

# RE01 256KB グループ

R01AN5484JJ0100

Rev.1.00

## R\_CORE 詳細仕様書

2020.07.02

### 要旨

本書では、RE01 256KB グループ CMSIS Driver Package のスタートアップルーチン R\_CORE の詳細仕様を説明します。

### 動作確認デバイス

RE01 256KB グループ

### 目次

1. 概要	3
2. ソフトウェアコンポーネントの内部構成	4
2.1 ファイル構成	4
3. ソフトウェアコンポーネントの内部動作	7
3.1 通常起動モード	7
3.2 EHC 起動モード	7
4. ソフトウェアユニット詳細情報	8
4.1 コンフィグレーション	8
4.1.1 メインクロック発振器停止(SYSTEM_CFG_MOSC_ENABLE)	10
4.1.2 メインクロック動作周波数設定(SYSTEM_CFG_MOSC_FREQUENCY_HZ)	10
4.1.3 メインクロック発振器駆動能力切り替え(SYSTEM_CFG_MOSC_DRIVE)	10
4.1.4 メインクロック発振器切り替え(SYSTEM_CFG_MOSC_CLOCK_SOURCE)	11
4.1.5 メインクロック発振器低消費発振機能有効化 (SYSTEM_CFG_MOSC_LOW_POWER_ENABLE)	11
4.1.6 メインクロック発振器安定待ち時間設定(SYSTEM_CFG_MOSC_WAIT_TIME)	11
4.1.7 HOCO 停止(SYSTEM_CFG_HOCO_ENABLE)	12
4.1.8 HOCO 発振周波数設定(SYSTEM_CFG_HOCO_FREQUENCY)	12
4.1.9 FLL 機能有効化(SYSTEM_CFG_FLL_ENABLE)	12
4.1.10 MOCO 停止(SYSTEM_CFG_MOCO_ENABLE)	13
4.1.11 LOCO 停止(SYSTEM_CFG_LOCO_ENABLE)	13
4.1.12 サブクロック発振器停止(SYSTEM_CFG_SOSC_ENABLE)	13
4.1.13 サブクロック発振器駆動能力切り替え(SYSTEM_CFG_SOSC_DRIVE)	13
4.1.14 サブクロック発振器ノイズフィルタ停止(SYSTEM_CFG_SOSC_NF_STOP)	14
4.1.15 クロックソース選択(SYSTEM_CFG_CLOCK_SOURCE)	14
4.1.16 システムクロック(ICLK)/周辺モジュールクロック(PCLKA)選択 (SYSTEM_CFG_ICLK_PCKA_DIV)	15
4.1.17 周辺モジュールクロック B(PCLKB)選択(SYSTEM_CFG_PCKB_DIV)	15
4.1.18 電源状態選択(SYSTEM_CFG_POWER_CONTROL_MODE)	16
4.1.19 EHC 起動モード設定(SYSTEM_CFG_EHC_MODE)	16
4.1.20 電圧検出 1 レベル選択(SYSTEM_CFG_EHC_LVD1LVL)	16
4.1.21 電圧検出 BAT レベル選択(SYSTEM_CFG_EHC_LVDBATLVL)	17
4.1.22 オプション設定メモリ 0(SYSTEM_CFG_OFS0)	17

