                                 | 4                      | E8 <sub>H</sub>                         | 01 <sub>H</sub> (=N)                                                                                                 | WD <sub>1</sub>                      | D0 <sub>H</sub>    |
| Multi Programming<br>8 bytes (N = 2), 16 bytes (N = 4),<br>32 bytes (N = 8), 64 bytes (N = 16),<br>128 bytes (N = 32) | N + 3                  | ED <sub>H</sub>                         | 02 <sub>H</sub> (=N)<br>04 <sub>H</sub> (=N)<br>08 <sub>H</sub> (=N)<br>10 <sub>H</sub> (=N)<br>20 <sub>H</sub> (=N) | WD <sub>1</sub> to WD <sub>N</sub>   | D0 <sub>H</sub>    |
| DMA Programming<br>N = 1 to 65535                                                                                     | N + 2                  | EA <sub>H</sub>                         | N                                                                                                                    | WD <sub>1</sub> to WD <sub>N</sub>   | —                  |
| Block Erasure                                                                                                         | 2                      | 20 <sub>H</sub>                         | D0 <sub>H</sub>                                                                                                      | —                                    | —                  |
| Area Erasure                                                                                                          | 2                      | 23 <sub>H</sub>                         | D0 <sub>H</sub>                                                                                                      | —                                    | —                  |
| Programming/Erasure Suspension                                                                                        | 1                      | B0 <sub>H</sub>                         | —                                                                                                                    | —                                    | —                  |
| Programming/Erasure Resumption                                                                                        | 1                      | D0 <sub>H</sub>                         | —                                                                                                                    | —                                    | —                  |
| Status Clearing                                                                                                       | 1                      | 50 <sub>H</sub>                         | —                                                                                                                    | —                                    | —                  |
| Forced Stop                                                                                                           | 1                      | B3 <sub>H</sub>                         | —                                                                                                                    | —                                    | —                  |
| Blank Checking                                                                                                        | 2                      | 71 <sub>H</sub>                         | D0 <sub>H</sub>                                                                                                      | —                                    | —                  |
| Property Programming<br>32 bytes (N = 8)                                                                              | 11                     | 45 <sub>H</sub>                         | 08 <sub>H</sub> (=N)                                                                                                 | WD <sub>1</sub> to WD <sub>8</sub>   | D0 <sub>H</sub>    |
| Property Erasure                                                                                                      | 2                      | 47 <sub>H</sub>                         | D0 <sub>H</sub>                                                                                                      | —                                    | —                  |
| Switch Programming<br>32 bytes (N = 8)                                                                                | 11                     | 85 <sub>H</sub>                         | 08 <sub>H</sub> (=N)                                                                                                 | WD <sub>1</sub> to WD <sub>8</sub>   | D0 <sub>H</sub>    |
| Switch Erasure                                                                                                        | 2                      | 87 <sub>H</sub>                         | D0 <sub>H</sub>                                                                                                      | —                                    | —                  |
| TAG Update                                                                                                            | 2                      | 83 <sub>H</sub>                         | D0 <sub>H</sub>                                                                                                      | —                                    | —                  |
| TAG Erasure                                                                                                           | 2                      | 89 <sub>H</sub>                         | D0 <sub>H</sub>                                                                                                      | —                                    | —                  |

**Note:** WD<sub>N</sub> (N = 1, 2,...): Nth 32-bit data to be programmed.

Note 1. 8-bit data is written by a command other than DMA Programming. By the DMA Programming Command, 16-bit data is written.

Note 2. 8-bit data is written by a command.

Once the flash sequencer starts processing any command other than Status Clearing, it sets the FRDY bit in the FSTATR register to 0, and when processing of the command is complete, it sets the FRDY bit to 1 (see **Section 4.14, FSTATR<sub>n</sub> — Flash Status Register (n = 0, 2)**). When the value of the FRDY bit changes from 0 to 1, a flash ready (FRDY) interrupt is generated.

## 6.2 Relationship between Flash Sequencer Status and FACI Commands

The FACI commands are accepted according to the mode/state of the flash sequencer.

The FACI command should be issued after the shift of the flash sequencer to the code flash programming/erasure mode or data flash programming/erasure mode and checking that the flash sequencer has shifted to the mode. To check the state of flash sequencer, use the FSTATR\_n and FASTAT\_n registers. In addition, error occurrence can be checked by the CMDLK bit in the FASTAT\_n register. It is the logical OR of the ERCDTCT, SWTDTCT, SECDTCT, ILGCOMERR, FESETERR, SECERR, OTERR, BPLDTCT, ILGLERR, ERSERR, PRGERR, FHVEERR, CFGDTCT, and TBLDTCT, bits of the FSTATR\_n register.

**Table 6.3** summarizes available flash sequencer commands in each operating mode.

**Table 6.3 Flash Sequencer Operation Mode and Available Commands**

| Operating Mode                      | FENTRYR_n         | Available Command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Read Mode                           | 0000 <sub>H</sub> | <ul style="list-style-type: none"> <li>Status Clearing</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Code flash programming/erasure Mode | 0001 <sub>H</sub> | <ul style="list-style-type: none"> <li>Programming</li> <li>Block Erasure</li> <li>Programming/Erasure Suspension</li> <li>Programming/Erasure Resumption</li> <li>Status Clearing</li> <li>Forced Stop</li> </ul>                                                                                                                                                                                                                                                                                                                           |
| Data flash programming/erasure Mode | 0080 <sub>H</sub> | <ul style="list-style-type: none"> <li>Programming</li> <li>Multi Programming</li> <li>DMA Programming</li> <li>Block Erasure</li> <li>Area Erasure</li> <li>Programming/Erasure Suspension</li> <li>Programming/Erasure Resumption</li> <li>Status Clearing</li> <li>Forced Stop</li> <li>Blank Checking*<sup>1</sup><br/>See <b>Section 6.3.17, Blank Checking Command</b></li> <li>Property Programming</li> <li>Property Erasure</li> <li>Switch Programming</li> <li>Switch Erasure</li> <li>TAG Update</li> <li>TAG Erasure</li> </ul> |

Note 1. Blank Checking to Hardware Property Area, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

**Table 6.4** shows the flash sequencer state and the acceptable FACI commands. The table assumes appropriate flash sequencer operation mode is set before issuing the command.

Table 6.4 Flash Sequencer State and Acceptable FACL Commands

|                                    | Programming/Multi Programming/Block Erasure/Area Erasure Command processing | Property Programming/Property Erasure/Switch Programming/Switch Erasure/TAG Update/TAG Erasure Command processing | Programming/Multi Programming/Block Erasure/Area Erasure Command suspension processing | Blank Checking Command processing | DMA Programming Command processing | While suspend Programming/Multi Programming Command | While suspend Block Erasure/Area Erasure Command | While suspend Block Erasure/Area Erasure Command, and Programming/Multi Programming Command processing | Command Lock state (FRDY = 1) | Command Lock state (FRDY = 0) | Forced Stop Command processing | Read mode | Other               |
|------------------------------------|-----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|-----------------------------------|------------------------------------|-----------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------------|-------------------------------|-------------------------------|--------------------------------|-----------|---------------------|
| FRDY bit                           | 0                                                                           | 0                                                                                                                 | 0                                                                                      | 0                                 | 0                                  | 1                                                   | 1                                                | 0                                                                                                      | 1                             | 0                             | 0                              | 1         | 1                   |
| SUSRDY bit                         | 1                                                                           | 0                                                                                                                 | 0                                                                                      | 0                                 | 0                                  | 0                                                   | 0                                                | 0                                                                                                      | 0                             | 0                             | 0                              | 0         | 0                   |
| ERSSPD bit                         | 0                                                                           | 0                                                                                                                 | 0/1                                                                                    | 0/1                               | 0                                  | 0                                                   | 1                                                | 1                                                                                                      | 0/1                           | 0/1                           | 0                              | 0/1       | 0                   |
| PRGSPD bit                         | 0                                                                           | 0                                                                                                                 | 0/1                                                                                    | 0/1                               | 0                                  | 1                                                   | 0                                                | 0                                                                                                      | 0/1                           | 0/1                           | 0                              | 0/1       | 0                   |
| FENTRYC or FENTRYD bit             | 1                                                                           | 1                                                                                                                 | 1                                                                                      | 1                                 | 1                                  | 1                                                   | 1                                                | 1                                                                                                      | 1                             | 1                             | 1                              | 0         | 1                   |
| CMDLK bit                          | 0                                                                           | 0                                                                                                                 | 0                                                                                      | 0                                 | 0                                  | 0                                                   | 0                                                | 0                                                                                                      | 1                             | 1                             | 0                              | 0/1       | 0                   |
| Programming                        | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | O <sup>*2</sup>                                     | O <sup>*2</sup>                                  | X                                                                                                      | X                             | X                             | X                              | X         | O                   |
| Multi Programming                  | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | O <sup>*1, *2</sup>                                 | O <sup>*1, *2</sup>                              | X                                                                                                      | X                             | X                             | X                              | X         | O <sup>*1</sup>     |
| DMA Programming                    | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | O <sup>*1, *2</sup>                                 | O <sup>*1, *2</sup>                              | X                                                                                                      | X                             | X                             | X                              | X         | O <sup>*1</sup>     |
| Block Erasure                      | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | O <sup>*2</sup>                                     | O <sup>*2</sup>                                  | X                                                                                                      | X                             | X                             | X                              | X         | O                   |
| Area Erasure                       | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | O <sup>*1, *2</sup>                                 | O <sup>*1, *2</sup>                              | X                                                                                                      | X                             | X                             | X                              | X         | O <sup>*1</sup>     |
| Programming/Erasure Suspension     | O                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | X                                                   | X                                                | X                                                                                                      | —                             | X                             | X                              | X         | —                   |
| Programming/Erasure Resumption     | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | O                                                   | O                                                | X                                                                                                      | X                             | X                             | X                              | X         | X                   |
| Status Clearing                    | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | O                                                   | O                                                | X                                                                                                      | O                             | X                             | X                              | O         | O                   |
| Forced Stop                        | O                                                                           | O                                                                                                                 | O                                                                                      | O                                 | O                                  | O                                                   | O                                                | O                                                                                                      | O                             | O                             | O                              | X         | O                   |
| Blank Checking                     | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | O <sup>*1</sup>                                     | O <sup>*1</sup>                                  | X                                                                                                      | X                             | X                             | X                              | X         | O <sup>*1</sup>     |
| Property Programming <sup>*3</sup> | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | X                                                   | X                                                | X                                                                                                      | X                             | X                             | X                              | X         | O <sup>*1</sup>     |
| Property Erasure <sup>*3</sup>     | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | X                                                   | X                                                | X                                                                                                      | X                             | X                             | X                              | X         | O <sup>*1</sup>     |
| Switch Programming <sup>*3</sup>   | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | X                                                   | X                                                | X                                                                                                      | X                             | X                             | X                              | X         | O <sup>*1</sup>     |
| Switch Erasure <sup>*3</sup>       | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | X                                                   | X                                                | X                                                                                                      | X                             | X                             | X                              | X         | O <sup>*1, *4</sup> |
| TAG Update <sup>*3</sup>           | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | X                                                   | X                                                | X                                                                                                      | X                             | X                             | X                              | X         | O <sup>*1, *5</sup> |
| TAG Erasure <sup>*3</sup>          | X                                                                           | X                                                                                                                 | X                                                                                      | X                                 | X                                  | X                                                   | X                                                | X                                                                                                      | X                             | X                             | X                              | X         | O <sup>*1, *5</sup> |

Remarks:

O: Acceptable

X: Not acceptable (enter Command Lock state)

—: Ignored (Not acceptable and not enter Command Lock state)

Note 1. Acceptable only in data flash programming/erasure mode.

Note 2. Acceptable when programming/erasure area is other than programming/erasure suspending area. See **Table 6.5** for details of programming/erasure command acceptable area.

Note 3. Acceptable only in FACL0. See **Table 6.6** for details of acceptable FACL Commands for Hardware property area.

Note 4. Not acceptable when VAF in TAG Area is not correct (Flag protection).

Note 5. Not acceptable when the CVA/SVA/BVA0 in Switch Area or VOFs in Configuration Setting Area/Security Setting Area/Block Protection Area are not correct (Flag protection).

**Table 6.5 Programming/Erasure Command Acceptable Area while Suspend Programming/Erasure Command within one FACL**

|                                                  |            |     | Programming, Multi Programming or DMA Programming Command acceptable area |     |            |     |     | Block Erasure or Area Erasure Command acceptable area |   |            |     |     |
|--------------------------------------------------|------------|-----|---------------------------------------------------------------------------|-----|------------|-----|-----|-------------------------------------------------------|---|------------|-----|-----|
|                                                  |            |     | Code Flash                                                                |     | Data Flash |     |     | Code Flash                                            |   | Data Flash |     |     |
|                                                  |            |     | m                                                                         | n   | i          | j   | exd | m                                                     | n | i          | j   | exd |
| Programming or Multi Programming suspending area | Code Flash | m   | X                                                                         | O   | O          | O   | O   | X                                                     | O | O          | O   | O   |
|                                                  |            | n   | O                                                                         | X   | O          | O   | O   | O                                                     | X | O          | O   | O   |
|                                                  | Data Flash | i   | O                                                                         | O   | O*2        | O   | O   | O                                                     | O | O*2        | O   | O   |
|                                                  |            | j   | O                                                                         | O   | O          | O*2 | O   | O                                                     | O | O          | O*2 | O   |
|                                                  |            | exd | O                                                                         | O   | O          | O   | O*2 | O                                                     | O | O          | O   | X   |
| Block Erasure or Area Erasure suspending area    | Code Flash | m   | O*1                                                                       | O   | O          | O   | O   | X                                                     | O | O          | O   | O   |
|                                                  |            | n   | O                                                                         | O*1 | O          | O   | O   | O                                                     | X | O          | O   | O   |
|                                                  | Data Flash | i   | O                                                                         | O   | O*2        | O   | O   | O                                                     | O | O*2        | O   | O   |
|                                                  |            | j   | O                                                                         | O   | O          | O*2 | O   | O                                                     | O | O          | O*2 | O   |
|                                                  |            | exd | O                                                                         | O   | O          | O   | X   | O                                                     | O | O          | O   | X   |

Remarks:

O: Acceptable

X: Not acceptable (enter Command Lock state)

m, n: Bank number\*3

i, j: Data Area number\*3

exd: Extended Data Area

Note 1. Acceptable when programming area is other than erasure suspending block. Flash sequencer enters "Command Lock" state and detects ILGLERR (not occur ILGCOMERR) when programming area is in suspending block.

Note 2. Acceptable when programming area is other than erasure suspending block or programming suspending area. Acceptable when erasure area is other than programming/erasure suspending block. Flash sequencer enters "Command Lock" state and occur ILGLERR (not occur ILGCOMERR).

Note 3. Available Bank / Data Area is dependent on each FACL. See the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

**Table 6.6 Acceptable FACL Commands for Hardware Property Area**

|                      | FACL0 | FACL2 |
|----------------------|-------|-------|
| Property Programming | O     | X     |
| Property Erasure     | O     | X     |
| Switch Programming   | O     | X     |
| Switch Erasure       | O     | X     |
| TAG Update           | O     | X     |
| TAG Erasure          | O     | X     |

Remarks:

O: Acceptable

X: Not acceptable (enter Command Lock state)

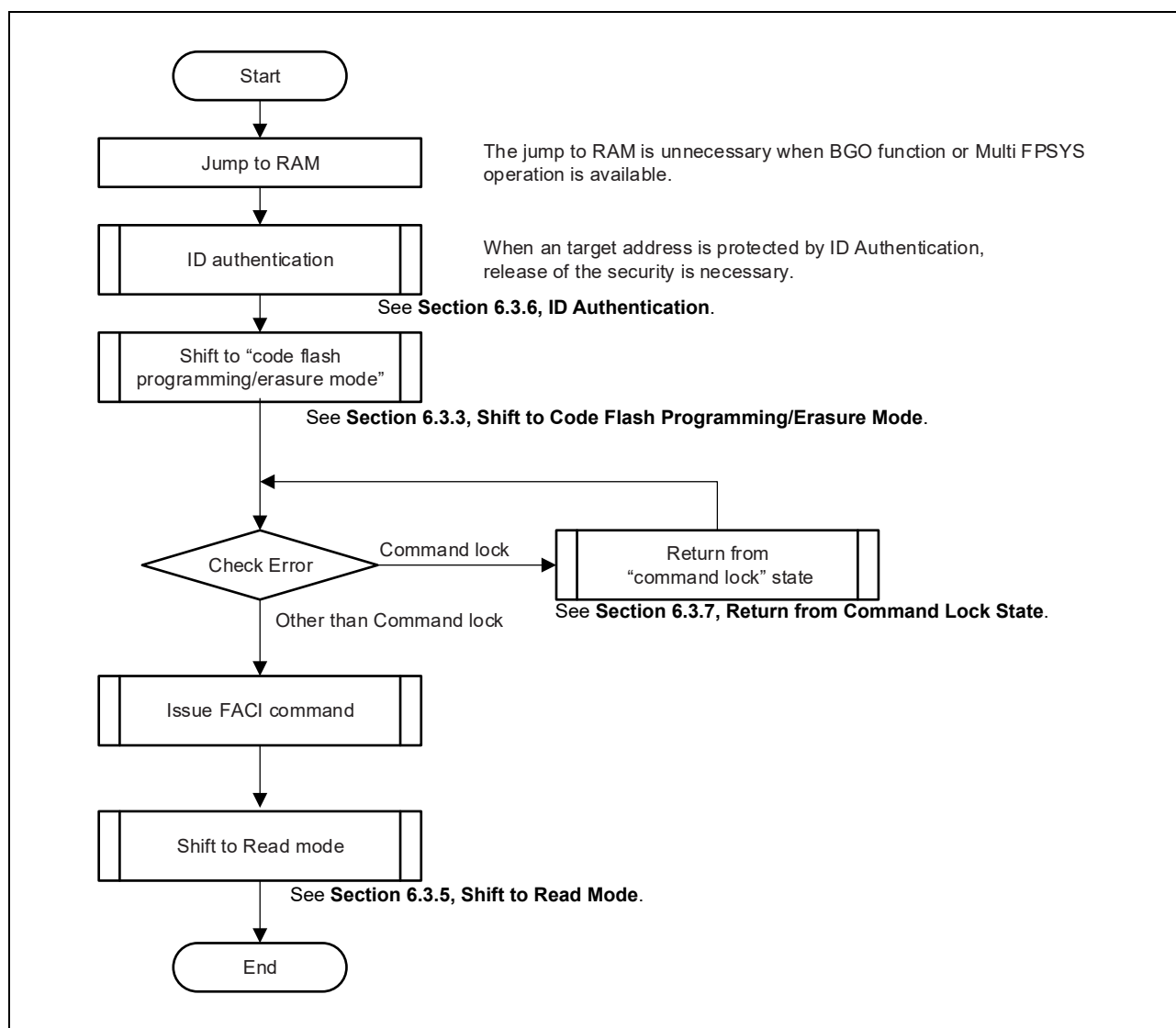
## 6.3 FACL Command

This section describes the overview of FACL command usage.

### 6.3.1 Overview of the Command Usage in Code Flash Programming/Erase Mode

The overview of the FACL command usage in code flash programming/erase mode is shown below.

**Table 6.3** lists the available commands in code flash programming/erase mode.

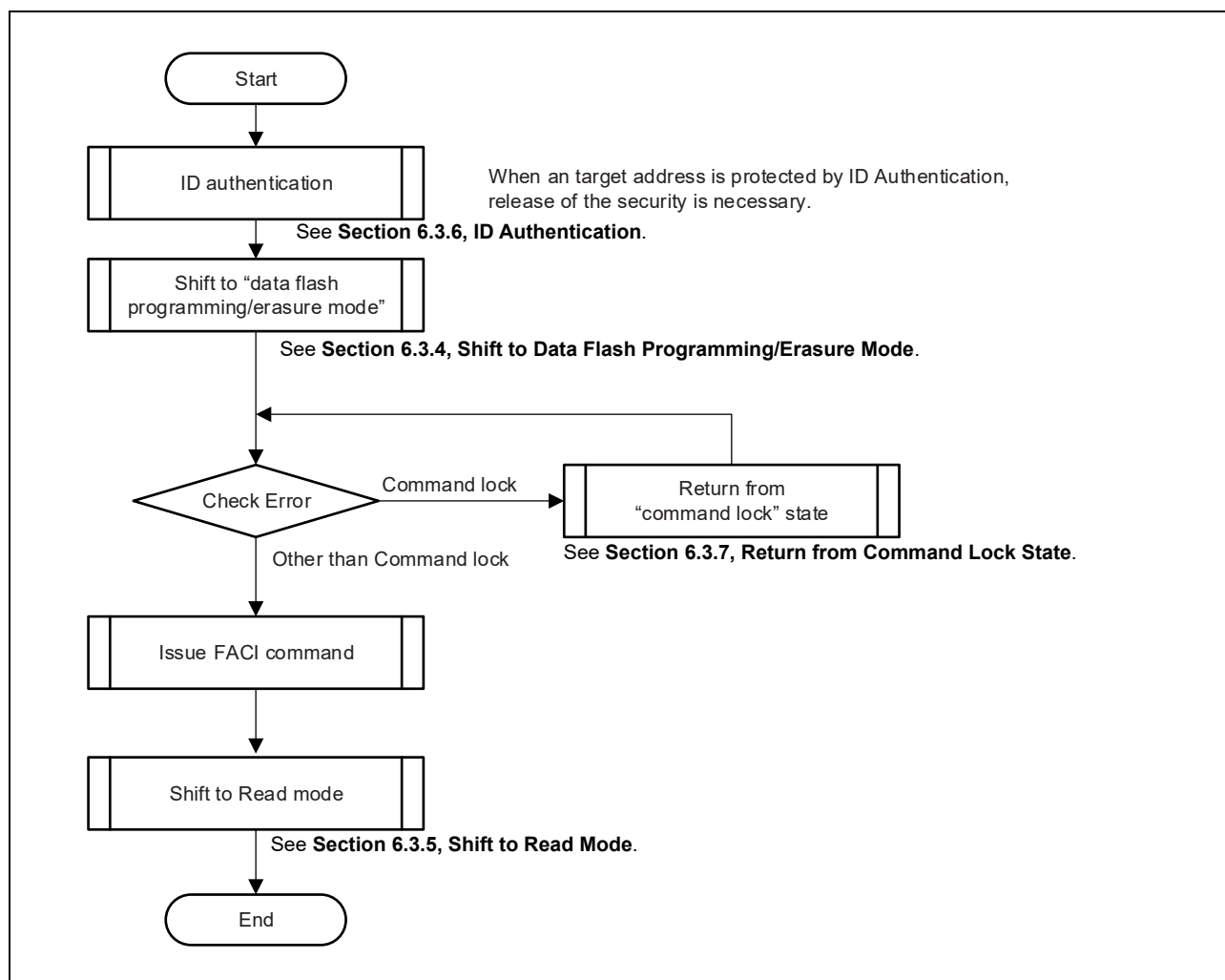


**Figure 6.1** Overview of Command Usage in Code Flash Programming/Erase Mode

### 6.3.2 Overview of the Command Usage in Data Flash Programming/Erase Mode

The overview of the FACL command usage in data flash programming/erase mode is shown below.

