RH850/U2B Group

Startup Application Note

Summary

This application note explains the startup process for the RH850/U2B series (hereafter referred to as U2B) of automotive microcomputers from Renesas Electronics.

This document and program are intended to promote understanding of the functions installed in the U2B and are not intended for mass production design.

In addition, it does not reflect the latest manuals, errata, technical updates, or development environment updates. When using the relevant functions, please treat this program as a reference only and use the latest documentation and development environment at your own risk.

Target Devices

- RH850/U2B Group
  - RH850/U2B24
  - RH850/U2B20
  - RH850/U2B10
  - RH850/U2B6

Target IDE

CS+(made by Renesas Electronics)

Version : CS+forCCE8.07.00g6 (For U2B6)

MULTI(made by Green Hills Software)

Version : MULTI v7.1.6

Table of contents

1. Overview

Startup processing is the processing from after U2B is released from reset until the user application is called.

This application note describes the startup processing of RH850/U2B using CS+ (hereafter referred to as CS+), an integrated development environment from Renesas Electronics Corporation, and MULTI (hereafter referred to as MULTI), an integrated development environment from Green Hills Software.

1.1 Precautions for Use

This application note describes the startup processing of the RH850/U2B.

The number of CPUs installed varies depending on the RH850/U2B series product.

The RH850/U2B24 has six CPUs installed, the RH850/U2B10 has four CPUs, and the U2B6 has three CPUs.

This application note describes the startup processing of the RH850/U2B6.

2. CS+

2.1 Startup files

shows a list of files related to Startup.

Table 2-1 Startup files(*n* = 0 ~ 2)

File name	Description
system_init.c	Clock gear-up process
boot.asm	Startup process
cstart*n*.asm	Startup process for user applications
vecttbl*n*.asm	Vector table
main*n*.c	Main process (user application)

2.2 Section settings

2.2.1 Sections list

An example of the main sections related to Startup is shown below. The address of each section is set in the project file (*.mtpj, *mtsp) in CS+.

Table 2-2 Startup sections (common, *n* = 0 ~ 2)

Section	Allocation Data
RESET_PE*n*	Reset vector, exception handler vector
EIINTTBL_PE*n*	EI level interrupt vector table for table reference method
.text	Program code

Table 2-3 Startup sections (PE0 ~ PE2, *n* = 0 ~ 2)

Section	Allocation Data
const	Read-only data
INIT_DSEC.const	Initialization table for sections with initial values
INIT_BSEC.const	Initialization table for sections without initial values
text.cmn	boot.asm/cstart*n*.asm transfer information
text	program code (cstart.*/main.c)
data	Data with initial values (ROM)
data.R	Data with initial values (RAM)
bss	Data without initial value
stack.bss	Stack

2.2.2 Sections setting method

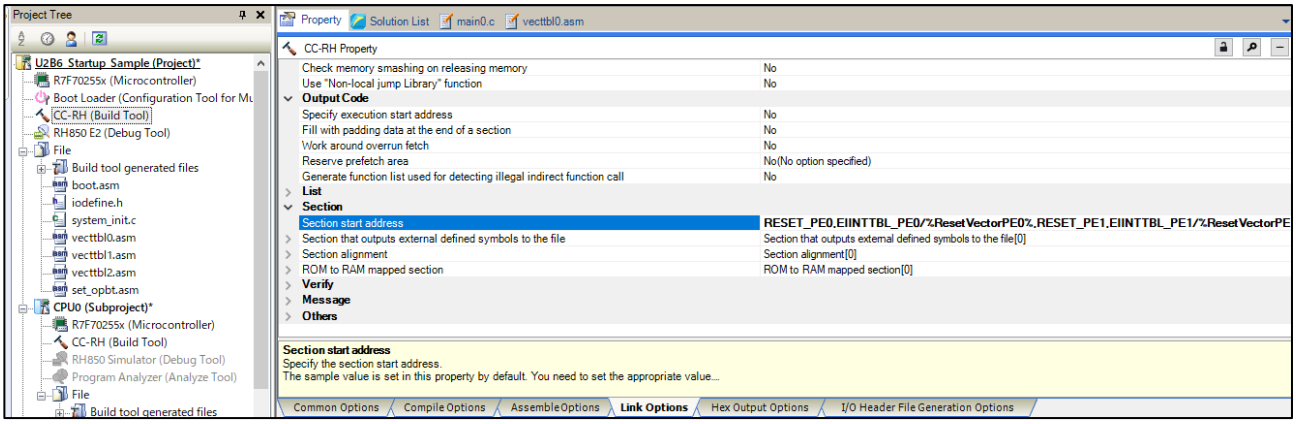
This section describes how to set up a section using CS+.

2.2.2.1 Code Flash

Code Flash is the area to store application code. For detailed information about flash memory areas and settings, refer to Section 63 "Flash Memory" and Section 4 "Address Space" in the U2B-E User's Manual.

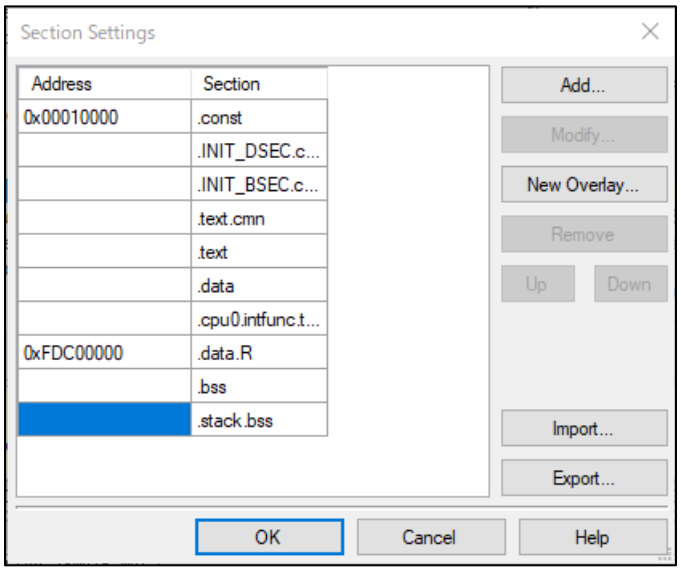
To change sections, select tabs "CC-RH (Build Tool)" to "Link Options" and then "Section" to "Section start address".

Figure 2-1 Section setting



Once selected, the following window will appear, so change the address or section by clicking "Add" or "Modify".

Figure 2-2 Sections setting



2.2.2.2 RAM

RAM (Random Access Memory) is memory for temporarily storing data. For detailed information about RAM areas and settings, refer to Section 64 "RAM" and Section 4 "Address Space" in the U2B-E User's Manual.

The method for changing it is the same as in the previous section. Pay attention to the start address of the section.

2.3 Startup process (CS+)

2.3.1 Conditional assembly control instruction definitions

The definitions for the conditional assembly control instructions are shown in Table 2-4. The conditional assembly control instructions are defined in boot.asm.

Table 2-4 List of definitions for conditional assembly control instructions

Definition name	Value	Description
USE_TABLE_REFERENCE_METHOD		Defines whether to use the table reference method as the interrupt vector method. The initial value is 1 (use table reference method).
	0	Does not use the table reference method.
	1	Using the table reference method。

2.3.2 Overall flow

The overall flow of the startup process is shown below.

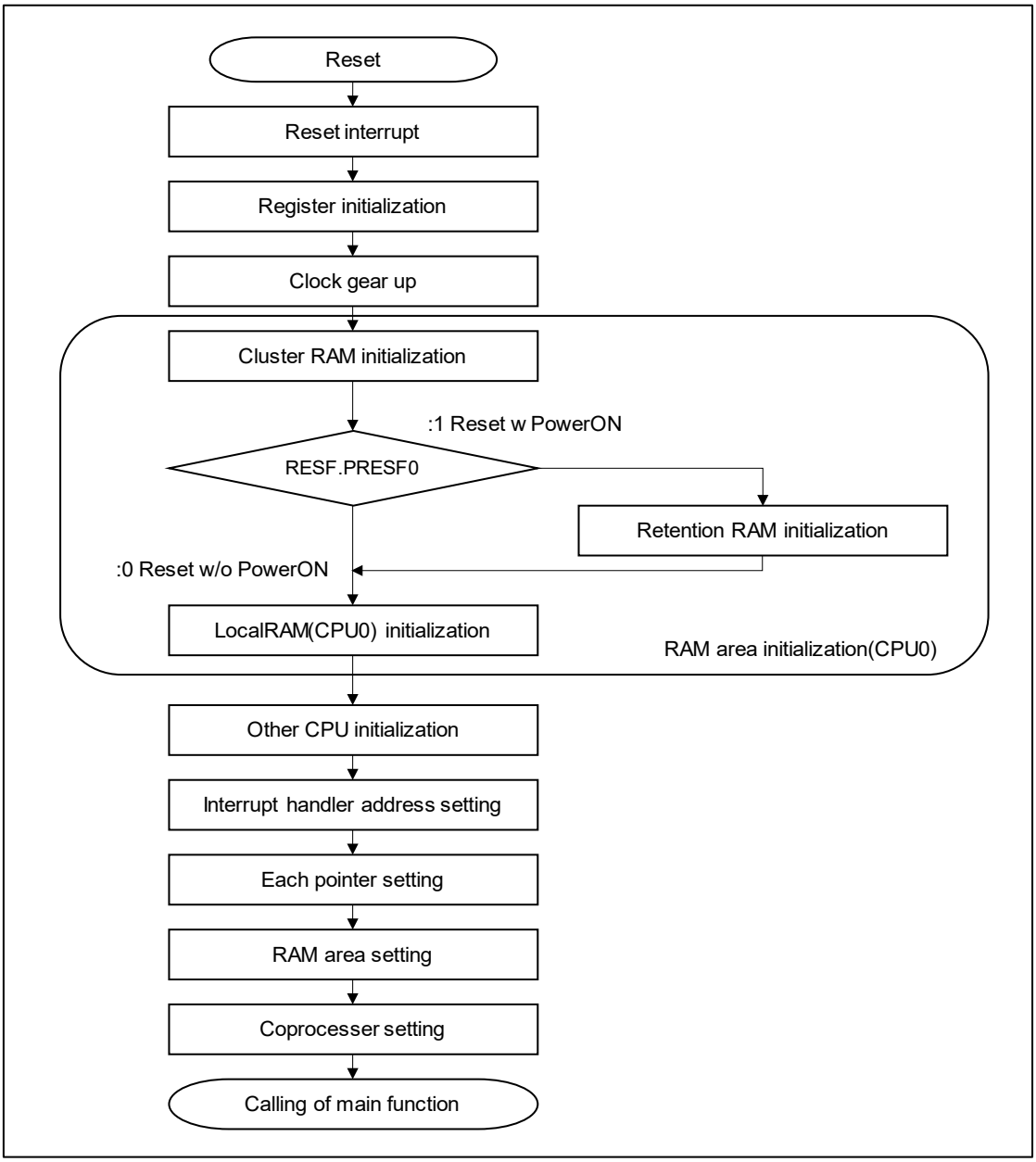


Figure 2-3 Startup flowchart

2.3.3 Processing overview

An overview of each treatment is shown in Table 2-5.

Table 2-5 Implementation list of each process (*n* = 0 ~ 2)

#	Process name	Label	Implementation files	Note
1	Reset interrupt processing	-	vecttbl*n*.asm	Implemented with interrupt vector table
2	Register initialization	_start	boot.asm	
3	Clock gear-up settings	_hdwinit_PE0	boot.asm system_init.c	Process only if the processing PE is PE0
4	RAM area initialization	_hdwinit_PEn _RetentionRAMClear _LocalRAMCPU0Clear	boot.asm	RetentionRAM is processed only when the processing PE is PE0.
5	Other CPU initialization	_WakeUpOtherCPU	boot.asm system_init.c	Process only if the processing PE is PE0
6	Interrupt handler address setting	_set_table_reference_method	boot.asm	
7	Each pointer setting	_cstart_pm*n*	cstart*n*.asm	Processing at each PE
8	RAM area setting	_cstart_pm*n*	cstart*n*.asm	Processing at each PE
9	Coprocessor Settings	_cstart_pmn	cstart*n*.asm	Processing at each PE

2.3.4 Details of each process

We will explain the details of each process.

Unless otherwise specified, the same process will be performed for all PE0 to PE2.

2.3.4.1 Reset interrupt

At power-on, the device transitions to the address of the RESET vector (RESET section).

The RESET vector transitions to __start processing.

PE0 starts when the device itself is powered on.

An example of program code is shown in Figure 2-4.

```
.section    "RESET_PE0", text
.align      512

jr32        __start      ; RESET
```

Figure 2-4 Reset interrupt program code example

2.3.4.2 Register initialization

Initialize the general-purpose and system registers of each PE.

The registers to be initialized are listed in Table 2-6. The setting values are examples, so initialize them to the most appropriate values for your system.

Table 2-6 Initialization register list

#	Register classification	Register name	Example of settings	Description
1	Program register	r1~r31	0	
2	Basic system registers	EIPC	0	
3		FEPC	0	
4		CTPC	0	
5		EIWR	0	
6		FEWR	0	
7		INTBR	0	
8		EBASE	0	
9		MEA	0	
10		MEI	0	
11		RBIP	0	
12	MPU function register	MCA	0	
13		MCS	0	
14		MCR	0	
15		MPIDX	0	
16		MPLA	0	
17		MPUA	0	
18		MPAT	0	
19		MPID0	0	
20		MPID1	0	
21		MPID2	0	
22		MPID3	0	
23		MPID4	0	
24		MPID5	0	
25		MPID6	0	
26		MPID7	0	
27		MCI	0	
28	Cache operation function register	ICTAGL	0	
29		ICTAGH	0	
30		ICDATL	0	
31		ICDATH	0	
32		ICERR	0	
33	FPU function register	FPSR	0x00220000	
34		FPEPC	0	
35		FPST	0	

#	Register classification	Register name	Example of settings	Description
36		FPCC	0	
37	Virtualization support function system registers	HVSB	0	
38	Guest context register	GMEIPC	0	
39		GMFEPC	0	
40		GMEBASE	0	
41		GMINTBP	0	
42		GMEIWR	0	
43		GMFEWR	0	
44		GMMEA	0	
45		GMMEI	0	

An example of the program code is shown in Figure 2-5.

```
        $nowarning
        mov         r0, r1
        $warning
        mov         r0, r2
(skip)
        ldsr r0, 0, 0 ; SR0,0 EIPC
        ldsr r0, 2, 0 ; SR2,0 FEPC
        ldsr r0, 16, 0 ; SR16,0 CTPC
(skip)
```

Figure 2-5 Example of register initialization program code

2.3.4.3 Clock gear up

After PE0 starts up, change the system clock sources (CLK_SYS_CLEAN and CLK_SYS_SSCG) to CLK_PLLO and CLK_SSCGO to gear up the clock.

This process will be executed only if all of the following conditions are met.

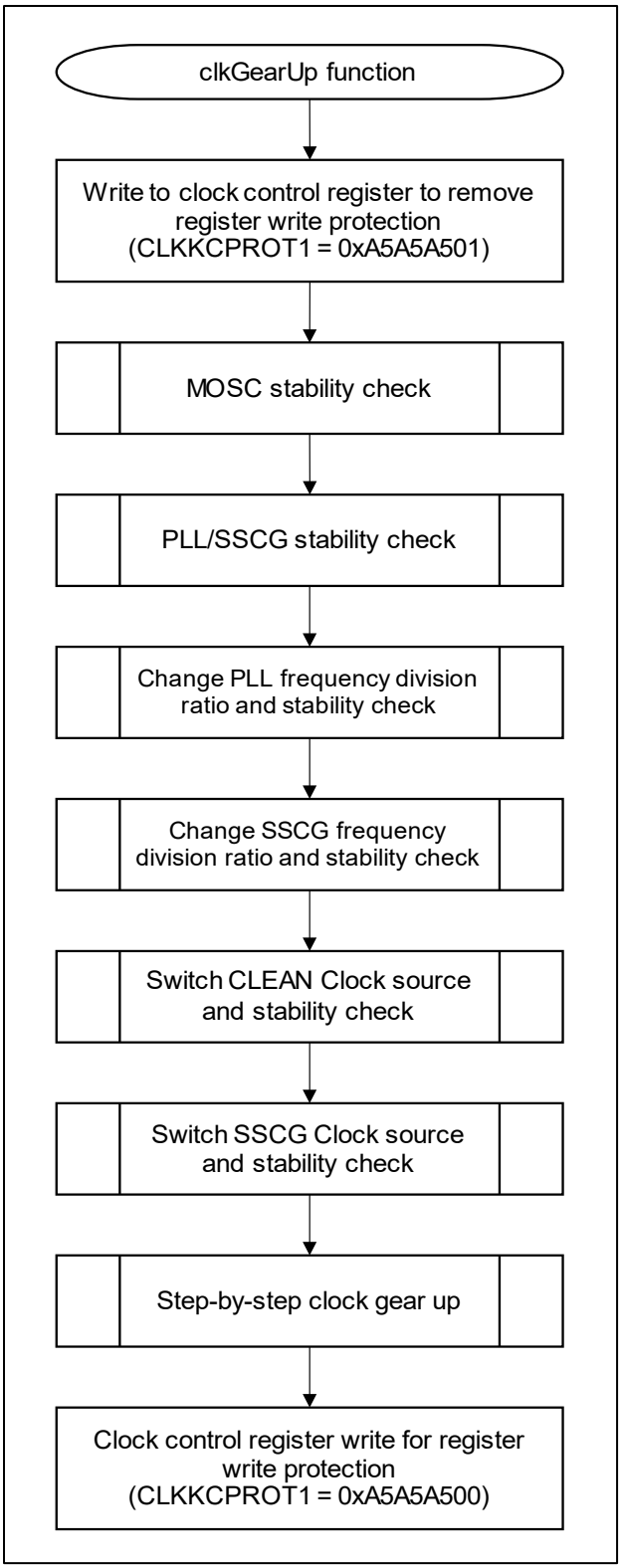
- The processing PE is PE0 (PEID register bit2:0 (PEID) = 0)

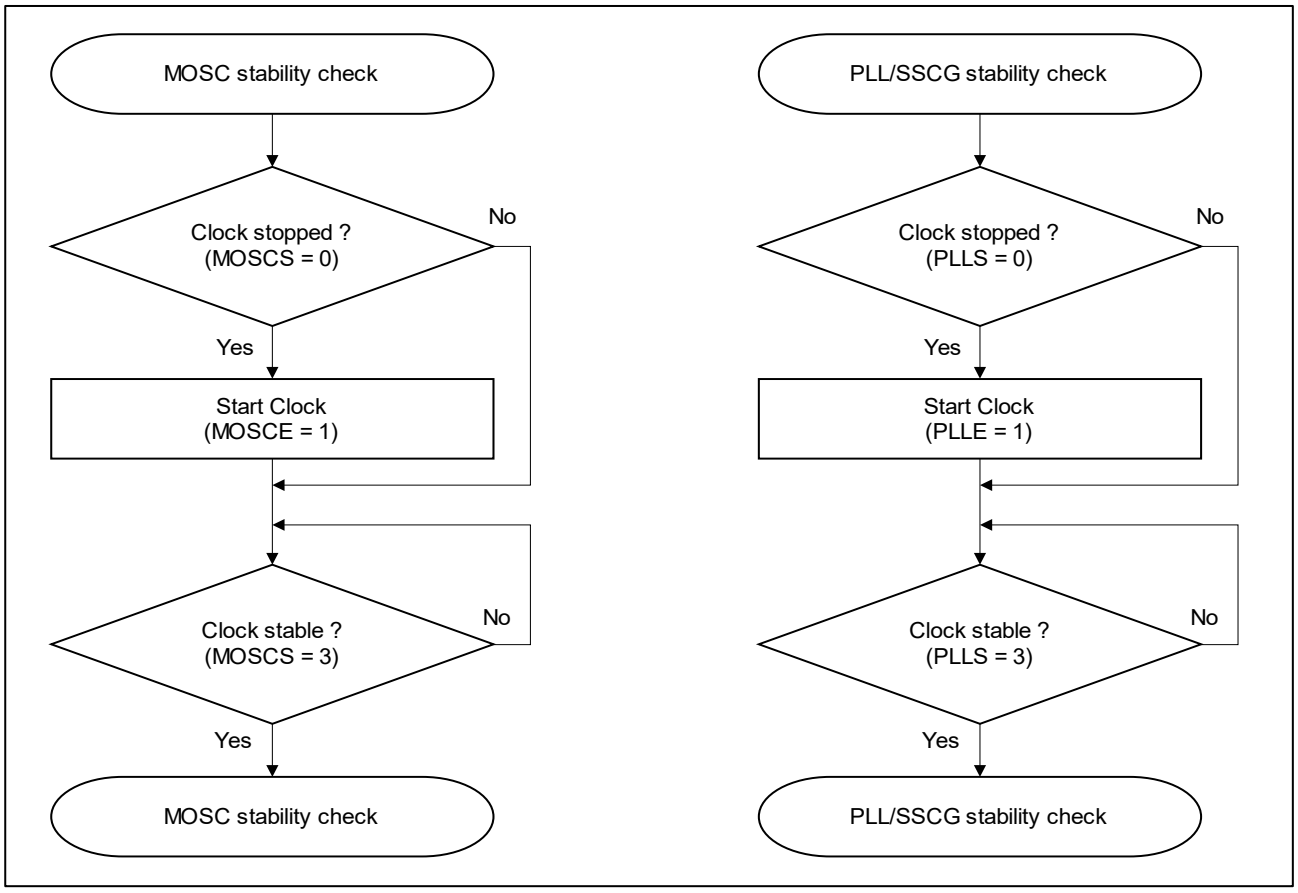
An example of the program code is shown in Figure 2-6. The clock gear-up function is called and the processing is performed within the function implemented in system_init.c.

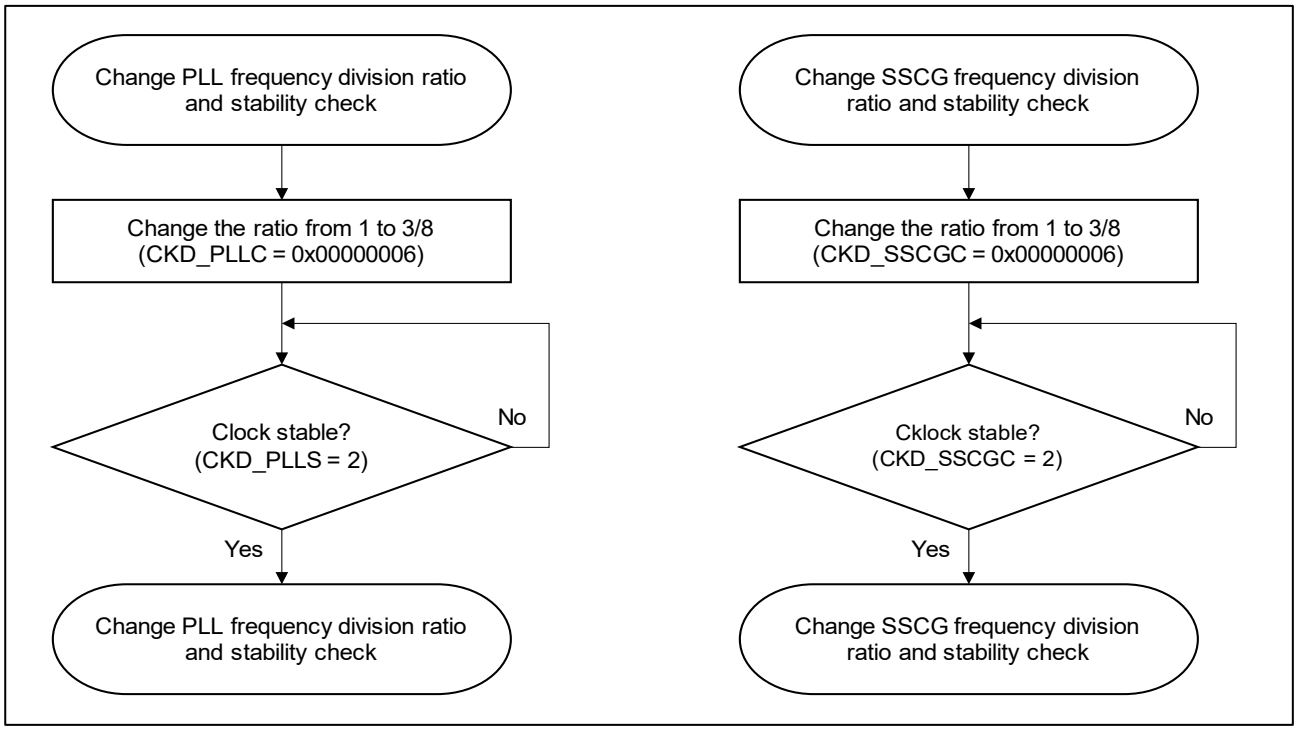
```
jarl    _hdwinit,    lp      ; initiaize hardware (clock gear up)
```