4.1.23 オプション設定メモリ 1(SYSTEM_CFG_OFS1).....	17
4.1.24 オプション設定メモリ 2(SYSTEM_CFG_SECMPU_xxx) .....	18
4.1.25 オプション設定メモリ 3(SYSTEM_CFG_ID_CODE_PROTECTION_n)(n=1-4) .....	19
4.1.26 オプション設定メモリ 4(SYSTEM_CFG_AWS).....	19
4.1.27 プリプロセッサエラーチェック .....	20
4.2 定数/マクロ .....	21
4.2.1 オプション設定メモリ (OptionSettingMemory).....	21
4.2.2 シリアルプログラマ ID 設定 (OptionSettingID) .....	21
4.2.3 アクセスウィンドウ設定 (OptionSettingAWS).....	21
4.2.4 システムクロックソース定義マクロ (SYSTEM_CLOCK_SEL).....	22
4.2.5 システムクロック定義マクロ (SYSTEM_CLOCK) .....	22
4.2.6 システムクロックウェイトサイクル定義 .....	23
4.3 関数仕様 .....	24
4.3.1 system_clock_init 関数 .....	24
4.3.2 system_mosc_start 関数 .....	29
4.3.3 system_mosc_stop 関数 .....	30
4.3.4 system_hoco_start 関数 .....	31
4.3.5 system_hoco_stop 関数 .....	32
4.3.6 system_moco_stop 関数 .....	33
4.3.7 system_loco_start 関数 .....	34
4.3.8 system_loco_stop 関数 .....	35
4.3.9 system_sosc_start 関数 .....	36
4.3.10 system_sosc_stop 関数 .....	39
4.3.11 system_normal_to_boost 関数.....	40
4.3.12 system_normal_to_vbb 関数.....	41
4.3.13 Reset_Handler 関数 .....	42
4.3.14 SystemInit 関数 .....	43
4.4 付録 .....	44
4.4.1 プリプロセッサエラー一覧 .....	44
5. EHC 起動モード .....	48
5.1 ファイル構成 .....	48
5.2 EHC 起動モード遷移 .....	51
5.3 コンフィグレーション .....	52
5.4 関数仕様 .....	52
5.4.1 r_ehc_Startup 関数 .....	54
5.4.2 ehc_Startup_ramfunc 関数.....	58
5.4.3 NMI_Handler_EHC 関数 .....	62
5.4.4 IEL1_IRQHandler_EHC 関数 .....	63

## 1. 概要

本書における略語一覧と関連文書一覧を示します。

表 1-1 略語一覧

名称	略語
RE01 グループ ユーザーズマニュアル ハードウェア編	UMH
RENESAS CMSIS-Core	R_CORE

表 1-2 関連文書一覧

文書名	文書番号
RE01 グループ 256KB フラッシュメモリ搭載製品 ユーザーズマニュアル ハードウェア編	R01UH0894
RE01 1500KB,256KB グループ CMSIS パッケージを用いた開発スタートアップガイド	R01AN4660

## 2. ソフトウェアコンポーネントの内部構成

### 2.1 ファイル構成

R\_CORE は、CMSIS Driver Package のシステム初期化(CMSIS-Core)に該当し、ベンダ独自ファイル格納ディレクトリ内の startup\_RE01\_256KB.c、system\_RE01\_256KB.c、system\_RE01\_256KB.h、r\_core\_cfg.h の4個のファイルで構成されます。

各ファイルの役割を表 2-1 に示します。RE01 CMSIS Driver Package における R\_CORE のファイル構成を図 2-1 に示します。内部構成については図 2-2 に示します。

表 2-1 R\_CORE 各ファイルの役割

ファイル名	内容
startup_RE01_256KB.c	R_CORE スタートアップソースファイルです。 ベクタテーブル宣言、システム初期化関数コールを行います。 システム初期化をする場合は、本ファイルをビルドする必要があります。
system_RE01_256KB.c	R_CORE システム初期化ソースファイルです。 システム初期化関数の実体を用意します。 システム初期化をする場合は、本ファイルをビルドする必要があります。
system_RE01_256KB.h	R_CORE システム初期化ヘッダファイルです。 ユーザが参照可能なマクロ／型／プロトタイプ宣言を用意します。 システム初期化をする場合は、本ファイルをインクルードする必要があります。
r_core_cfg.h	R_CORE コンフィグレーション定義ファイルです。 ユーザが設定可能なコンフィグレーション定義を用意します。