**Table 6.3** lists the available commands in data flash programming/erase mode.



**Figure 6.2** Overview of Command Usage in Data Flash Programming/Erase Mode

### 6.3.3 Shift to Code Flash Programming/Erase Mode

To use the FACL commands relating the code flash memory, operation should be shifted to the code flash programming/erase mode. Set the FENTRYRC bit in FENTRYR\_n register to 1 to shift to the code flash programming/erase mode.

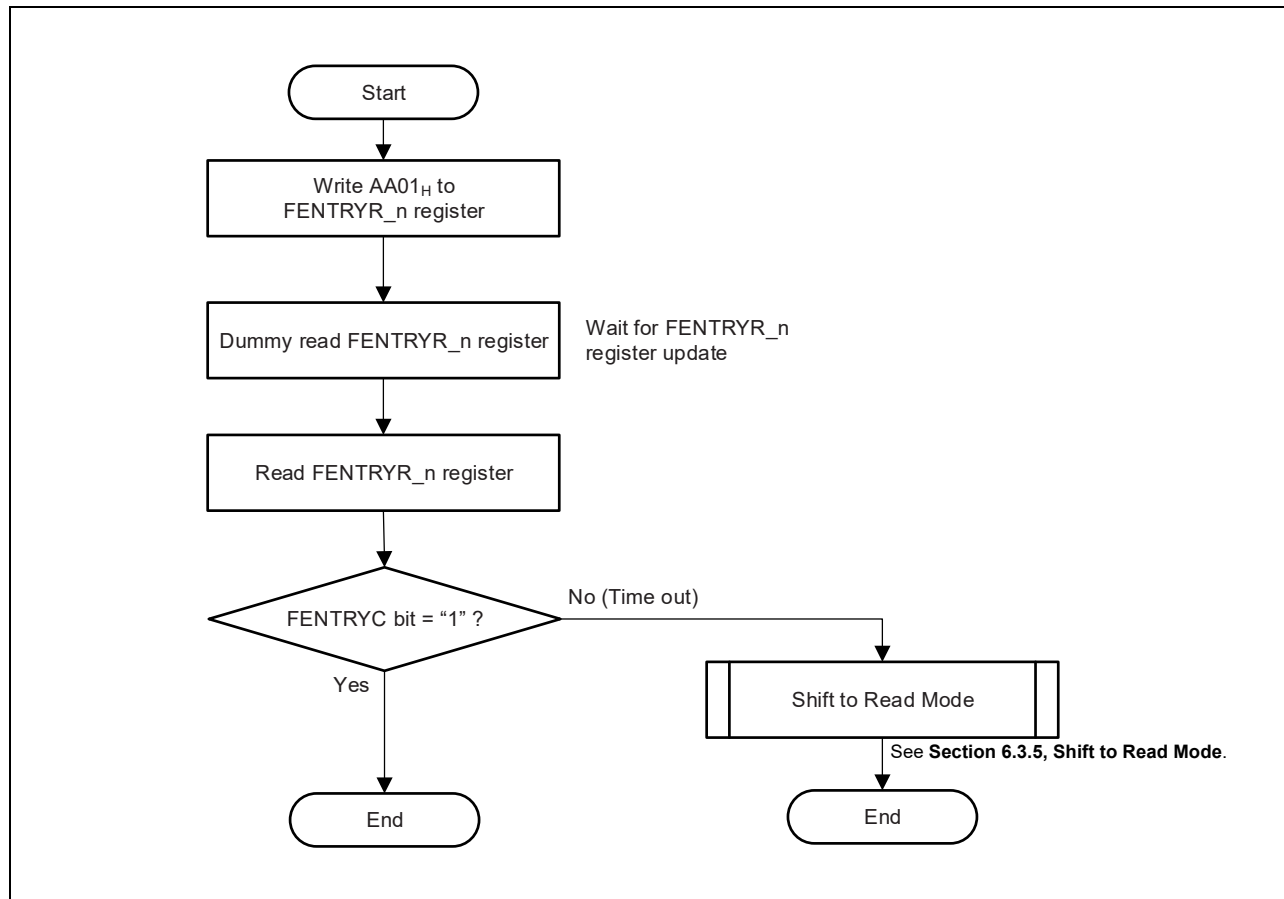


Figure 6.3 Flow of Shift to Code Flash Programming/Erase Mode



### 6.3.4 Shift to Data Flash Programming/Erase Mode

To use the FACL commands relating the data flash memory, operation should be shifted to the data flash programming/erase mode. Set the FENTRYRD bit in FENTRYR\_n register to 1 to shift to the data flash programming/erase mode.

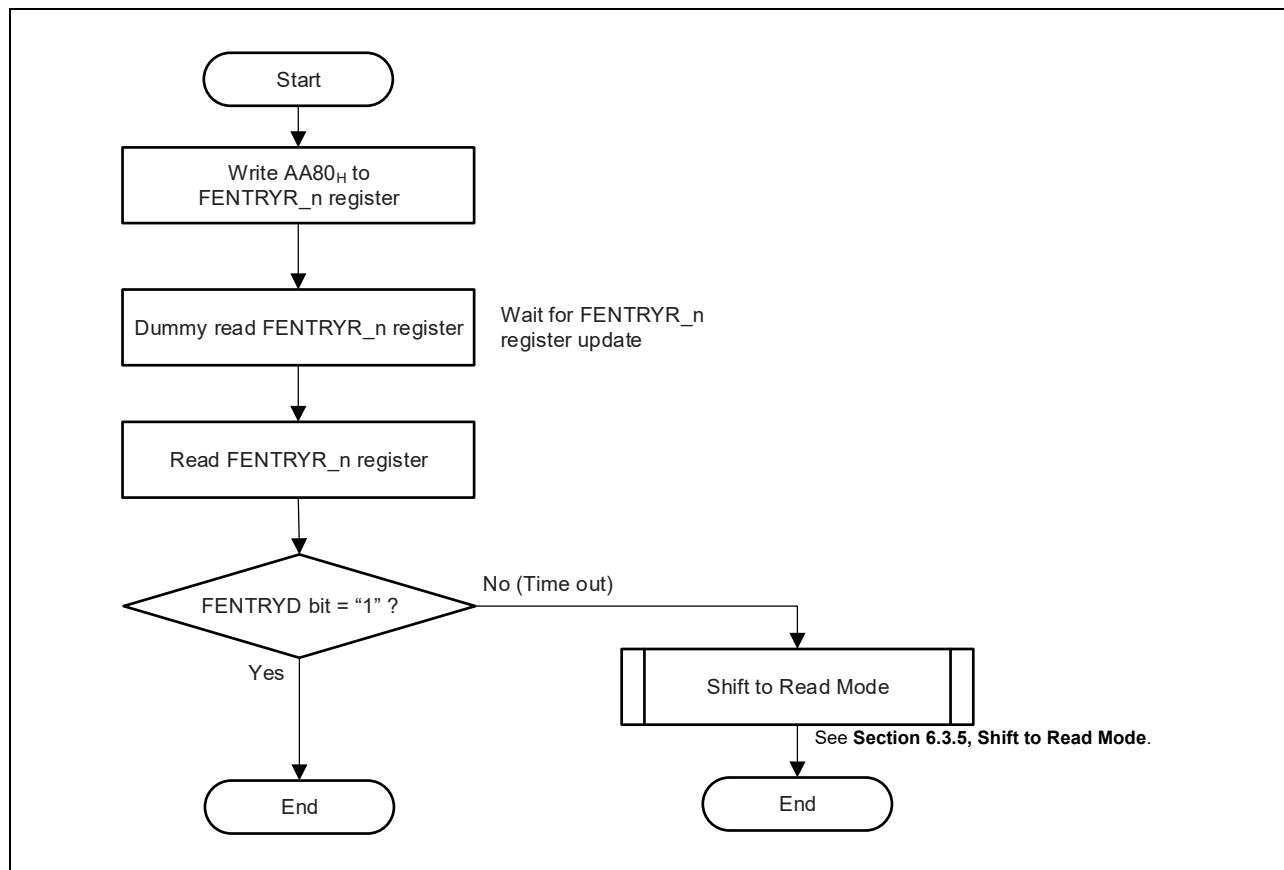


Figure 6.4 Flow of Shift to Data Flash Programming/Erase Mode

### 6.3.5 Shift to Read Mode

To read the flash memory without using the BGO function, the operation should be shifted to the read mode.

To shift to the read mode, set the FENTRYR\_n register to 0000<sub>H</sub>.

When entering the read mode, the flash sequencer processing should be completed and the operation is better to in other than command lock state.

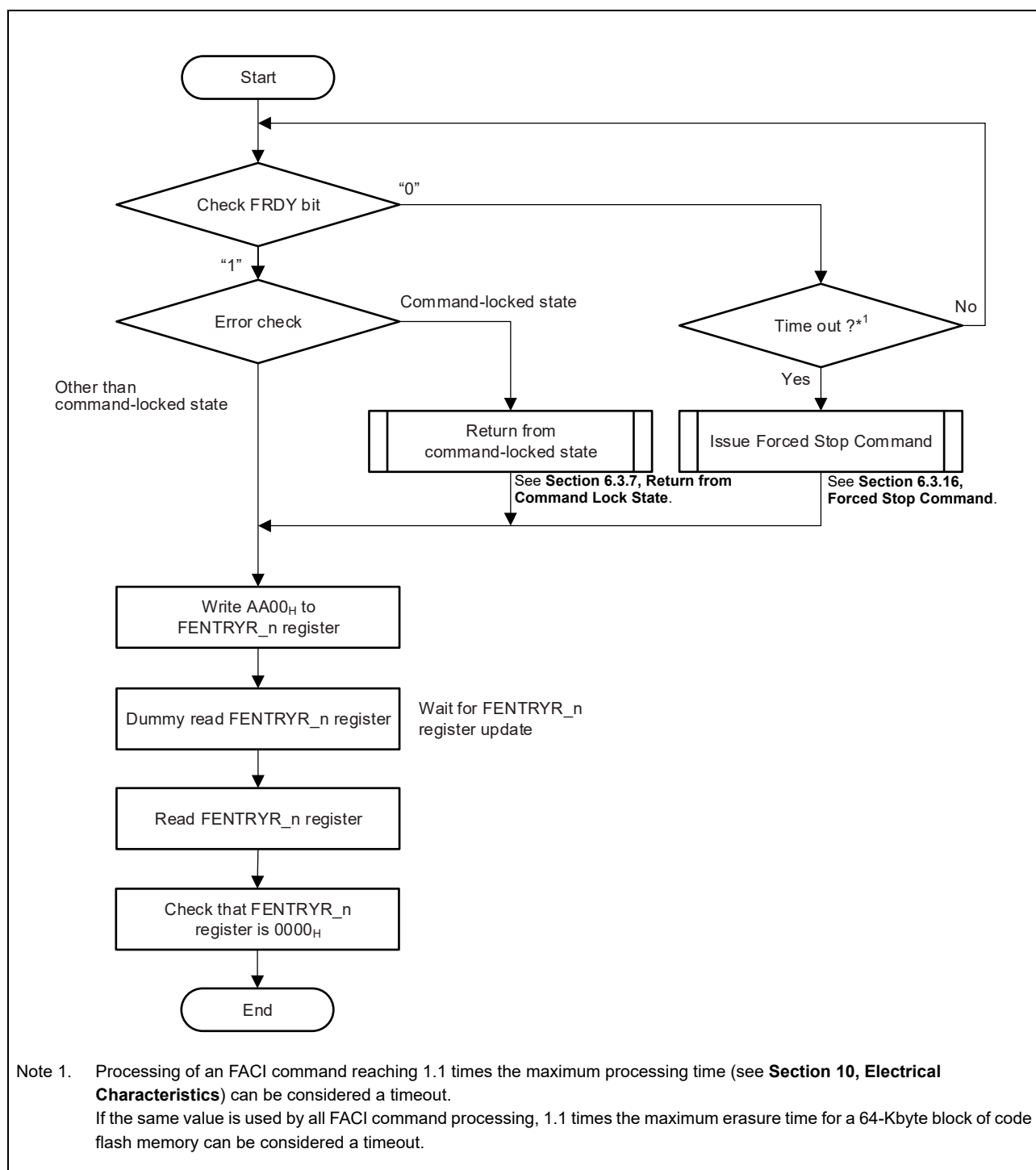
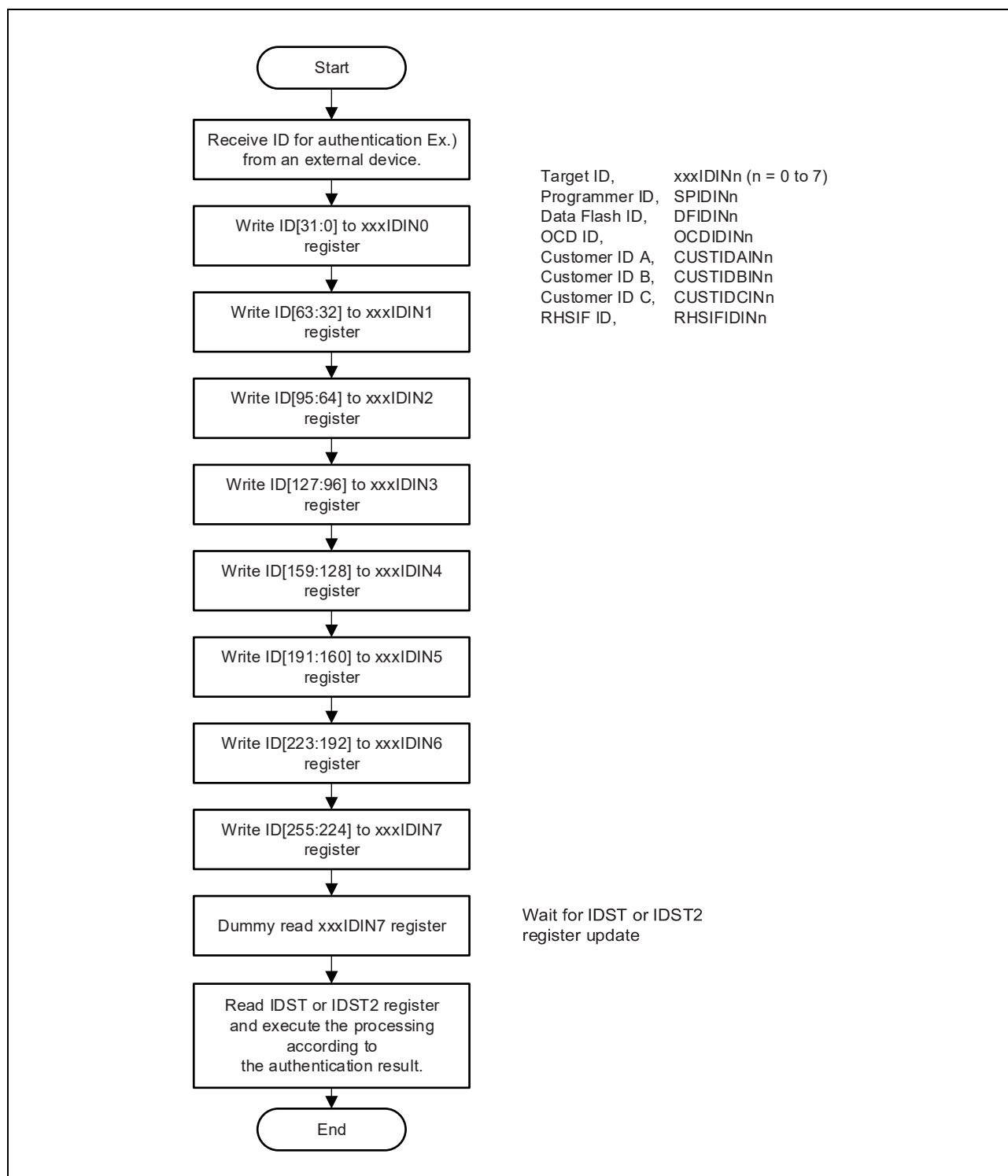


Figure 6.5 Flow of Shift to Read Mode

### 6.3.6 ID Authentication

When a target address is protected by ID Authentication, Release of the security is necessary. For details of Address to protect in each ID, see the *RH850/U2B Group User's Manual: Hardware Section 58, Basic Hardware Protection*.

**Figure 6.6** shows the each ID compare method, and how the compare result is checked by IDST or IDST2.



**Figure 6.6** Flow of ID Authentication

### 6.3.7 Return from Command Lock State

When the flash sequencer enters the command lock state, FACL commands cannot be accepted. To release the command lock state, use the Status Clearing Command, Forced Stop Command, or FASTAT\_n register.

When the command lock state is detected by checking an error before issuing the Programming/Erase Suspension Command, the FRDY bit in the FSTATR\_n register may hold 0 without completing the command processing. If the processing is not completed within the maximum programming/erase time specified by electrical characteristics, it is determined as timeout and the flash sequencer should be stopped by the Forced Stop Command.

The FHVEERR bits in the FSTATR\_n register can be changed from 1 to 0 by the Status Clearing Command. All bits to be the command lock source can be changed from 1 to 0 by the Status Clearing or Forced Stop Command.

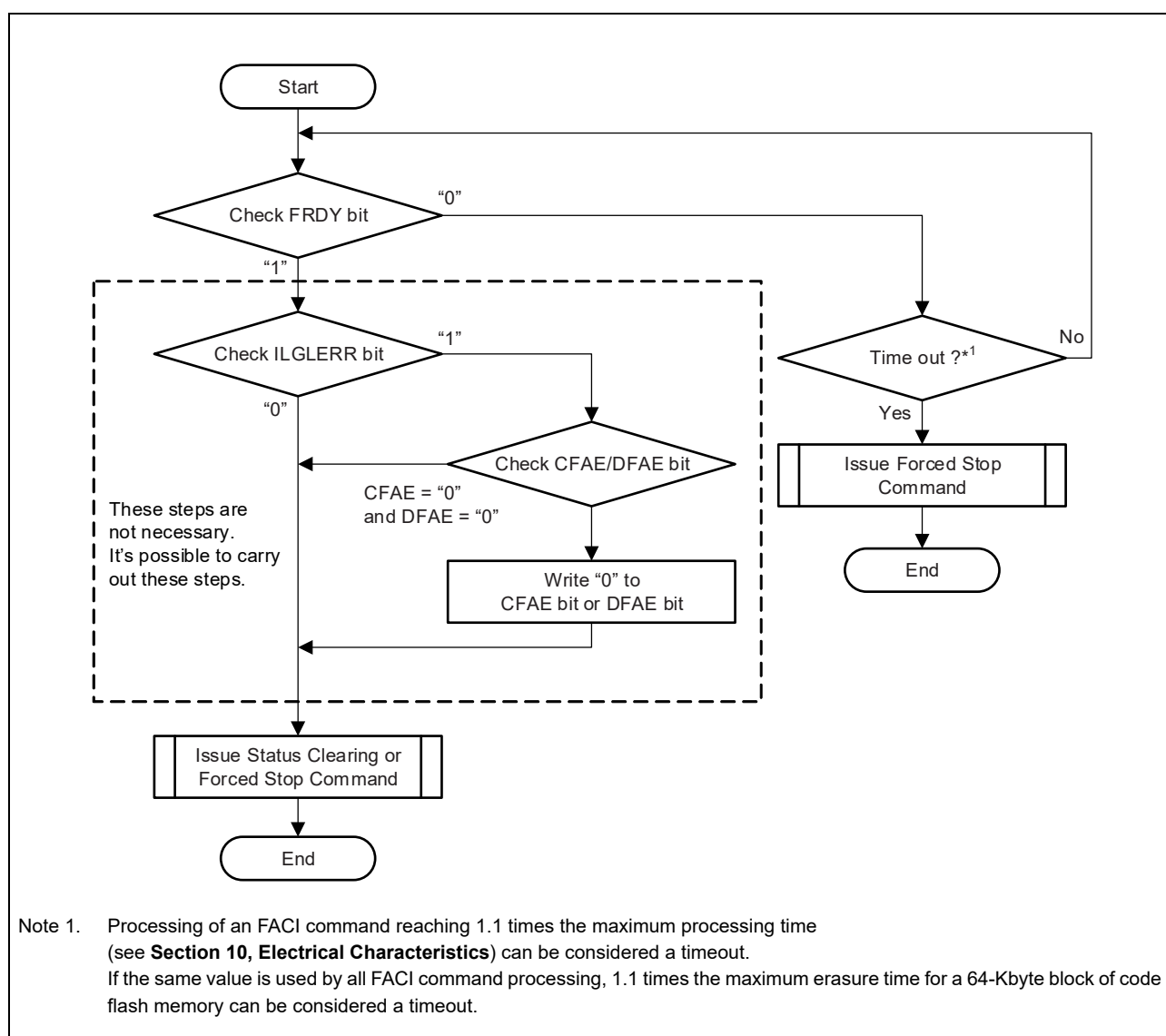


Figure 6.7 Return from Command Lock State

### 6.3.8 Issuing of Programming Command

The programming command is used to write to User Area, User Boot Area, Data Area and Extended Data Area.