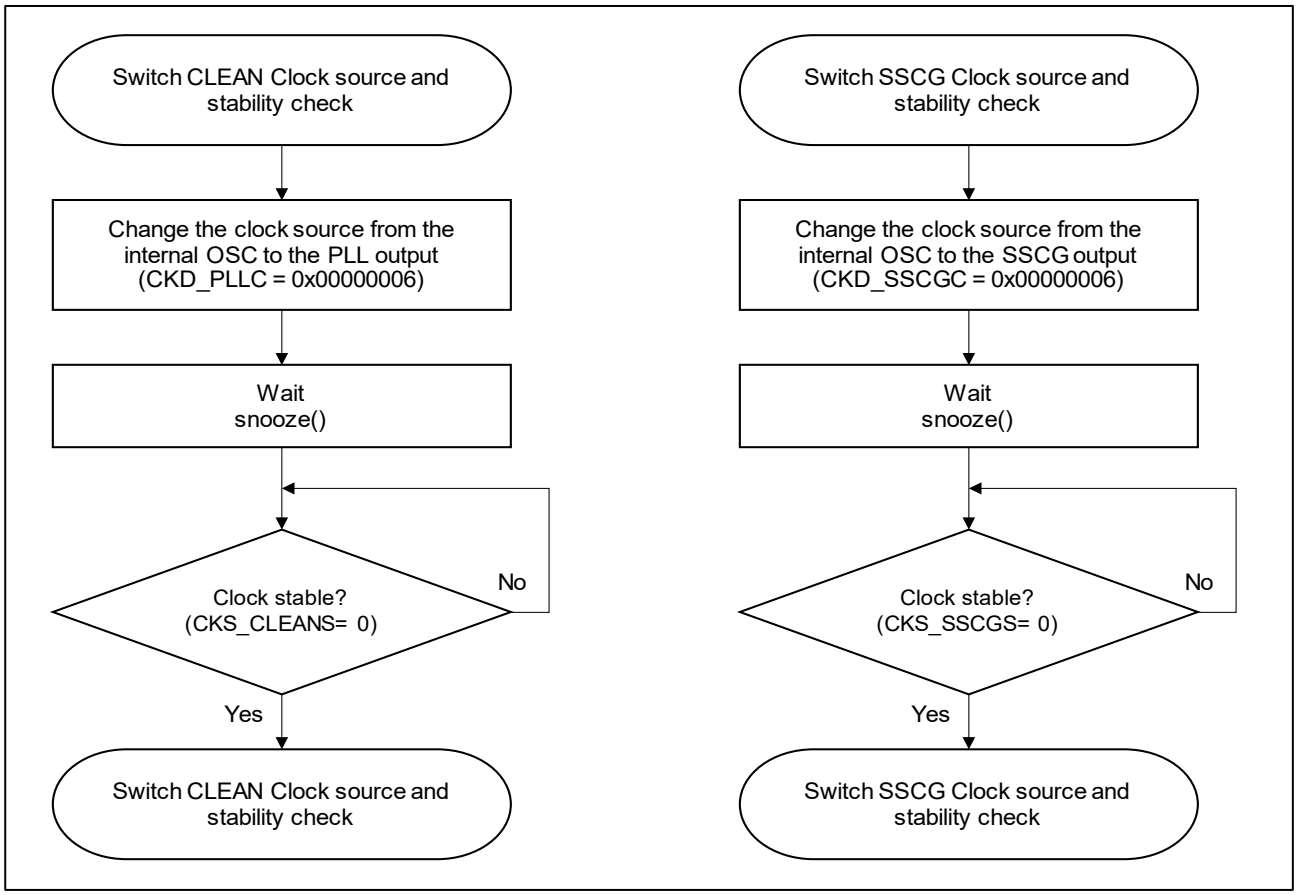
Figure 2-6 Example of clock gear up program code

The next section shows the clock gear up setting flow in this article.


clkGearUp function
Write to clock control register to remove register write protection (CLKKCPROT1 = 0xA5A5A501)
MOSC stability check
PLL/SSCG stability check
Change PLL frequency division ratio and stability check
Change SSCG frequency division ratio and stability check
Switch CLEAN Clock source and stability check
Switch SSCG Clock source and stability check
Step-by-step clock gear up
Clock control register write for register write protection (CLKKCPROT1 = 0xA5A5A500)


MOSC stability check
Clock stopped ?
(MOSCS = 0)
No
Yes
Start Clock
(MOSCE = 1)
Clock stable ?
(MOSCS = 3)
No
Yes
MOSC stability check
PLL/SSCG stability check
Clock stopped ?
(PLLS = 0)
No
Yes
Start Clock
(PLLE = 1)
Clock stable ?
(PLLS = 3)
No
Yes
PLL/SSCG stability check


Change PLL frequency division ratio
and stability check
Change the ratio from 1 to 3/8
(CKD_PLLC = 0x00000006)
Clock stable?
(CKD_PLLS = 2)
No
Yes
Change PLL frequency division ratio
and stability check
Change SSCG frequency division
ratio and stability check
Change the ratio from 1 to 3/8
(CKD_SSCGC = 0x00000006)
Cklock stable?
(CKD_SSCGC = 2)
No
Yes
Change SSCG frequency division
ratio and stability check


Switch CLEAN Clock source and stability check
Change the clock source from the internal OSC to the PLL output (CKD_PLLC = 0x00000006)
Wait snooze()
Clock stable? (CKS_CLEANS= 0)
No
Yes
Switch CLEAN Clock source and stability check
Switch SSCG Clock source and stability check
Change the clock source from the internal OSC to the SSCG output (CKD_SSCGC = 0x00000006)
Wait snooze()
Clock stable? (CKS_SSCGS= 0)
No
Yes
Switch SSCG Clock source and stability check

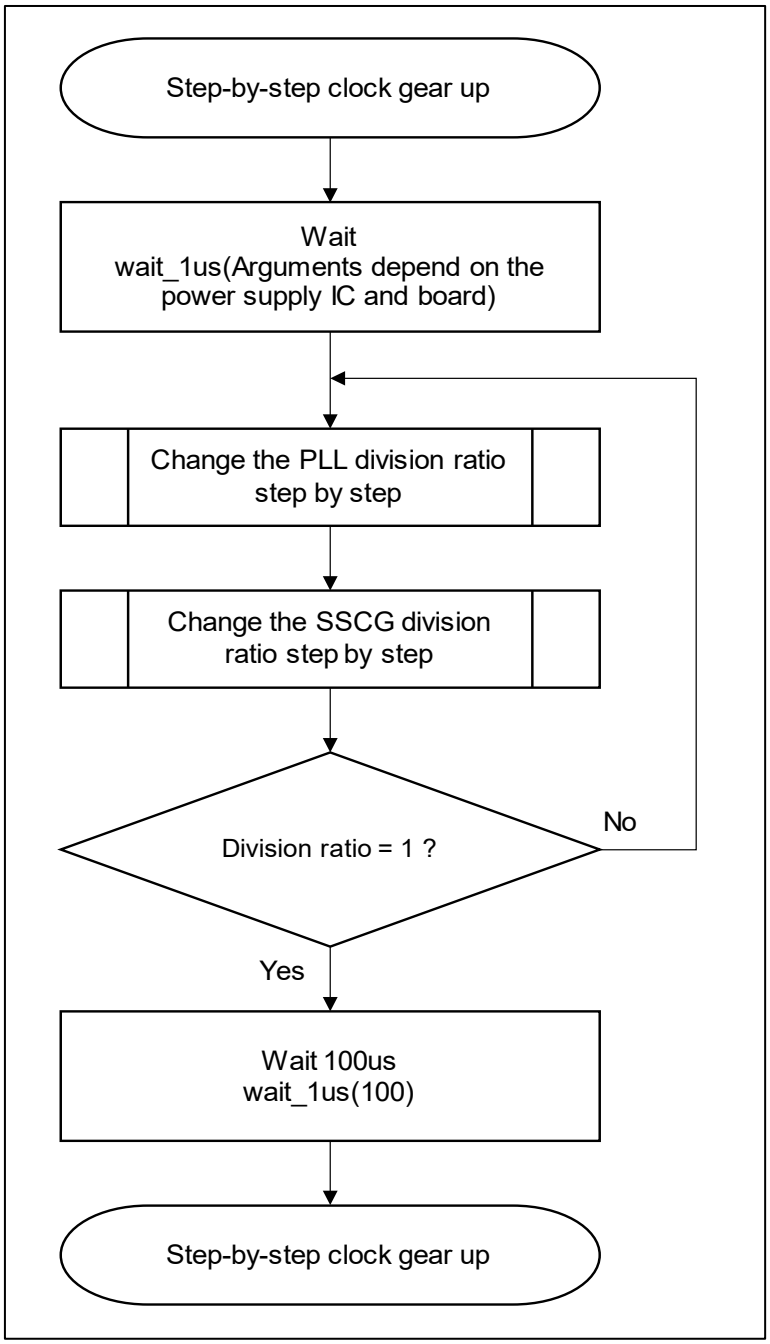


Figure 2-7 Clock gear up flows

2.3.4.4 RAM area initialization

This initializes LRAM (Local RAM) and CRAM (Cluster RAM).

In this project, the following RAM areas are initialized.

The RAM that each PE initializes is as follows:

Table 2-7 RAM list

PE	Initialization target	U2B24	U2B20	U2B10	U2B6
PE0	LRAM(self)	0xFDE0_0000 -0xFDE0_FFFF	0xFDE0_0000 -0xFDE0_FFFF	0xFDE0_0000 -0xFDE0_FFFF	0xFDE0_0000 -0xFDE0_FFFF
	CRAM0	0xFE00_0000 -0xFE05_7FFF	0xFE00_0000 -0xFE05_7FFF	0xFE00_0000 -0xFE05_7FFF 0xFE06_0000 -0xFE07_FFFF	0xFE00_0000 -0xFE05_7FFF
	CRAM0(Retention)	-	-	0xFE05_8000 -0xFE05_FFFF	0xFE05_8000 -0xFE05_FFFF
	CRAM1	0xFE08_0000 -0xFE0F_FFFF	0xFE08_0000 -0xFE0F_FFFF	0xFE08_0000 -0xFE0D_FFFF	-
	CRAM1(Retention)	-	-	0xFE0E_0000 -0xFE0F_FFFF	-
	CRAM2	0xFE10_0000 -0xFE15_FFFF	0xFE10_0000 -0xFE15_FFFF	-	-
	CRAM2(Retention)	0xFE16_0000 -0xFE17_FFFF	0xFE16_0000 -0xFE17_FFFF	-	-
	LRAM(CPU0)	0xFDC0_0000 -0xFDC0_FFFF	0xFDC0_0000 -0xFDC0_FFFF	0xFDC0_0000 -0xFDC0_FFFF	0xFDC0_0000 -0xFDC0_FFFF
PE1	LRAM(CPU1)	0xFDA0_0000 -0xFDA0_FFFF	0xFDA0_0000 -0xFDA0_FFFF	0xFDA0_0000 -0xFDA0_FFFF	0xFDA0_0000 -0xFDA0_FFFF
PE2	LRAM(CPU2)	0xFD80_0000 -0xFD80_FFFF	0xFD80_0000 -0xFD80_FFFF	0xFD80_0000 -0xFD80_FFFF	0xFD80_0000 -0xFD80_FFFF
PE3	LRAM(CPU3)	0xFD60_0000 -0xFD60_FFFF	0xFD60_0000 -0xFD60_FFFF	-	-
PE4	LRAM(CPU4)	0xFD40_0000 -0xFD40_FFFF	0xFD40_0000 -0xFD40_FFFF	-	-
PE5	LRAM(CPU5)	0xFD20_0000 -0xFD20_FFFF	0xFD20_0000 -0xFD20_FFFF	-	-

Some RAM, such as DTSRAM and MMCA RAM, are initialized by the RAM Initialization function of the hardware. For details on the RAM Initialization function, see "RH850/U2B-E Group User's Manual, 11.5.6 RAM Initialization."

A program code example is shown below.

```
        ; local ram address
        LOCAL_RAM_SELF_ADDR .set 0xFDE00000
        LOCAL_RAM_SELF_END .set 0xFDE0FFFF
(skip)
        ; cluster ram address
        CLUSTER_RAM0_1_ADDR0 .set 0xFE000000
        CLUSTER_RAM0_1_END0 .set 0xFE137FFF
(skip)
        ; clear Cluster RAM0 _1
        mov CLUSTER_RAM0_1_ADDR0, r6
        mov CLUSTER_RAM0_1_END0, r7
        jarl _zeroclr4, lp
        ; clear Local RAM(CPU0)
        mov LOCAL_RAM_SELF_ADDR, r6
        mov LOCAL_RAM_SELF_END, r7
        jarl _zeroclr4, lp
(skip)
_zeroclr4:
        br .L.zeroclr4.2
.L.zeroclr4.1:
        st.w r0, [r6]
        add 4, r6
.L.zeroclr4.2:
        cmp r6, r7
        bh .L.zeroclr4.1
```

Figure 2-8 Example of RAM initialization program code (PE0) (CS+)

2.3.4.5 Other CPU initialization

After RAM initialization processing for CPU0 is completed, the wakeUp_Othercpu function is called to start up and initialize CPU1 to CPU6.

```
_wakeUpOtherCPU:
    jarl    _wakeUp_Othercpu, lp    ; WakeUp Other CPU
```

Figure 2-9 Example of other CPU initialization program code

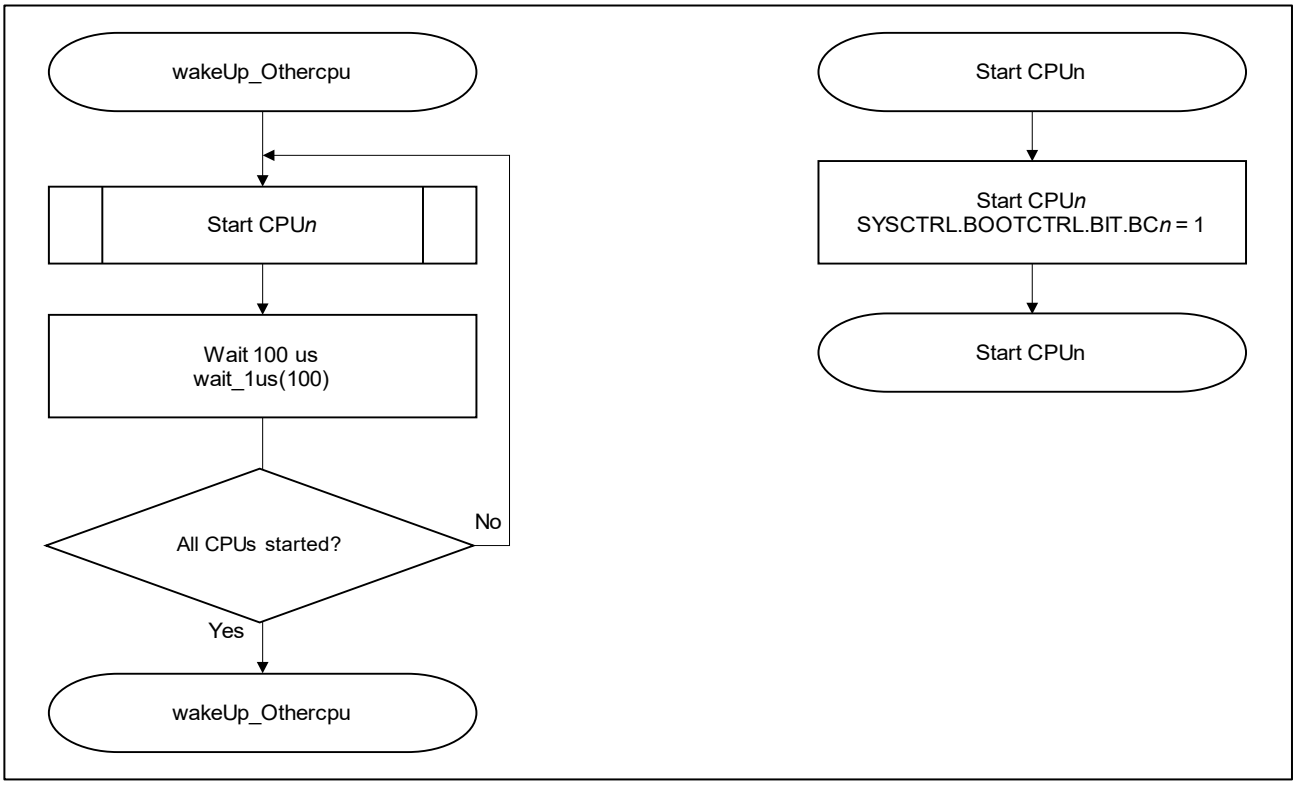


Figure 2-10 wakeUpOtherCPU function internal processing flowchart (*n* = 1 ~ 2)

2.3.4.6 Interrupt handler address setting

Set the base address of the EI level interrupt handler address table for table lookup in the INTBP register of each PE. In this sample code, the start address of the EIINTTBL_PEx section is set as the base address for the table lookup.

This process will be executed only if all of the following conditions are met.

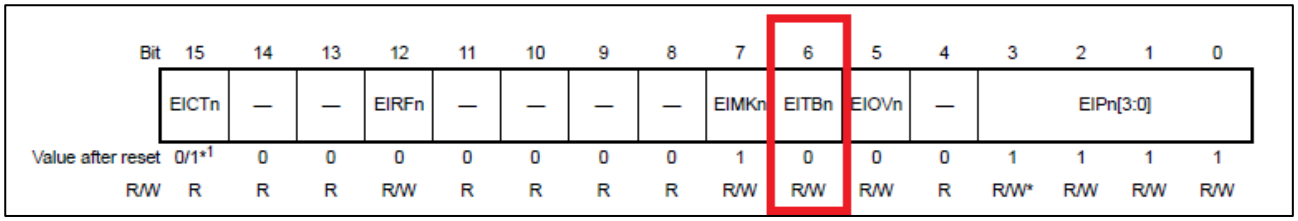
- USE_TABLE_REFERENCE_METHOD is 1.

An example of the program code is shown in Figure **2-11**.

```
$ifdef USE_TABLE_REFERENCE_METHOD
    mov     #__sEIINTTBL_PE0, r6
    jarl    _set_table_reference_method, lp     ; set table reference method
$endif
```

Figure 2-11 Example of interrupt handler address setting program code

For the EI level interrupt EIINTn, you can select either the direct branch method or the table reference method. The direct branch method is the default. To change to the table lookup method, select it with the interrupt control register EICn. The sixth bit, EITBn, is the interrupt vector method selection bit; if it is 0, it is the direct branch method, and if it is 1, it is the table reference method. Select 1 here.



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EICTn	—	—	EIRFn	—	—	—	—	EIMKn	EITBn	EIOVn	—	EIPn[3:0]			
Value after reset	0/1*1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R	R/W*	R/W	R/W	R/W

Figure 2-12 Interrupt control register EICn (n = channel number)

Below is a sample code example.

```
$ifdef USE_TABLE_REFERENCE_METHOD
;-------------------------------------------------------------------------
;set table reference method
;-------------------------------------------------------------------------
      ; interrupt control register address
      ICBASE      .set      0xFFFC0000

      .section    ".text", text
      .align      2
_set_table_reference_method:
      ldsr        r6, 4, 1          ; set INTBP

      ; Some interrupt channels use the table reference method.
      mov         ICBASE, r10       ; get interrupt control register address
      set1        6, 0[r10]         ; set INT0 as table reference
      set1        6, 2[r10]         ; set INT1 as table reference
      set1        6, 4[r10]         ; set INT2 as table reference

      jmp         [lp]
$endif
```

Figure 2-13 Example of table reference mode setting code

2.3.4.7 Each pointer setting

After setting the interrupt handler address, we move on to the processing implemented in cstartn.asm.

Here, we set the stack pointer, global pointer, and element pointer.

An example of the program code for PE0 is shown in Figure 2-14.

```
STACKSIZE   .set        0x5000
            .section    ".stack.bss",    bss
            .align      4
            .ds         (STACKSIZE)
            .align      4
            _stacktop:

STACKSIZE   .set        0x5000
            .section    ".stack.bss", bss
             .public     _stacktop_pm0
(skip)
            mov         #_stacktop, sp   ;  set sp register
            mov         #__gp_data, gp   ;  set gp register
            mov         #__ep_data, ep   ;  set ep register
```

Figure 2-14 Example of program code for setting each pointer

2.3.4.8 RAM area setting

The .data section (RAM section with initial values) and the .bss section (RAM section without initial values) are initialized by __INITSCT_RH processing.

By calling the __INITSCT_RH processing with the start and end addresses of the RAM section initialization table with initial values set in the parameter registers (r6, r7) and the start and end addresses of the RAM section initialization table without initial values set in the parameter registers (r8, r9), data is copied from ROM to the .data section and the .bss section is cleared to 0.

The RAM section initialization table with initial values must be placed in the .INIT_DSEC.const section, and the start address of the source ROM section, the end address of the source ROM section, and the start address of the destination RAM section are set in the table.

The RAM section initialization table without initial values must be placed in the .INIT_BSEC.const section, and the start address of the RAM section to be cleared and the end address of the RAM section to be cleared are set in the table.

An example of the program code is shown in Figure 2-15.

```
;-----------------------------------------------------------------------
;section initialize table
;-----------------------------------------------------------------------
    .section    ".INIT_DSEC.const", const
    .align      4
    .dw         #__s.data, #__e.data, #__s.data.R

    .section    ".INIT_BSEC.const", const
    align       4
    .dw         #__s.bss,#__e.bss


    mov         #__s.INIT_DSEC.const, r6
    mov         #__e.INIT_DSEC.const, r7
    mov         #__s.INIT_BSEC.const, r8
    mov         #__e.INIT_BSEC.const, r9
    jarl32      __INITSCT_RH, lp            ; initialize RAM area
```

Figure 2-15 Example of RAM area setting sample code

Some RAMs, such as DTSRAM and GTM RAM, are initialized by the RAM Initialization function of the hardware. For details of the RAM Initialization function, please refer to the user's manual of the device.

2.3.4.9 Coprocessor Settings

When using a fully-connected neural network that uses the floating-point unit (FPU) and extended floating-point unit (FXU) built into the U2B-E, the following settings must be made.

An example of the program code is shown in Figure 2-16.

```
    ; enable FPU
$if 1; disable this block when not using FPU
    stsr    6, r10, 1       ; r10 <- PID
    shl     21, r10
    shr     30, r10
    bz      .L1             ; detecting FPU
    stsr    5, r10, 0       ; r10 <- PSW
    movhi   0x0001, r0, r11
    or      r11, r10
    ldsr    r10, 5, 0       ; enable FPU

    movhi   0x0002, r0, r11
    ldsr    r11, 6, 0       ; initialize FPSR
    ldsr    r0, 7, 0        ; initialize FPEPC
.L1:
$endif

; enable FXU
$if 1 ; disable this block when not using FXU
    sts     r6, r11, 1      ; r11 <- PID
    mov     r11, r10
    shr     24, r11
    cmp     6, r11
    bl      .L2
    shl     20, r10
    shr     31, r10
    bz      .L2             ; detecting FXU
    sts     r5, r10, 0      ; r10 <- PSW
    movhi   0x0002, r0, r11
    or      r11, r10
    ldsr    r10, 5, 0       ; enable FXU

    movhi   0x0002, r0, r11
    ldsr    r11, 6, 10      ; initialize FXSR
    ldsr    r0, 8, 10       ; initialize FXST
    ldsr    r0, 10, 10      ; initialize FXCFG
.L2:
$endif
```

Figure 2-16 Example of coprocessor configuration sample code

2.4 Table reference interrupt method (CS+)

This section describes the table reference interrupt method.

2.4.1 Table reference interrupt configuration

The RH850 has two types of interrupts: direct vector and table reference.

With the table reference method, the word data stored in the handler address is read and a jump is made to the address pointed to by that word data.

Interrupt vectors are written in vecttbl*n*.asm (*n*=0~2).

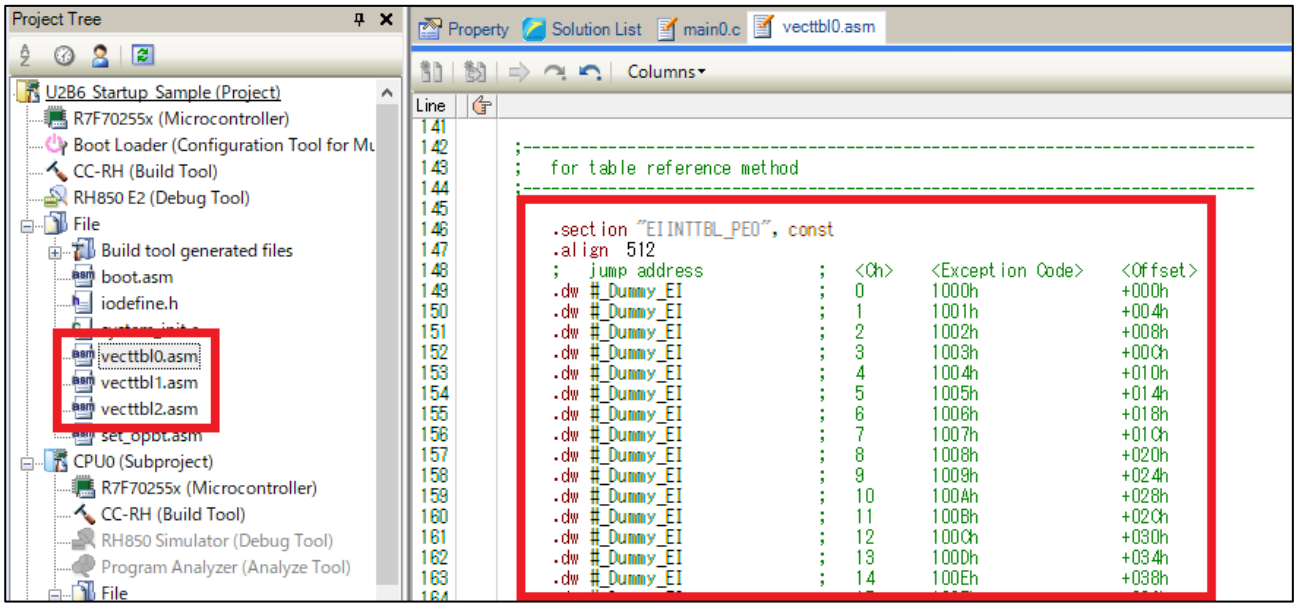


Figure 2-17 Vector table

2.4.2 Writing interrupt functions (Pragma interrupt)

This section explains how to write the definition of a function to be executed when an interrupt occurs.