Before issuing the programming command, set the start address of target area to the FSADDR\_n register.

Writing D0<sub>H</sub> to the FACI command-issuing area at the final access of the FACI command-issuing starts the programming command processing. If the target area of programming command processing contains the area not for writing, write FFFF FFFF<sub>H</sub> to the corresponding area.

When using the CAN-FD/CAN bootstrap function in serial programming mode, to switch the target area of programming of the code flash memory, the setting of FAREASELC register must be changed. The FAREASELC register must be set before issuing the programming command.

If issuing the programming command is kept while the FACI internal data buffer is full, wait is generated in Peripheral Bus and it may affect the communication performance of other peripheral IPs.

To avoid the wait generation, the DBFULL bit in FSTATR\_n should be 0 when FACI commands are issued. Unlike the previous products, the data buffer never become full in this product.

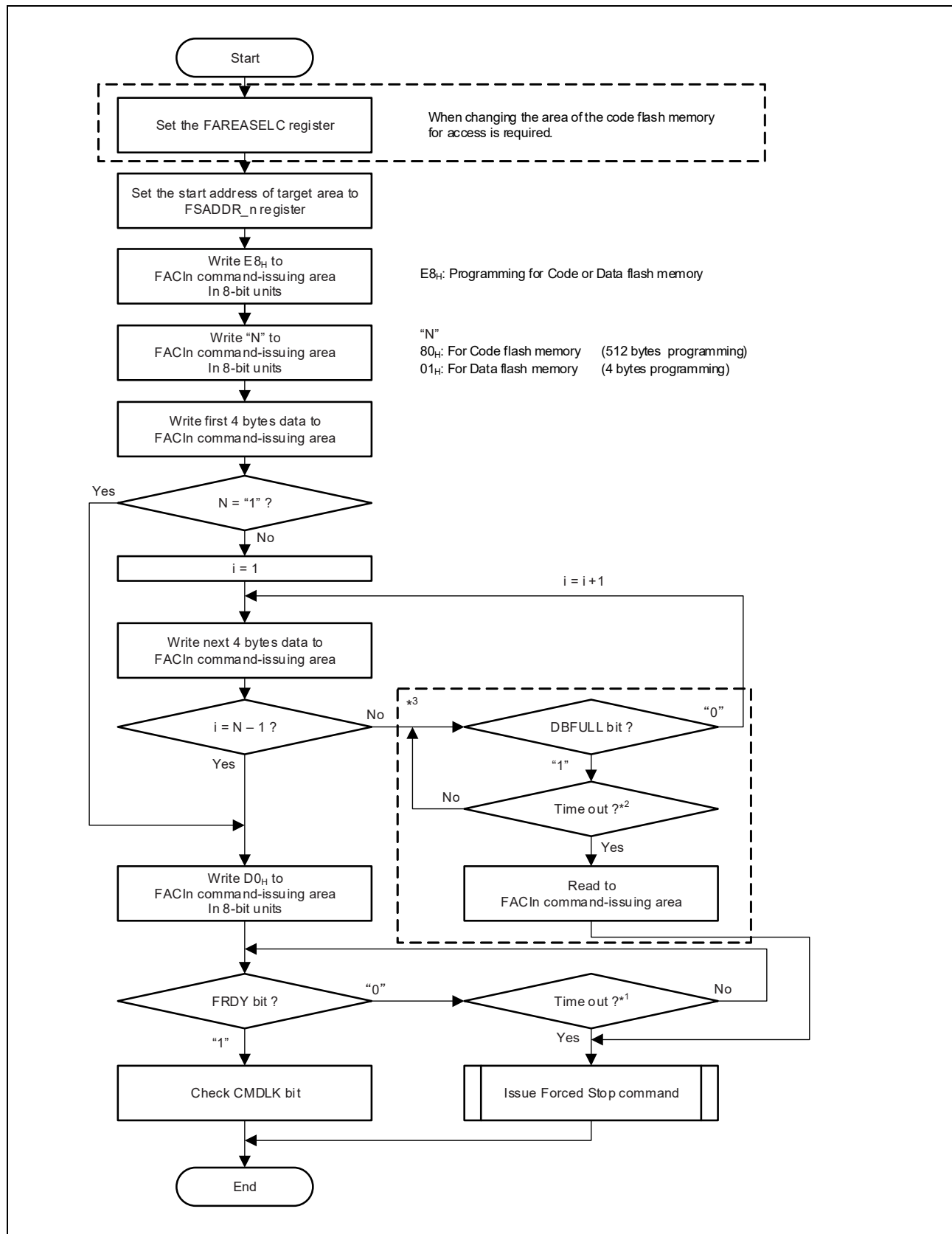


Figure 6.8 Programming Command Usage (1/2)

- Note 1. For the code flash memory, the time of 512-byte programming reaching 1.1 times is judged as a timeout.  
For the data flash memory, the time of 4-byte programming reaching 1.1 times is judged as a timeout.  
For the Extended Data Area, the time of 4-byte programming reaching 1.1 times is judged as a timeout.  
If the same value is used by all FACL command processing, the time of 512-byte programming of the code flash memory reaching 1.1 times is judged as a timeout. (the maximum time: see **Section 10, Electrical Characteristics**)
- Note 2. The time reaching 1 time or 5  $\mu$ s is judged as a timeout.
- Note 3. DBFULL bit never become "1", and these steps are not necessary. It's possible to carry out these steps for the compatibility of the previous product.

**Figure 6.8      Programming Command Usage (2/2)**

### 6.3.9 Issuing of Multi Programming Command

The Multi Programming Command is used to write to Data Area and Extended Data Area.

Before issuing the Multi Programming Command, set the start address of target area to the FSADDR\_n register.

Writing D0<sub>H</sub> to the FACI command-issuing area at the final access of the FACI command-issuing starts the programming command processing. If the target area of programming command processing contains the area not for writing, write FFFF FFFF<sub>H</sub> to the corresponding area.



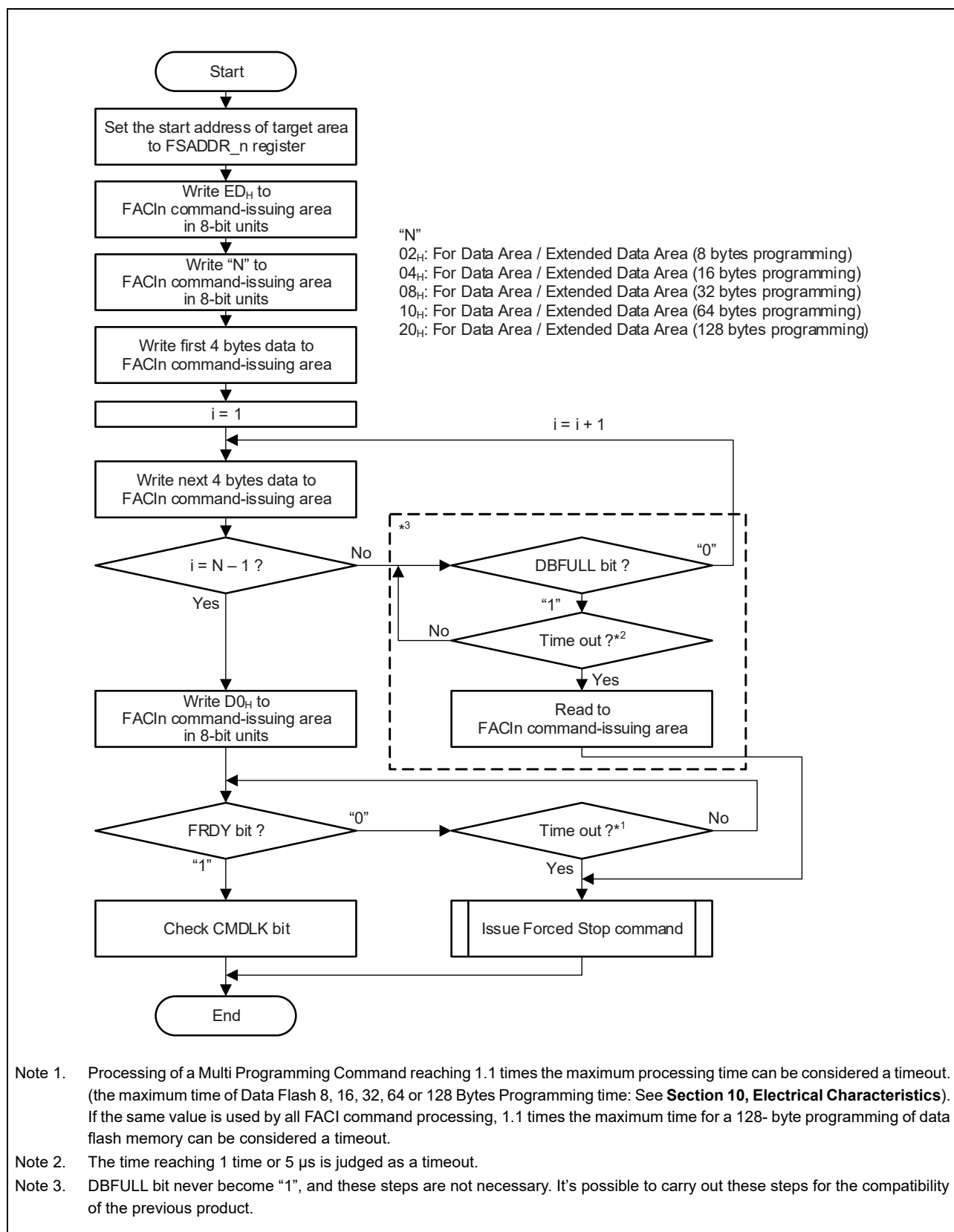


Figure 6.9 Multi Programming Command Usage

- Note 1. Processing of a Multi Programming Command reaching 1.1 times the maximum processing time can be considered a timeout. (the maximum time of Data Flash 8, 16, 32, 64 or 128 Bytes Programming time: See **Section 10, Electrical Characteristics**). If the same value is used by all FACL command processing, 1.1 times the maximum time for a 128-byte programming of data flash memory can be considered a timeout.
- Note 2. The time reaching 1 time or 5  $\mu$ s is judged as a timeout.
- Note 3. DBFULL bit never become "1", and these steps are not necessary. It's possible to carry out these steps for the compatibility of the previous product.

### 6.3.10 DMA Programming Command

The DMA Programming Command is used to program multiple 4-byte data transferred from the DMAC to the Data Area. Thus, a large amount of data can be programmed continuously with reduced CPU load.

Maximum program unit is limited within one Data Area\*<sup>1</sup>. Address boundary between different Data Area cannot be crossed.

Before issuing a DMA Programming Command, set the start address of target area to the FSADDR\_n register. Set the write data in the RAM and set the DMAC to perform DMA transfer from the pertinent area to the FACL command-issuing area. FACL issues a data transfer request to DMAC after reception of the DMA Programming Command and every time the writing of 4 bytes of data is completed. Set up the DMAC so that 4 bytes of data will be transferred for each data transfer request. For the usage of the DMAC, see the *RH850/U2B Group User's Manual: Hardware Section 7, sDMA Controller or Section 8, DTS Controller*.

**Note 1.** For details of one Data Area, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

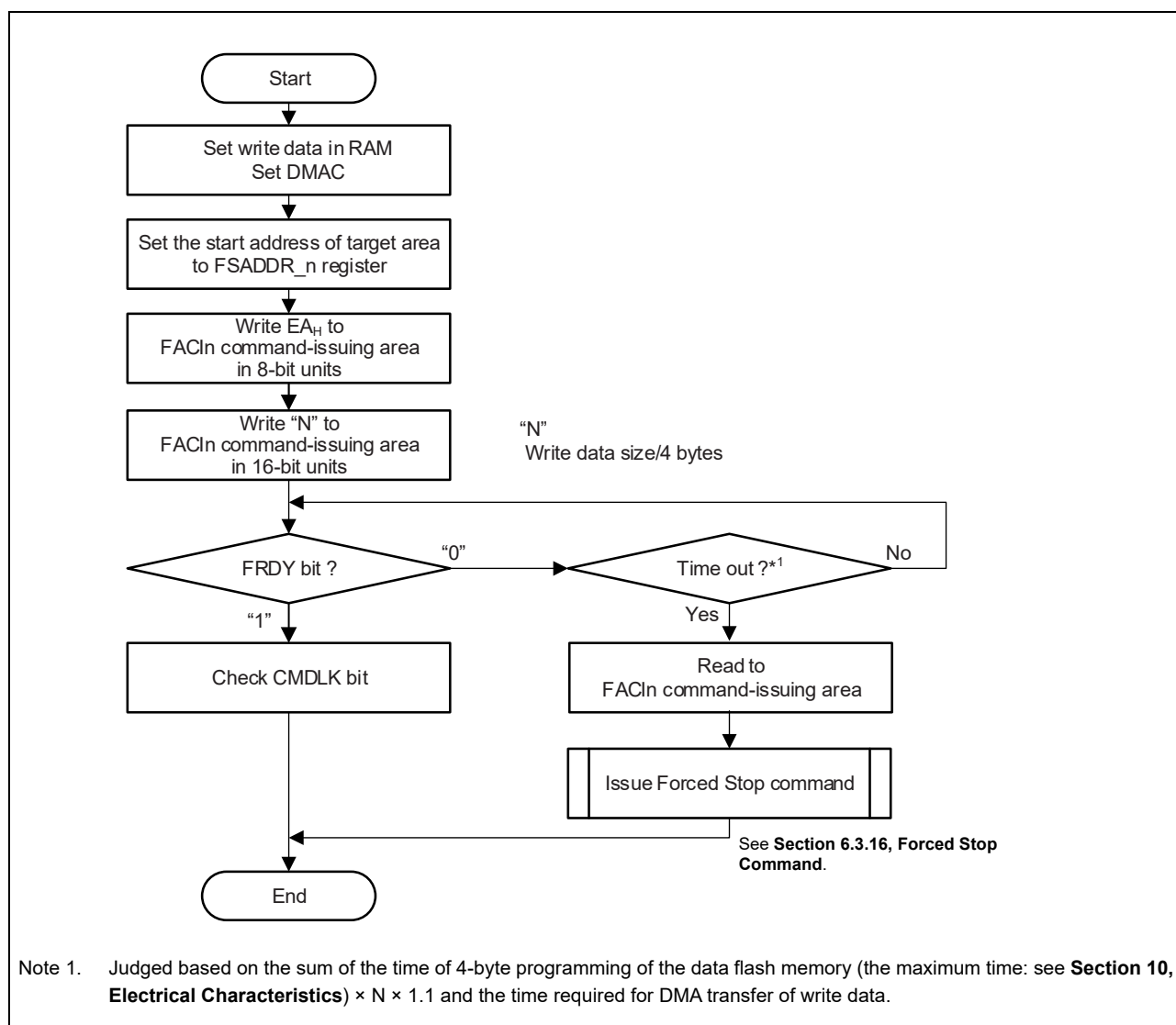
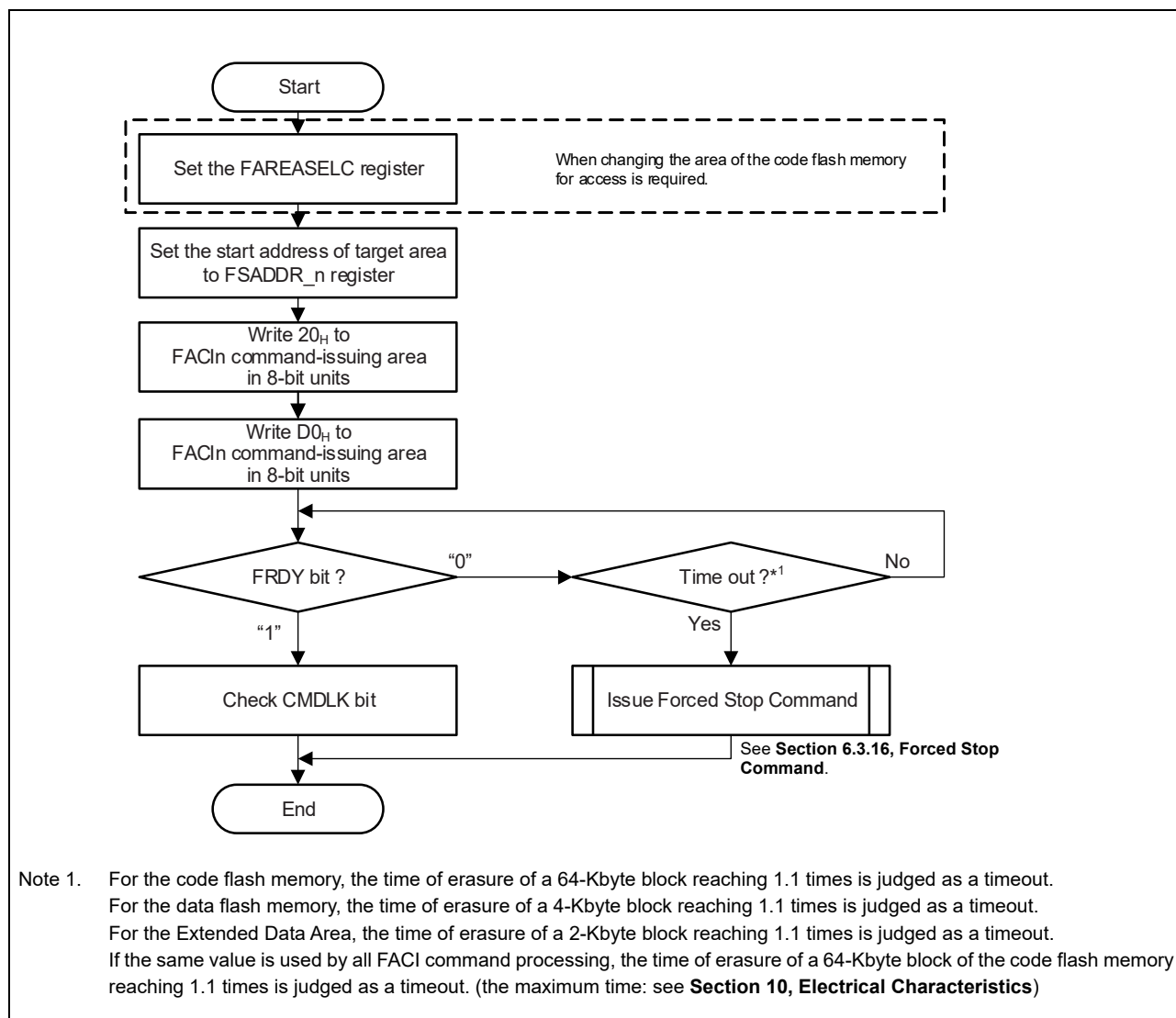


Figure 6.10 DMA Programming Command Usage

### 6.3.11 Block Erasure Command

The Block Erasure Command is used to erase the User Area, User Boot Area, Data Area and Extended Data Area. Before issuing the Block Erasure Command, set the start address of target area to the FSADDR\_n register. Writing 20<sub>H</sub> and D0<sub>H</sub> to the FACL command-issuing area starts the Block Erasure Command processing.

When using the CAN-FD/CAN bootstrap function in serial programming mode, to switch the target area of erasure of the code flash memory, the setting of FAREASELC register must be changed. The FAREASELC register must be set before issuing the block erasure command.