When writing an interrupt function in C, use the #pragma interrupt directive. This will cause the function written in C to be compiled as an interrupt function. For information on interrupt specifications, refer to the CS+ manual.

```
#pragma  interrupt  Function name (Interrupt Specifications [,Interrupt Specifications])
```

Note: The following procedure will result in a compilation error.

- Using arguments or return types other than Void type
- Calling an interrupt handler as if it were a normal function

2.4.3 How to handle interrupts on a CPU other than CPU0

The RH850/U2B-E has a multi-core configuration.

When programming an interrupt function in the CS+ multi-core development environment, the interrupt function is linked to the original project (boot loader project).

If the interrupt function is defined in the application project for each CPU, the interrupt function must be shared with the boot loader project.

To share an interrupt function with the boot loader project, follow the steps below.

① Interrupt function definition

The #pragma section directive places the function in the section for interrupt functions.

```
#pragma section .cpu1.intfunc

#pragma interrupt int_test1
void int_test1 (void) {
                    __nop();
```

Figure 2-18 Example of interrupt function definition code

② Registering a section for interrupt functions

In the build tool properties, specify the section for interrupt functions in “(For multi-core) Section that outputs external defined symbols to the file” on the “Section” tab of “Link Options”.

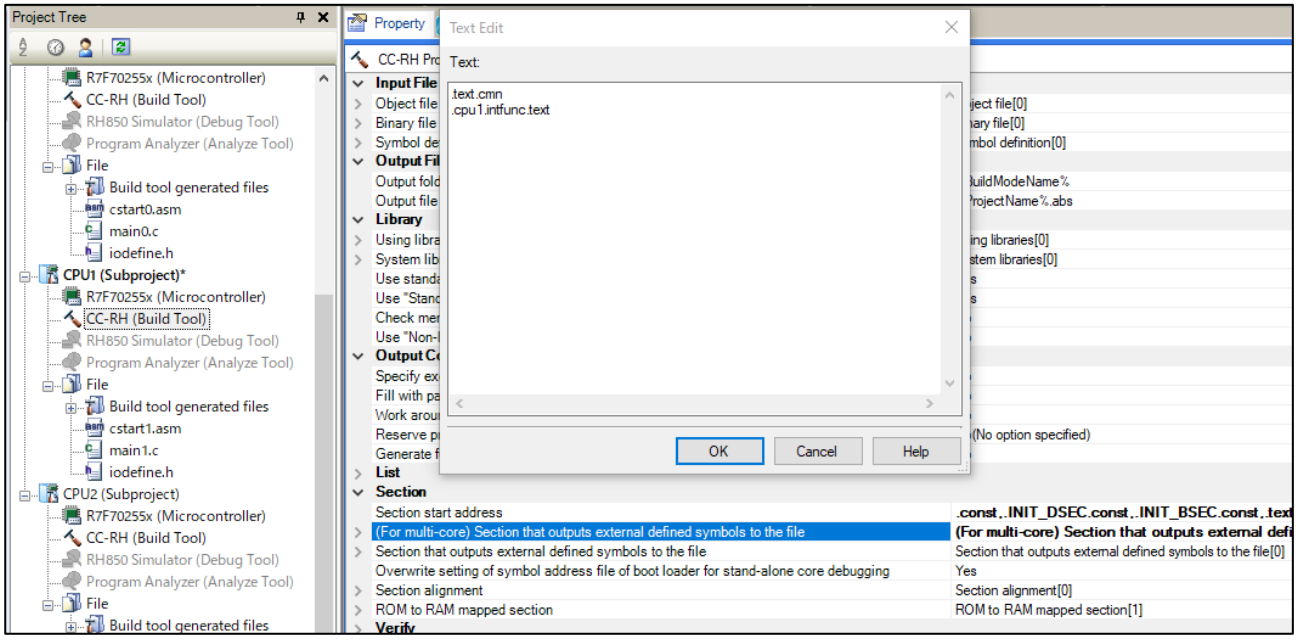


Figure 2-19 Example of section settings for multi-core interrupt functions

3. MULTI

3.1 Startup files

The list of files related to Startup is shown in Table 3-1.

Table 3-1 Files(*n* = 0 ~ 2)

File name	Directory	Description
system_init.c	root/startup	Clock gear up process
boot.850		Startup process
vecttbl_PE*n*.c		Vector table
eiint_vecttbl_PE*n*.c		
RH850_U2B6.ld		Memory and section settings
main_pe*n*.c	root/src/core*n*	Main process (user application)
intprg_PE*n*.c		Interrupt Functions

Note. root is the root of the project.

3.2 Section settings

3.2.1 Section list

Examples of the main sections related to Startup are shown in ~. In MULTI, the address of each section is set in the linker directive file (*.ld).

Table 3-2 Startup sections (ROM、*n* = 1 ~ 2)

Section	Allocation data
.RESET	Start of RESET vector
.EIINTTBL	Start of interrupt vector
.rozdata	Constant data in ZDA area
.robase	Initialize textpointer TP for SDA addressing
.rosdata	Constant datas in SDA16 area
.rodata	Constant datas in normal area
.rodata_EIInt	Constant data EIIntTbl
.rodata_IntBP	Constant data IntBPTbl
.text	Program code area
.startup	Startup code .text section
.fixaddr	GHS internal
.fixtype	
.secinfo	
.syscall	
.romdata	Constant data to initialize variables (copied to RAM at startup)
.romzdata	Constant data to initialize variables in ZDA area (copied to RAM at startup)
.romsdata01	Constant data to initialize variables in SDA area (copied to RAM at startup)
.romsdata	Constant data to initialize variables in SDA area (copied to RAM at startup)
.romtdata	Constant data to initialize variables in TDA area (copied to RAM at startup)
.EBASE_PEn	Start of EBASE
.EIINTTBL_PEn	Start of interrupt vector
.PEn_rozdata	Constant data in ZDA area
.PEn_robase	Initialize textpointer TP for SDA addressing
.PEn_rosdata	Constant data in SDA16 area
.PEn_rodata	Constant data in normal area
.PEn_text	Program code area

Table 3-3 Startup sections (Cluster RAM)

Section	Allocation data
.stack	Stack area
.sdabase	
.sdata01	Initialized data in SDA16 area
.sbss01	Zero initialized data in SDA16 area
.sdata	Initialized data in SDA23 area
.sbss	Zero initialized data in SDA23 area
.data	Initialized data
.bss	Zero initialized data
.zdata	Initialized data in ZDA area
.zbss	Zero initialized data in ZDA area
.ramtext	Initialized and zero-initialized data in TDA area
.tdata	Initialized and zero-initialized data in TDA area

Depending on the device or project, the names may differ, some sections may not be required, or other sections may be required. For details about sections, see the user's manual for MULTI or the device.

3.2.2 Section setting method

This section describes how to set sections using MULTI.

The following shows how to specify sections when specifying or adding sections in a program in assembler or C language.

When adding a section, in addition to specifying it below, you will also need to modify section.ld.

- Assembler

  .section *Section name* [,"Context(a|b|w|z)"] [> Location address]

  Section name：Specify the name of the section

  Context：Specify the context. When specifying multiple contexts, specify them consecutively within double quotes.

  Location Address：Specifies the address where the section is located.

The attributes that can be specified are described below.

Table 3-4 List of contexts that can be specified in .section

#	Context	Description
1	a	This means that the section has memory allocated to it that is not used for debugging or symbol information.
2	b	This means the section can have BSS semantics. In a .bss section, normal data pseudo-ops such as .word and .byte are allowed, but all values specified by these pseudo-ops are discarded by the assembler. Only the size of the section is recorded in the ELF output file, the contents of the section are omitted. When the section is downloaded to the target, space is allocated for the section, but no data is downloaded to the section. Instead, the startup code initializes all bytes in the section to zero.
3	w	The section is writable.
4	z	The section contains executable code.

- C language

  #pragma ghs section [Section type = ”Section name”]

  Section type：Specify the type of section to change the allocation.

  Section name：Specify a name for the section.

  The section types that can be specified are shown below.

Table 3-5 List of section types that can be specified with #pragma ghs section

#	Section type	Default programs section
1	bss	.bss
2	data	.data
3	text	.text
4	rodata	.rodata
5	sbss	.sbss
6	sdata	.sdata
7	rosdata	.rosdata
8	zbss	.zbss
9	zdata	.zdata
10	rozdata	.rozdata

3.3 Startup process (MULTI)

3.3.1 Conditional assembly control instruction definitions

The definitions for the conditional assembly control instructions are shown in Table 3-6. The conditional assembly control instruction definitions are defined in boot.850.

Table 3-6 List of definitions for conditional assembly control instructions

Definition name	Value	Description
USE_TABLE_REFERENCE_METHOD		Defines whether to use the table reference method as the interrupt vector method. The initial value is 1 (use table reference method).
	0	Does not use the table reference method.
	1	Using the table reference method。

3.3.2 Overall flow

The overall flow of the startup process is shown below.

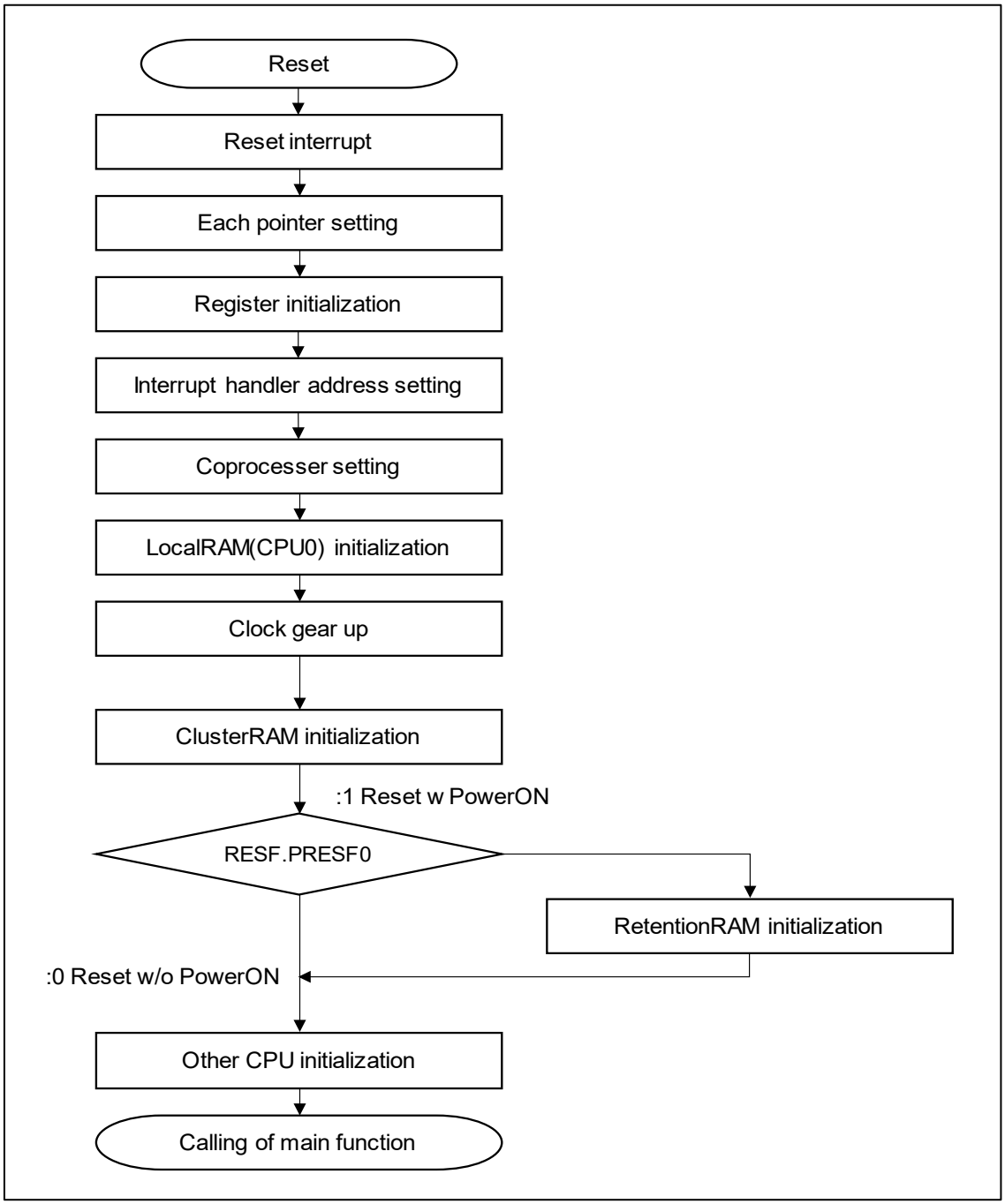


Figure 3-1 Startup flowchart

3.3.3 Processing overview

An overview of each treatment is shown in Table **3-7**.

Table 3-7 Implementation list of each process (*n* = 0 ~ 2)

#	Process name	Label	Implementation files	Note
1	Reset interrupt process	-	vecttbl_PE0.850	Implemented with interrupt vector table
2	Each pointer setting	_cstart_pm*n*	boot.850	
3	Register initialization	_startUp850	boot.850	
4	Interrupt handler address setting		boot.850	
5	Coprocessor setting	__skip_coresync_init	boot.850	
6	Clock gear up setting	_hdwinit_PE0	boot.850 system_init.c	Process only if the processing PE is PE0
7	RAM area initialization	_hdwinit_PEn _RetentionRAMClear _LocalRAMCPU0Clear	boot.850	RetentionRAM is processed only when the processing PE is PE0.
8	Other CPU initiaization	_WakeUpOtherCPU	boot.850 system_init.c	Process only if the processing PE is PE0
9	Calling the main function	-	-	

3.3.4 Details of each process

We will explain the details of each process.

Unless otherwise specified, the same process will be performed for all PE0 to PE6.

3.3.4.1 Reset interrupt

When power is turned on, the address of the RESET vector (RESET section) is transitioned to.

The RESET vector transitions to __start processing.

PE0 starts when the device itself is powered on.

An example of program code is shown in Figure 3-2.

```
    .section ".RESET", .text
    .align      4
    .global     _reset
_reset:
    .extern     __startUp850
    jr32        __startUp850    --RESET
```

Figure 3-2 Example of reset interrupt program code

3.3.4.2 Each pointer setting

Here we set the stack pointer, global pointer, and text pointer.

An example of program code for PE0 is shown in Figure 3-3.

```
 -- SP(r3)
movhi   hi( ___ghsend_stack - 8 ), zero, sp
movea   lo( ___ghsend_stack - 8 ), sp, sp       ---- set stack pointer

-- GP(r4)
movhi   hi(__gp), zero, gp
movea   lo(__gp), gp, gp              ---- set gp

-- GP(r5)
movhi   hi(__tp), zero, tp
movea   lo(__tp), tp, tp              ---- set tp
```

Figure 3-3 Example of program code for setting each pointer

3.3.4.3 Register initialization

Initialize the general-purpose registers and system registers of each PE.

Table 3-8 shows a list of registers to be initialized. The setting values are examples. Initialize with optimal values for your system.

Table 3-8 Initialization register list

#	Register classification	Register name	Example of settings	Description
1	Program register	r1~r31	0	
2	Basic system registers	EIPC	0	
3		FEPC	0	
4		CTPC	0	
5		EIWR	0	
6		FEWR	0	
7		INTBR	0	
8		EBASE	0	
9		MEA	0	
10		MEI	0	
11		RBIP	0	
12	MPU function register	MCA	0	
13		MCS	0	
14		MCR	0	
15		MPIDX	0	
16		MPLA	0	
17		MPUA	0	
18		MPAT	0	
19		MPID0	0	
20		MPID1	0	
21		MPID2	0	
22		MPID3	0	
23		MPID4	0	
24		MPID5	0	
25		MPID6	0	
26		MPID7	0	
27		MCI	0	
28	Cache operation function register	ICTAGL	0	
29		ICTAGH	0	
30		ICDATL	0	
31		ICDATH	0	
32		ICERR	0	
33	FPU function register	FPSR	0x00220000	
34		FPEPC	0	
35		FPST	0	

#	Register classification	Register name	Example of settings	Description
36		FPCC	0	
37	Virtualization support function system registers	HVSB	0	
38	Guest context register	GMEIPC	0	
39		GMFEPC	0	
40		GMEBASE	0	
41		GMINTBP	0	
42		GMEIWR	0	
43		GMFEWR	0	
44		GMMEA	0	
45		GMMEI	0	

An example of the program code is shown in Figure 3-4.

```
        $nowarning
        mov         r0, r1
        $warning
        mov         r0, r2
(skip)
        ldsr r0, 0, 0 ; SR0,0 EIPC
        ldsr r0, 2, 0 ; SR2,0 FEPC
        ldsr r0, 16, 0 ; SR16,0 CTPC
(skip)
```

Figure 3-4 Example of register initialization program code

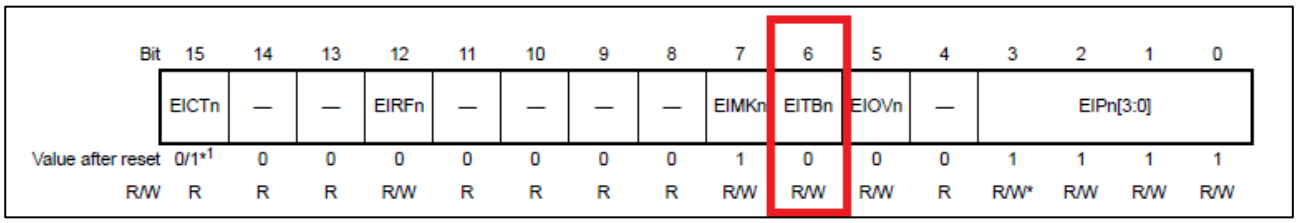
3.3.4.4 Interrupt handler address setting

Set the base address of the EI level interrupt handler address table for table lookup in the INTBP register of each PE.

This process will be executed only if all of the following conditions are met.

- USE_TABLE_REFERENCE_METHOD is 1.

For the EI level interrupt EIINTn, you can select either the direct branch method or the table reference method. The direct branch method is the default. To change to the table lookup method, select it with the interrupt control register EICn. The sixth bit, EITBn, is the interrupt vector method selection bit; if it is 0, it is the direct branch method, and if it is 1, it is the table reference method. Select 1 here.



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EICTn	—	—	EIRFn	—	—	—	—	EIMKn	EITBn	EIOVn	—	EIPn[3:0]			
Value after reset	0/1*1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R	R/W*	R/W	R/W	R/W

Figure 3-5 Interrupt control register EICn (n = channel number)

Below is a sample code example.

```
#if (USE_TABLE_REFERENCE_METHOD==1)
    di

    stsr    0, r10, 2    --get PEID

    -- IntBP
    mov     4, r11
    mulu    r10, r11, r0
    mov     __INTBPTBL_ADDR    ,r12
    add     r12, r11
    ld.w    0[r11],    r13
    ldsr    r13,  4, 1    -- INTBP

    -- then set 0 to PSW.EBV -> RBASE=EBASE
    mov     0xFFFF7FFF,        r11
    stsr    5,    r12,    0    -- load PSW
    and     r11,    r12    -- Clear EBV
    ldsr    r12,    5,    0    -- set PSW

    -- EBASE
    mov     4, r11
    mulu    r10, r11, r0
    mov     __EIINTTBL_ADDR    ,r12
    add     r12, r11
    ld.w    0[r11],    r13
    ldsr    r13,  3, 1    -- EBASE

    stsr    2, r11, 1    -- RBASE
    cmp     r13, r11
    be      __EBASE_INIT_END
    synci

    -- then set 1 to PSW.EBV -> RBASE!=EBASE
    mov     0x00008000,        r11
    stsr    5,    r12,    0        -- load PSW
    or    r11,    r12              -- set EBV
    ldsr    r12,  5,    0          -- set PSW

__EBASE_INIT_END:
```

Figure 3-6 Example of table reference mode setting

3.3.4.5 Coprocessor settings

When using a fully-connected neural network that uses the floating-point unit (FPU) and extended floating-point unit (FXU) built into U2B, the following settings must be made.

An example of the program code is shown in Figure 3-7.

```
-- FPU --
stsr        5, r10, 0       --r10 <- PSW

movhi       0x0001, r0, r11
or          r11, r10
ldsr        r10, 5, 0       --enable FPU
synci

movhi       0x0022, r0, r11
ldsr        r11, 6, 0       --initialize FPSR
ldsr        r0,  7, 0       --initialize FPEPC
ldsr        r0,  8, 0       --initialize FPST
ldsr        r0,  9, 0       --initialize FPCC


 -- FXU Check
stsr        6, r10, 1       -- get PID
mov         0x0800,r11
and         r11, r10
cmp         r10, r0
bz          __start_PE

-- FXU--

stsr        5, r10, 0       --r10 <- PSW
movhi       0x0002, r0, r11
or          r11, r10
ldsr        r10, 5, 0       --enable  FXU
synci

movhi       0x0022, r0, r11
ldsr        r11, 6, 10      --initialize FXSR
ldsr        r0,  8, 10      --initialize FXST
ldsr        r0, 12, 10      --initialize FXXC
ldsr        r0, 13, 10      --initialize FXXP
```

Figure 3-7 Example of coprocessor configuration sample code

3.3.4.6 Clock gear up

After PE0 starts up, change the system clock sources (CLK_SYS_CLEAN and CLK_SYS_SSCG) to CLK_PLLO and CLK_SSCGO to gear up the clock.

This process will be executed only if all of the following conditions are met.

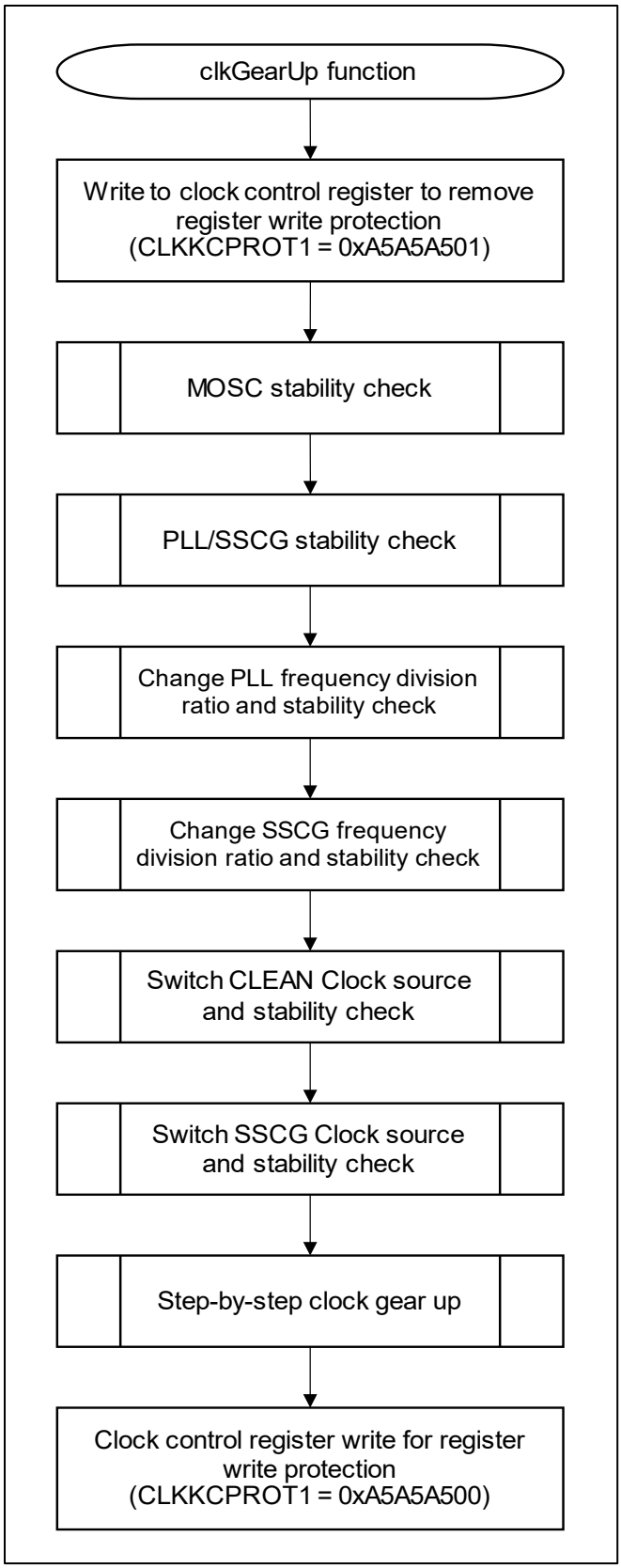
- The processing PE is PE0 (PEID register bit2:0 (PEID) = 0)

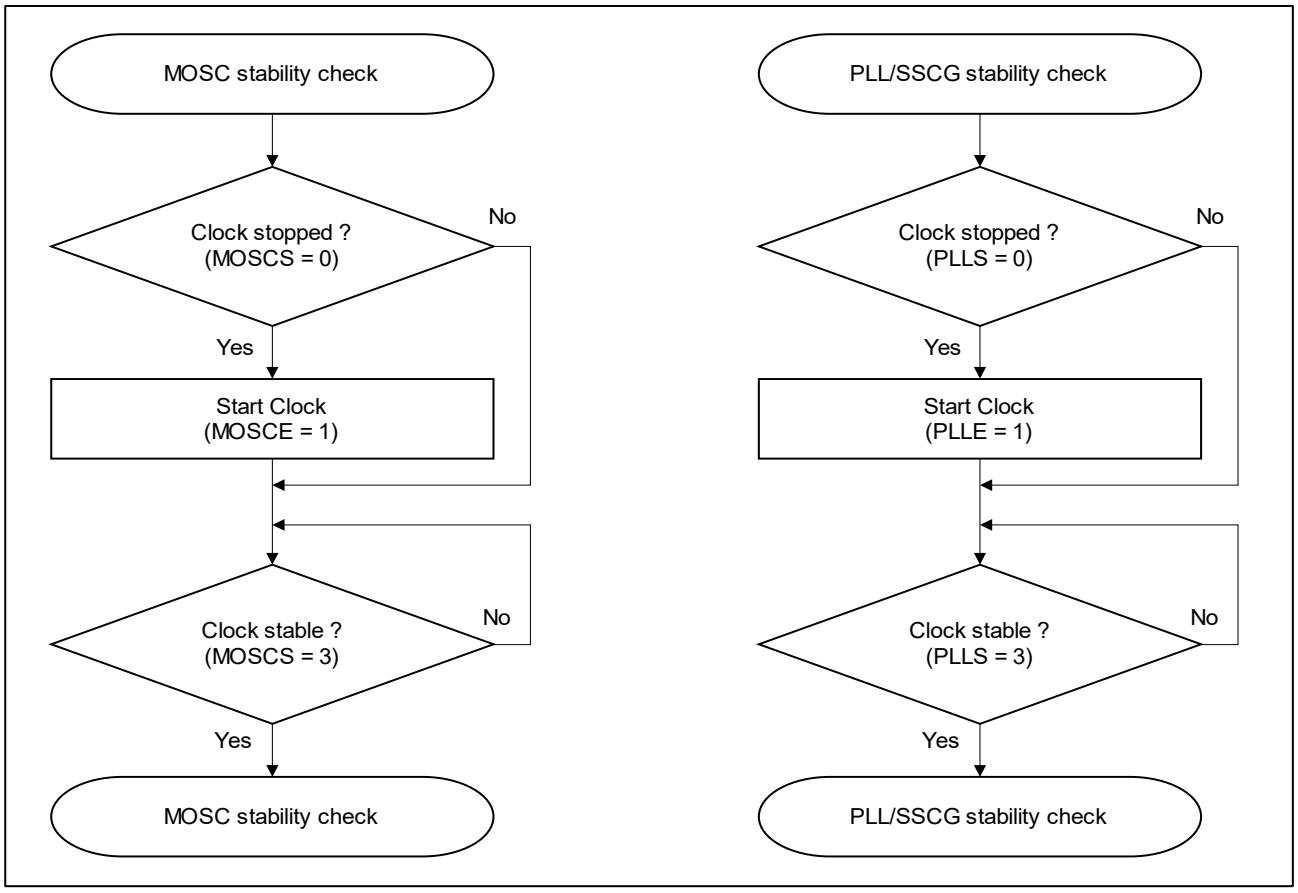
An example of the program code is shown in Figure 3-8. The clock gear-up function is called and the processing is performed within the function implemented in system_init.c.

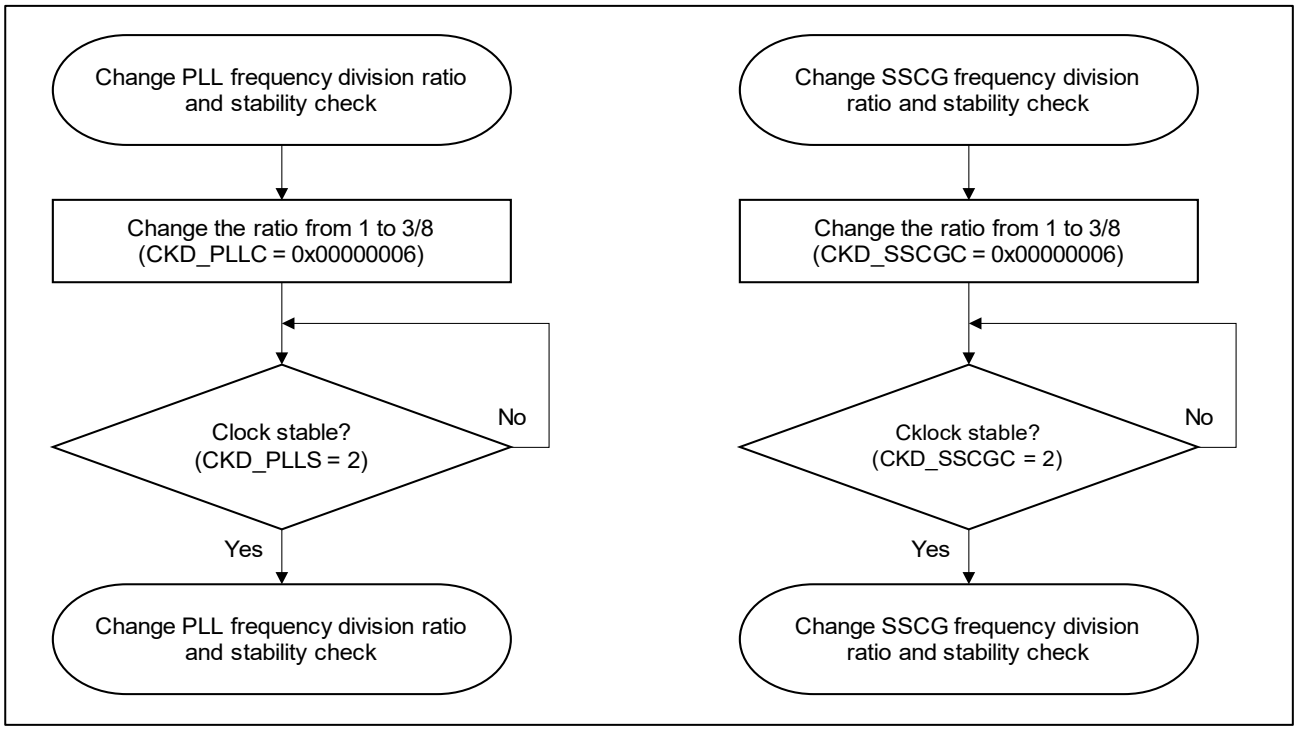
```
jarl   _hdwinit, lp   --initialize hardware(clock gear up)
```

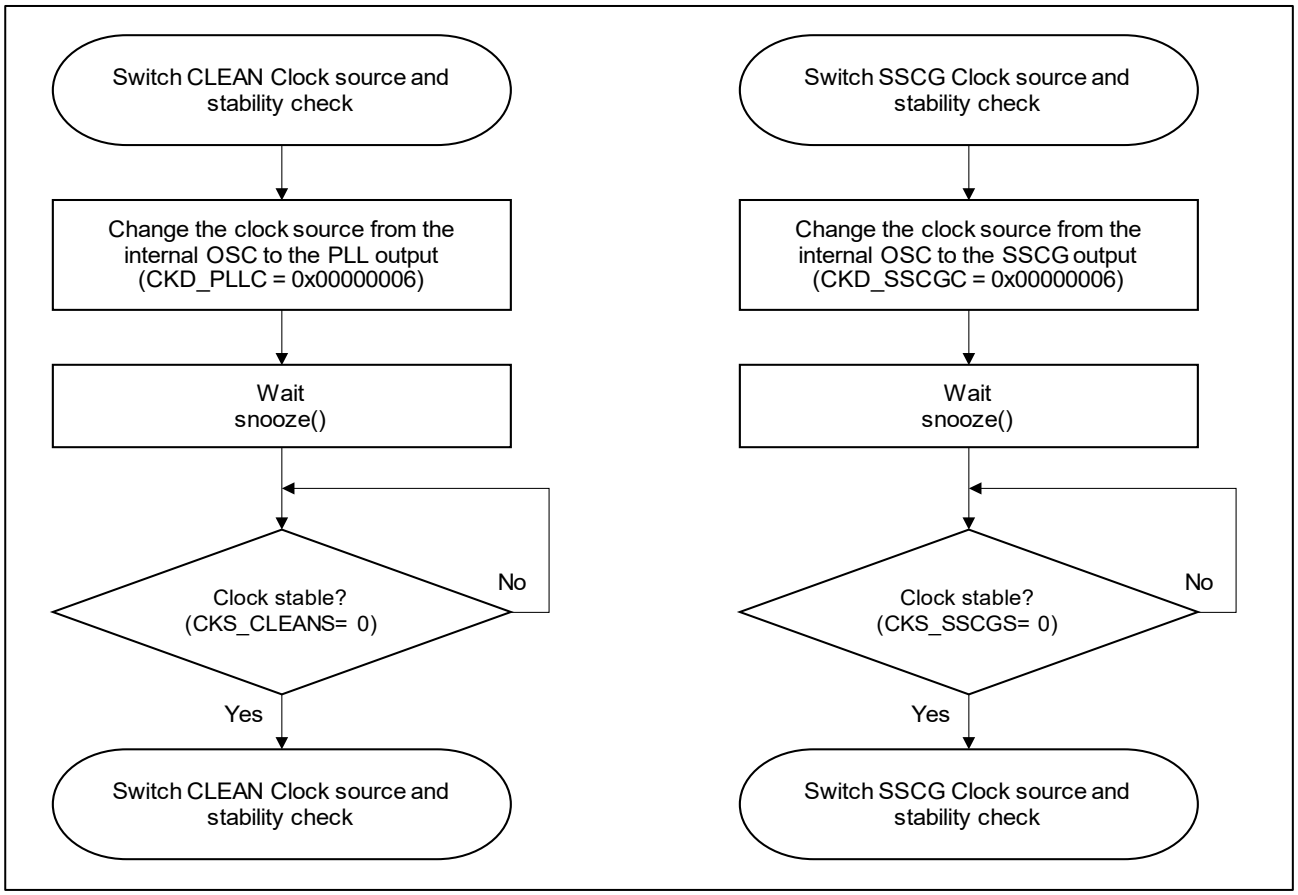
Figure 3-8 Example of clock gear up program code

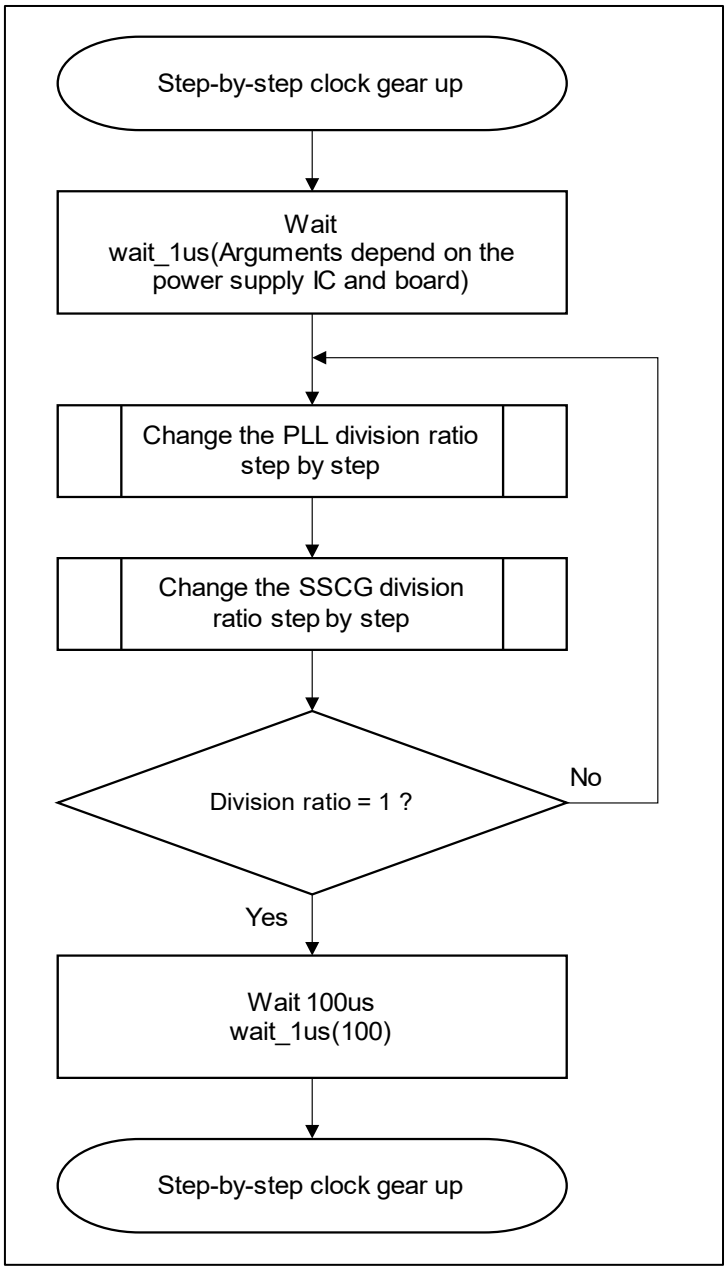
The next section shows the clock gear up setting flow in this article.


clkGearUp function
Write to clock control register to remove register write protection (CLKKCPROT1 = 0xA5A5A501)
MOSC stability check
PLL/SSCG stability check
Change PLL frequency division ratio and stability check
Change SSCG frequency division ratio and stability check
Switch CLEAN Clock source and stability check
Switch SSCG Clock source and stability check
Step-by-step clock gear up
Clock control register write for register write protection (CLKKCPROT1 = 0xA5A5A500)


MOSC stability check
Clock stopped ?
(MOSCS = 0)
No
Yes
Start Clock
(MOSCE = 1)
Clock stable ?
(MOSCS = 3)
No
Yes
MOSC stability check
PLL/SSCG stability check
Clock stopped ?
(PLLS = 0)
No
Yes
Start Clock
(PLLE = 1)
Clock stable ?
(PLLS = 3)
No
Yes
PLL/SSCG stability check


Change PLL frequency division ratio
and stability check
Change the ratio from 1 to 3/8
(CKD_PLLC = 0x00000006)
Clock stable?
(CKD_PLLS = 2)
No
Yes
Change PLL frequency division ratio
and stability check
Change SSCG frequency division
ratio and stability check
Change the ratio from 1 to 3/8
(CKD_SSCGC = 0x00000006)
Cklock stable?
(CKD_SSCGC = 2)
No
Yes
Change SSCG frequency division
ratio and stability check


Switch CLEAN Clock source and stability check
Change the clock source from the internal OSC to the PLL output (CKD_PLLC = 0x00000006)
Wait snooze()
Clock stable? (CKS_CLEANS= 0)
No
Yes
Switch CLEAN Clock source and stability check
Switch SSCG Clock source and stability check
Change the clock source from the internal OSC to the SSCG output (CKD_SSCGC = 0x00000006)
Wait snooze()
Clock stable? (CKS_SSCGS= 0)
No
Yes
Switch SSCG Clock source and stability check


Step-by-step clock gear up
Wait
wait_1us(Arguments depend on the
power supply IC and board)
Change the PLL division ratio
step by step
Change the SSCG division
ratio step by step
Division ratio = 1 ?
No
Yes
Wait 100us
wait_1us(100)
Step-by-step clock gear up

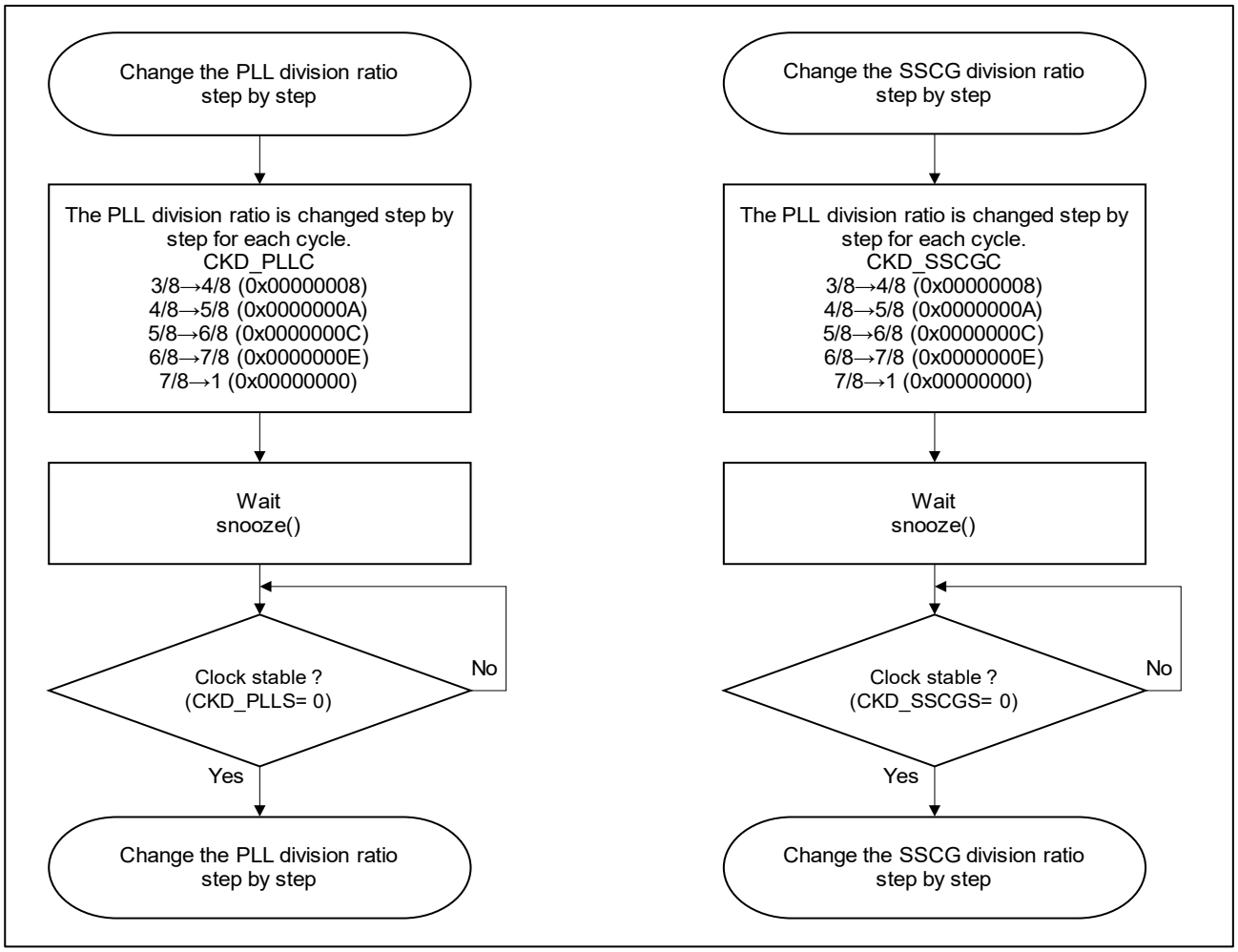


Figure 3-9 Clock gear up flow

3.3.4.7 RAM area initialization

This initializes LRAM (Local RAM) and CRAM (Cluster RAM).

In this project, the following RAM areas are initialized.

The RAM that each PE initializes is as follows:

Table 3-9 RAM list

PE	Initialization target	U2B24	U2B20	U2B10	U2B6
PE0	LRAM(self)	0xFDE0_0000 -0xFDE0_FFFF	0xFDE0_0000 -0xFDE0_FFFF	0xFDE0_0000 -0xFDE0_FFFF	0xFDE0_0000 -0xFDE0_FFFF
	CRAM0	0xFE00_0000 -0xFE05_7FFF	0xFE00_0000 -0xFE05_7FFF	0xFE00_0000 -0xFE05_7FFF 0xFE06_0000 -0xFE07_FFFF	0xFE00_0000 -0xFE05_7FFF
	CRAM0(Retention)	-	-	0xFE05_8000 -0xFE05_FFFF	0xFE05_8000 -0xFE05_FFFF
	CRAM1	0xFE08_0000 -0xFE0F_FFFF	0xFE08_0000 -0xFE0F_FFFF	0xFE08_0000 -0xFE0D_FFFF	-
	CRAM1(Retention)	-	-	0xFE0E_0000 -0xFE0F_FFFF	-
	CRAM2	0xFE10_0000 -0xFE15_FFFF	0xFE10_0000 -0xFE15_FFFF	-	-
	CRAM2(Retention)	0xFE16_0000 -0xFE17_FFFF	0xFE16_0000 -0xFE17_FFFF	-	-
	LRAM(CPU0)	0xFDC0_0000 -0xFDC0_FFFF	0xFDC0_0000 -0xFDC0_FFFF	0xFDC0_0000 -0xFDC0_FFFF	0xFDC0_0000 -0xFDC0_FFFF
PE1	LRAM(CPU1)	0xFDA0_0000 -0xFDA0_FFFF	0xFDA0_0000 -0xFDA0_FFFF	0xFDA0_0000 -0xFDA0_FFFF	0xFDA0_0000 -0xFDA0_FFFF
PE2	LRAM(CPU2)	0xFD80_0000 -0xFD80_FFFF	0xFD80_0000 -0xFD80_FFFF	0xFD80_0000 -0xFD80_FFFF	0xFD80_0000 -0xFD80_FFFF
PE3	LRAM(CPU3)	0xFD60_0000 -0xFD60_FFFF	0xFD60_0000 -0xFD60_FFFF	-	-
PE4	LRAM(CPU4)	0xFD40_0000 -0xFD40_FFFF	0xFD40_0000 -0xFD40_FFFF	-	-
PE5	LRAM(CPU5)	0xFD20_0000 -0xFD20_FFFF	0xFD20_0000 -0xFD20_FFFF	-	-

Some RAM, such as DTSRAM and MMCA RAM, are initialized by the RAM Initialization function of the hardware. For details on the RAM Initialization function, see "RH850/U2B Group User's Manual, 11.5.6 RAM Initialization."

An example of the program code is shown in Figure 3-10.

```
--stack initialize for _hdwinit
--LocalRAM initialize for _hdwinit
mov     ___ghsbegin_start_LRAM_self  ,r6
mov     ___ghsbegin_end_LRAM_self - 1,r7
(skip)
--clear Cluster RAM
mov      ___ghsbegin_start_CRAM0  ,r6
mov     ___ghsbegin_end_CRAM0 - 1,r7
jarl     _zeroclr4, lp
(skip)
--  Reset Factor Result
--  If (RESF.PRESF0 == 1 )  Retention Area clear.
--  The other Case, Retention Area not clear.
mov     RESF_Reg, r6
ld.w     0[r6], r7
mov     r7, r6
mov     0x00000001, r8
and     r8, r7
cmp     r8, r7
be      _RetentionRAMClear     -- PRESF0 = 1    Power On Reset

mov     r6, r7
mov     0x00000002, r8
and     r8, r7
cmp     r8, r7
be      _ClusterRAMInitialValueCopy     -- SRES1F0 = 1   Power Off Standby Reset
(skip)
```

Figure 3-10 Example of RAM initialization program code (PE0) (MULTI)

3.3.4.8 Other CPU initialization

After RAM initialization processing for CPU0 is complete, the wakeUp_Othercpu function is called to start up and initialize CPU1 through CPU6.

An example program code and flow are shown below.

```
_wakeUpOtherCPU:
    jarl    _wakeUp_Othercpu, lp    --WakeUp Other CPU
```

Figure 3-11 Example of other CPU initialization program code

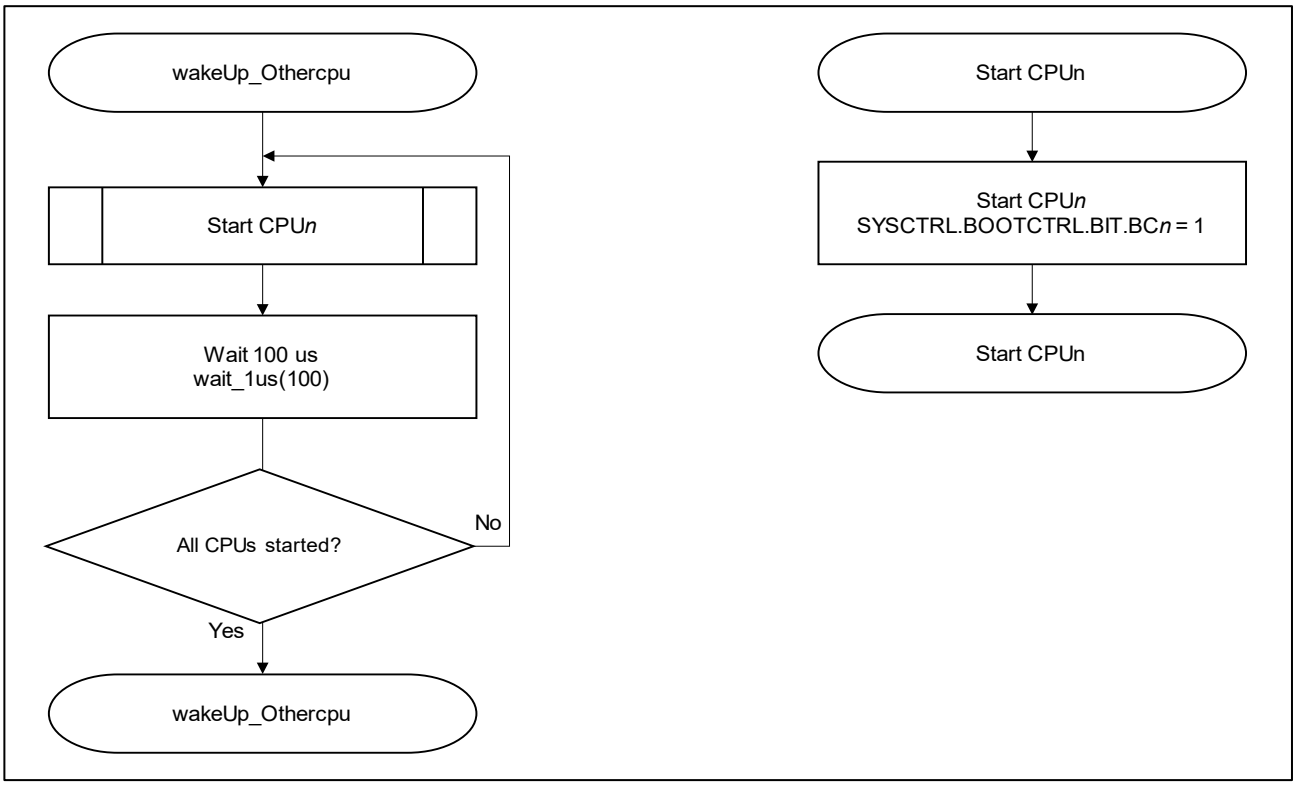


Figure 3-12 wakeUpOtherCPU function internal processing flowchart (*n* = 1 ~ 2)

3.4 Table reference interrupt method (MULTI)

This section describes the table reference interrupt method.