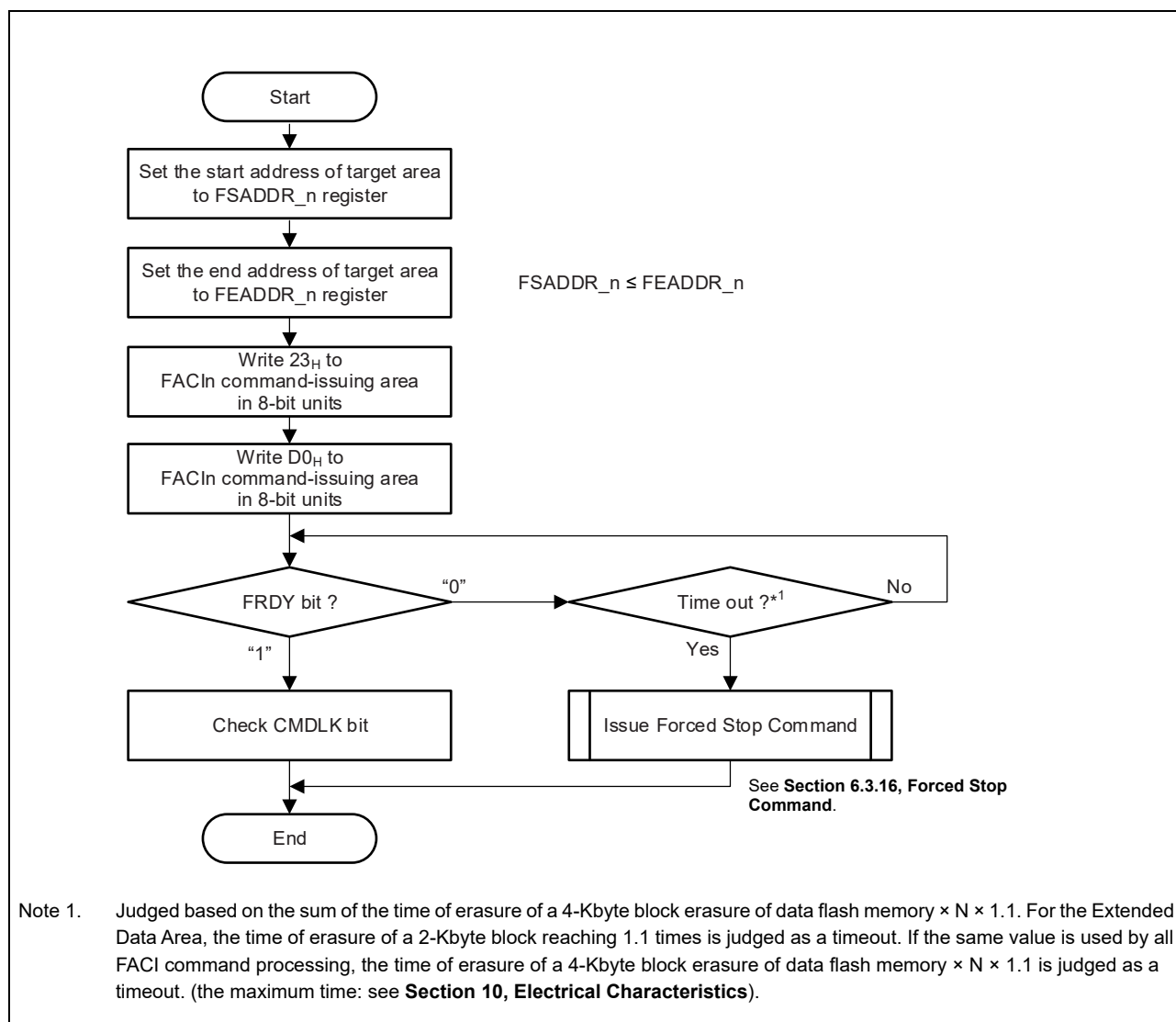
**Figure 6.11 Block Erasure Command Usage**

### 6.3.12 Area Erasure Command

The Area Erasure Command is used to erase the Data Area and Extended Data Area. Erase unit are N blocks ( $4K \times N$  Bytes) or one block ( $2K \times 1$  Bytes) for Extended Data Area. Before issuing the Area Erasure Command, set the first address of target area to the FSADDR\_n register and set the end address of target area to FEADDR\_n register. Writing 23<sub>H</sub> and D0<sub>H</sub> to the FACL command-issuing area starts the Area Erasure Command processing. Area Erasure Command only performs within one Data Area<sup>\*1</sup>. If setting of FSADDR\_n and FEADDR\_n are inconsistent, FACL detects error and flash sequencer enters command lock state.

Address boundary between different Data Area cannot be crossed.

**Note 1.** For details of Data Area, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.



**Figure 6.12** Area Erasure Command Usage

### 6.3.13 Programming/Erasure Suspension Command

The Programming/Erasure Suspension Command is used for suspending Programming, Multi Programming, Block Erasure or Area Erasure Command processing. Before issuing the Programming/Erasure Suspension Command, check that CMDLK bit is 0 to ensure that Programming, Multi Programming, Block Erasure or Area Erasure Command processing is being performed correctly. In addition, check that the SUSRDY bit is 1 to ensure that the Programming/Erasure Suspension Command is acceptable. After issuing the Programming/Erasure Suspension Command, check CMDLK bit to ensure no error has occurred.

If an error has occurred, the CMDLK bit is set to 1. If Programming, Multi Programming, Block Erasure or Area Erasure Command processing is complete within the period from when the SUSRDY bit is ensured to be 1 until the Programming/Erasure Suspension Command is accepted, no error occurs, hence no transition to a suspended state (the FRDY bit is 1 and both ERSSPD and PRGSPD bits are 0).

Once the Programming/Erasure Suspension Command is accepted and Programming, Multi Programming, Block Erasure or Area Erasure Command processing is normally suspended, flash sequencer enters a suspended state and that FRDY bit is 1 and ERSSPD or PRGSPD bit is 1. After issuing the Programming/Erasure Suspension Command and ensuring that flash sequencer has entered suspend state, determine which operation to perform in the succeeding process. If the Programming/Erasure Resumption Command is issued in the succeeding process while flash sequencer has not entered a suspended state, an illegal command error occurs and flash sequencer shifts to the command lock state (see **Section 8.2, Error Protection**).

When the operation shifts to the programming/erasure suspend state, programming/erasure is enabled for certain areas in code flash/data flash. (See **Table 6.5** in **Section 6.2, Relationship between Flash Sequencer Status and FACL Commands**.) In addition, when the FENTRYR\_n register is cleared, the operation shifts to the read mode.

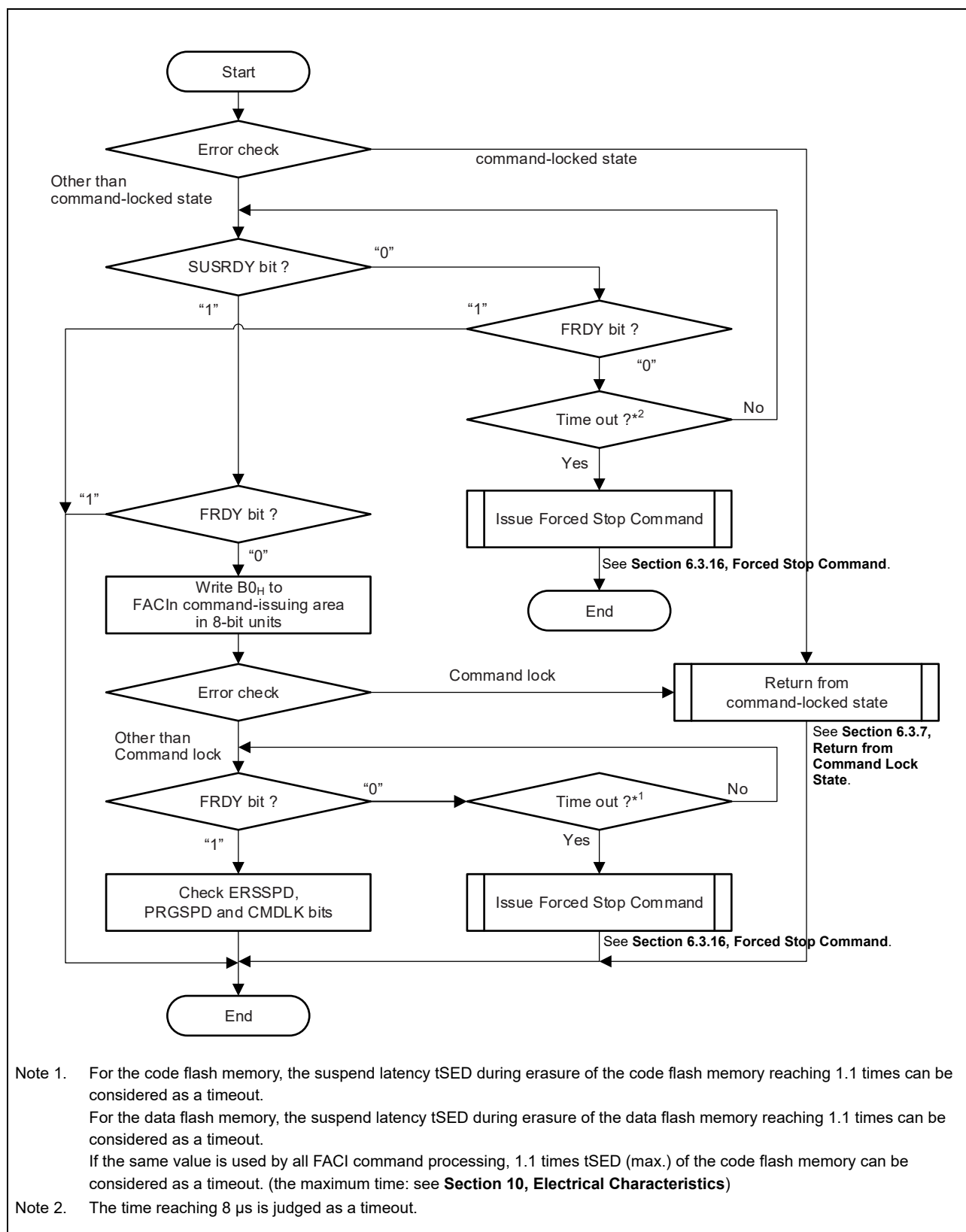
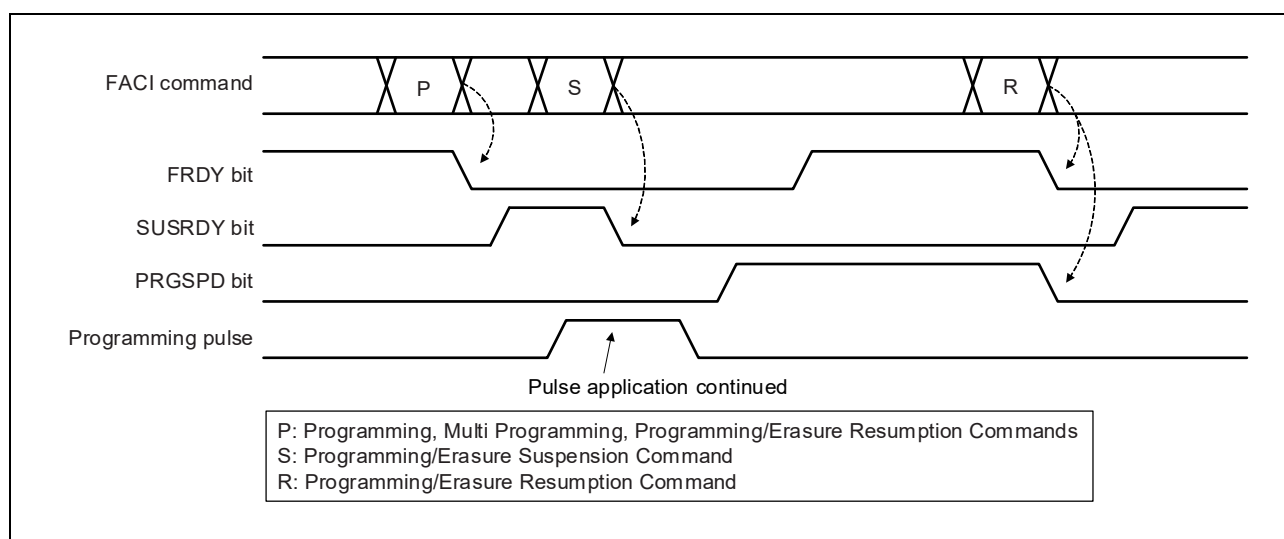


Figure 6.13 Programming/Erasure Suspension Command Usage

### (1) Suspend programming command

If a Programming/Erasure Suspension Command is issued while the flash memory is being programmed, the flash sequencer suspends programming. **Figure 6.14** shows the suspending operation. Once the flash sequencer enters a state where it is ready to accept the Programming/Erasure Suspension Command after the start of programming, SUSRDY bit is set to 1. If the Programming/Erasure Suspension Command is issued, the flash sequencer accepts the command and clears SUSRDY bit. If the flash sequencer accepts the command while applying a programming pulse, the flash sequencer continues applying the pulse. After a specified pulse application time has elapsed, the flash sequencer completes applying the pulse, suspends programming, and sets the FSTATR\_n.PRGSPD bit to 1. Once the suspension process is complete, the flash sequencer sets the FRDY bit to 1 and enters programming suspended state. If the flash sequencer accepts the Programming/Erasure Resumption Command in this state, the flash sequencer clears the FRDY and PRGSPD bits to 0 and restarts programming.

**Figure 6.14** gives an overview of operation for suspending programming command processing. Upon accepting programming command, the flash sequencer clears FRDY bit to 0 and starts programming.

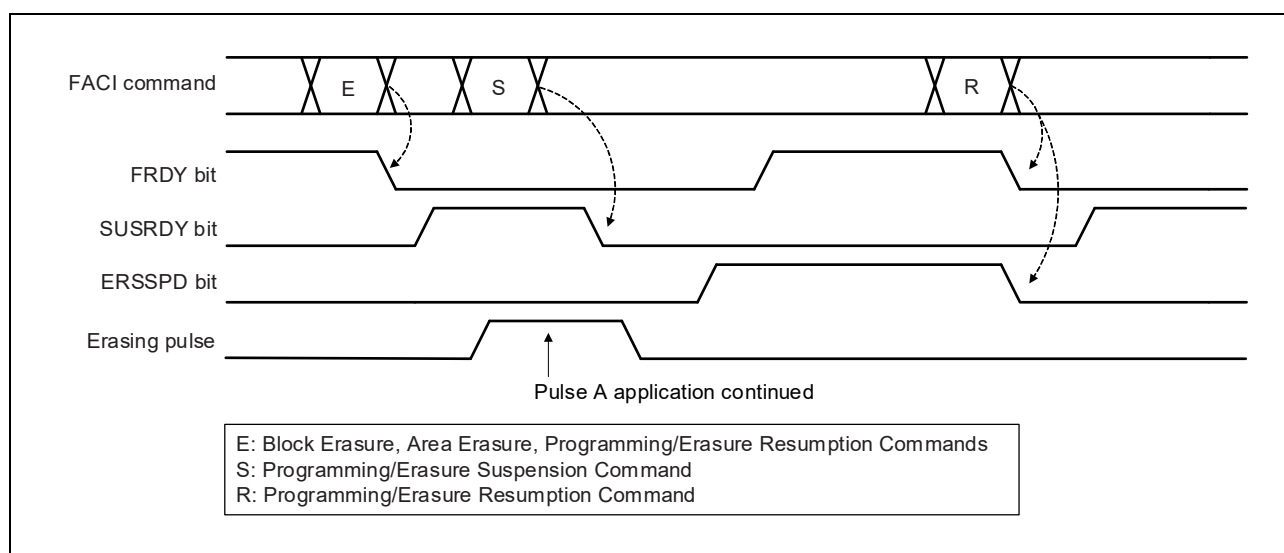


**Figure 6.14 Suspend Programming Command**

## (2) Suspend erase command

If a Programming/Erase Suspension Command is issued while the flash memory is being erased, the flash sequencer suspends erasure. **Figure 6.15** shows the suspending operation. Once the flash sequencer enters a state where it is ready to accept the Programming/Erase Suspension Command after the start of erasing, SUSRDY bit is set to 1. If the Programming/Erase Suspension Command is issued, the flash sequencer accepts the command and clears SUSRDY bit. If the flash sequencer accepts the command while applying a erasing pulse, the flash sequencer continues applying the pulse. After a specified pulse application time has elapsed, the flash sequencer completes applying the pulse, suspends erasing, and sets the FSTATR\_n.ERSSPD bit to 1. Once the suspension process is complete, the flash sequencer sets the FRDY bit to 1 and enters erasing suspended state. If the flash sequencer accepts the Programming/Erase Resumption Command in this state, the flash sequencer clears the FRDY and ERSSPD bits to 0 and restarts erasing.

**Figure 6.15** gives an overview of operation for suspending Block Erasure or Area Erasure command processing. Upon accepting Block Erasure or Area Erasure command, the flash sequencer clears FRDY bit to 0 and starts erasing.



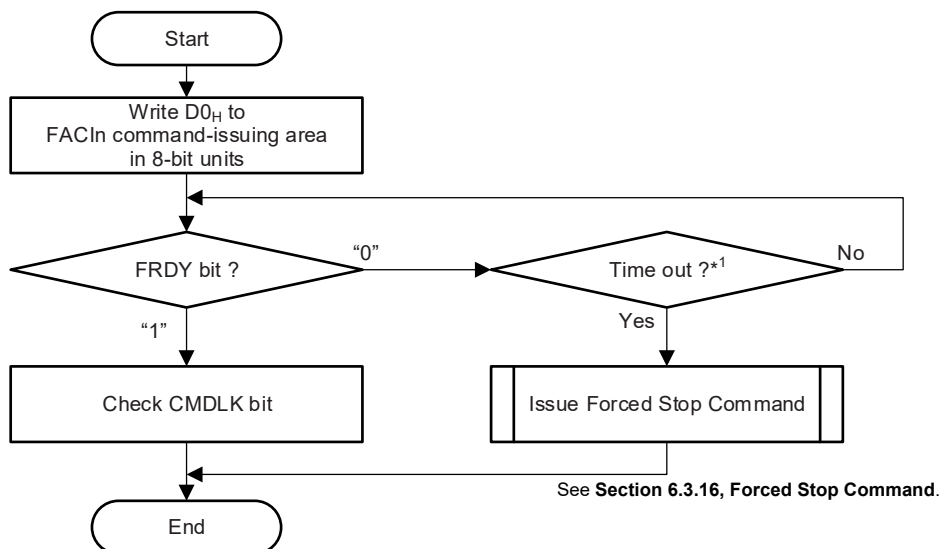
**Figure 6.15 Suspend Erase Command**



### 6.3.14 Programming/Erasure Resumption Command

The Programming/Erasure Resumption Command is used for resuming a Programming, Multi Programming, Block Erasure or Area Erasure Command processing that has been suspended.

If the FENTRYR\_n setting has been modified during suspension, issue a Programming/Erasure Resumption Command only after resetting FENTRYR\_n to the previous value that was held before the Programming/Erasure Suspension Command was issued.



Note 1. Processing of an FACL command reaching 1.1 times the maximum processing time can be considered a timeout. (the maximum time: see **Section 10, Electrical Characteristics**)  
If the same value is used by all FACL command processing, 1.1 times the maximum erasure time for a 64-Kbyte block of code flash memory can be considered a timeout.

**Figure 6.16 Programming/Erasure Resumption Command Usage**

### 6.3.15 Status Clearing Command

The Status Clearing Command is used to clear the command lock state. (See **Section 6.3.7, Return from Command Lock State.**)

The specific bits in FASTAT\_n and FSTATR\_n register can be cleared by the status clearing command.

See **Section 4.8, FASTAT\_n — Flash Access Status Register (n = 0, 2)** and **Section 4.14, FSTATR\_n — Flash Status Register (n = 0, 2)** for register bits cleared by the status clearing command.