The RH850 has two types of interrupts: direct vector and table reference.

With the table lookup method, the word data stored in the handler address is read and a jump is made to the address pointed to by that word data.

The interrupt vectors are written in intprg_pen.c (*n*＝0~2).

3.4.1 Writing interrupt function (Pragma interrupt)

This section explains how to write the definition of a function to be executed when an interrupt occurs.

When writing an interrupt function in C language, use the #pragma ghs interrupt directive. This will cause the function written in C language to be compiled as an interrupt function. For information about parameters, refer to the GHS compiler manual.

An example of program code is shown below.

```
#pragma ghs interrupt ([parameter])
__interrupt void Function name (void)
```

Figure 3-13 Example of interrupt function program code

Note: The following procedure will result in a compilation error.

- Using arguments or return types other than Void type
- Calling an interrupt handler as if it were a normal function

3.4.2 How to handle interrupts on a CPU other than CPU0

The RH850/U2B-E has a multi-core configuration.

When running interrupt processing from a CPU other than CPU0 in MULTI, you can reference the interrupt vector defined in the vector table by using the #pragma ghs reference command.

```
#pragma ghs reference Interrupt vector name
__interrupt void function name (void)
```

Figure 3-14 Example of interrupt function definition code

Note: The interrupt vector name and the function name must be identical.

4. How to write option bytes

This section explains how to edit the option byte (hereinafter referred to as OPBT).

4.1 4.1 Data preparation in CS+

4.1.1 Section settings

The section name and the address where the set section is located must be specified in the "Configuration Setting Area" of the flash memory where the reset vector base address and option bytes are located.

The figure shows the code that uses the .dw pseudo instruction to set the reset vector and option bytes, and the section name is set in line 5. This code is written in set_opbt.asm.

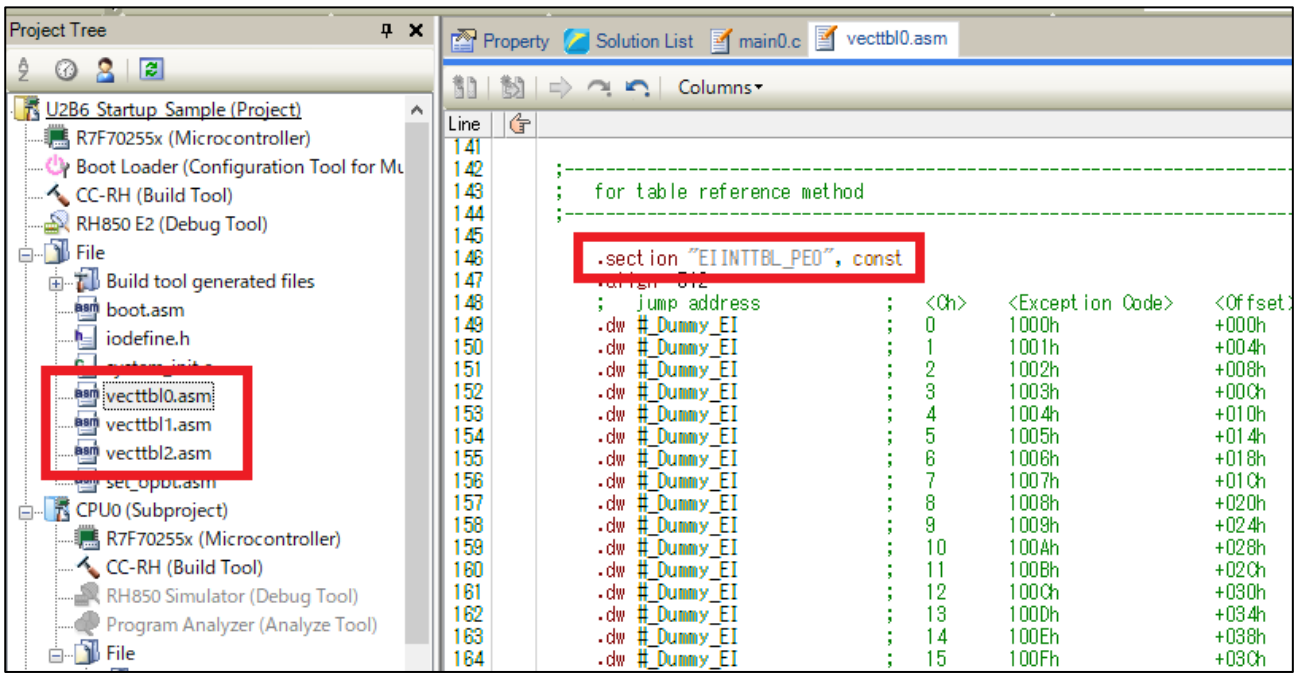


Figure 4-1 Setting of section name

Next, set the section address. Because the address to be set is "Reset Vector (PE0)" immediately after setting the section name (line 8 of set_opbt.asm), set the address to FF32 1380H of "Reset Vector (PE0)" in the "Configuration Setting Area".

Table 63.11 Base address of Configuration Setting Area in case of Area 0 is valid (FSWASTAT_0.CFGVA=0)

Base Address Name <CSAk_base> (k = f, b)	Base Address	Bus Group
<CSAf_base>	FF32 0800_H (Configuration Setting Area 0)	P-Bus Group 1
<CSAb_base>	FF32 1000_H(Configuration Setting Area 1)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 63.12 Base address of Configuration Setting Area in case of Area 1 is valid (FSWASTAT_0.CFGVA=1)

Base Address Name <CSAk_base> (k = f, b)	Base Address	Bus Group
<CSAf_base>	FF32 0800_H(Configuration Setting Area 1)	P-Bus Group 1
<CSAb_base>	FF32 1000_H (Configuration Setting Area 0)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 63.48 Configuration Setting Area (1/3)

Name	Address*9	State at the shipping*1	Write Protection ID*2	Read Protection ID*3	CSAVOF/ CSAVOFC Number
Valid Option Byte Flag (CSAVOF0) *6 *8	<CSAk_base>+ 0000_H	5AA5 A55A$_H$	Can not Write	—	—
Reserved*7	<CSAk_base>+ (0004_H to $001C_H$)	Erased	Can not Write	—	—
Valid Option Byte Flag (CSAVOF8-19) *6 *8	<CSAk_base>+ (0020_H to $004C_H$)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*7	<CSAk_base>+ (0050_H to $00FC_H$)	Erased	Can not Write	—	—
VOF Program completion flag (CSAVOFC0) *6 *8	<CSAk_base>+ 0100_H	5AA5 A55A$_H$	Can not Write	—	—
Reserved*7	<CSAk_base>+ (0104_H to $011C_H$)	Erased	Can not Write	—	—
VOF Program completion flag (CSAVOFC8-19) *6 *8	<CSAk_base>+ (0120_H to $014C_H$)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*7	<CSAk_base>+ (0150_H to $01FC_H$)	Erased	Can not Write	—	—
OTP Setting of Configuration setting Area*4	<CSAk_base>+ (0200_H to 0208_H)	All FFFF FFFF$_H$	—	—	0
Reserved (need to be programmed)*5	<CSAk_base>+ ($020C_H$ to $021C_H$)	All FFFF FFFF$_H$	—	—	0
Reserved*7	<CSAk_base>+ (0220_H to $02FC_H$)	Erased	Can not Write	—	—
Software Configuration Option Byte	<CSAk_base>+ 0300_H + 04_H x n (n = 0 to 31)	All 0000 0000$_H$	Customer ID A	—	8 to 11
Reset Vector (PE0)	<CSAk_base>+ 0380_H	0000 0000$_H$	Customer ID A	—	12
Reset Vector (PE1)	<CSAk_base>+ 0384_H	0000 0000$_H$	Customer ID A	—	12
Reset Vector (PE2)	<CSAk_base>+ 0388_H	0000 0000$_H$	Customer ID A	—	12
Reset Vector (PE3)	<CSAk_base>+ $038C_H$	0000 0000$_H$	Customer ID A	—	12
Reset Vector (PE4)	<CSAk_base>+ 0390_H	0000 0000$_H$	Customer ID A	—	12
Reset Vector (PE5)	<CSAk_base>+ 0394_H	0000 0000$_H$	Customer ID A	—	12

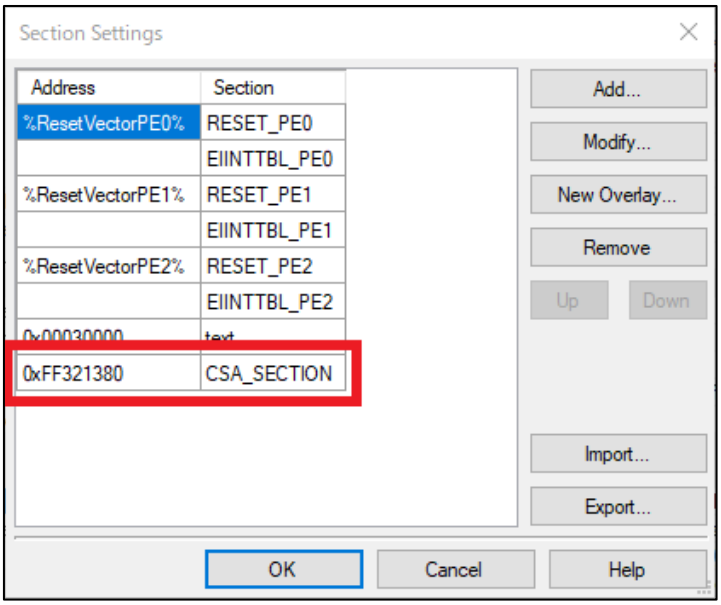


Figure 4-2 Section address setting

4.1.2 Data preparation

The .dw pseudo instruction sets the reset vector and option bytes in the Configuration Setting Area. The .dw pseudo instruction initializes memory in 4-byte units. Note that the comment "For U2B24-E" indicates where the value should be changed depending on the package used. For details about the dw pseudo instruction, see the CS+ help.

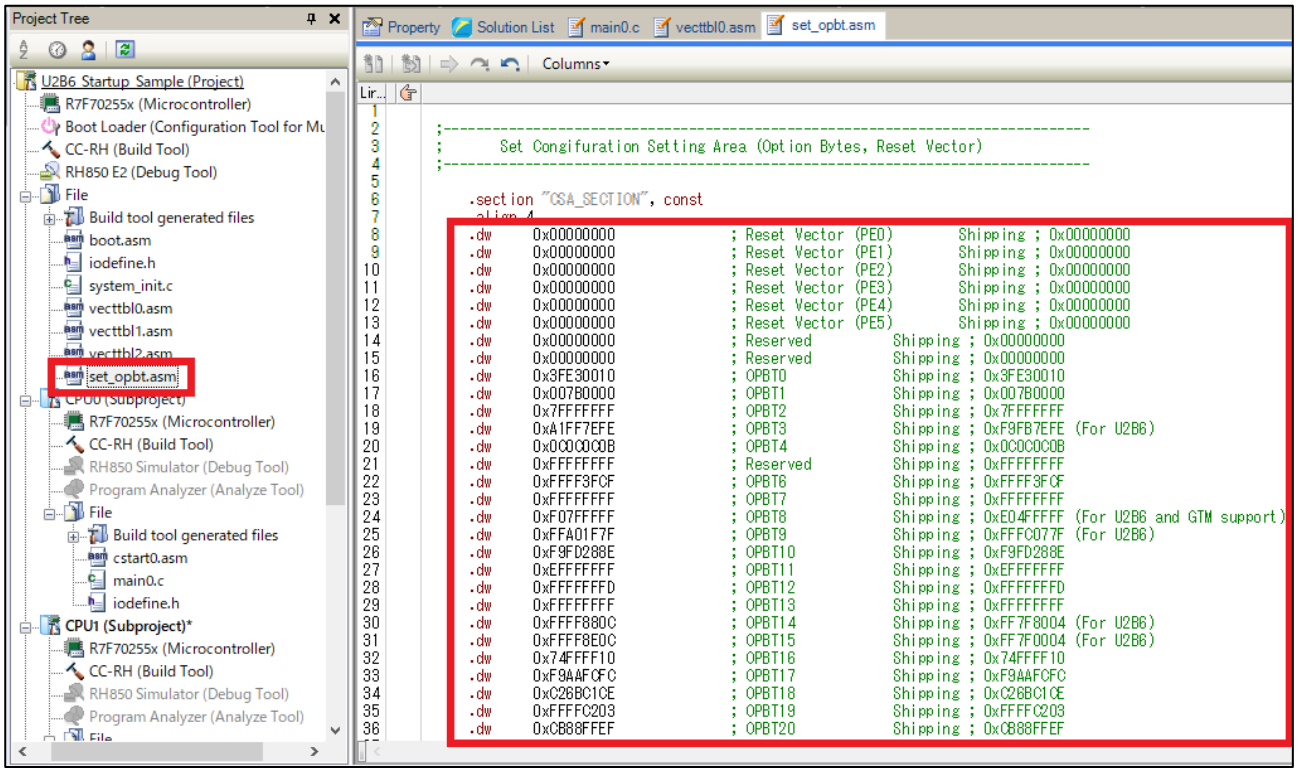


Figure 4-3 Prepare reset vector and option byte data

4.1.3 Writing by CS+

4.1.3.1 Build download

To set the option bytes, follow the procedure below.

The set option bytes will take effect after the next reset is released.

① Run the build

Run the build and make sure there are no errors.

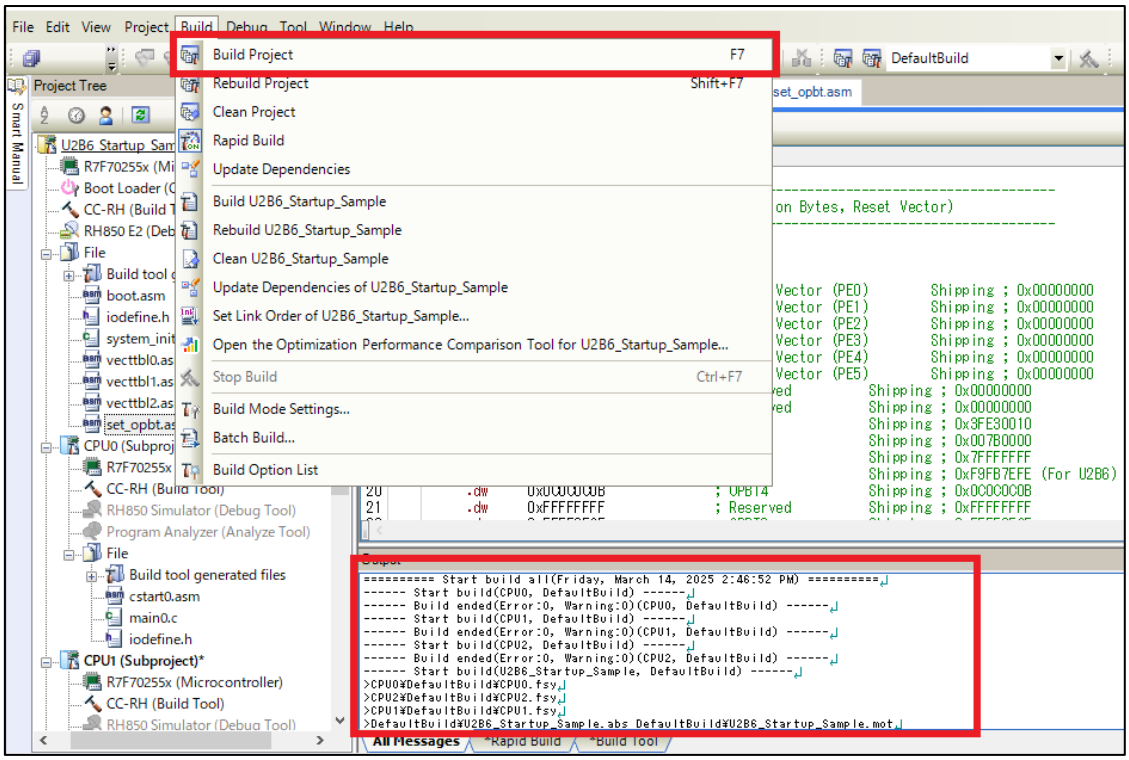


Figure 4-4 Run the build

② Enable rewriting of option bytes

Set this to "Yes" to allow rewriting of the "Configuration Setting Area".

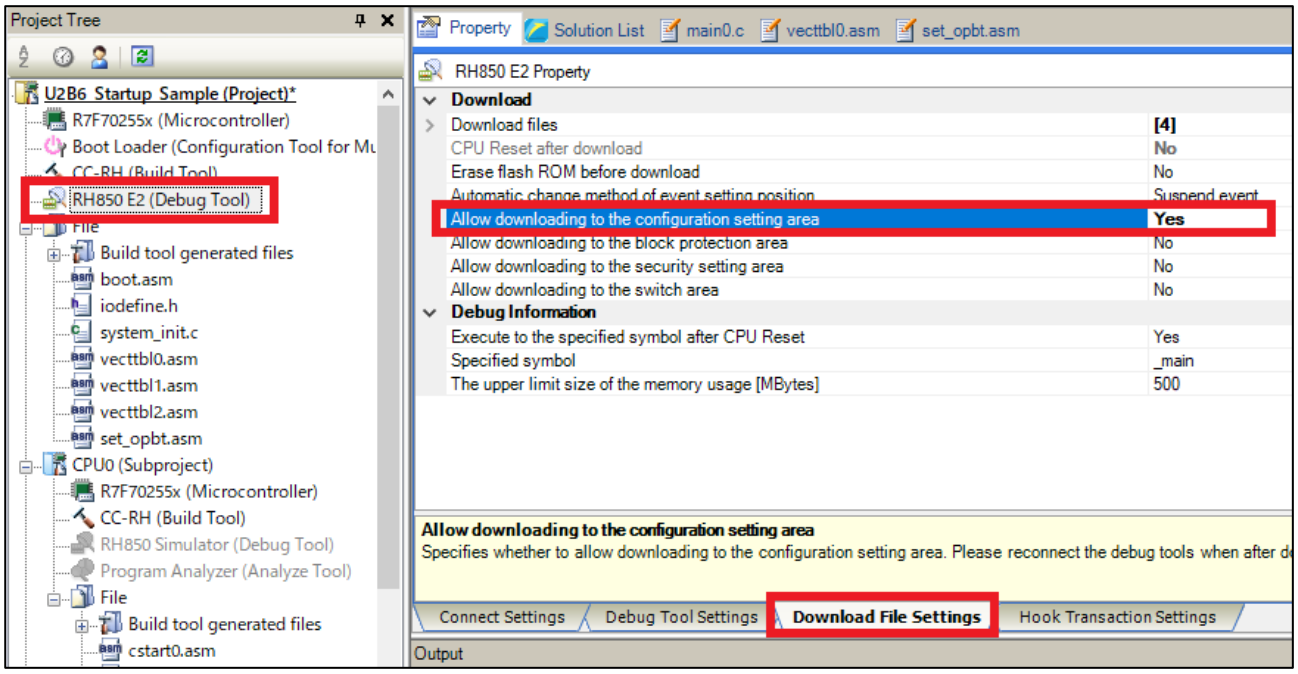


Figure 4-5 Reset vector and option byte rewrite permission settings

③ Execute download

Execute the download. When executed, the value will be written to the "Configuration Setting Area" of the flash memory.

If the download is successful, the pop-up screen shown in will be displayed and the permission setting will be changed to "No."

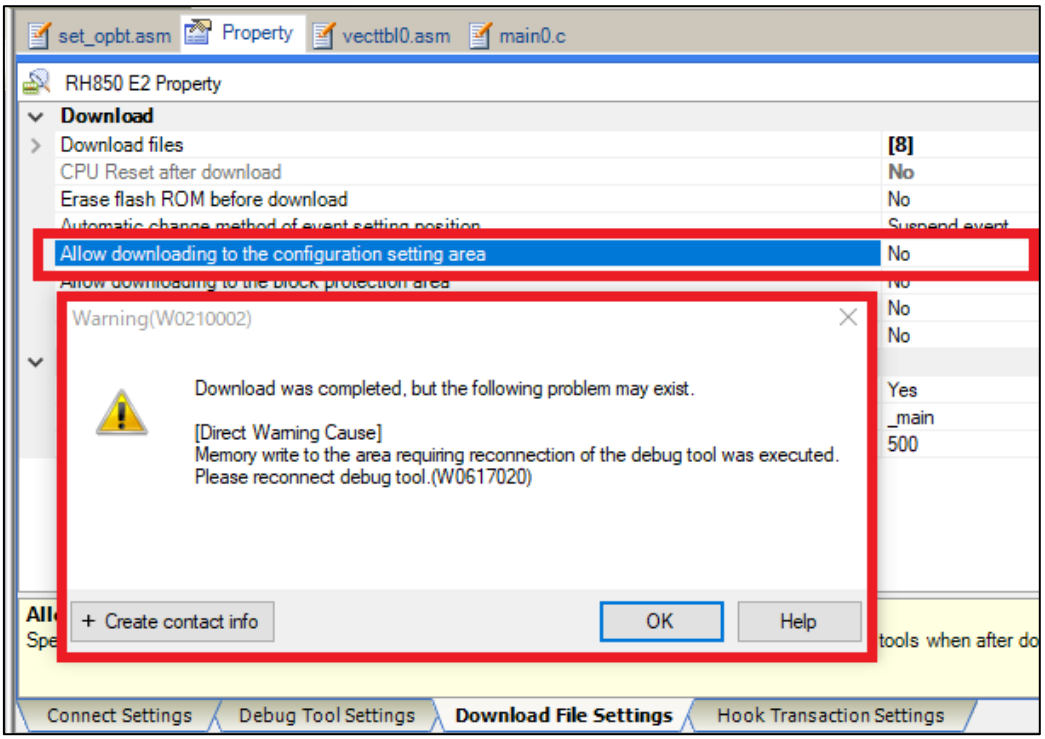


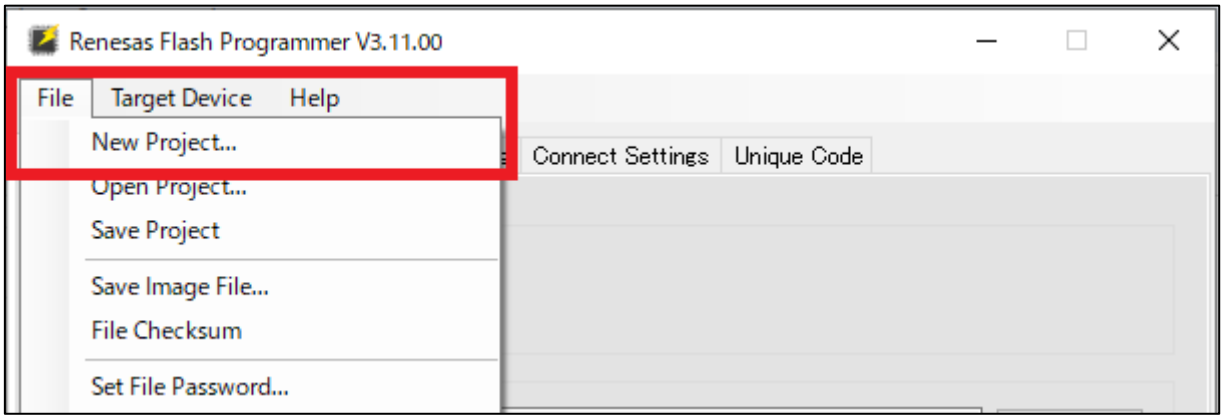
Figure 4-6 Download

This completes the rewriting of the option bytes.

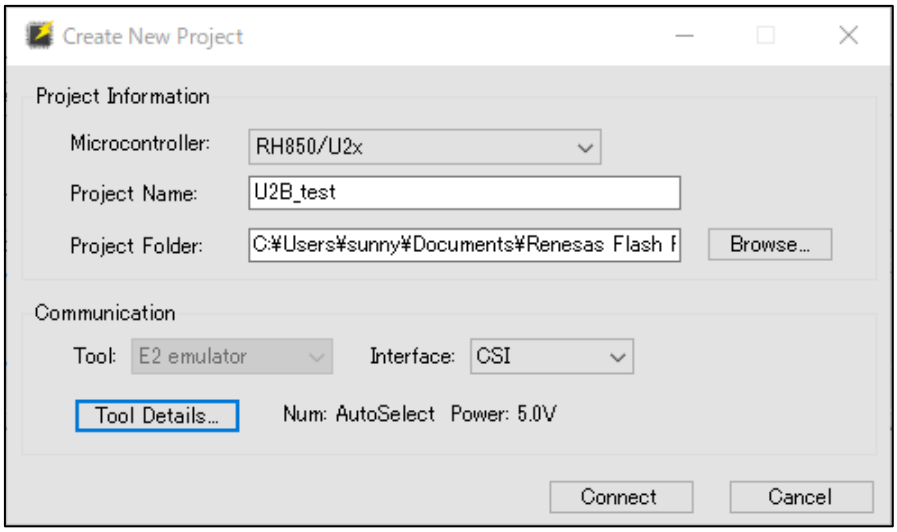
4.1.4 Writing by Renesas Flash Programmer

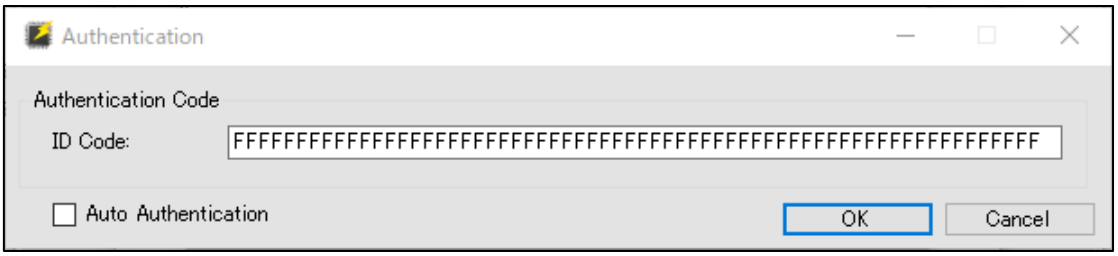
The option bytes can also be written using the Renesas Flash Programmer (hereinafter referred to as RFP).