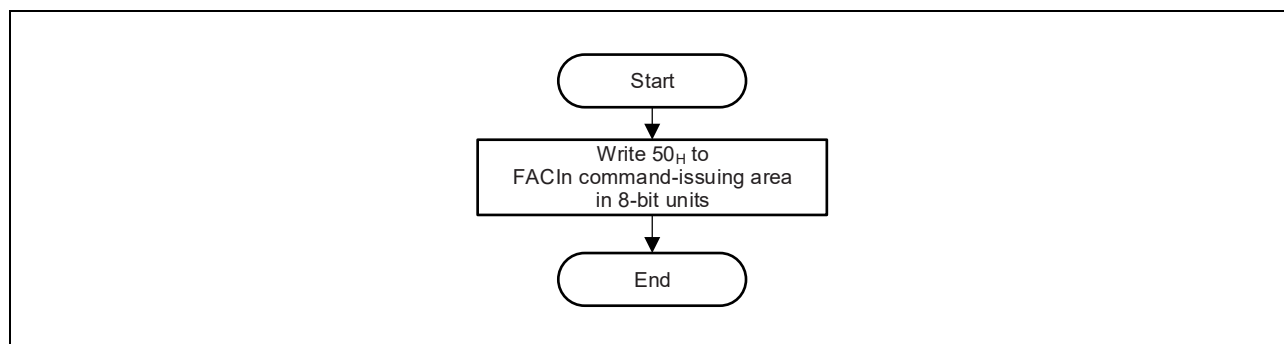
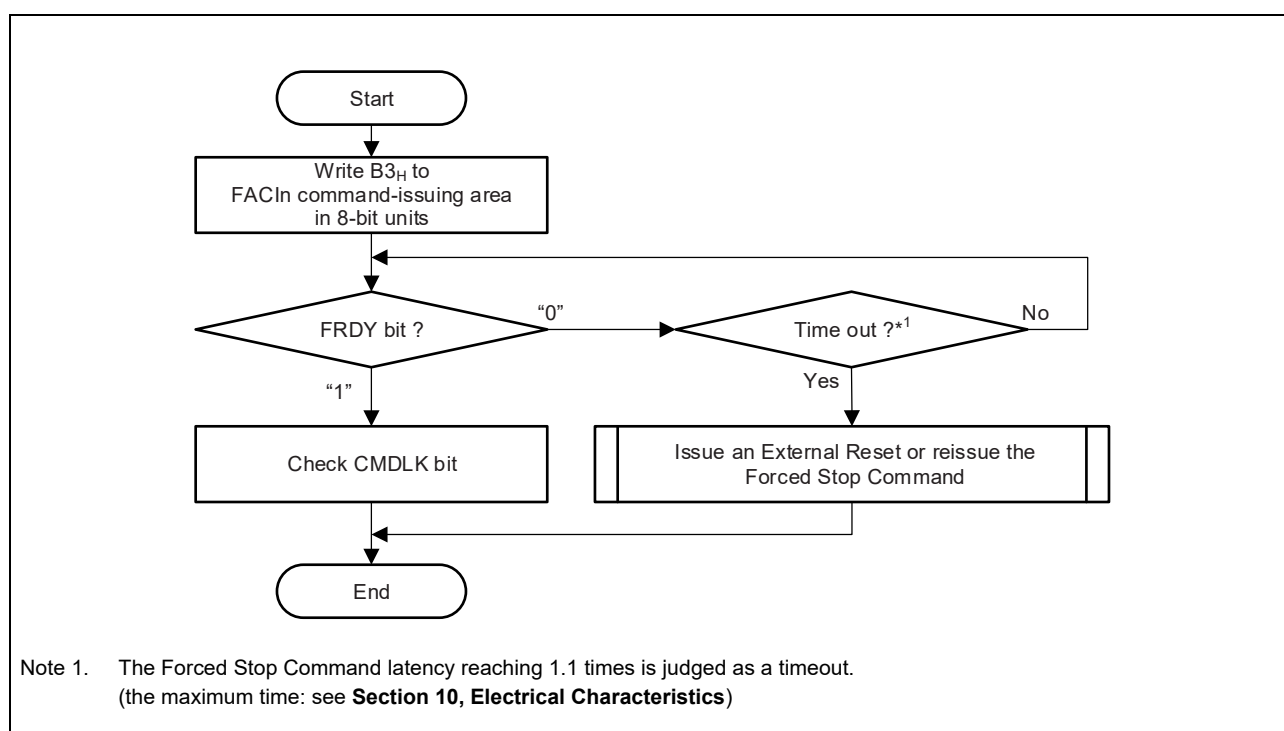


Figure 6.17 Status Clearing Command Usage

### 6.3.16 Forced Stop Command

The Forced Stop Command is used to abort the command being processed by the flash sequencer. While the processing speed of this command is faster than that of Programming/Erase Suspension Commands, it does not guarantee any result of the stopped command operation such as data in programmed or erased area. Furthermore, it is not possible to resume the suspended processing later. The suspended programming or erasure is counted as one from the perspective of programming endurance.

When a Forced Stop Command is issued, the whole FCU and a part of FACLn are initialized as well as the FASTAT\_n and FSTATR\_n registers. This enables Forced Stop Command in recovery from a command lock state or during handling of timeout in flash sequencer operation mode. (See **Section 6.3.7, Return from Command Lock State.**)



**Figure 6.18 Forced Stop Command Usage**

#### Issuing the Forced Stop Command while Another Command is Being Issued

If the Forced Stop Command is used to suspend processing when a timeout of the programming command occurs when checking the DBFULL bit, when a timeout for injection of ECC errors occurs when checking the DBFULL or EBFULL bit, or when a timeout of the DMA Programming Command occurs, writing to the FACL command issuing area may be handled as writing of data by the programming command. If this is the case, read the FACL command issuing area to intentionally lock commands and issue the Forced Stop Command by following the procedure for returning from the command-locked state.

Locking commands is possible in any case where the unit for reading the FACL command issuing area is 8, 16, or 32 bits.

### 6.3.17 Blank Checking Command

The values of the code/data flash memory in which no data are programmed after erasure (non-programmed state) are all-bits = "1". Reading from an area of non-programmed state can lead the detection of an ECC error and address parity error (code flash memory). Thus, the Blank Checking Command is available to confirm the non-programmed state of Data Area. In addition, this product has Blank Check Area<sup>\*1</sup>. The reading of this area is available to confirm the non-programmed state of code/data flash memory. For details of the blank checking of code flash, see the *RH850/U2B Group User's Manual: Hardware Section 63 Flash Memory*.

Before issuing the Blank Checking Command, set addressing mode, start address, and end address to FBCCNT\_n, FSADDR\_n, and FEADDR\_n registers, respectively. When blank check addressing mode is set to incremental mode (i.e. FBCCNT\_n.BCDIR = 0), address specified in FSADDR\_n should be equal to or smaller than address in FEADDR\_n. Conversely, address in FSADDR\_n should be equal to or larger than address in FEADDR\_n when blank check addressing mode is set to decremental mode (i.e. FBCCNT\_n.BCDIR = 1).

If setting of BCDIR, FSADDR\_n, and FEADDR\_n are inconsistent, FACI detects error and flash sequencer enters command lock state. (See **Section 8.2, Error Protection**).

Blank Checking Command only performs within one Data Area<sup>\*1</sup>. Then, make sure to set blank check area not crossing boundary of different Data Area.

Write 71<sub>H</sub> and D0<sub>H</sub> to the FACI command-issuing area to start Blank Checking Command processing.

Completion of command processing can be confirmed by FRDY bit of FSTATR\_n register. At the end of processing, the result of blank checking is stored in the BCST bit in the FBCSTAT\_n register. If non-blank data exists within blank checked area, flash sequencer stops blank checking operation. In this case, address of non-blank data is indicated to FPSADDR\_n register.

The erasure state can be checked by this command only for an area for which erasure processing has been correctly completed. If erasure is not correctly completed (for example, due to a reset input or power shutdown), the erasure state cannot be checked by this command.

**Note 1.** For details of Data Area and Blank Check Area, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

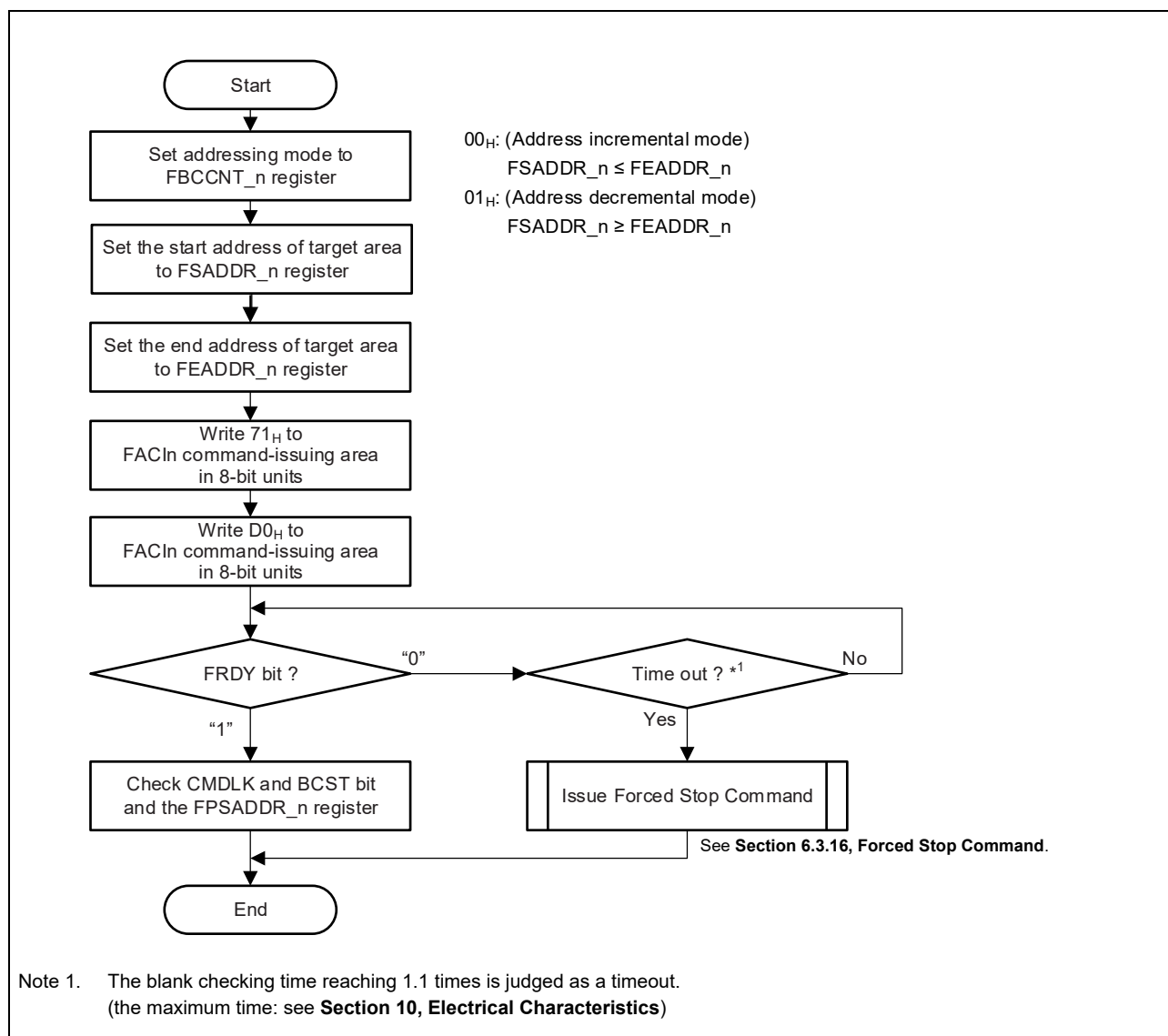


Figure 6.19 Blank Checking Command Usage

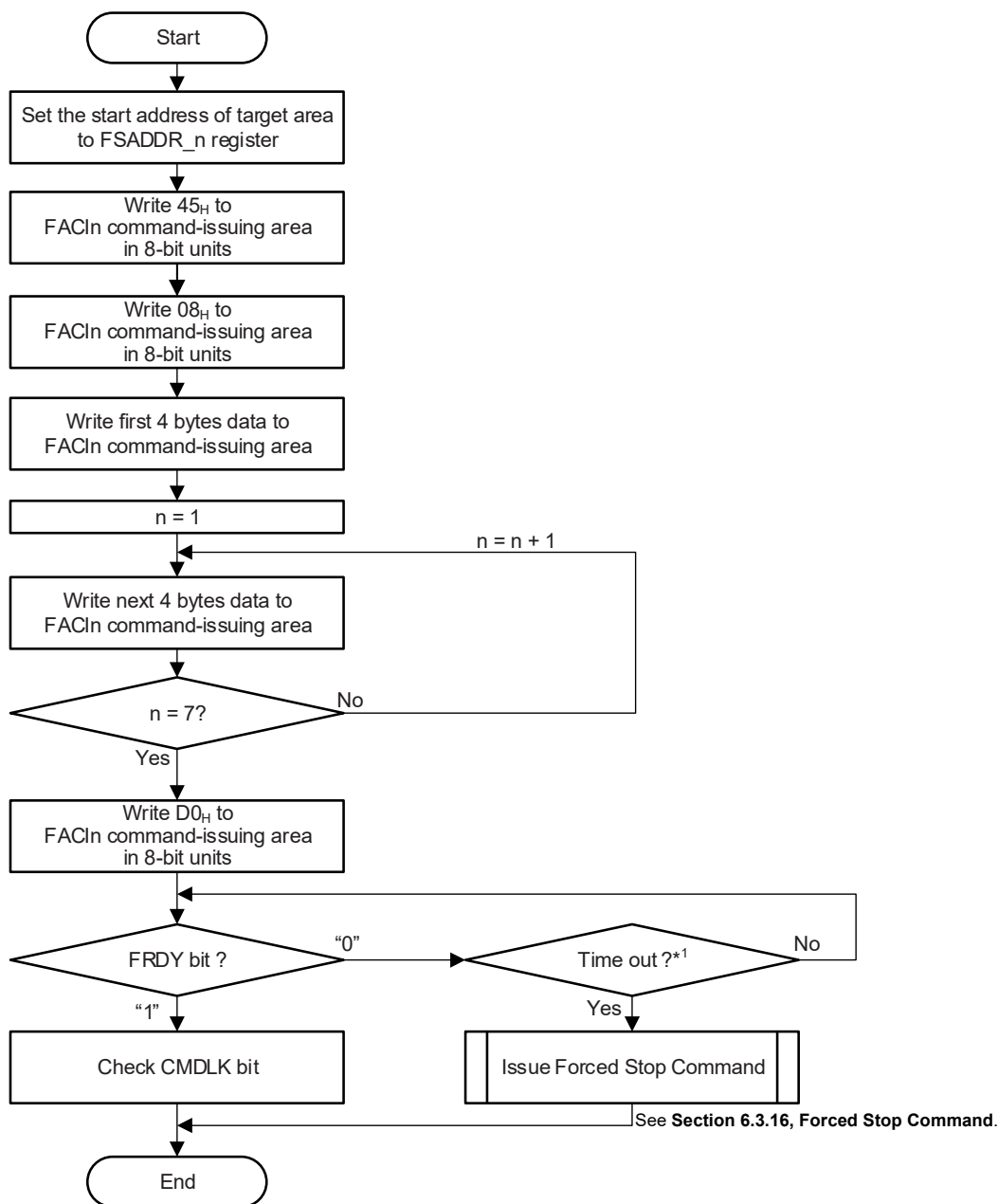
### 6.3.18 Property Programming Command

The Property Programming Command is used to write to the back side (invalid) of Configuration Setting Area (Flash Option Byte, OTP settings for each 4 bytes of Configuration Setting Area), Security Setting Area (Each ID's, Each ID's related Option Byte, OTP settings for each 4bytes of Security Setting Area), and Block Protection Area (OTP Settings for each block in User Area and User Boot Area, Block Protection Settings, etc.).

The Property Programming Command can write data by 32-byte unit.

Before issuing the Property Programming Command set the start address of target area to the FSADDR\_n register.

Writing D0<sub>H</sub> to the FACL command-issuing area at the final access of the FACL command issue starts the Property Programming Command processing.



Note 1. The property Programming maximum time of reaching 1.1 times is judged as a timeout.  
(the maximum time: see **Section 10, Electrical Characteristics**)

**Figure 6.20 Property Programming Command Usage**

### 6.3.19 Property Erasure Command

The Property Erasure Command is used to erase the back side (invalid) of Configuration Setting Area (Flash Option Byte, OTP settings for each 4 bytes of Configuration Setting Area), Security Setting Area (Each ID's, Each ID's related Option Byte, OTP settings for each 4bytes of Security Setting Area), and Block Protection Area (OTP Settings for each block in User Area and User Boot Area, Block Protection Settings, etc.). Before issuing the Property Erasure Command, set the start address of target area to the FSADDR\_n register. Writing 47<sub>H</sub> and D0<sub>H</sub> to the FACLn command-issuing area in 8-bit units starts the Property Erasure Command processing.

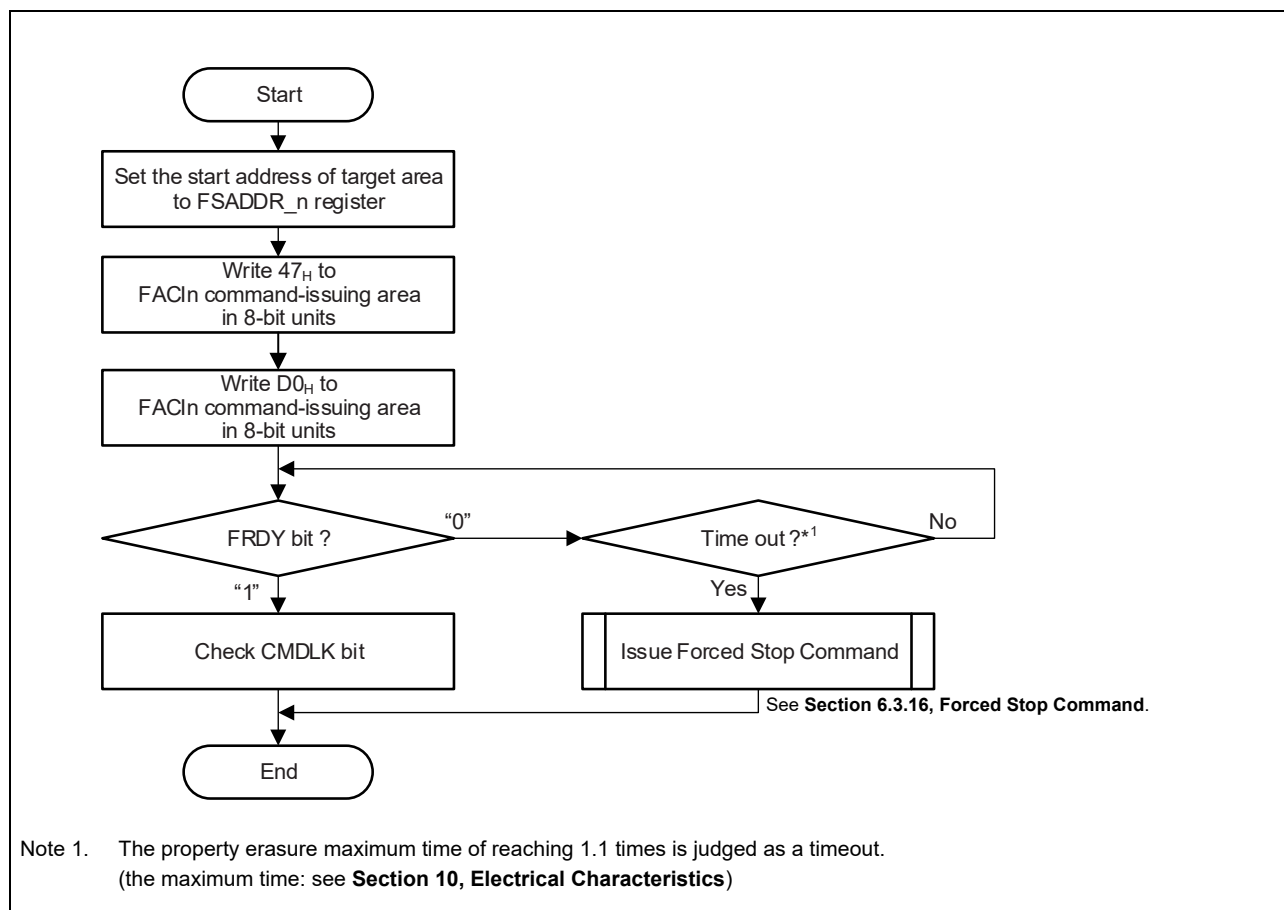
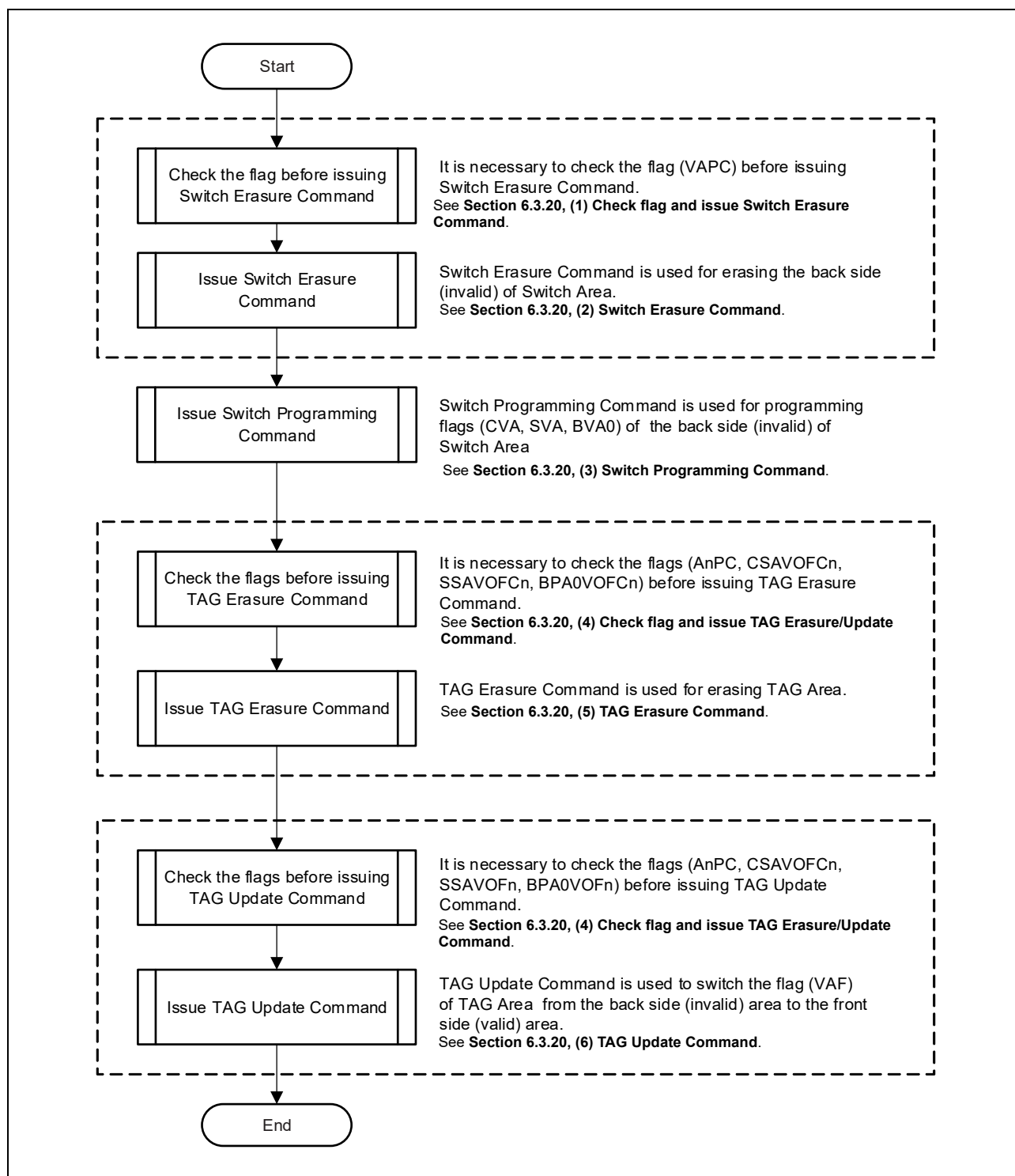


Figure 6.21 Property Erasure Command Usage



### 6.3.20 Switching of Hardware Property Area

The front side (valid) area of Configuration Setting Area, Security Setting Area, and Block Protection Area are decided by CVA, SVA, BVA0 in Switch Area. The front side (valid) of Switch Area is decided by flags in Switch Area 0, Switch Area 1, and TAG Area. **Figure 6.22** shows FACI command operation for switching of Hardware Property Area. If specified areas are not updated completely, FACI doesn't execute these steps.

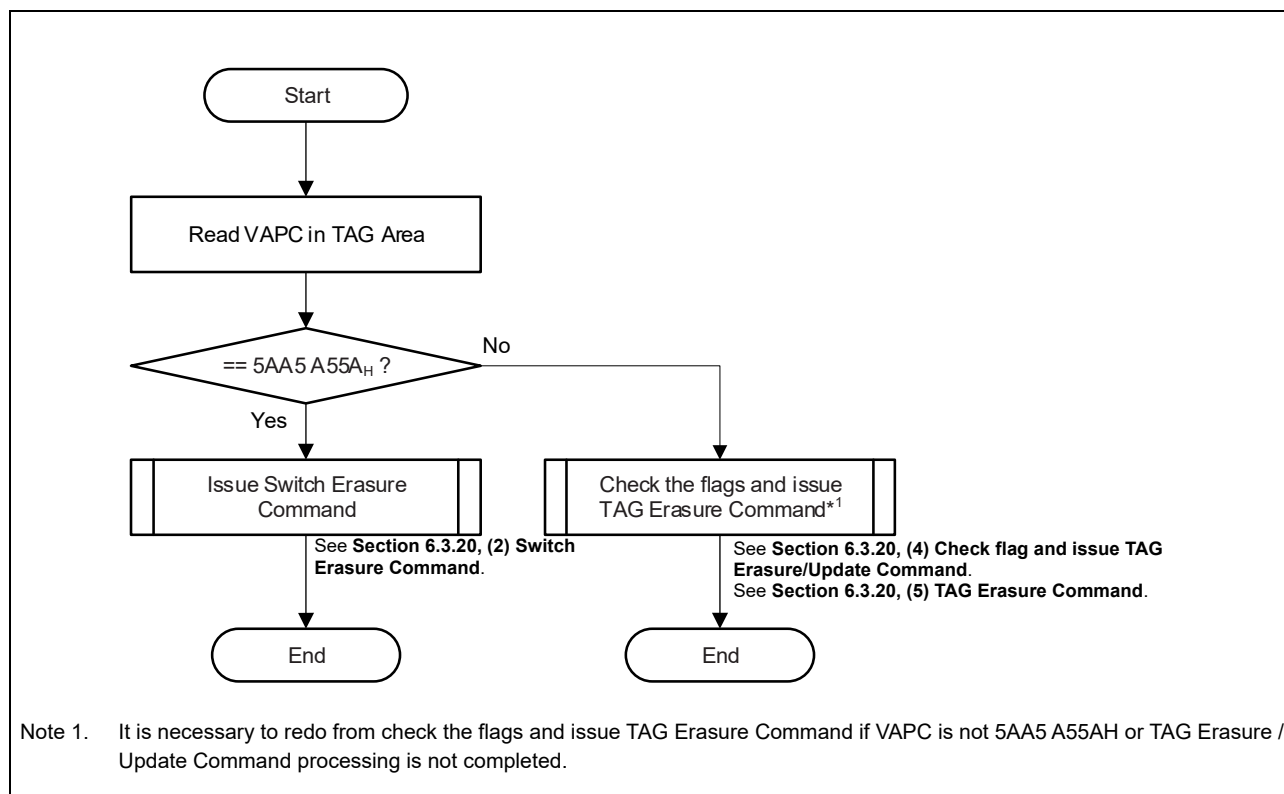


**Figure 6.22** Switching Usage of Hardware Property Area

**(1) Check flag and issue Switch Erasure Command**

It is necessary to check VAPC before issuing Switch Erasure Command. For details of VAPC, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

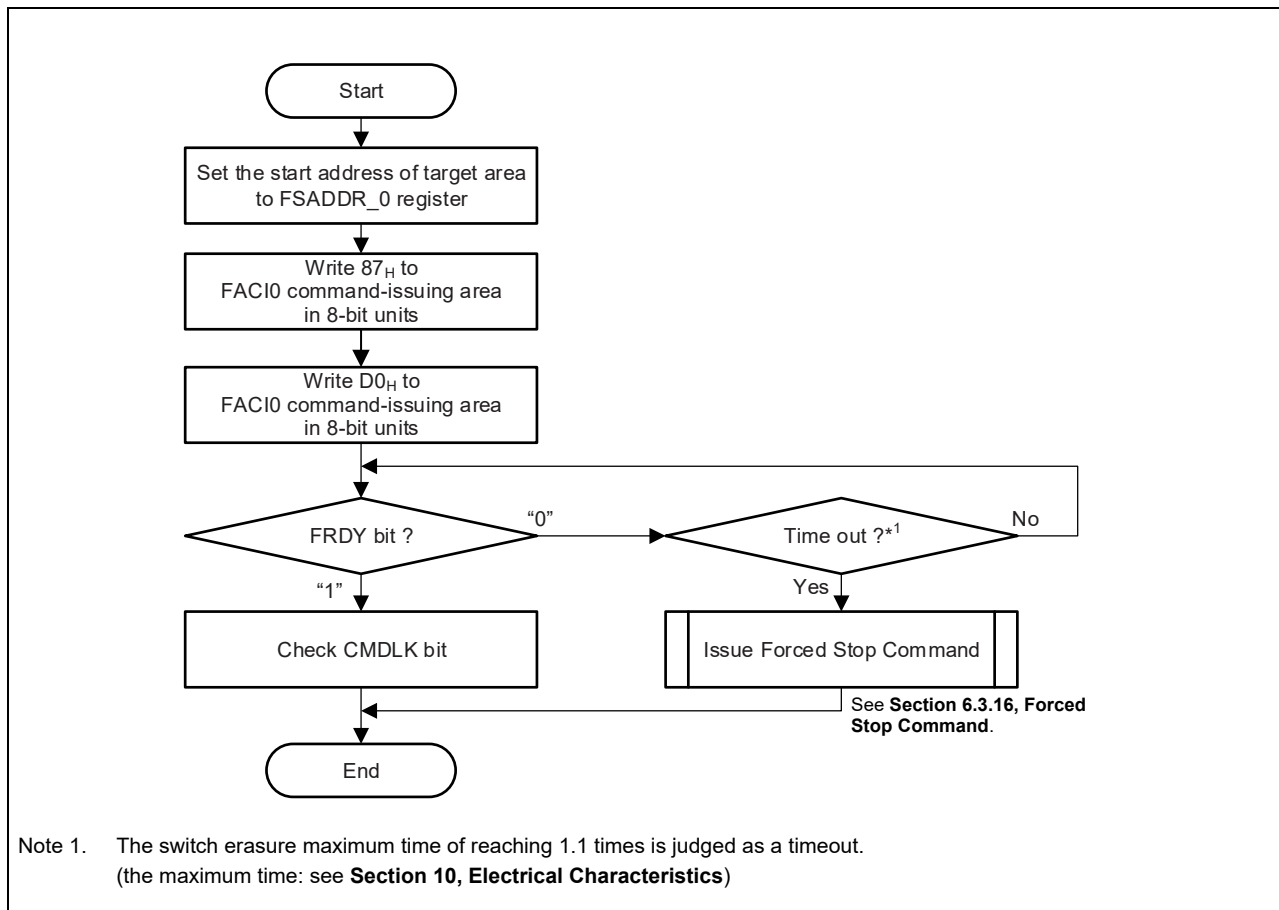
If the check result of flags before Switch Erasure Command is ignored, it does not guarantee switching of Hardware Property Area.



**Figure 6.23 Check the flag before issuing Switch Erasure Command**

## (2) Switch Erasure Command

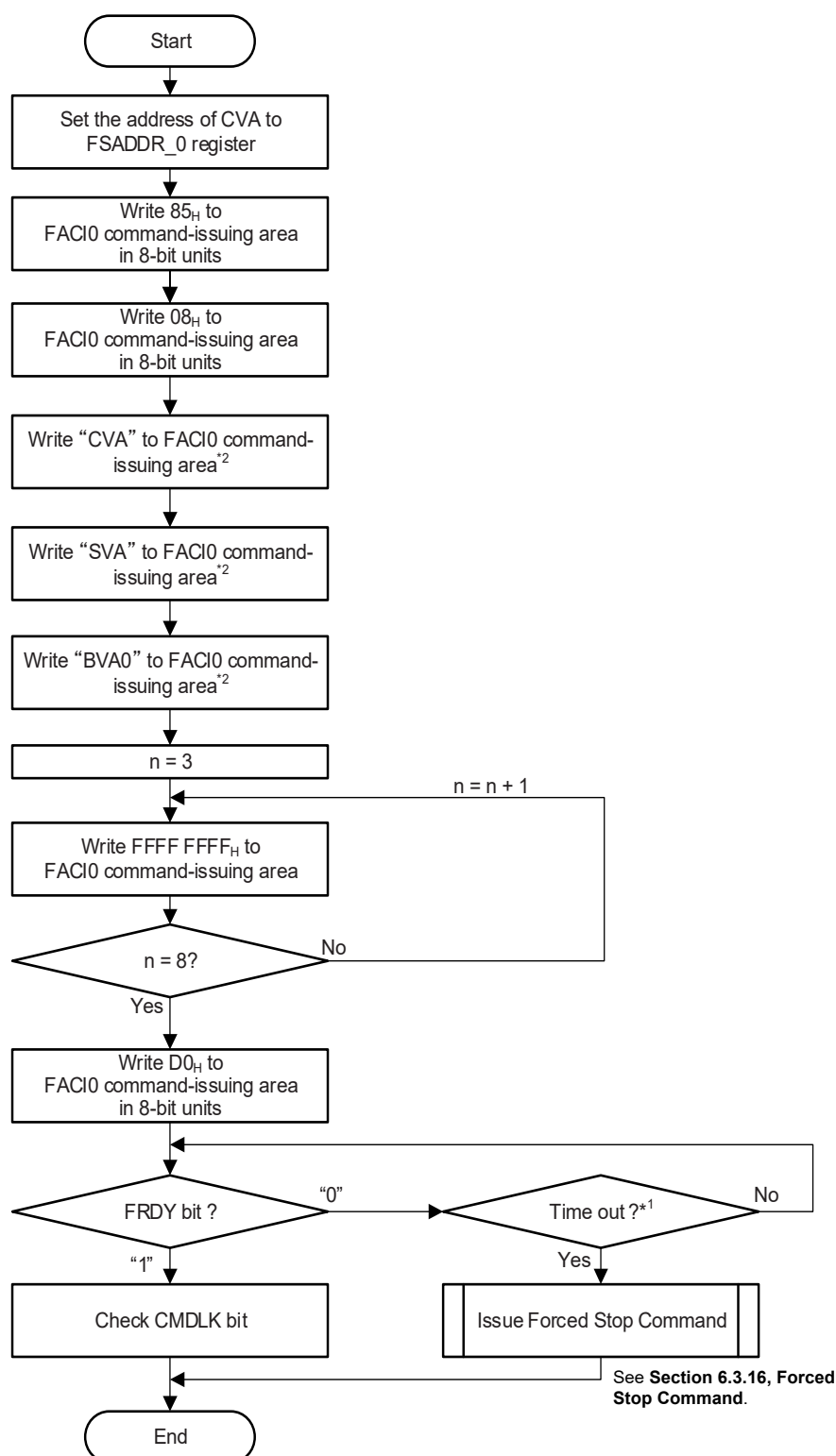
The Switch Erasure Command is used to erase the back side (invalid) of Switch Area. Before issuing the Switch Erasure Command, set the start address of target area to the FSADDR\_0 register. Writing 87<sub>H</sub> and D0<sub>H</sub> to the FACL command-issuing area starts the Switch Erasure Command processing.



**Figure 6.24** Switch Erasure Command Usage

**(3) Switch Programming Command**

The Switch Programming Command is used to write to the back side (invalid) of Switch Area. Before issuing the Switch Programming Command, set the address of CVA to the FSADDR\_0 register. The Switch Programming Command can write data by 32-byte unit. Writing D0<sub>H</sub> to the FACL command-issuing area at the final access of the FACL command issue starts the Switch Programming Command processing.



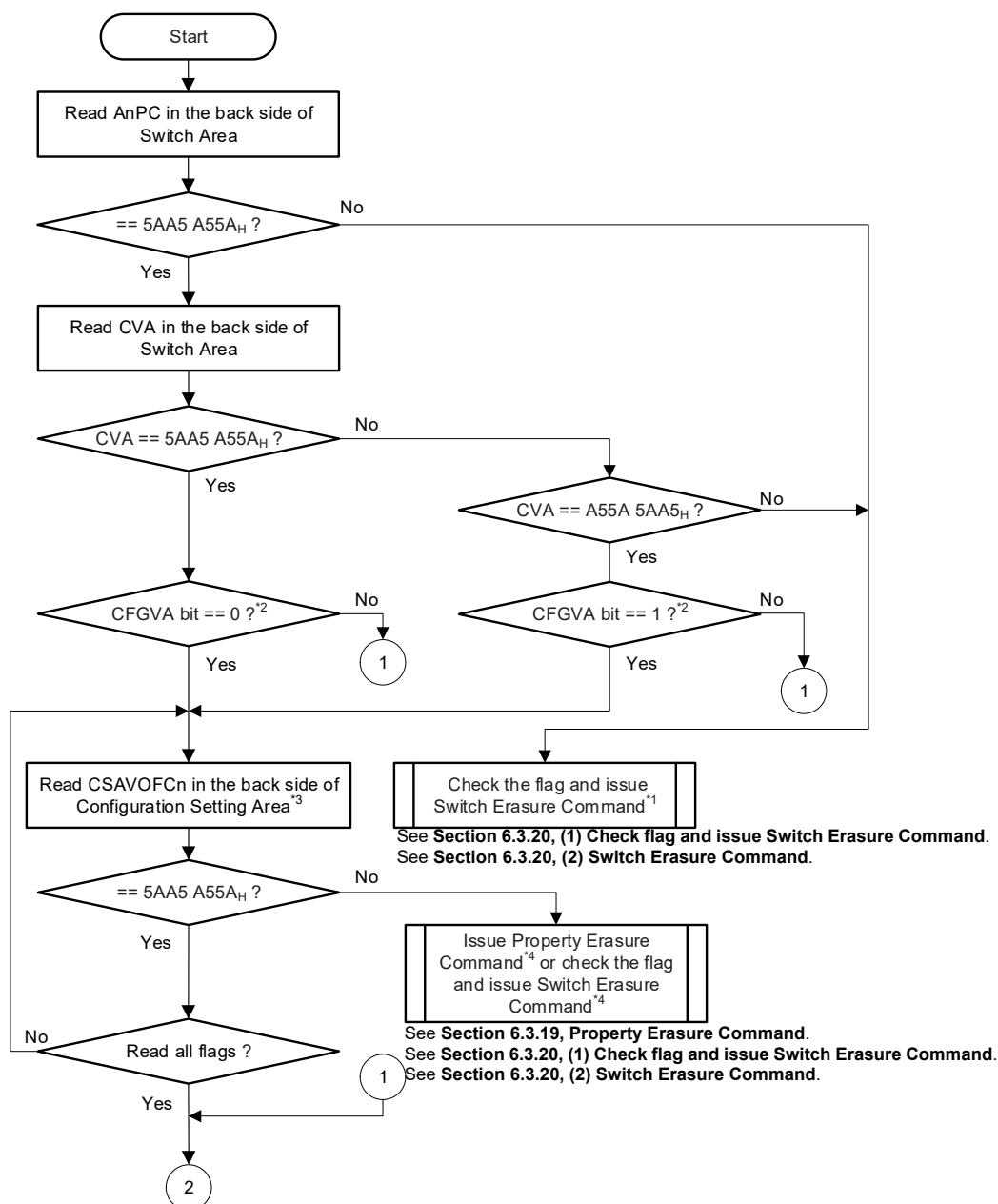
Note 1. The switch Programming maximum time of reaching 1.1 times is judged as a timeout.  
(the maximum time: see **Section 10, Electrical Characteristics**)

Note 2. Configuration Setting Area, Security Setting Area and Block Protection Area switch front side(valid) by each valid area flags.  
When the valid area flag is A55A\_5AA5H, the front side (valid) is area 0. When the valid area flag is 5AA5\_A55AH, the front side (valid) is area 1.

**Figure 6.25 Switch Programming Command Usage**

**(4) Check flag and issue TAG Erasure/Update Command**

It is necessary to check AnPC, CSAVOFCn, SSAVOFCn, BPA0VOFCn before issuing TAG Erasure Command and TAG Update Command. The detail of AnPC, CSAVOFCn, SSAVOFCn, BPA0VOFCn, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.



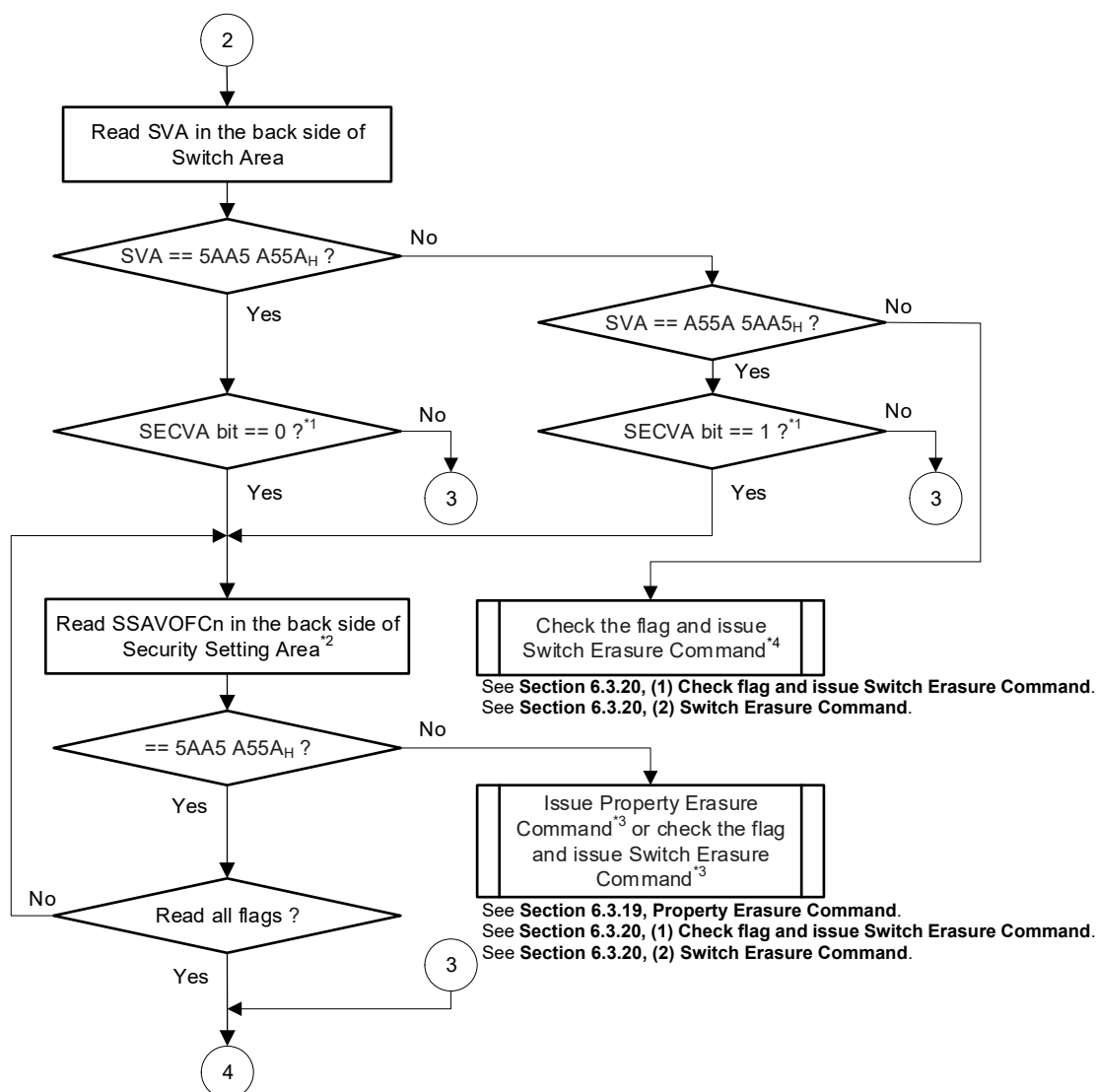
Note 1. When AnPC is not 5AA5 A55A<sub>H</sub>, or CVA is neither 5AA5 A55A<sub>H</sub> nor A55A 5AA5<sub>H</sub> before issuing TAG Erasure Command or TAG Update Command, Switch Programming Command processing is not completed. It is necessary to redo from check the flag and issue Switch Erasure Command.