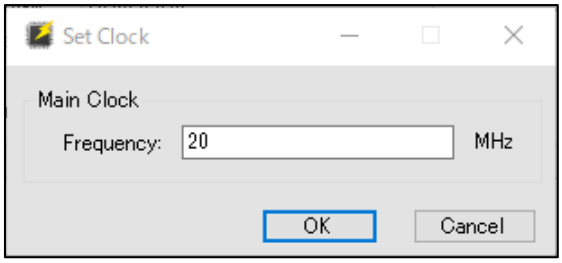
① Launch RFP and select "File" → " New Project" to create a project.



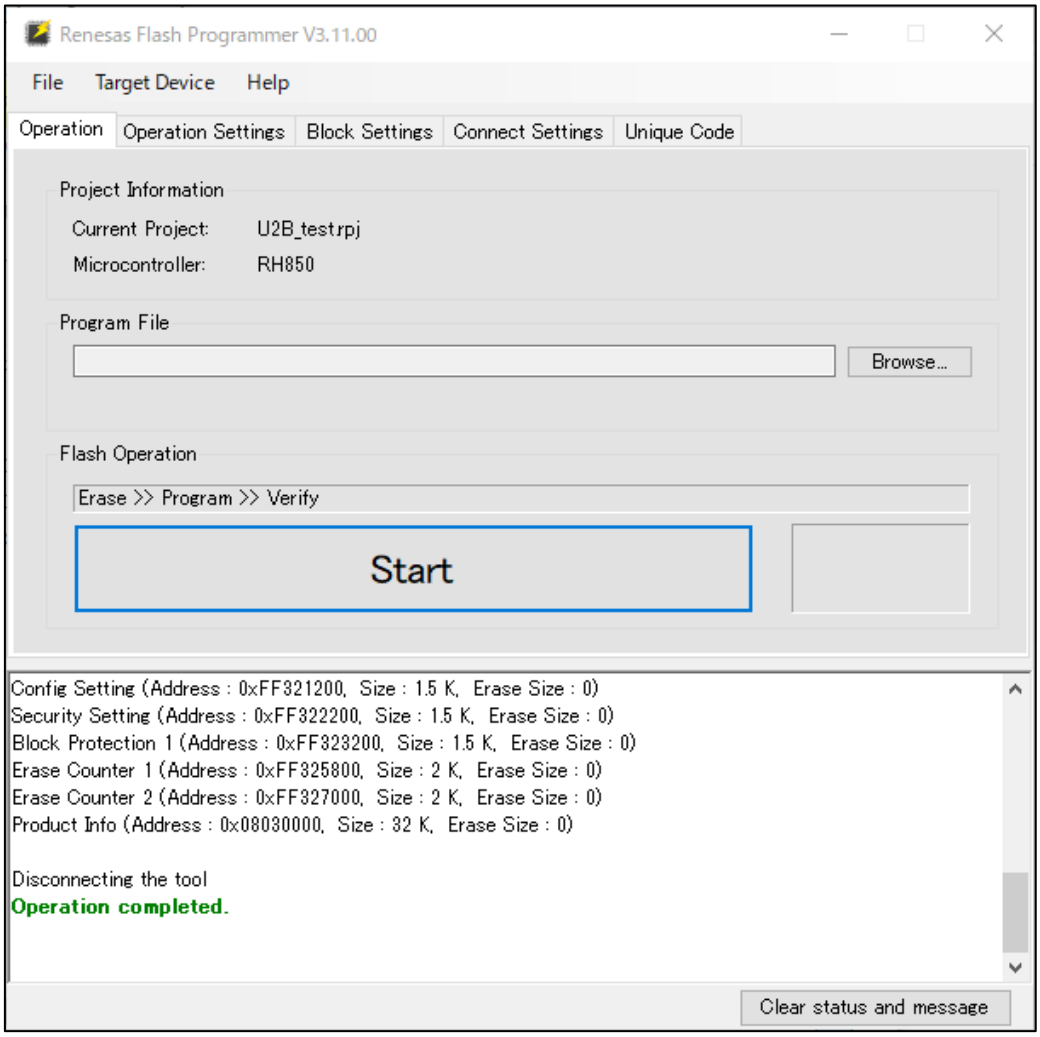
② A window for creating a new project will appear, so please configure it according to the environment you are using. This APN has been tested with a 5.0 V power supply from the emulator.



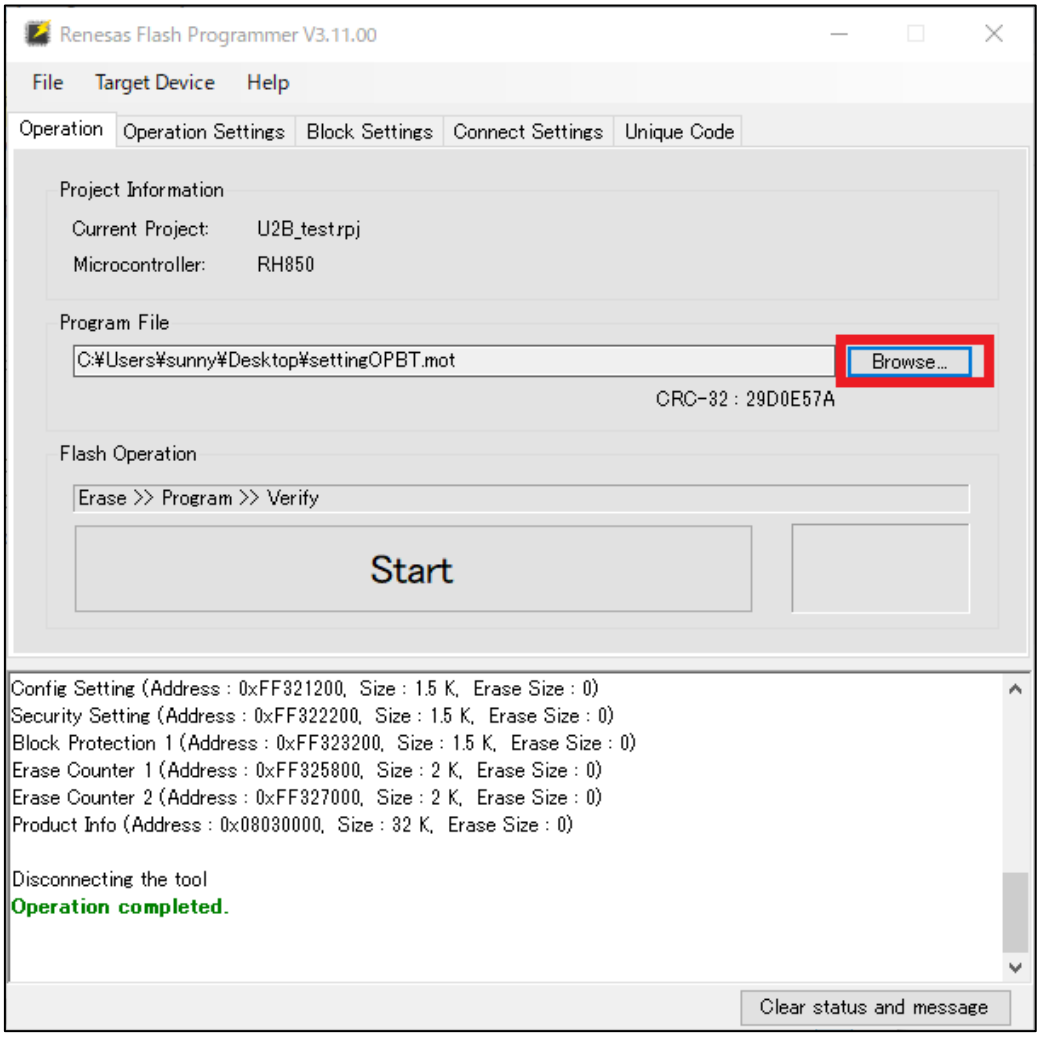




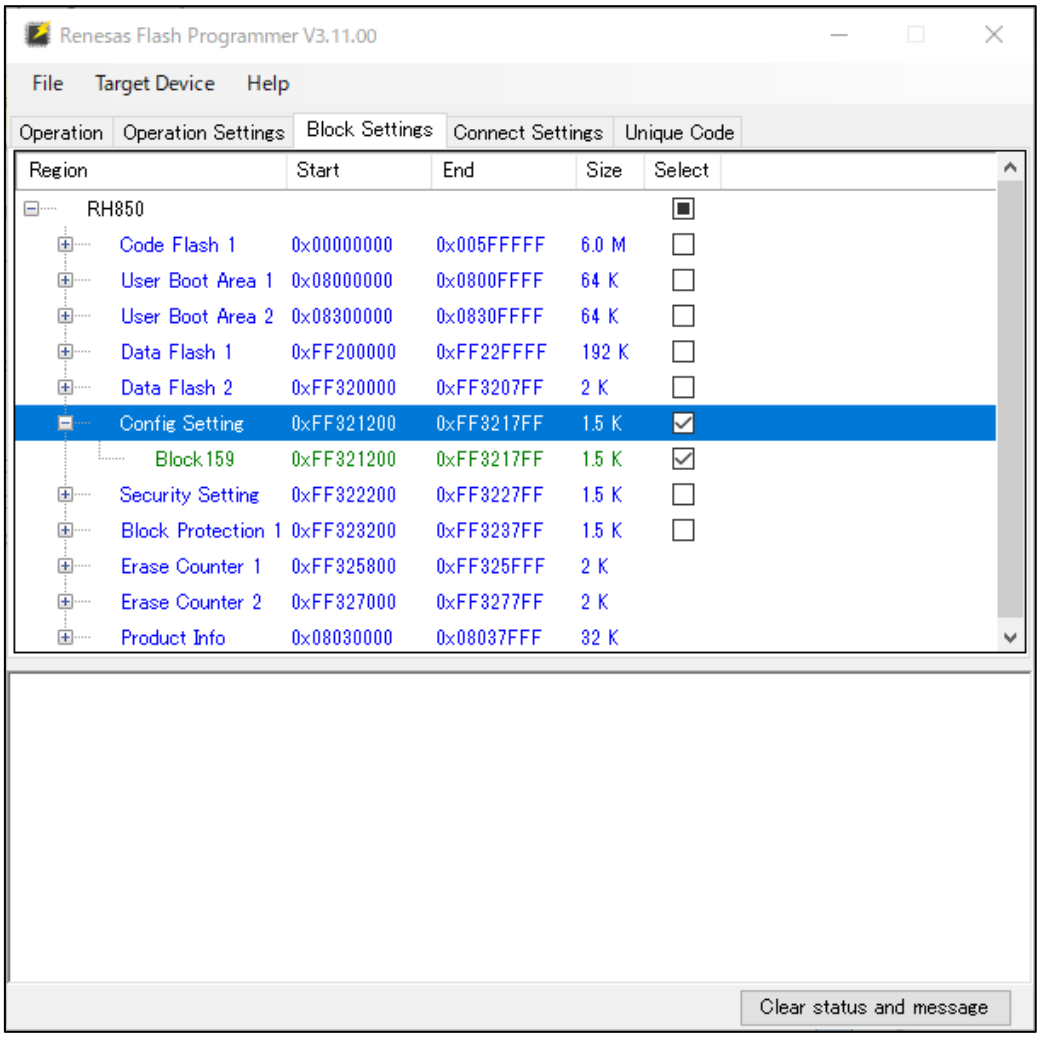
③ If the creation is successful, you will receive a notification like the one below.



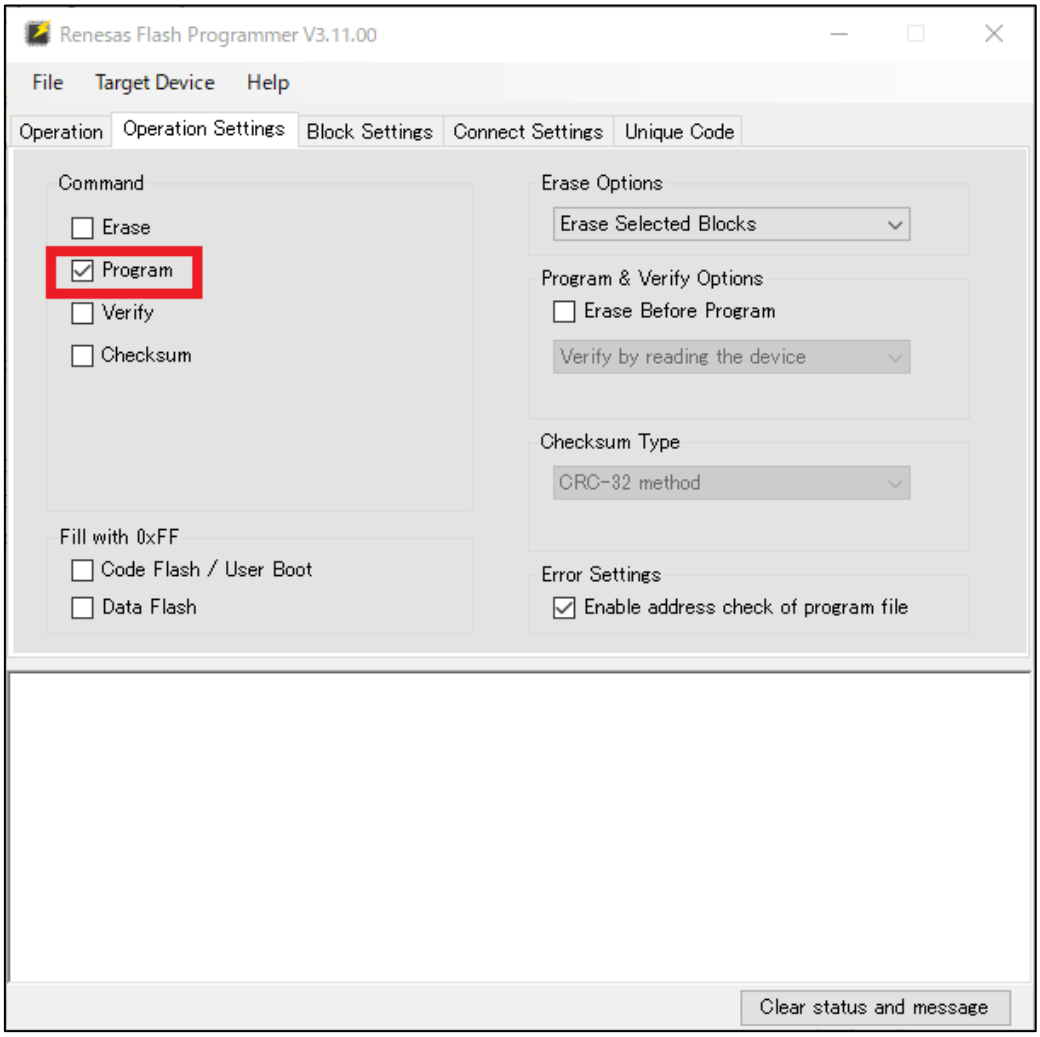
④ Select the specified file that describes the option bytes to be written by clicking "Browse".



⑤ From the "Block Settings" tab, select the block to be rewritten. The option byte for U2B (this APN is U2B6 compliant) is in the Config Setting area.

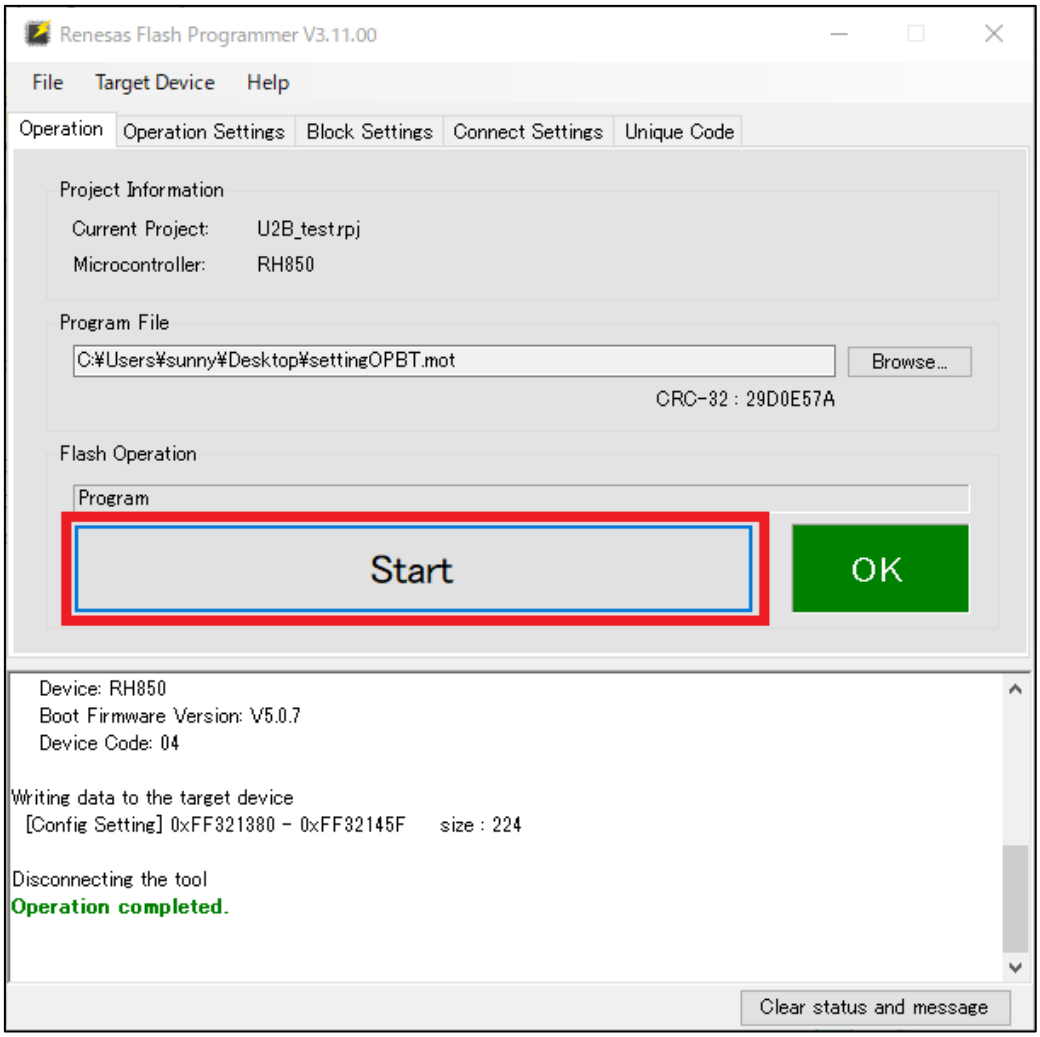


⑥ From the "Operation Settings" tab, select only the "Program" command.



⑦ When you are ready, click the "Start" button in the "Operation" tab to begin the process.

When the process is completed successfully, the memory area that was processed will be notified as a message like the one below.



4.2 Overview of option bytes

This section provides an overview of the option bytes.

Table 4-1 shows the list of option bytes. The address value is the <CSAk_base> address plus an offset.

Table 4-1 Option byte list

Name	Address	Initial value	Overview
Reset Vector (PE0)	0x0380	0x0000_0000	Reset vector base address
Reset Vector (PE1)	0x0384	0x0000_0000	
Reset Vector (PE2)	0x0388	0x0000_0000	
Reset Vector (PE3)	0x038C	0x0000_0000	
Reset Vector (PE4)	0x0390	0x0000_0000	
Reset Vector (PE5)	0x0394	0x0000_0000	
Reserved	0x0398	0x0000_0000	
Reserved	0x039C	0x0000_0000	
OPBT0	0x03A0	0x3FE3_0010	For watch dog timer
OPBT1	0x03A4	0x007B_0000	
OPBT2	0x03A8	0x7FFF_FFFF	
OPBT3	0x03AC	(1)	Enable BIST and PE
OPBT4	0x03B0	0x0C0C_0C0B	For power supplies
Reserved	0x03B4	0xFFFF_FFFF	
OPBT6	0x03B8	0xFFFF_3FCF	Initialization of IP internal RAM
OPBT7	0x03BC	0xFFFF_FFFF	Initialization of MSPI_RAM
OPBT8	0x03C0	(2)	For clocks
OPBT9	0x03C4	(1)	For Hyper-visor
OPBT10	0x03C8	0xF9FD_288E	For main clock
OPBT11	0x03CC	0xEFFF_FFFF	Settings for map mode by code-flash
OPBT12	0x03D0	0xFFFF_FFFD	
OPBT13	0x03D4	0xFFFF_FFFF	Settings for SGMI
OPBT14	0x03D8	(3)	
OPBT15	0x03DC	(3)	Settings for power circuits such as SVR
OPBT16	0x03E0	0x74FF_FF10	
OPBT17	0x03E4	0xF9AA_FCFC	
OPBT18	0x03E8	0xC26B_C1CE	
OPBT19	0x03EC	0xFFFF_C203	
OPBT20	0x03F0	0xCB88_FFEF	
OPBT21	0x03F4	0x0655_03CF	
OPBT22	0x03F8	0xFD94_FE31	
OPBT23	0x03FC	0xFFFF_FDFC	
OPBT24	0x0400	(1)	PE Redundancy Bit

Name	Address	Initial value	Overview
OPBT25	0x0404	0xFFC0_F3FF	Settings for power circuits such as SVR
OPBT26	0x0408	0xFFFF_0010	For watch dog timer
OPBT27	0x040C	0xFFFF_0000	
OPBT28	0x0410	0xFFFF_FFFF	Settings for power circuits such as SVR
Reserved	0x0414-0x041C	0xFFFF_FFFF	
Reserved	0x0420	0x6000_0055	
Reserved	0x0424	0x6000_0056	
Reserved	0x0428	0xF0F5_EC01	
Reserved	0x042C	0x0A18_0001	
OPBT36	0x0430	(1)	SSCG setting value, PE redundant bit
OPBT37	0x0434	0x02EF_FF55	Clock setting value
OPBT38	0x0438	0x0000_0000	For DFP
OPBT39	0x043C	0x0000_0000	
OPBT40	0x0440	0xFF00_FCFF	CAN and other communication settings

Table 4-2 List of option bytes for each product (1)

プロダクト	G4MH コア設定	OPBT3	OPBT9	OPBT24	OPBT36
U2B24	Performance	0xC1FF_7EFE	0xFFF0_3F7F	0xFFFF_FFF0	0xC184_C8FF
	Safety	0xE1FF_7EFE	0xFFF0_1F7F	0xFFFF_FFE0	0xE184_C8FF
U2B20	Performance	0xC1FF_7EFE	0xFFFC_3F7F	0xFFFF_FFFC	0xC184_C8FF
	Safety	0xE9FF_7EFE	0xFFE0_1F7F	0xFFFF_FFE0	0xE984_C8FF
U2B10	Performance	0xF1FB_7EFE	0xFFFC_0F7F	0xFFFF_FFFC	0xF184_C8FF
	Safety	0xF9FB_7EFE	0xFFFC_077F	0xFFFF_FFF0	0xF984_C8FF
U2B6	-	0xF9FB_7EFE	0xFFFC_077F	0xFFFF_FFFC	0xF984_C8FF

Table 4-3 List of option bytes for each product (2)

プロダクト	GTM-IP 設定	OPBT8
U2B24	Support	0xF47F_FFFF
	No support	0xF57F_FFFF
U2B20	Support	0xE44F_FFFF
	No support	0xE54F_FFFF
U2B10	Support	0xE04F_FFFF
	No support	0xE14F_FFFF
U2B6	Support	0xE04F_FFFF
	No support	0xE14F_FFFF

Table 4-4 List of option bytes for each product (3)

プロダクト	OPBT14	OPBT15
U2B24	0xFFFF_880C	0xFFFF_880C
U2B20	0xFFFF_880C	0xFF7F_0004
U2B10	0xFF7F_8004	0xFF7F_0004
U2B6	0xFF7F_8004	0xFF7F_0004

4.2.1 OPBT0

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: *For setting these data, see the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03AD_H$

Value after reset: Specified by the user

Value at the shipping: $3FE3\ 0010_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPWD RUNA	OPWD WMSA	OPWDOVFA [2:0]			OPWD INTA	—	OPWDWSA[1:0]		—	—	WDTBR ES_EN	—	OPWD VACA	—	OPWD ENA
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	OPWDWOSTA[15:0]															
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.58 OPBT0 Contents (1/2)

Bit Position	Bit Name	Function
31	OPWDRUNA	This bit sets the start mode of WDTBA 0: WDTBA software trigger start mode 1: WDTBA default start mode
30	OPWDWMSA	This bit selects Window Open function mode of WDTBA 0: Window Size of Window Open function is set by WDTBAWS[1:0]. WDTBATIT outputs When the counter reaches 75% of the overflow setting defined by WDTBAMD.WDTBAOVF[2:0] 1: Window Size of Window Open function is set by WDTBAWOST WDTBATIT outputs When WDTBAWIS matches WDT counter.
29 to 27	OPWDOVFA[2:0]	These bits select the overflow interval time of WDTBA.