Note 2. For details of CFGVA bit in FSWASTAT\_0, see **Section 4.19, FSWASTAT\_0 — Switch Area Status Register**.

Note 3. It is necessary to check all CSAVOFCn in the back side of Configuration Setting Area.

Note 4. When CSAVOFCn is not 5AA5 A55A<sub>H</sub> before issuing TAG Erasure Command, CVA is wrong value or Property Programming Command processing is not completed.  
When CSAVOFCn is not 5AA5 A55A<sub>H</sub> before issuing TAG Update Command, Property Programming Command processing is not completed.  
If CVA is wrong value, it is necessary to redo from check the flag and issue Switch Erasure Command.  
If Property Programming Command processing is not completed, it is necessary to redo from issue Property Erasure Command.

**Figure 6.26 Check the flag before issuing TAG Erasure/Update Command (1)**



Note 1. For details of SECVA bit in FSWASTAT\_0, see **Section 4.19, FSWASTAT\_0 — Switch Area Status Register**.

Note 2. It is necessary to check all SSAVOFCn in the back side of Security Setting Area.

Note 3. When SSAVOFCn is not 5AA5 A55A<sub>H</sub> before issuing TAG Erasure Command, SVA is wrong value or Property Programming Command processing is not completed.

When SSAVOFCn is not 5AA5 A55A<sub>H</sub> before issuing TAG Update Command, Property Programming Command processing is not completed.

If SVA is wrong value, it is necessary to redo from check the flag and issue Switch Erasure Command.

If Property Programming Command processing is not completed, it is necessary to redo from issue Property Erasure Command.

Note 4. When SVA is neither 5AA5 A55A<sub>H</sub> nor A55A 5AA5<sub>H</sub> before issuing TAG Erasure Command or TAG Update Command, Switch Programming Command processing is not completed. It is necessary to redo from check the flag and issue Switch Erasure Command.

**Figure 6.27 Check the flag before issuing TAG Erasure/Update Command (2)**



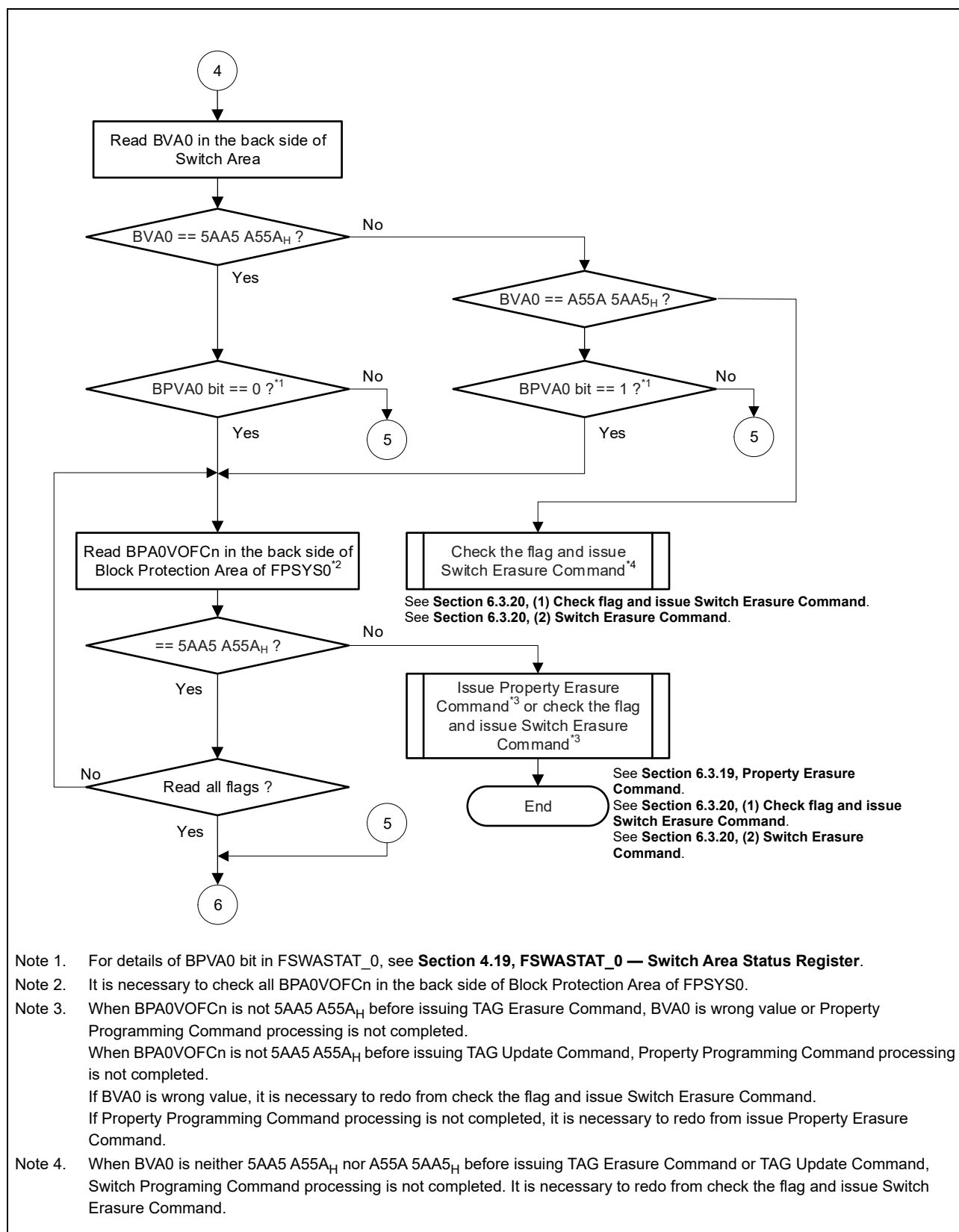
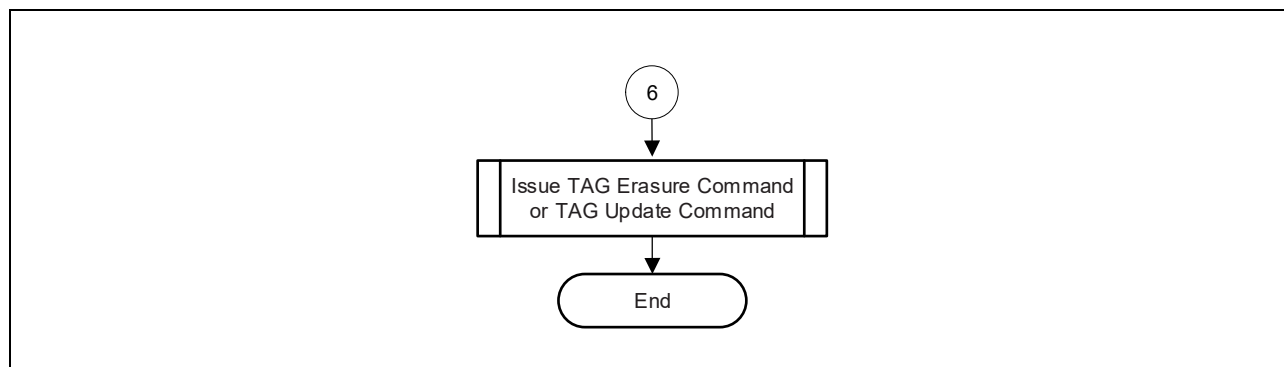


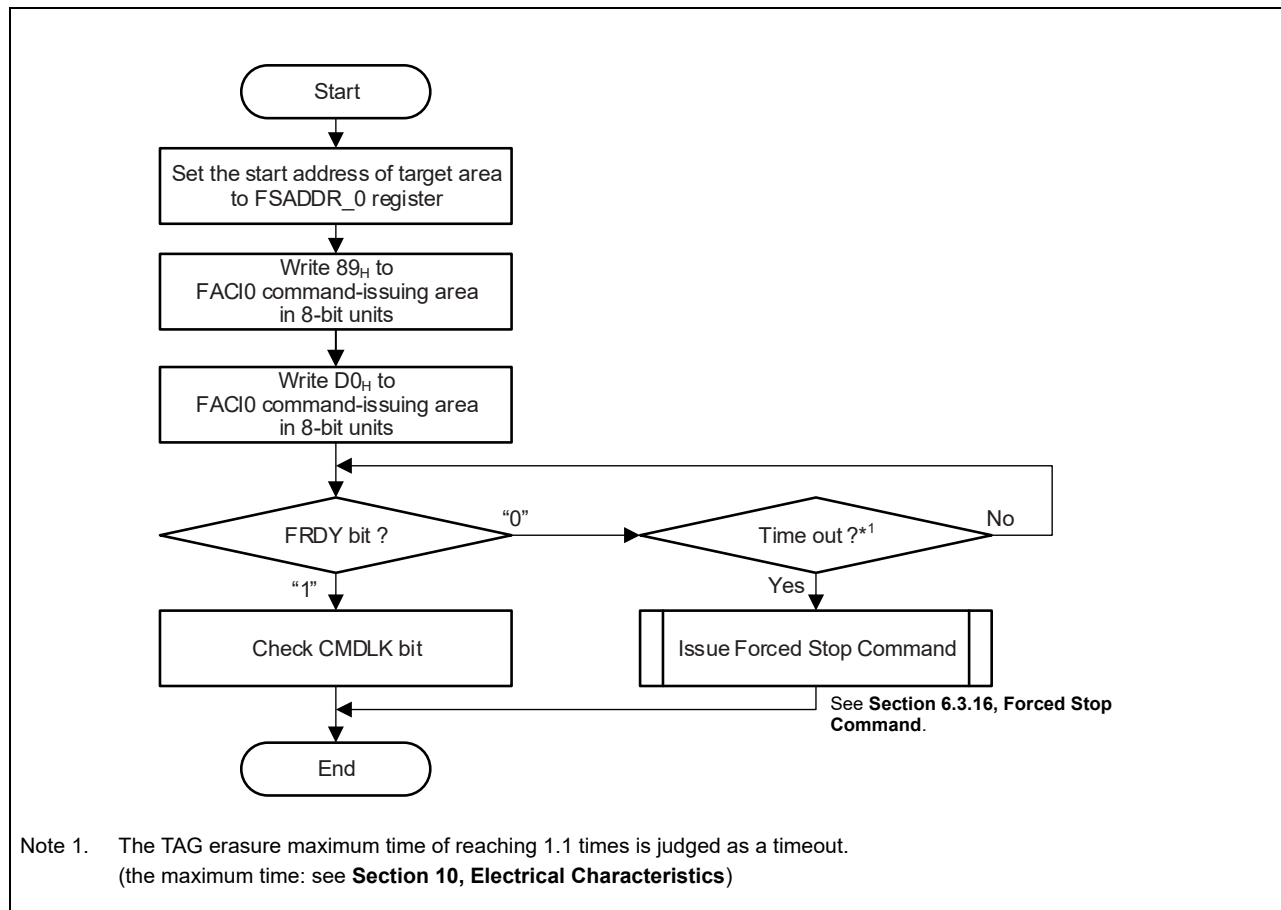
Figure 6.28 Check the flag before issuing TAG Erasure/Update Command (3)



**Figure 6.29** Check the flag before issuing TAG Erasure/Update Command (4)

**(5) TAG Erasure Command**

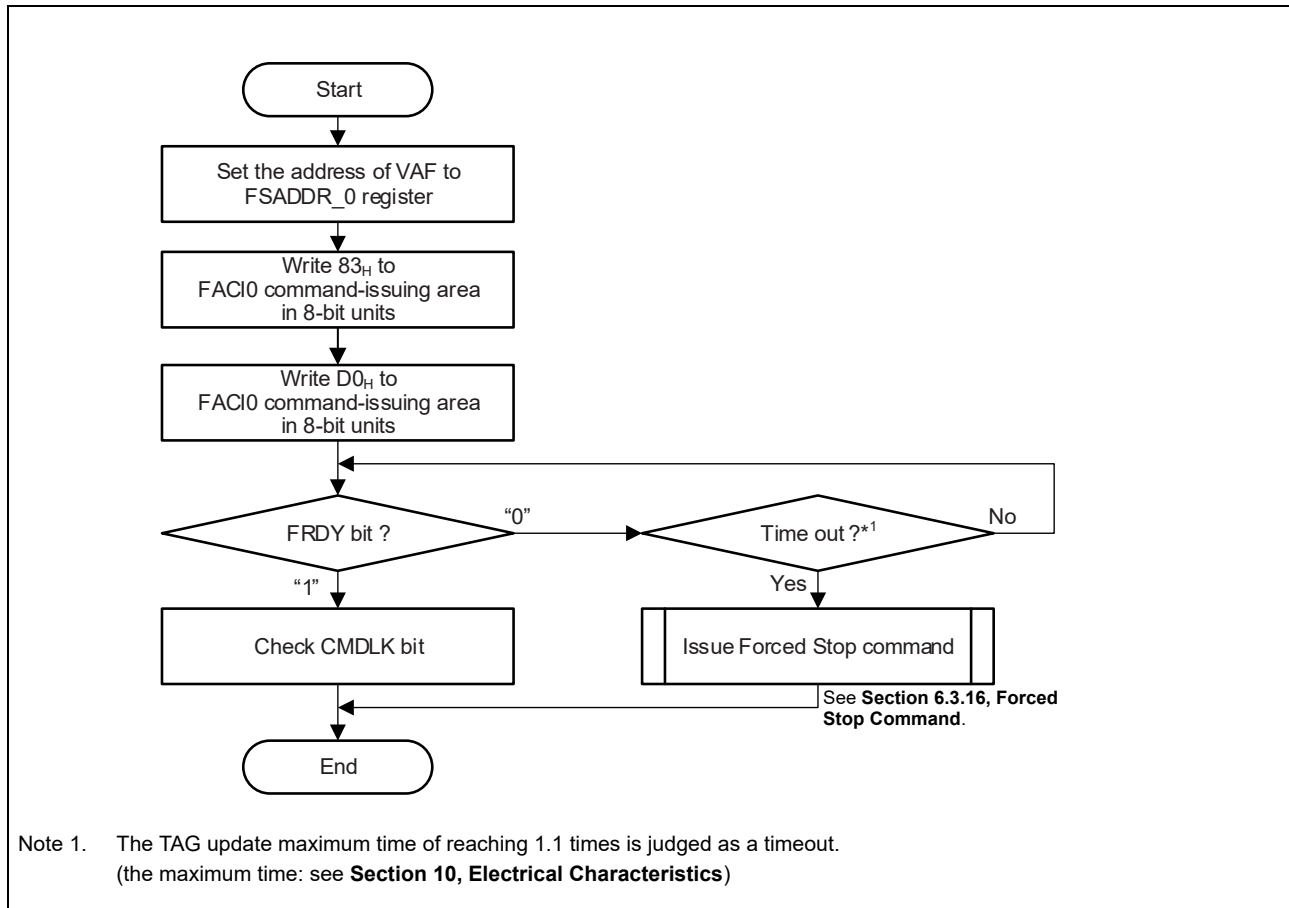
The TAG Erasure Command is used for the change of the data in TAG Area. Before issuing the TAG Erasure Command, set the start address of target area to the FSADDR\_0 register. Writing 89<sub>H</sub> and D0<sub>H</sub> to the FACL command-issuing area starts the TAG Erasure Command processing.



**Figure 6.30 TAG Erasure Command Usage**

**(6) TAG Update Command**

The TAG Update Command is used for the change of the data in TAG Area. Before issuing the TAG Update Command, set the address of VAF to the FSADDR\_0 register. Writing 83<sub>H</sub> and D0<sub>H</sub> to the FACL command-issuing area starts the TAG Update Command processing.



**Figure 6.31 TAG Update Command Usage**

### 6.3.21 Recovering Switch Area Status

It is necessary to recover switch area status by user program when switch area status is dirty (FSWASTAT\_0.SWAS = 1).

It is also necessary to recover switch area status by user program when switching of Hardware Property Area is interrupted or FACL commands issued in Switching of Hardware Property Area end with error.

The overview of recovering switch area status is shown below.

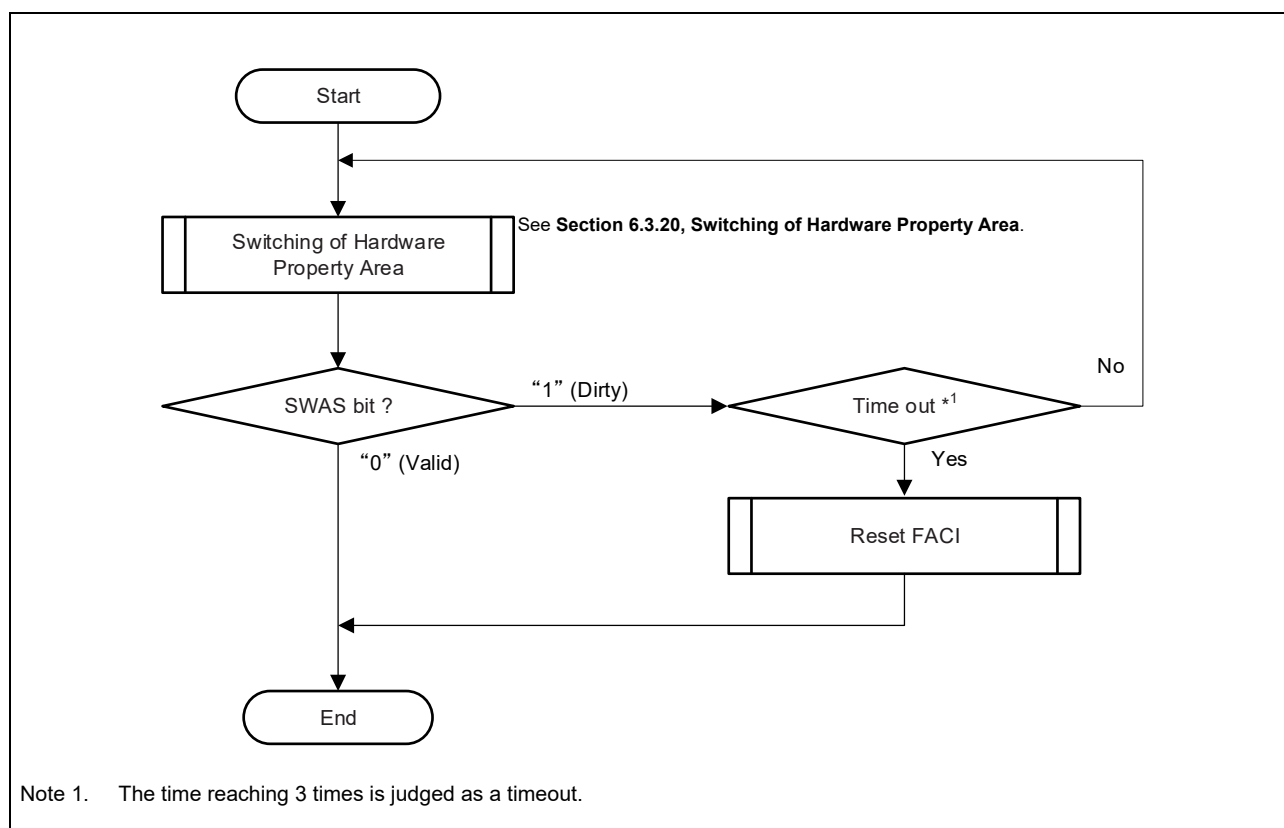


Figure 6.32 Flow of Recovering Switch Area Status.

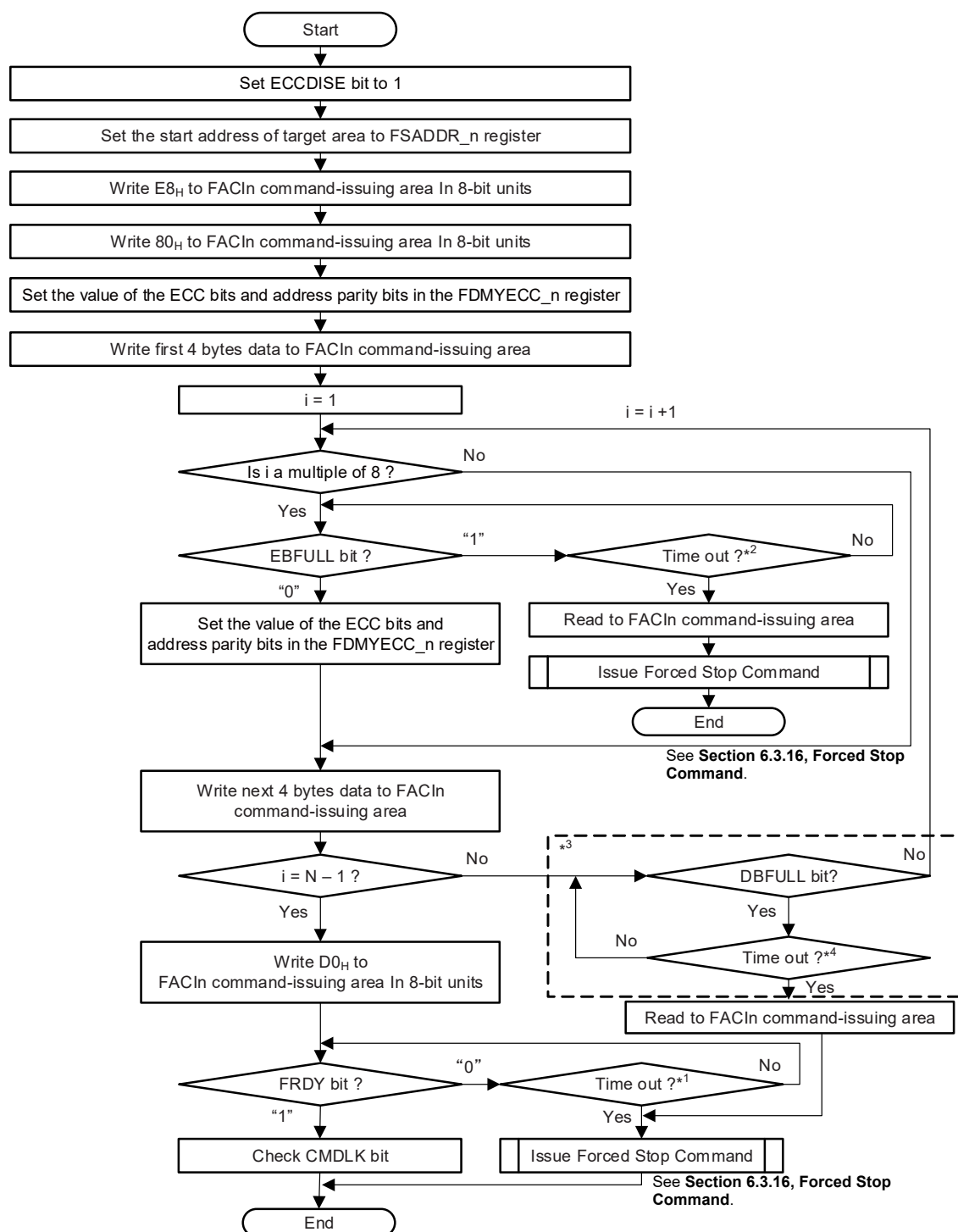
### 6.3.22 Injecting ECC Errors for the Flash Memory

Any value of the ECC bits and address parity bits in the FDMYECC\_n register can be written to the flash memory by using a programming command. The function of injecting ECC error to Data Area or Extended Data Area can be used only by a programming command in 4 bytes units.

Before writing the value set in the FDMYECC\_n register to the flash memory, set the ECCDISE bit in the FECCTMD\_n register to 1. In addition, set the values for the ECC bits and address parity bits in the FDMYECC\_n register before writing the data to the FACL command-issuing area. In the case of the code flash memory, the unit (512 bytes) for writing in response to the programming command differs from the unit (32 bytes) for which the ECC bits and address parity bits are to be added for the data.

Therefore, every time 32 bytes of data are written to the FACL command-issuing area, change the setting in the FDMYECC\_n register. In the case of the data flash memory, since the unit (4 bytes) for writing by the programming command is the same as that for the unit (4 bytes) of data for which the ECC bits are to be added, only change the setting in the FDMYECC\_n register once before issuing the programming command.

Issuing the command for writing to the FDMYECC\_n register while the EBFULL bit in the FSTATR\_n register is 1 may lead to a wait being generated on Peripheral Bus, which will affect performance in communication with other peripheral IP modules. To avoid the generation of such a wait, write to the FDMYECC\_n register while the EBFULL bit in the FSTATR\_n register is 0.



Note 1. The time of 512-byte programming (the maximum time: see **Section 10, Electrical Characteristics**) reaching 1.1 times is judged as a timeout.

Note 2. The time reaching 2 times or 5  $\mu$ s is judged as a timeout.

Note 3. DBFULL bit never become "1", and these steps are not necessary for Code Flash Programming Command. It's possible to carry out these steps for the compatibility of the previous product.

Note 4. The time reaching 1 time or 5  $\mu$ s is judged as a timeout.

**Figure 6.33 Injecting an ECC Error for the Code Flash Memory**

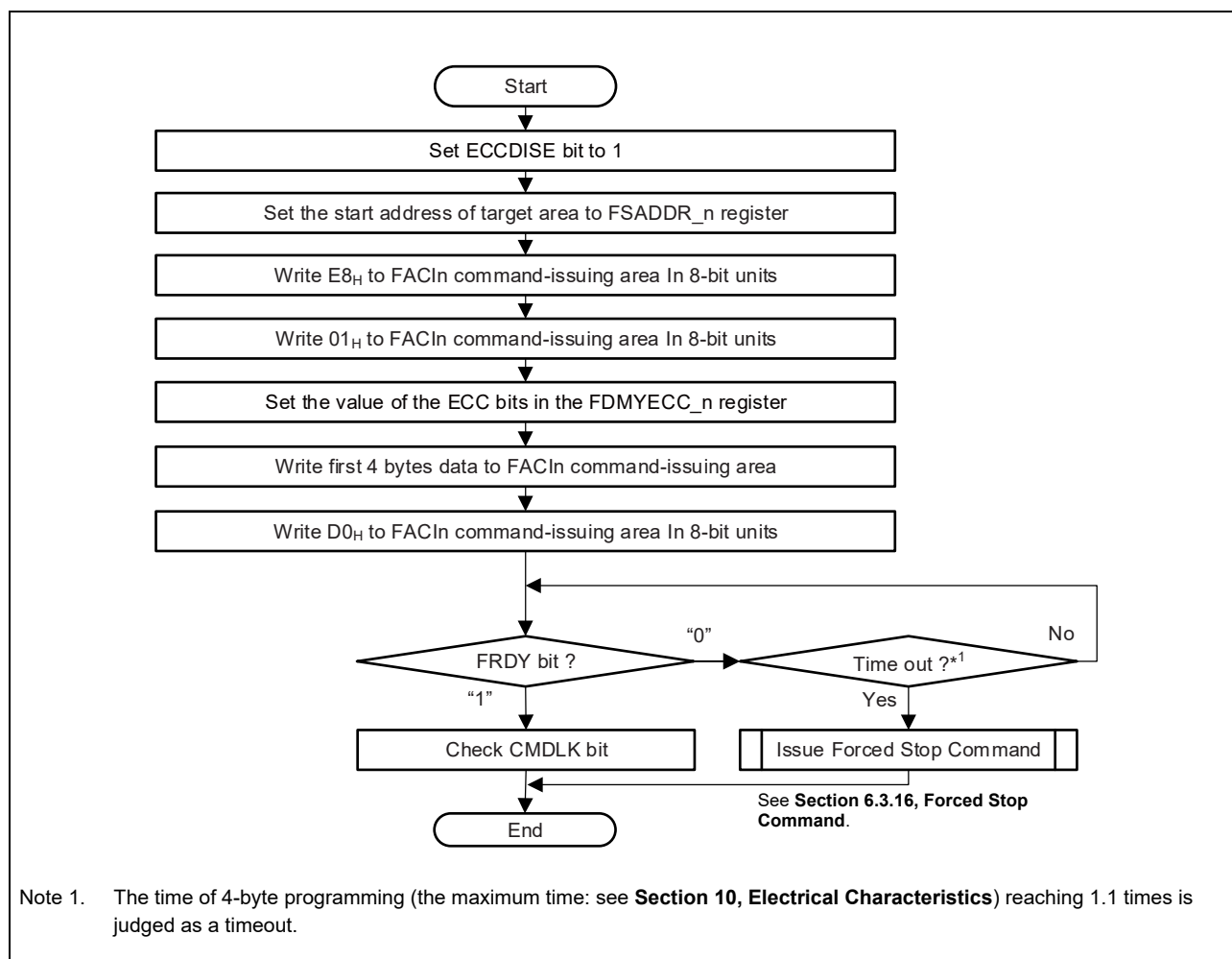


Figure 6.34 Injecting an ECC Error for the Data Flash Memory



## Section 7      Security Function

For details of the security function, see the *RH850/U2B Group User's Manual: Hardware Section 58, Basic Hardware Protection*.

## Section 8 Protection Function

### 8.1 Software Protection

Software protection function disables flash sequencer command operation according to register settings. If an attempt is made to issue flash sequencer command against software protection, flash sequencer enters command lock state.

#### 8.1.1 Protection by FENTRYR

When FENTRYR\_n register is set to 0000<sub>H</sub>, flash sequencer is set to read mode. The FACY commands except for the Status Clear Command cannot be accepted in read mode, and flash sequencer enters command lock state.

#### 8.1.2 Code Flash Valid area protection

Code Flash Valid area protection is function for front side of User Area in Double Map Mode. The front side of User Area can be protected from Programming, Block Erasure, and Programming/Erase Resumption Command when CVAPROT bit in FCVAPROT\_n register is 1.

When flash sequencer command has been issued to protected area, flash sequencer enters command lock state.

### 8.2 Error Protection

Error protection function detects an illegal FACY command issued, an illegal access, or a flash sequencer malfunction, and disables FACY command acceptance (command lock state). While flash sequencer is in the command lock state, flash memory cannot be programmed or erased. To cancel command lock state, issue Status Clearing or Forced Stop Command. Status Clearing Command can be used only when FRDY bit is 1. Forced Stop Command can be used regardless of FRDY bit value. While the CMDLKIE bit in FAEINT\_n register is 1, a flash access error (FLERR) interrupt is generated if flash sequencer enters command lock state (the CMDLK bit of the FASTAT\_n register is 1).

If flash sequencer enters command lock state during programming or erasure processing by the command other than Programming/Erase Suspension, the flash sequencer continues programming or erasure processing. In this state, programming or erasure processing cannot be suspended by the Programming/Erase Suspension Command. If a command is issued in command lock state, ILGLERR bit becomes 1 and the other bits retain the values set due to the previous error detection.

**Table 8.1** shows error protection types and status bit values after error detection.

Table 8.1 Error Protection Type (1/3)

| Error Type              | Description                                                                                                                                                                           | SECDTCT | BPLDTCT | ILGCOMERR | FESETERR | SECERR | OTERR | ILGLERR | ERSERR | PRGERR | FHVEERR | CFGDTCT | TBLDTCT | ERCDTCT | SWTDTCT | CFAE | DFAE |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|---------|-----------|----------|--------|-------|---------|--------|--------|---------|---------|---------|---------|---------|------|------|
| FENTRYR setting error   | The value set in FENTRYR_n is not 0000 <sub>H</sub> , 0001 <sub>H</sub> , or 0080 <sub>H</sub> .                                                                                      | 0/1     | 0/1     | 0/1       | 1        | 0/1    | 0/1   | 1       | 0/1    | 0/1    | 0/1     | 0/1     | 0/1     | 0/1     | 0/1     | 0/1  | 0/1  |
|                         | The FENTRYR_n setting for resuming operation does not match that for suspending operation.                                                                                            | 0       | 0       | 0         | 0        | 0      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
| Illegal command error   | An undefined code has been specified in the first cycle of command.                                                                                                                   | 0       | 0       | 1         | 0        | 0      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
|                         | The value specified in the last access of the multiple- access FACL command is not D0 <sub>H</sub> (except for DMA Programming).                                                      | 0       | 0       | 1         | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0/1  | 0/1  |
|                         | The value (N) specified in the second write access of FACL command in the Programming, Multi Programming, DMA Programming, Property Programming, Switch Programming Command is wrong. | 0       | 0       | 1         | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0/1  | 0/1  |
|                         | Blank Checking Command has been issued with inconsistent BCDIR, FSADDR, and FEADDR settings*1.                                                                                        | 0       | 0       | 1         | 0        | 0      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
|                         | Area Erasure Command has been issued with inconsistent FSADDR and FEADDR settings*1.                                                                                                  | 0       | 0       | 1         | 0        | 0      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
|                         | Area Erasure Command or Blank Checking Command has been issued with the settings of FSADDR and FEADDR crossing over the different Data Area boundary*2.                               | 0       | 0       | 1         | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0/1  |
|                         | FACL command has been issued with the access size different from the specification. (See Table 6.2.)                                                                                  | 0       | 0       | 1         | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0/1  | 0/1  |
|                         | FACL command has been issued against FACL command not acceptable mode. (See Table 6.3.)                                                                                               | 0       | 0       | 1         | 0        | 0      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
|                         | FACL command has been issued when command acceptance conditions are not satisfied in code flash programming/ erasure mode and data flash programming/erasure mode. (See Table 6.4.)   | 0/1     | 0/1     | 1         | 0        | 0/1    | 0/1   | 1       | 0/1    | 0/1    | 0/1     | 0/1     | 0/1     | 0/1     | 0/1     | 0/1  | 0/1  |
|                         | Flag protection for Switch Area and TAG Area. (See Section 6, FACL Command)                                                                                                           | 0/1     | 0/1     | 1         | 0        | 0      | 0     | 1       | 0      | 0      | 0       | 0/1     | 0       | 0       | 0/1     | 0    | 0    |
| Erase error             | An error has occurred during flash memory erasure.                                                                                                                                    | 0       | 0       | 0         | 0        | 0      | 0     | 0       | 1      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
| Programming Error       | An error has occurred during flash memory programming.                                                                                                                                | 0       | 0       | 0         | 0        | 0      | 0     | 0       | 0      | 1      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
| Code flash access error | FACL command has been issued to wrong address in code flash programming/erasure mode. (See Section 4.8, FASTAT_n — Flash Access Status Register (n = 0, 2))                           | 0       | 0       | 0         | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 1    | 0    |
|                         | Programming, Block Erasure, Programming/Erasure Resumption Command has been issued to valid area when code flash valid area protection is configured to enable.                       | 0       | 0       | 0         | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 1    | 0    |

Table 8.1 Error Protection Type (2/3)

| Error Type                           | Description                                                                                                                                                                                                                      | SECDTCT | BPLDTCT | ILGCOMER | FESETERR | SECERR | OTERR | ILGLERR | ERSERR | PRGERR | FHVEERR | CFGDTCT | TBLDTCT | ERCDTCT | SWTDTCT | CFAE | DFAE |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|---------|----------|----------|--------|-------|---------|--------|--------|---------|---------|---------|---------|---------|------|------|
| Data flash access error              | Programming, Multi Programming, DMA Programming or Block Erasure Command has been issued to wrong address in data flash programming/erasure mode. (See <b>Section 4.8, FASTAT_n — Flash Access Status Register (n = 0, 2).</b> ) | 0       | 0       | 0        | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 1    |
|                                      | Address boundary between different Data Area*2 is crossed by DMA Programming in data flash programming/erasure mode. (See <b>Section 6.3.10, DMA Programming Command.</b> )                                                      | 0       | 0       | 0        | 0        | 0      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 1    |
|                                      | Area Erasure or Blank Checking Command has been issued to wrong address in data flash programming/erasure mode. (See <b>Section 4.8, FASTAT_n — Flash Access Status Register (n = 0, 2).</b> )                                   | 0       | 0       | 1        | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 1    |
|                                      | Property Programming or Property Erasure Command has been issued to wrong address in data flash programming/erasure mode. (See <b>Section 4.8, FASTAT_n — Flash Access Status Register (n = 0, 2).</b> )                         | 0       | 0       | 0        | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 1    |
|                                      | Switch Programming or Switch Erasure Command has been issued to wrong address in data flash programming/erasure mode. (See <b>Section 4.8, FASTAT_n — Flash Access Status Register (n = 0, 2).</b> )                             | 0       | 0       | 0        | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 1    |
|                                      | TAG Update or TAG Erasure Command has been issued to wrong address in data flash programming/erasure mode. (See <b>Section 4.8, FASTAT_n — Flash Access Status Register (n = 0, 2).</b> )                                        | 0       | 0       | 0        | 0        | 0/1    | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 1    |
| Security                             | A command has been issued against the block which is configured to enable OTP                                                                                                                                                    | 0       | 0       | 0        | 0        | 1      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
|                                      | Code Flash access protection error                                                                                                                                                                                               | 0       | 0       | 0        | 0        | 1      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
|                                      | Data Flash access protection error                                                                                                                                                                                               | 0       | 0       | 0        | 0        | 1      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0/1  |
|                                      | Switch Area or TAG Area access protection error                                                                                                                                                                                  | 0       | 0       | 0        | 0        | 1      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0/1  |
|                                      | The error that security function of ICUMHB caused. (See the <i>RH850/U2B Group Security User's Manual: Hardware.</i> )                                                                                                           | 0       | 0       | 0        | 0        | 1      | 0     | 1       | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0/1  | 0/1  |
| Other                                | FACI command has been issued when command acceptance conditions are not satisfied in read mode. (See <b>Table 6.4.</b> )                                                                                                         | 0/1     | 0/1     | 0/1      | 0        | 0/1    | 1     | 1       | 0/1    | 0/1    | 0/1     | 0/1     | 0/1     | 0/1     | 0/1     | 0/1  | 0/1  |
|                                      | FACI command-issuing area has been read.                                                                                                                                                                                         | 0/1     | 0/1     | 0/1      | 0/1      | 0/1    | 1     | 1       | 0/1    | 0/1    | 0/1     | 0/1     | 0/1     | 0/1     | 0/1     | 0/1  | 0/1  |
| FHVE setting error                   | FHVE3CNT bit in FHVE3FPn register has changed to 0 while command processing is provided by the flash sequencer.                                                                                                                  | 0       | 0       | 0        | 0        | 0      | 0     | 0       | 0      | 0      | 1       | 0       | 0       | 0       | 0       | 0    | 0    |
| Security Setting Area ECC error      | 2-bit error has been detected during Security Setting Area access.                                                                                                                                                               | 1       | 0       | 0/1      | 0        | 0      | 0     | 0/1     | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
| Block Protection Area ECC error      | 2-bit error has been detected during Block Protection Area access.                                                                                                                                                               | 0       | 1       | 0/1      | 0        | 0      | 0     | 0/1     | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0    | 0    |
| Configuration Setting Area ECC error | 2-bit error has been detected during Configuration Setting Area access.                                                                                                                                                          | 0       | 0       | 0/1      | 0        | 0      | 0     | 0/1     | 0      | 0      | 0       | 1       | 0       | 0       | 0       | 0    | 0    |

Table 8.1 Error Protection Type (3/3)

| Error Type                    | Description                                                                    | SECDTCT | BPLDTCT | ILGCOMERR | FESETERR | SECERR | OTERR | ILGLERR | ERSERR | PRGERR | FHVEERR | CFGDTCT | TBLDTCT | ERCDTCT | SWTDTCT | CFAE | DFAE |
|-------------------------------|--------------------------------------------------------------------------------|---------|---------|-----------|----------|--------|-------|---------|--------|--------|---------|---------|---------|---------|---------|------|------|
| P/E Parameter Table ECC error | Data with 2-bit error has been detected during P/E parameter table access.     | 0       | 0       | 0         | 0        | 0/1    | 0     | 0/1     | 0      | 0      | 0       | 0       | 1       | 0       | 0       | 0    | 0    |
| Switch Area ECC error         | Data with 2-bit error has been detected during Switch Area or TAG Area access. | 0       | 0       | 0         | 0        | 0      | 0     | 0       | 0      | 0      | 0       | 0       | 0       | 0       | 1       | 0    | 0    |
| Erase Counter Area ECC error  | Data with 2-bit error has been detected during Erase Counter area access.      | 0       | 0       | 0         | 0        | 0      | 0     | 0       | 0      | 0      | 0       | 0       | 0       | 1       | 0       | 0    | 0    |

Note 1. Inconsistent FSADDR and FEADDR settings are follow.

- FSADDR[20:0] > FEADDR[20:0] (Area Erasure Command )
- FSADDR[20:0] > FEADDR[20:0] (Blank Checking Command (BCDIR=0))
- FSADDR[20:0] < FEADDR[20:0] (Blank Checking Command (BCDIR=1))

Note 2. Data Area size, see the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

## 8.3 Boot Program Protection

### 8.3.1 User Boot Protection

The User Boot Area can be programmed/erased by the serial programming. Since this area is usually write-protected for the self-programming, it can be used to store programs such as a boot program safely.

---

## Section 9 Usage Notes

See Usage Notes in the *RH850/U2B Group User's Manual: Hardware Section 63, Flash Memory*.

## Section 10 Electrical Characteristics

For details of the electrical characteristics, see the *RH850/U2B Group User's Manual: Hardware Section 66, Electrical Characteristics*.

---

RH850/U2B Group Flash Memory User's Manual: Hardware

Publication Date: Rev.0.40 February 26, 2021  
Rev.1.30 July 31, 2026

Published by: Renesas Electronics Corporation

---



# RH850/U2B Group Flash Memory