OPWDOVFA[2:0]	Overflow Interval Time
000_B	2^9 / WDTBTCKI
001_B	2^{10} / WDTBTCKI
010_B	2^{11} / WDTBTCKI
011_B	2^{12} / WDTBTCKI
100_B	2^{13} / WDTBTCKI
101_B	2^{14} / WDTBTCKI
110_B	2^{15} / WDTBTCKI
111_B	2^{16} / WDTBTCKI

26	OPWDINTA	This bit enables or disables a 75% interrupt request of WDTBA (WDTBATIT). 0: WDTBATIT disabled 1: WDTBATIT enabled
25	Reserved	Set the value of valid area at the shipping.

Table 63.58 OPBT0 Contents (2/2)

Bit Position	Bit Name	Function
24, 23	OPWDWSA[1:0]	These bits select the window open period of WDTBA. OPWDWSA[1:0] / Window Open Period 00_B 25% 01_B 50% 10_B 75% 11_B 100%
22, 21	Reserved	Set the value of valid area at the shipping.
20	WDTBRES_EN	WDTB Reset Enable 1: Enable 0: Disable
19	Reserved	Set the value of valid area at the shipping.
18	OPWDVACA	This bit specifies the trigger register for the generation of counter re-start triggers to keep the counter from overflowing. 0: WDTBAWDTE (fixed) 1: WDTBAEVAC (variable)
17	Reserved	Set the value of valid area at the shipping.
16	OPWDENA	This bit enables/disables the WDTBA 0: WDTBA is disabled 1: WDTBA is enabled
15 to 0	OPWDWOSTA[15:0]	These bits specify the initial value of WDTBA Window Open Start Register for setting the start timing of the window open.

For details of these data, see **Section 33, Window Watchdog Timer (WDTB).**

4.2.2 OPBT1

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: *For setting these data, see the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03A4_H$

Value after reset: Specified by the user

Value at the shipping: $007B\ 0000_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	OPWDRUN5	OPWDRUN4	OPWDRUN3	OPWDRUN2	OPWDRUN1	OPWDRUN0	OPWDWMS0	OPWDOVF[2:0]			OPWDINT0	OPWDVAC	OPWDWS0[1:0]	
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	OPWDWISA[15:0]															
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.59 OPBT1 Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	Set the value of valid area at the shipping.
29	OPWDRUN5	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] This bit sets the start mode of WDTB5. 0: WDTB5 software trigger start mode 1: WDTB5 default start mode [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
28	OPWDRUN4	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] This bit sets the start mode of WDTB4. 0: WDTB4 software trigger start mode 1: WDTB4 default start mode [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
27	OPWDRUN3	[For U2B24-FCC/U2B20-FCC/U2B10-FCC/U2B24/U2B20/U2B10] This bit sets the start mode of WDTB3. 0: WDTB3 software trigger start mode 1: WDTB3 default start mode [For U2B6-FCC/U2B6] Set the value of valid area at the shipping.
26	OPWDRUN2	This bit sets the start mode of WDTB2. 0: WDTB2 software trigger start mode 1: WDTB2 default start mode
25	OPWDRUN1	This bit sets the start mode of WDTB1. 0: WDTB1 software trigger start mode 1: WDTB1 default start mode
24	OPWDRUN0	This bit sets the start mode of WDTB0. 0: WDTB0 software trigger start mode 1: WDTB0 default start mode

Table 63.59 OPBT1 Contents (2/2)

Bit Position	Bit Name	Function
23	OPWDWMS0	This bit selects Window Open function mode of WDTB0 0: Window Size of Window Open function is set by WDTB0WS[1:0]. WDTB0TIT outputs When the counter reaches 75% of the overflow setting defined by WDTB0MD.WDTB0OVF[2:0] 1: Window Size of Window Open function is set by WDTB0WOST WDTB0TIT outputs When WDTB0WIS matches WDT counter.
22 to 20	OPWDOVF[2:0]	These bits select the overflow interval time of WDTBn. [For U2B24-FCC/U2B20-FCC/U2B24/U2B20] n = 0 to 5 [For U2B10-FCC/U2B10] n = 0 to 3 [For U2B6-FCC/U2B6] n = 0 to 2
19	OPWDINT0	This bit enables or disables interrupt request of WDTB0 (WDTB0TIT). 0: WDTB0TIT disabled 1: WDTB0TIT enabled
18	OPWDVAC	This bit specifies the trigger register of WDTBn for the generation of counter re-start triggers to keep the counter from overflowing. [For U2B24-FCC/U2B20-FCC/U2B24/U2B20] n = 0 to 5 [For U2B10-FCC/U2B10] n = 0 to 3 [For U2B6-FCC/U2B6] n = 0 to 2 0: WDTBnWDTE (fixed) 1: WDTBnEVAC (variable)
17 to 16	OPWDWS0[1:0]	These bits select the window open period of WDTB0.
15 to 0	OPWDWISA[15:0]	These bits specify the initial value of WDTBA Interrupt Output Timing Setting Register.

OPWDOVF[2:0]	Overflow Interval Time
000_B	2^9 / WDTBTCKI
001_B	2^{10} / WDTBTCKI
010_B	2^{11} / WDTBTCKI
011_B	2^{12} / WDTBTCKI
100_B	2^{13} / WDTBTCKI
101_B	2^{14} / WDTBTCKI
110_B	2^{15} / WDTBTCKI
111_B	2^{16} / WDTBTCKI

OPWDWS0[1:0]	Window Open Period
00_B	25%
01_B	50%
10_B	75%
11_B	100%

For details of these data, see **Section 33, Window Watchdog Timer (WDTB).**

4.2.3 OPBT2

※UM に詳細な記述が無い為割愛。

4.2.4 OPBT3

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data

Access: For setting these data, see the RH850/U2B Group Flash Memory User's Manual: Hardware.

Address: <CSAk_base> + $03AC_H$

Value after reset: Specified by the user

Value at the shipping: See **Table 63.49, Relation between each device type and the shipping value of Option Bytes (1)** (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	PE5_DISABLE	PE4_DISABLE	PE3_DISABLE	PE2_DISABLE	PE1_DISABLE	—	—	—	—	—	—	PE2_FPSIMD_EN	—	PE0_FPSIMD_EN
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	HWBIST	—	—	—	TESTSET[1:0]		LBISTSEL[1:0]		—	—	—	—	—	—	STMSEL1	STMSEL0
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.60 OPBT3 Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	Set the value of valid area at the shipping.
29	PE5_DISABLE	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] PE5_DISABLE bit 0: PE5 enable 1: PE5 disable [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
28	PE4_DISABLE	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] PE4_DISABLE bit 0: PE4 enable 1: PE4 disable [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
27	PE3_DISABLE	[For U2B24-FCC/U2B20-FCC/U2B10-FCC/U2B24/U2B20/U2B10] PE3_DISABLE bit 0: PE3 enable 1: PE3 disable [For U2B6-FCC/U2B6] Set the value of valid area at the shipping.
26	PE2_DISABLE	PE2_DISABLE bit 0: PE2 enable 1: PE2 disable
25	PE1_DISABLE	PE1_DISABLE bit 0: PE1 enable 1: PE1 disable
24 to 19	Reserved	Set the value of valid area at the shipping.
18	PE2_FPSIMD_EN	[For U2B24-FCC/U2B20-FCC/U2B10-FCC/U2B24/U2B20/U2B10] PE2_FP-SIMD Enable bit 0: PE2_FPSIMD disable 1: PE2_FPSIMD enable [For U2B6-FCC/U2B6] Set the value of valid area at the shipping.
17	Reserved	Set the value of valid area at the shipping.

Table 63.60 OPBT3 Contents (2/2)

Bit Position	Bit Name	Function
16	PE0_FPSIMD_EN	PE0_FP-SIMD Enable bit 0: PE0_FPSIMD disable 1: PE0_FPSIMD enable
15	HWBIST	Power On BIST enable 0: BIST is skipped 1: BIST is executed
14 to 12	Reserved	Set the value of valid area at the shipping.
11, 10	TESTSET[1:0]	BIST selection 00_B: Prohibited 01_B: LBIST Only 10_B: MBIST Only 11_B: LBIST and MBIST For details, see **Section 55, Functional Safety**.
9, 8	LBISTSEL[1:0]	BIST scenario selection 00_B: LBIST scenario 1/MBIST scenario 1 01_B: LBIST scenario 2/MBIST scenario 1 10_B: LBIST scenario 3/MBIST scenario 1 11_B: Prohibited For details, see **Section 55, Functional Safety**.
7 to 2	Reserved	Set the value of valid area at the shipping.
1, 0	STMSEL1, STMSEL0	These bits select operating mode and startup area. When FLMD0 pin is 0, the operating mode and startup area are selected depending on the combination of the STMSEL1 and STMSEL0. For details, see **Section 5, Operating Mode**.

4.2.5 OPBT4

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see the RH850/U2B Group Flash Memory User's Manual: Hardware.

Address: <CSAIk_base> + $03B0_H$

Value after reset: Specified by the user

Value at the shipping: $0C0C\ 0C0B_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ISOVDDFLTW [1:0]		ISOVDDCLKSEL[1:0]		ISOVDD FLTEN	—	ISOVDD HDE	ISOVDD LDE	VCCFTW [1:0]		VCCCLKSEL [1:0]		VCCFL TEN	—	VCCHD E	VCCLD E
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	E0VCCFLTW [1:0]		E0VCCCLKSEL [1:0]		E0VCC FLTEN	—	E0VCC HDE	E0VCC LDE	VDD2FLTW[1:0]		VDD2CLKS EL[1:0]		VD D2FLTE N	DSDET EN	VDD2H DE	VDD2L DE
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.61 OPBT4 Contents (1/2)

Bit Position	Bit Name	Function
31, 30	ISOVDDFLTW[1:0]	VMON configurations for ISOVDD Select the minimum filtering width of digital noise filter.*1
29, 28	ISOVDDCLKSEL[1:0]	VMON configurations for ISOVDD Select a clock of the digital noise filter.*1
27	ISOVDDFLTEN	VMON configurations for ISOVDD Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF.
26	Reserved	Set the value of valid area at the shipping.
25	ISOVDDHDE	VMON configurations for ISOVDD ISOVDD High voltage detection enable.
24	ISOVDDLDE	VMON configurations for ISOVDD ISOVDD Low voltage detection enable.
23, 22	VCCFLTW[1:0]	VMON configurations for VCC Select the minimum filtering width of digital noise filter.
21, 20	VCCCLKSEL[1:0]	VMON configurations for VCC Select a clock of the digital noise filter.
19	VCCFLTEN	VMON configurations for VCC Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF.
18	Reserved	Set the value of valid area at the shipping.
17	VCCHDE	VMON configurations for VCC VCC High voltage detection enable.
16	VCCLDE	VMON configurations for VCC VCC Low voltage detection enable.
15, 14	E0VCCFLTW[1:0]	VMON configurations for E0VCC Select the minimum filtering width of digital noise filter.
13, 12	E0VCCCLKSEL[1:0]	VMON configurations for E0VCC Select a clock of the digital noise filter.
11	E0VCCFLTEN	VMON configurations for E0VCC Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF.

Table 63.61 OPBT4 Contents (2/2)

Bit Position	Bit Name	Function
10	Reserved	Set the value of valid area at the shipping.
9	E0VCCHDE	VMON configurations for E0VCC E0VCC High voltage detection enable.
8	E0VCCLDE	VMON configurations for E0VCC E0VCC Low voltage detection enable.
7, 6	VDD2FLTW[1:0]	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] VMON configurations for AWOVDD/ISOVDD2 Select the minimum filtering width of digital noise filter. [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
5, 4	VDD2CLKSEL [1:0]	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] VMON configurations for AWOVDD/ISOVDD2 Select a clock of the digital noise filter. [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
3	VDD2FLTEN	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] VMON configurations for AWOVDD/ISOVDD2 Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF. [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
2	DSDETEN	VMON control during DeepSTOP 0: VMON disable (STOP) 1: VMON continue
1	VDD2HDE	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] VMON configurations for AWOVDD/ISOVDD2 AWOVDD/ISOVDD2 High voltage detection enable. [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
0	VDD2LDE	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] VMON configurations for AWOVDD/ISOVDD2 AWOVDD/ISOVDD2 Low voltage detection enable. [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.

Note 1. When using the SVR and VMON functions with U2B-FCC, the VMON function may pick up the error excessive with the power supply voltage fluctuations in during FBIST, depending on the VMON configuration settings. Please set the appropriate values to the VMON configuration.

For details of these data, see **Section 13, Power Supply Voltage Monitor.**

4.2.6 OPBT6

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see the RH850/U2B Group Flash Memory User's Manual: Hardware.

Address: <CSAk_base> + $03B8_H$

Value after reset: Specified by the user

Value at the shipping: FFFF $3FCF_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	STAC_DTSRAM [1:0]		STAC_DPRAM [1:0]		STAC_DFE[1:0]		—	—	—	—	STAC_GTM [1:0]	
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.62 OPBT6 Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	Set the value of valid area at the shipping.
11, 10	STAC_DTSRAM [1:0]	RAM Initialization Mode for DTSRAM. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
9, 8	STAC_DPRAM[1:0]	RAM Initialization Mode for DPRAM. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
7, 6	STAC_DFE[1:0]	Ram Initialization Mode for DFE $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
5 to 2	Reserved	Set the value of valid area at the shipping.
1, 0	STAC_GTM[1:0]	RAM Initialization Mode for GTM. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled

For details of these data, see **Section 11, Reset Controller.**

4.2.7 OPBT7

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see the RH850/U2B Group Flash Memory User's Manual: Hardware.

Address: <CSAk_base> + $03BC_H$

Value after reset: Specified by the user

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	STAC_ETN[1:0]		STAC_EMU3S1[1:0]		STAC_EMU3S0[1:0]		STAC_DFP[1:0]		STAC_MMCA[1:0]		STAC_MSPI9[1:0]		STAC_MSPI8[1:0]	
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	STAC_MSPI7[1:0]		STAC_MSPI6[1:0]		STAC_MSPB5[1:0]		STAC_MSPI4[1:0]		STAC_MSPI3[1:0]		STAC_MSPI2[1:0]		STAC_MSPI1[1:0]		STAC_MSPI0[1:0]	
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.63 OPBT7 Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	Set the value of valid area at the shipping.
29, 28	STAC_ETN[1:0]	RAM Initialization Mode for ETNE0(RSWITCH) $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
27, 26	STAC_EMU3S1[1:0]	Ram zeroing Initialization Mode for EMU3S1 $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
25, 24	STAC_EMU3S0[1:0]	Ram zeroing Initialization Mode for EMU3S0 $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
23, 22	STAC_DFP[1:0]	Ram zeroing Initialization Mode for DFP $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
21, 20	STAC_MMCA[1:0]	RAM Initialization Mode for MMCA. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
19, 18	STAC_MSPI9[1:0]	RAM Initialization Mode for MSPI9. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
17, 16	STAC_MSPI8[1:0]	RAM Initialization Mode for MSPI8. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled

Table 63.63 OPBT7 Contents (2/2)

Bit Position	Bit Name	Function
15, 14	STAC_MSPI7[1:0]	RAM Initialization Mode for MSPI7. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
13, 12	STAC_MSPI6[1:0]	RAM Initialization Mode for MSPI6. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
11, 10	STAC_MSPI5[1:0]	RAM Initialization Mode for MSPI5. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
9, 8	STAC_MSPI4[1:0]	RAM Initialization Mode for MSPI4. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
7, 6	STAC_MSPI3[1:0]	RAM Initialization Mode for MSPI3. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
5, 4	STAC_MSPI2[1:0]	RAM Initialization Mode for MSPI2. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
3, 2	STAC_MSPI1[1:0]	RAM Initialization Mode for MSPI1. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled
1, 0	STAC_MSPI0[1:0]	RAM Initialization Mode for MSPI0. $X0_B$: Disabled 01_B: Prohibited 11_B: Enabled

For details of these data, see **Section 11, Reset Controller.**

4.2.8 OPBT8

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03C0_H$

Value after reset: Specified by the user

Value at the shipping: See **Table 63.50, Relation between each device type and the shipping value of Option Bytes (2)** (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	ETNC1_IF_SEL[1:0]		ETNC0_IF_SEL[1:0]		ATU_GTM_SEL	CKSEL_GTM	CLMA1SEL	ETNC1_UNIT_SEL	ETNC0_UNIT_SEL	CKSEL_RHSB3[1:0]		CKSEL_RHSB2[1:0]	
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	CKSEL_RHSB1[1:0]		CKSEL_RHSB0[1:0]		CKSEL_HRPWM[1:0]		—	—	—	—	—	—	CKSEL_SSCG1	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.64 OPBT8 Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	Reserved	Set the value of valid area at the shipping.
28, 27	ETNC1_IF_SEL[1:0]	ETNC1 MII/RMII/SGMII/RevMII select 00: MII 01: RMII 10: SGMII 11: RevMII
26, 25	ETNC0_IF_SEL[1:0]	ETNC0 MII/RMII/SGMII/RevMII select 00: MII 01: RMII 10: SGMII 11: RevMII
24	ATU_GTM_SEL	ATU, GTM select 0: ATU: disable, GTM: enable 1: ATU: enable, GTM: disable
23	CKSEL_GTM	Clock source select for GTM 1: CLKC_UHSB 0: CLKC_SBUS
22	CLMA1SEL	Select clock monitor target for CLMA1 0: CLK_HSIOSC / 20 1: CLK_WDT
21	ETNC1_UNIT_SEL	ETNC1 TSNES/RSWITCH select 0: TSNES 1: RSWITCH
20	ETNC0_UNIT_SEL	ETNC0 TSNES/RSWITCH select 0: TSNES 1: RSWITCH
19, 18	CKSEL_RHSB3[1:0]	CLK_RHSB_C3 frequency setting bit 11: CLKC_HSB (80MHz) 10: CLKC_UHSB (160MHz) 01: Setting prohibited 00: CLKC_CPU (400MHz)

Table 63.64 OPBT8 Contents (2/2)

Bit Position	Bit Name	Function
17, 16	CKSEL_RHSB2[1:0]	CLK_RHSB_C2 frequency setting bit 11: CLKC_HSB (80MHz) 10: CLKC_UHSB (160MHz) 01: Setting prohibited 00: CLKC_CPU (400MHz)
15, 14	CKSEL_RHSB1[1:0]	CLK_RHSB_C1 frequency setting bit 11: CLKC_HSB (80MHz) 10: CLKC_UHSB (160MHz) 01: Setting prohibited 00: CLKC_CPU (400MHz)
13, 12	CKSEL_RHSB0[1:0]	CLK_RHSB_C0 frequency setting bit 11: CLKC_HSB (80MHz) 10: CLKC_UHSB (160MHz) 01: Setting prohibited 00: CLKC_CPU (400MHz)
11 to 10	CKSEL_HRPWM[1:0]	Clock source select for HR-PWM 11: CLKC_HSB Set when HRPWM is used with TSG3 10: CLKC_UHSB Set when HRPWM is used with 160MHz GTM 0x: CLKC_SBUS Set when HRPWM is used with 200MHz GTM
9 to 4	Reserved	Set the value of valid area at the shipping.
3	CKSEL_SSCG1	OPBT Clock source select for DFP 1: SSCG1 0: SSCG
2 to 0	Reserved	Set the value of valid area at the shipping.

4.2.9 OPBT9

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAIk_base> + $03C4_H$

Value after reset: Specified by the user

Value at the shipping: See **Table 63.49, Relation between each device type and the shipping value of Option Bytes (1)** (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	PE4_DCLS_DIS	PE3_DCLS_DIS	PE2_DCLS_DIS	PE1_DCLS_DIS	PE0_DCLS_DIS
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	PE5_HVE	PE4_HVE	PE3_HVE	PE2_HVE	PE1_HVE	PE0_HVE	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.65 OPBT9 Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	Reserved	Set the value of valid area at the shipping.
20	PE4_DCLS_DIS	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] PE4 DCLS Disable bit 0: DCLS Enabled 1: DCLS Disabled [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
19	PE3_DCLS_DIS	[For U2B24-FCC/U2B24] PE3 DCLS Disable bit 0: DCLS Enabled 1: DCLS Disabled [For U2B20-FCC/U2B10-FCC/U2B6-FCC/U2B20/U2B10/U2B6] Set the value of valid area at the shipping.
18	PE2_DCLS_DIS	[For U2B24-FCC/U2B20-FCC/U2B10-FCC/U2B24/U2B20/U2B10] PE2 DCLS Disable bit 0: DCLS Enabled 1: DCLS Disabled [For U2B6-FCC/U2B6] Set the value of valid area at the shipping.
17	PE1_DCLS_DIS	PE1 DCLS Disable bit 0: DCLS Enabled 1: DCLS Disabled
16	PE0_DCLS_DIS	PE0 DCLS Disable bit 0: DCLS Enabled 1: DCLS Disabled
15, 14	Reserved	Set the value of valid area at the shipping.
13	PE5_HVE	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] PE5 Hypervisor Enable 0: Hypervisor Disabled 1: Hypervisor Enabled [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.

Table 63.65 OPBT9 Contents (2/2)

Bit Position	Bit Name	Function
12	PE4_HVE	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] PE4 Hypervisor Enable 0: Hypervisor Disabled 1: Hypervisor Enabled [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
11	PE3_HVE	[For U2B24-FCC/U2B20-FCC/U2B10-FCC/U2B24/U2B20/U2B10] PE3 Hypervisor Enable 0: Hypervisor Disabled 1: Hypervisor Enabled [For U2B6-FCC/U2B6] Set the value of valid area at the shipping.
10	PE2_HVE	PE2 Hypervisor Enable 0: Hypervisor Disabled 1: Hypervisor Enabled
9	PE1_HVE	PE1 Hypervisor Enable 0: Hypervisor Disabled 1: Hypervisor Enabled
8	PE0_HVE	PE0 Hypervisor Enable 0: Hypervisor Disabled 1: Hypervisor Enabled
7 to 0	Reserved	Set the value of valid area at the shipping.

4.2.10 OPBT10

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03C8_H$

Value after reset: Specified by the user

Value at the shipping: $F9FD\ 288E_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	MOSC_EXCLK_INPUT	—	MOSC_FREQ[2:0]			—	MOSC_AMP_SEL_A[2:0]			—	MOSC_AMP_SEL_B[2:0]		
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	MOSC_CAP_SEL[3:0]				—	MOSC_RD_SEL_A[2:0]			—	MOSC_RD_SEL_B[2:0]			—	—	MOSC_SHTSTBY_A	MOSC_SHTSTBY_B
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.66 OPBT10 Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	Reserved	Set the value of valid area at the shipping.
28	MOSC_EXCLKINPUT	Main OSC input clock select. 0: Direct clock input to X1 (EXCLK mode). Main OSC amplifier is disabled. 1: Normal crystal oscillation. Main OSC amplifier is enabled.
27	Reserved	Set the value of valid area at the shipping.
26 to 24	MOSC_FREQ[2:0]	Main oscillation frequency selection bit 3'b011:40MHz 3'b010:24MHz 3'b001:20MHz 3'b000:16MHz 3'b1xx:25MHz
23	Reserved	Set the value of valid area at the shipping.
22 to 20	MOSC_AMP_SEL_A[2:0]	Main OSC trimming configuration These bits control OSC drivability during oscillation destabilization.
19	Reserved	Set the value of valid area at the shipping.
18 to 16	MOSC_AMP_SEL_B[2:0]	Main OSC trimming configuration These bits control OSC drivability during oscillation stabilization.
15 to 12	MOSC_CAP_SEL[3:0]	Main OSC trimming configuration These bits control internal capacitance.
11	Reserved	Set the value of valid area at the shipping.
10 to 8	MOSC_RD_SEL_A[2:0]	Main OSC trimming configuration These bits control Damping resistor during oscillation destabilization.
7	Reserved	Set the value of valid area at the shipping.
6 to 4	MOSC_RD_SEL_B[2:0]	Main OSC trimming configuration These bits control Damping resistor during oscillation stabilization.

Table 63.66 OPBT10 Contents (2/2)

Bit Position	Bit Name	Function
3, 2	Reserved	Set the value of valid area at the shipping.
1	MOSC_SHTSTBY_A	Main OSC trimming configuration This bit controls OSC drivability during oscillation destabilization. MOSC_SHTSTBY_A must be set to 1.
0	MOSC_SHTSTBY_B	Main OSC trimming configuration This bit controls OSC drivability during oscillation stabilization. MOSC_SHTSTBY_B must be set to 0.

For details of these data, see **Section 66.2.2, Input Voltage Characteristics** and **Section 66.3.4, Clock Timing.**

4.2.11 OPBT11

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAIk_base> + $03CC_H$

Value after reset: Specified by the user

Value at the shipping: EFFF $FFFF_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CKDIVMD[1:0]		—	START UPPLL	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.67 OPBT11 Contents

Bit Position	Bit Name	Function
31, 30	CKDIVMD[1:0]	Products of CPU Frequency & CPU System Clock Setting $0x_B$: 240 MHz 10_B: 320 MHz 11_B: 400 MHz
29	Reserved	Set the value of valid area at the shipping.
28	STARTUPPLL	Start Up of Main OSC and PLL after reset released except DeepSTOP Reset. This setting has no effect in Serial Programming Mode. 0: Main OSC and PLL are enabled 1: Main OSC and PLL are disabled
27 to 0	Reserved	Set the value of valid area at the shipping.

For details of these data, see **Section 15, Clock Controller.**

4.2.12 OPBT12

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03D0_H$

Value after reset: Specified by the user

Value at the shipping: FFFF $FFFD_H$ (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MAPMODE[1:0]	
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.68 OPBT12 Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	Set the value of valid area at the shipping.
1, 0	MAPMODE[1:0]	Code Flash Memory Mapping Mode Select 00_B: Double Map Mode. 01_B: Single Map Mode. 10_B: setting prohibited. 11_B: setting prohibited.

4.2.13 OPBT13

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAIk_base> + $03D4_H$

Value after reset: Specified by the user

Value at the shipping: FFFF $FFFF_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	DBMAP SW2	DBMAP SW1	DBMAP SW0
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.69 OPBT13 Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	Set the value of valid area at the shipping.
2	DBMAPSW2	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] Double Map Mode Switching of Cluster 2 0: Bank F is mapped in valid area. 1: Bank E is mapped in valid area. [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
1	DBMAPSW1	[For U2B24-FCC/U2B20-FCC/U2B10-FCC/U2B24/U2B20/U2B10] Double Map Mode Switching of Cluster 1 0: Bank D is mapped in valid area. 1: Bank C is mapped in valid area. [For U2B6-FCC/U2B6] Set the value of valid area at the shipping.
0	DBMAPSW0	Double Map Mode Switching of Cluster 0 0: Bank B is mapped in valid area. 1: Bank A is mapped in valid area.

4.2.14 OPBT14

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03D8_H$

Value after reset: Specified by the user

Value at the shipping: See **Table 63.51, Relation between each device type and the shipping value of Option Bytes (3)** (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SGMII0_PICNTSEL[7:0]							
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	—	SGMII0 POL_INV_TX	SGMII0 POL_INV_RX	—	—	—	—	—	—	—	SGMII0_RISRCREN[1:0]	
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.70 OPBT14 Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	Set the value of valid area at the shipping.
23 to 16	SGMII0_PICNTSEL[7:0]	Phase Interporator Control Select of SGMII0 bit[7] Prevention of High frequency Jitter tolerance degradation 0: OFF 1: ON bit[6:5] Setting of frequency deviation(UDCNTF) 00_B:200ppm 01_B:600ppm 10_B:6700ppm 11_B:8000ppm bit[4:3] Setting of frequency detection loop counter(UDCNTPF) 00_B: 2 01_B: 4 10_B: 8 11_B:16 bit[2:0] Setting of phase detect counter 000_B: 8 001_B:10 010_B:12 011_B:14 100_B:16 101_B:18 110_B:20 111_B:22
15 to 11	Reserved	Set the value of valid area at the shipping.
10	SGMII0_POL_INV_TX	SGMII0 transmitter polarity inversion (U2B-FCC(U2B24/U2B20 Mode)/U2B24/U2B20) 0: Polarity of ETH0_SG_TXD_P/ETH0_SG_TXD_N pin is inverted. 1: Polarity of ETH0_SG_TXD_P/ETH0_SG_TXD_N pin is not inverted.

Table 63.70 OPBT14 Contents (2/2)

Bit Position	Bit Name	Function
9	SGMII0_POL_INV_RX	SGMII0 receiver polarity inversion (U2B-FCC(U2B24/U2B20 Mode)/U2B24/U2B20) 0: Polarity of ETH0_SG_RXD_P/ETH0_SG_RXD_N pin is inverted. 1: Polarity of ETH0_SG_RXD_P/ETH0_SG_RXD_N pin is not inverted.
8 to 2	Reserved	Set the value of valid area at the shipping.
1, 0	SGMII0_RISRCREN[1:0]	SGMII0 receiver termination selectors 00_B: Hi-Z 01_B: Setting prohibited 10_B: 50ohm + 50ohm between ETH0_SG_RXD_P and ETH0_SG_RXD_N pins (DC coupled) 11_B: 50ohm to Vcm (AC coupled with IEEE802.3-2015 compliant capacitor)

4.2.15 OPBT15

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03DC_H$

Value after reset: Specified by the user

Value at the shipping: See **Table 63.51, Relation between each device type and the shipping value of Option Bytes (3)** (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SGMII1_PICNTSEL[7:0]							
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	—	SGMII1_POL_INV_TX	SGMII1_POL_INV_RX	—	—	—	—	—	—	—	SGMII1_RISRCREN[1:0]	
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.71 OPBT15 Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	Set the value of valid area at the shipping.
23 to 16	SGMII1_PICNTSEL[7:0]	Phase Interporator Control Select of SGMII1 bit[7] Prevention of High frequency Jitter tolerance degradation 0: OFF 1: ON bit[6:5] Setting of frequency deviation (UDCNTF) 00_B:200ppm 01_B:600ppm 10_B:6700ppm 11_B:8000ppm bit[4:3] Setting of frequency detection loop counter (UDCNTPF) 00_B: 2 01_B: 4 10_B: 8 11_B:16 bit[2:0] Setting of phase detect counter 000_B: 8 001_B:10 010_B:12 011_B:14 100_B:16 101_B:18 110_B:20 111_B:22
15 to 11	Reserved	Set the value of valid area at the shipping.
10	SGMII1_POL_INV_TX	SGMII1 transmitter polarity inversion (U2B-FCC(U2B24/U2B20 Mode)/U2B24/U2B20) 0: Polarity of ETH1_SG_TXD_P/ETH1_SG_TXD_N pin is inverted. 1: Polarity of ETH1_SG_TXD_P/ETH1_SG_TXD_N pin is not inverted.

Table 63.71 OPBT15 Contents (2/2)

Bit Position	Bit Name	Function
9	SGMII1_POL_INV_RX	SGMII1 receiver polarity inversion (U2B-FCC(U2B24/U2B20 Mode)/U2B24/U2B20) 0: Polarity of ETH1_SG_RXD_P/ETH1_SG_RXD_N pin is inverted. 1: Polarity of ETH1_SG_RXD_P/ETH1_SG_RXD_N pin is not inverted.
8 to 2	Reserved	Set the value of valid area at the shipping.
1, 0	SGMII1_RISRCREN[1:0]	SGMII1 receiver termination selectors 00_B: Hi-Z 01_B: Setting prohibited 10_B: 50ohm + 50ohm between ETH1_SG_RXD_P and ETH1_SG_RXD_N pins (DC coupled) 11_B: 50ohm to Vcm (AC coupled with IEEE802.3-2015 compliant capacitor)

4.2.16 OPBT16

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*
Address: <CSAk_base> + $03E0_H$
Value after reset: Specified by the user
Value at the shipping: $74FF\ FF10_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVRENABLE	—	SVRAJPRDSR[5:0]						—	SVRAJDTN[2:0]			—	SVRAJDTP[2:0]		
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	—	—	—	—	SVRADNSMP[7:0]							
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.72 OPBT16 Contents

Bit Position	Bit Name	Function
31	SVRENABLE	SVR Enable setting. 0: Disabled (default) 1: Enabled **CAUTION** Make sure that the all SVR parameters to be set to OPBT16-23, 25 and 28 are correct before enabling SVR. Otherwise, the output voltage of Power MOSFET may be unintentional value.
30	Reserved	Set the value of valid area at the shipping.
29 to 24	SVRAJPRDSR[5:0]	SVR Adjusting slew rate (drive ability).
23	Reserved	Set the value of valid area at the shipping.
22 to 20	SVRAJDTN[2:0]	SVR Adjusting dead time of Lo-side:OFF->Hi-side:ON.
19	Reserved	Set the value of valid area at the shipping.
18 to 16	SVRAJDTP[2:0]	SVR Adjusting dead time of Hi-side:OFF->Lo-side:ON.
15 to 8	Reserved	Set the value of valid area at the shipping.
7 to 0	SVRADNSMP[7:0]	SVR ADC Sampling time.

For details of these data, see **Section 12, Power Supply Circuit.**

4.2.17 OPBT17

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03E4_H$

Value after reset: Specified by the user

Value at the shipping: $F9AA\ FCFC_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVRADTHRESHE[7:0]								SVRADTHRESH[7:0]							
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	SVRMAXDUTY[7:0]								—	—	SVRMINSKIPDUTY[1:0]		—	—	SVRFSWMODE[1:0]	
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.73 OPBT17 Contents

Bit Position	Bit Name	Function
31 to 24	SVRADTHRESHE[7:0]	SVR ADC conversion completion state setting.
23 to 16	SVRADTHRESH[7:0]	SVR ADC sampling start state setting.
15 to 8	SVRMAXDUTY[7:0]	SVR Maximum on-duty setting.
7, 6	Reserved	Set the value of valid area at the shipping.
5, 4	SVRMINSKIPDUTY[1:0]	SVR Skip minimum pulse setting.
3, 2	Reserved	Set the value of valid area at the shipping.
1, 0	SVRFSWMODE[1:0]	SVR Switching frequency setting.

For details of these data, see **Section 12, Power Supply Circuit.**

4.2.18 OPBT18

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03E8_H$

Value after reset: Specified by the user

Value at the shipping: $C26B\ C1CE_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SVRKIVSCL[13:0]													
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	SVRKPVSCL[13:0]													
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.74 OPBT18 Contents

Bit Position	Bit Name	Function
31, 30	Reserved	Set the value of valid area at the shipping.
29 to 16	SVRKIVSCL[13:0]	SVR Scaled KI value of VPID.
15, 14	Reserved	Set the value of valid area at the shipping.
13 to 0	SVRKPVSCL[13:0]	SVR Scaled KP value of VPID.

For details of these data, see **Section 12, Power Supply Circuit.**

4.2.19 OPBT19

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03EC_H$

Value after reset: Specified by the user

Value at the shipping: FFFF $C203_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	SVRKDVSCL[13:0]													
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.75 OPBT19 Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	Set the value of valid area at the shipping.
13 to 0	SVRKDVSCL[13:0]	SVR Scaled KD value of VPID.

For details of these data, see **Section 12, Power Supply Circuit.**

4.2.20 OPBT20

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03F0_H$

Value after reset: Specified by the user

Value at the shipping: $CB88\ FFEF_H$ (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVRENABLEB	—	SVRAJPRDSRB[5:0]						—	SVRAJDTNB[2:0]			—	SVRAJDTPB[2:0]		
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	—	—	—	—	SVRADNSMPB[7:0]							
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.76 OPBT20 Contents

Bit Position	Bit Name	Function
31	SVRENABLEB	Program inverted value of SVRENABLE.
30	Reserved	Set the value of valid area at the shipping.
29 to 24	SVRAJPRDSRB[5:0]	Program inverted value of SVRAJPRDSR[5:0].
23	Reserved	Set the value of valid area at the shipping.
22 to 20	SVRAJDTNB[2:0]	Program inverted value of SVRAJDTN[2:0].
19	Reserved	Set the value of valid area at the shipping.
18 to 16	SVRAJDTPB[2:0]	Program inverted value of SVRAJDTP[2:0].
15 to 8	Reserved	Set the value of valid area at the shipping.
7 to 0	SVRADNSMPB[7:0]	Program inverted value of SVRADNSMP[7:0].

For details of these data, see **Section 12, Power Supply Circuit.**

4.2.21 OPBT21

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03F4_H$

Value after reset: Specified by the user

Value at the shipping: 0655 $03CF_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVRADTHRESHEB[7:0]								SVRADTHRESHB[7:0]							
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	SVRMAXDUTYB[7:0]								—	—	SVRMINSKIPDUTYB[1:0]		—	—	SVRFSWMODEB[1:0]	
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.77 OPBT21 Contents

Bit Position	Bit Name	Function
31 to 24	SVRADTHRESHEB [7:0]	Program inverted value of SVRADTHRESHE[7:0].
23 to 16	SVRADTHRESHB [7:0]	Program inverted value of SVRADTHRESH[7:0].
15 to 8	SVRMAXDUTYB[7:0]	Program inverted value of SVRMAXDUTY[7:0].
7, 6	Reserved	Set the value of valid area at the shipping.
5, 4	SVRMINSKIPDUTYB [1:0]	Program inverted value of SVRMINSKIPDUTY[1:0].
3, 2	Reserved	Set the value of valid area at the shipping.
1, 0	SVRFSWMODEB [1:0]	Program inverted value of SVRFSWMODE[1:0].

For details of these data, see **Section 12, Power Supply Circuit.**

4.2.22 OPBT22

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03F8_H$

Value after reset: Specified by the user

Value at the shipping: $FD94\ FE31_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SVRKIVSCLB[13:0]													
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	SVRKPVSCLB[13:0]													
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.78 OPBT22 Contents

Bit Position	Bit Name	Function
31, 30	Reserved	Set the value of valid area at the shipping.
29 to 16	SVRKIVSCLB[13:0]	Program inverted value of SVRKIVSCL[13:0].
15, 14	Reserved	Set the value of valid area at the shipping.
13 to 0	SVRKPVSCLB[13:0]	Program inverted value of SVRKPVSCL[13:0].

For details of these data, see **Section 12, Power Supply Circuit.**

4.2.23 OPBT23

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $03FC_H$

Value after reset: Specified by the user

Value at the shipping: FFFF $FDFC_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	SVRKDVSCLB[13:0]													
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.79 OPBT23 Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	Set the value of valid area at the shipping.
13 to 0	SVRKDVSCLB[13:0]	Program inverted value of SVRKDVSCL[13:0].

For details of these data, see **Section 12, Power Supply Circuit.**

4.2.24 OPBT24

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + 0400_H

Value after reset: Specified by the user

Value at the shipping: See **Table 63.49, Relation between each device type and the shipping value of Option Bytes (1)** (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	—	—	—	—	—	—	—	PE4_DCLS_DIS_REDUN	PE3_DCLS_DIS_REDUN	PE2_DCLS_DIS_REDUN	PE1_DCLS_DIS_REDUN	PE0_DCLS_DIS_REDUN
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.80 OPBT24 Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	Set the value of valid area at the shipping.
4	PE4_DCLS_DIS_REDUN	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] Redundant bit for PE4 DCLS Disable bit The bit have to be same as PE4_DCLS_DIS [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
3	PE3_DCLS_DIS_REDUN	[For U2B24-FCC/U2B24] Redundant bit for PE3 DCLS Disable bit The bit have to be same as PE3_DCLS_DIS [For U2B20-FCC/U2B10-FCC/U2B6-FCC/U2B20/U2B10/U2B6] Set the value of valid area at the shipping.
2	PE2_DCLS_DIS_REDUN	[For U2B24-FCC/U2B20-FCC/U2B10-FCC/U2B24/U2B20/U2B10] Redundant bit for PE2 DCLS Disable bit The bit have to be same as PE2_DCLS_DIS [For U2B6-FCC/U2B6] Set the value of valid area at the shipping.
1	PE1_DCLS_DIS_REDUN	Redundant bit for PE1 DCLS Disable bit The bit have to be same as PE1_DCLS_DIS
0	PE0_DCLS_DIS_REDUN	Redundant bit for PE0 DCLS Disable bit The bit have to be same as PE0_DCLS_DIS

4.2.25 OPBT25

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + 0404_H

Value after reset: Specified by the user

Value at the shipping: FFC0 F3FF$_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	SVRAJSSCGF[1:0]		SVRAJSSCGD[1:0]		SVRENSSCG	SVRENDCDCHZ
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	SVRAJSSCGDITH[1:0]		STOPCRDMD[1:0]		—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.81 OPBT25 Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	Set the value of valid area at the shipping.
21 to 20	SVRAJSSCGF[1:0]	SVR(DCDC) the modulation frequency of SSCG 00: Fsw/Fm=500 01: Fsw/Fm=250 10: Fsw/Fm=125 11: Fsw/Fm=62.5
19 to 18	SVRAJSSCGD[1:0]	SVR(DCDC) the peak deviation of SSCG 00: 2% 01: 4% 10: 6% 11: Setting prohibited
17	SVRENSSCG	SVR(DCDC) Enable SSCG 1: ON 0: OFF
16	SVRENDCDCHZ	SVR(DCDC) Set SVRPGATE/SVRNGATE to Hiz 0: Fixed 1: Hiz
15 to 12	Reserved	Set the value of valid area at the shipping.
11 to 10	SVRAJSSCGDITH[1:0]	The control signal for dither of SSCG 00: off 01: weak 10: medium 11: strong
9 to 8	STOPCRDMD[1:0]	Current reduction mode in STOP mode 0*: Not current reduction mode (Keeping SVR in PWM mode) 10: Current reduction mode 1 (Keeping SVR in PWM mode) 11: Current reduction mode 2 (SVR is changed from PWM mode to PFM mode)
7 to 0	Reserved	Set the value of valid area at the shipping.

4.2.26 OPBT26

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + 0408_H

Value after reset: Specified by the user

Value at the shipping: FFFF 0010_H (valid area)
Erased (Invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	OPWDWOST0[15:0]															
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.82 OPBT26 Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	Set the value of valid area at the shipping.
15 to 0	OPWDWOST0[15:0]	These bits specify the initial value of WDTB0 Window Open Start Register for setting the start timing of the window open.

4.2.27 OPBT27

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAIk_base>+ $040C_H$

Value after reset: Specified by the user

Value at the shipping: $FFFF\ 0000_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	OPWDWIS0[15:0]															
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.83 OPBT27 Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	Set the value of valid area at the shipping.
15 to 0	OPWDWIS0[15:0]	These bits specify the initial value of WDTB0 Interrupt Output Timing Setting Register.

4.2.28 OPBT28

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAIk_base> + 0410_H

Value after reset: Specified by the user

Value at the shipping: FFFF $FFFF_H$ (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	SVRAJSSCGFB [1:0]		SVRAJSSCGDB [1:0]		SVRENSSCGB	SVRENDCDCHZB
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	—	—	—	—	SVRAJSSCGDITHB[1:0]		—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.84 OPBT28 Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	Set the value of valid area at the shipping.
21 to 20	SVRAJSSCGFB [1:0]	Program inverted value of SVRAJSSCGF[1:0].
19 to 18	SVRAJSSCGDB [1:0]	Program inverted value of SVRAJSSCGD[1:0].
17	SVRENSSCGB	Program inverted value of SVRENSSCG.
16	SVRENDCDCHZB	Program inverted value of SVRENDCDCHZ.
15 to 12	Reserved	Set the value of valid area at the shipping.
11 to 10	SVRAJSSCGDITHB[1:0]	Program inverted value of SVRAJSSCGDITH[1:0].
9 to 0	Reserved	Set the value of valid area at the shipping.

4.2.29 OPBT36

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + 0430_H

Value after reset: Specified by the user

Value at the shipping: See **Table 63.45, Relation between each device type and the shipping value of Option Bytes (1)** (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	PE5_DISABLE_REDUN	PE4_DISABLE_REDUN	PE3_DISABLE_REDUN	PE2_DISABLE_REDUN	PE1_DISABLE_REDUN	—	SSCG1_DIS	SSCG1_M[1:0]		SSCG1_NI[6:2]				
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	SSCG1_NI[1:0]		SSCG1_FVV[2:0]			SSCG1_P[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.85 OPBT36 Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	Set the value of valid area at the shipping.
29	PE5_DISABLE_REDUN	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] Redundant bit for PE5_DISABEL bit The bit have to be same as PE5_DISABLE [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
28	PE4_DISABLE_REDUN	[For U2B24-FCC/U2B20-FCC/U2B24/U2B20] Redundant bit for PE4_DISABEL bit The bit have to be same as PE4_DISABLE [For U2B10-FCC/U2B6-FCC/U2B10/U2B6] Set the value of valid area at the shipping.
27	PE3_DISABLE_REDUN	[For U2B24-FCC/U2B20-FCC/U2B10-FCC/U2B24/U2B20/U2B10] Redundant bit for PE3_DISABEL bit The bit have to be same as PE3_DISABLE [For U2B6-FCC/U2B6] Set the value of valid area at the shipping.
26	PE2_DISABLE_REDUN	Redundant bit for PE2_DISABEL bit The bit have to be same as PE2_DISABLE
25	PE1_DISABLE_REDUN	Redundant bit for PE1_DISABEL bit The bit have to be same as PE1_DISABLE
24	Reserved	Set the value of valid area at the shipping.
23	SSCG1_DIS	SSCG1 disable signal 1: SSCG1 disable 0: SSCG1 enable If this bit is set to 1, SSCG1 cannot be enabled by either OPBT(STARTUPPLL) or Register(PLLE).
22 to 21	SSCG1_M[1:0]	OPBT to M-Value of SSCG1
20 to 14	SSCG1_NI[6:0]	OPBT to NI-Value of SSCG1

Table 63.85 OPBT36 Contents (2/2)

Bit Position	Bit Name	Function
13 to 11	SSCG1_FVV[2:0]	FVV-Value of SSCG1
10 to 8	SSCG1_P[2:0]	P-Value of SSCG1
7 to 0	Reserved	Set the value of valid area at the shipping.

4.2.30 OPBT37

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + 0434_H

Value after reset: Specified by the user

Value at the shipping: $02EF\ FF55_H$ (valid area)
Erased (Invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DFP_DCLS_DIS_SYS	DFP_DCLS_DIS_VLM	DFP_DCLS_DIS_VPU	DFP_DCLS_DIS_ROC	DFP_DCLS_DIS_SPU	DFP_DCLS_DIS_CCU	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	CKSEL_RHSBDW3[1:0]		CKSEL_RHSBDW2[1:0]		CKSEL_RHSBDW1[1:0]		CKSEL_RHSBDW0[1:0]		—	CKSEL_RHSBUP3	—	—	—	—	—	CKSEL_RHSBUP0
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.86 OPBT37 Contents (1/2)

Bit Position	Bit Name	Function
31	DFP_DCLS_DIS_SYS	DFP DCLS Disable bit for SYS 0: DCLS Enabled 1: DCLS Disabled
30	DFP_DCLS_DIS_VLM	DFP DCLS Disable bit for VLM 0: DCLS Enabled 1: DCLS Disabled
29	DFP_DCLS_DIS_VPU	DFP DCLS Disable bit for VPU 0: DCLS Enabled 1: DCLS Disabled
28	DFP_DCLS_DIS_ROC	DFP DCLS Disable bit for ROC 0: DCLS Enabled 1: DCLS Disabled
27	DFP_DCLS_DIS_SPU	DFP DCLS Disable bit for SPU 0: DCLS Enabled 1: DCLS Disabled
26	DFP_DCLS_DIS_CCU	DFP DCLS Disable bit for CCU 0: DCLS Enabled 1: DCLS Disabled
25 to 16	Reserved	Set the value of valid area at the shipping.
15 to 14	CKSEL_RHSBDW3[1:0]	CLK_RHSB_DW3 frequency setting bit 11: CLKC_SHSB (133MHz) 10: CLKC_UHSB (160MHz) 0x: CLKC_SBUS (200MHz)
13 to 12	CKSEL_RHSBDW2[1:0]	CLK_RHSB_DW2 frequency setting bit 11: CLKC_SHSB (133MHz) 10: CLKC_UHSB (160MHz) 0x: CLKC_SBUS (200MHz)
11 to 10	CKSEL_RHSBDW1[1:0]	CLK_RHSB_DW1 frequency setting bit 11: CLKC_SHSB (133MHz) 10: CLKC_UHSB (160MHz) 0x: CLKC_SBUS (200MHz)

Table 63.86 OPBT37 Contents (2/2)

Bit Position	Bit Name	Function
9 to 8	CKSEL_RHSBDW0[1:0]	CLK_RHSB_DW0 frequency setting bit 11: CLKC_SHSB (133MHz) 10: CLKC_UHSB (160MHz) 0x: CLKC_SBUS (200MHz)
7	Reserved	Set the value of valid area at the shipping.
6	CKSEL_RHSBUP3	CLK_RHSB_UP3 frequency setting bit 1: CLKC_CPU (400MHz) 0: CLK_PLL (800MHz)
5 to 1	Reserved	Set the value of valid area at the shipping.
0	CKSEL_RHSBUP0	CLK_RHSB_UP0 frequency setting bit 1: CLKC_CPU (400MHz) 0: CLK_PLL (800MHz)

4.2.31 OPBT38

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2B Group Flash Memory User's Manual: Hardware.*
Address: <CSAk_base> + 0438_H
Value after reset: Specified by the user
Value at the shipping: $0000\ 0000_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DFP_init_boothart[31:16]															
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	DFP_init_boothart[15:0]															
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.87 OPBT38 Contents

Bit Position	Bit Name	Function
31 to 0	DFP_init_boothart[31:0]	[For U2B-FCC only] DFP_init_boothart[31:0]

4.2.32 OPBT39

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $043C_H$

Value after reset: Specified by the user

Value at the shipping: $0000\ 0000_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DFP_resetvec[31:16]															
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	DFP_resetvec[15:0]															
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.88 OPBT39 Contents

Bit Position	Bit Name	Function
31 to 0	DFP_resetvec[31:0]	[For U2B-FCC only] DFP reset vector

4.2.33 OPBT40

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary to program these data.

Access: For setting these data, see *the RH850/U2B Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + $043C_H$

Value after reset: Specified by the user

Value at the shipping: $0000\ 0000_H$ (valid area)
Erased (invalid area)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DFP_resetvec[31:16]															
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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
	DFP_resetvec[15:0]															
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
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. This value is dependent on the value in the flash memory which is specified by the user.

Table 63.88 OPBT39 Contents

Bit Position	Bit Name	Function
31 to 0	DFP_resetvec[31:0]	[For U2B-FCC only] DFP reset vector

Table 63.89 OPBT42 Contents

Bit Position	Bit Name	Function
21	LVDS_ERT5	LVDS (P13_4, P13_5) termination resistor control 0: Disable the termination register 1: Enable the termination register
20	LVDS_ERT4	LVDS (P22_0, P22_1) termination resistor control 0: Disable the termination register 1: Enable the termination register
19	LVDS_ERT3	LVDS (P21_2, P21_3) termination resistor control 0: Disable the termination register 1: Enable the termination register
18	LVDS_ERT2	LVDS (P21_0, P21_1) termination resistor control 0: Disable the termination register 1: Enable the termination register
17	LVDS_ERT1	LVDS (P14_9, P14_10) termination resistor control 0: Disable the termination register 1: Enable the termination register
16	LVDS_ERT0	LVDS (P13_0, P13_1) termination resistor control 0: Disable the termination register 1: Enable the termination register
15 to 0	Reserved	Set the value of valid area at the shipping.

5. Revision

Revision Record

Rev.	Issue date	Revision	
		Page	Point
1.00	2025.5.30		First edition

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

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

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

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

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

Contact information

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