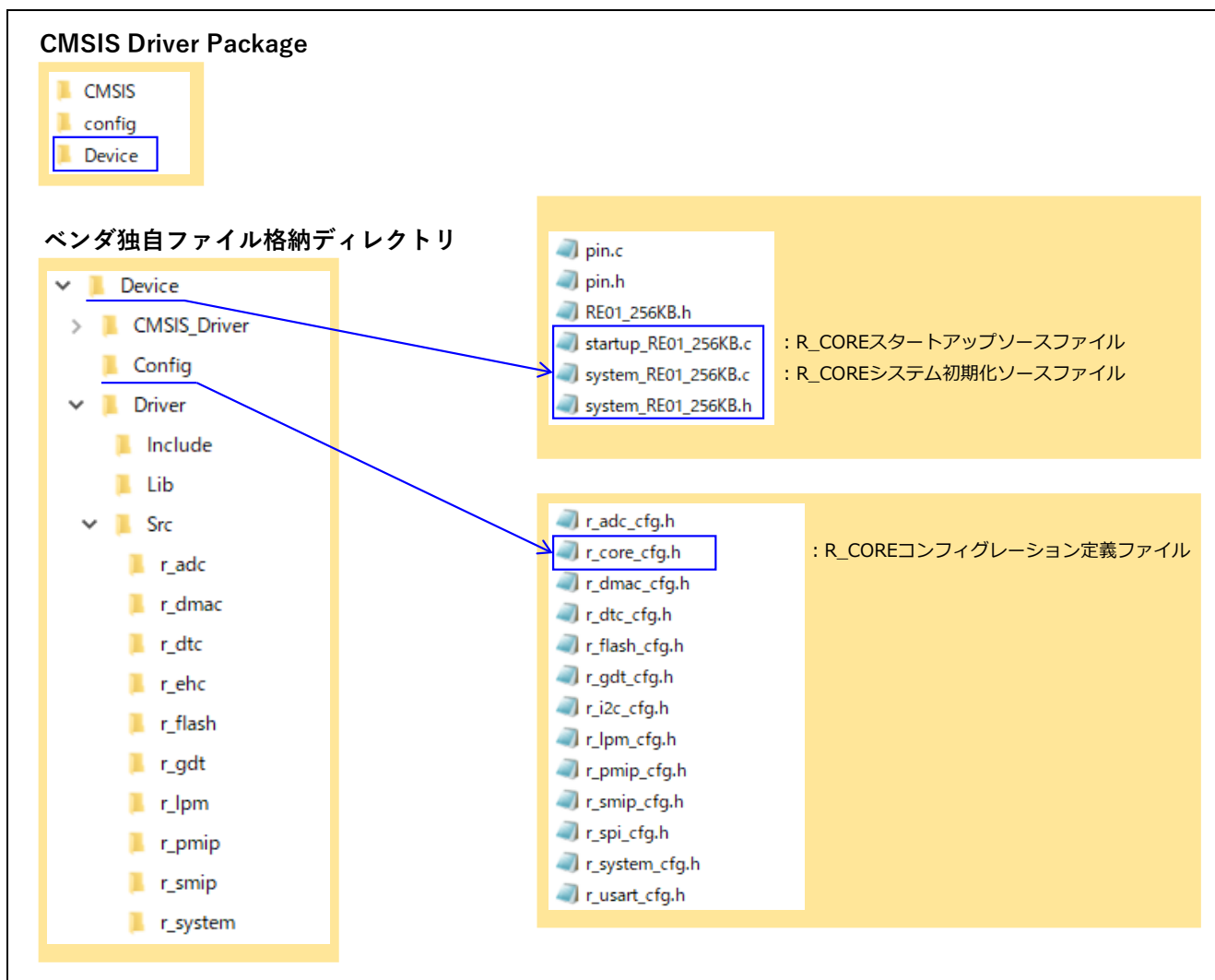


図 2-1 CMSIS Driver Package における R\_CORE のファイル構成

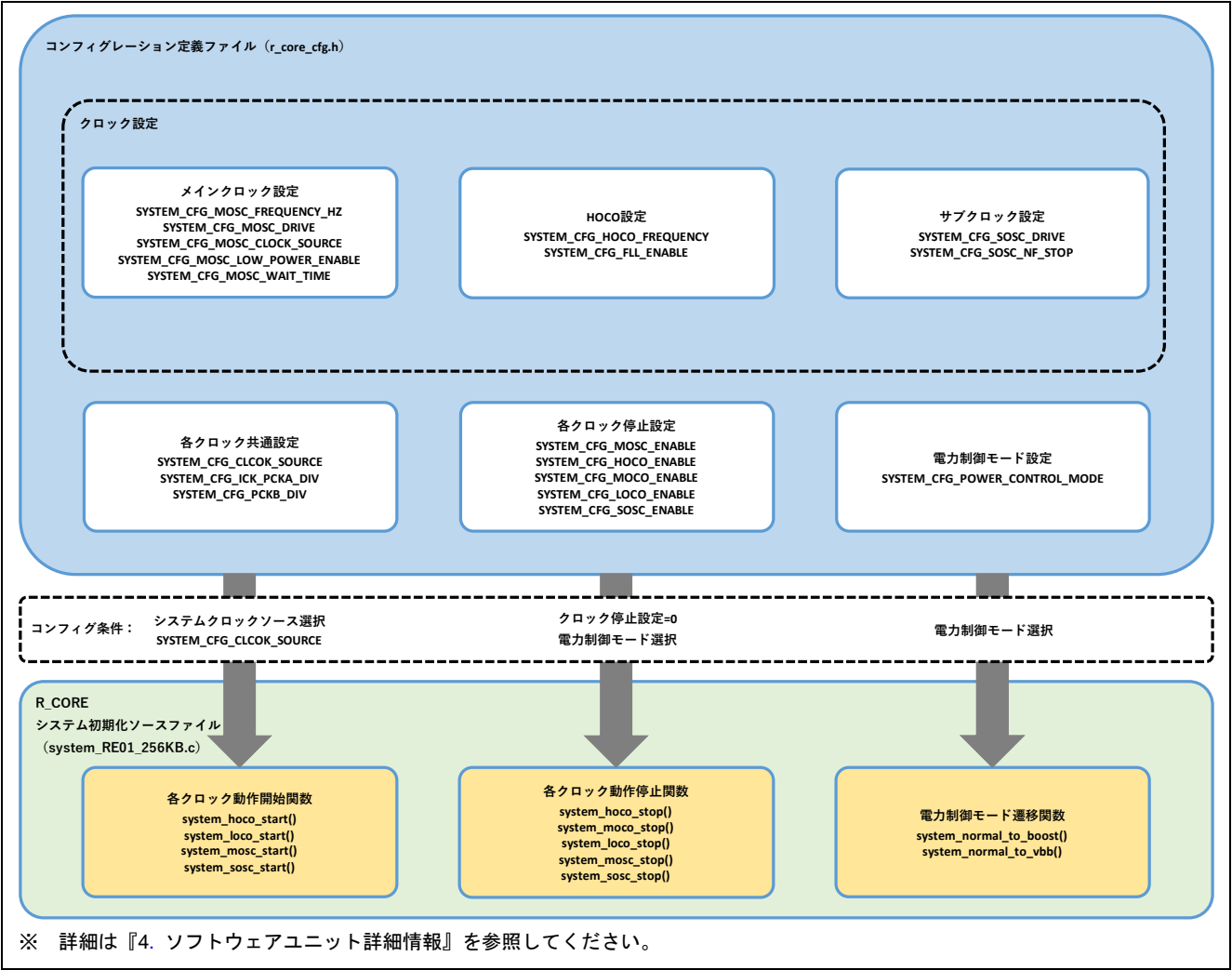


図 2-2 R\_CORE 関数と R\_CORE コンフィグ設定の関係

### 3. ソフトウェアコンポーネントの内部動作

R\_CORE は、初期動作として startup\_RE01\_256KB.c より Reset\_Handler 実行、SystemInit 関数をコールしてシステムの初期化を行います。詳細仕様については、『4.3.13 Reset\_Handler 関数』および『4.3.14 SystemInit 関数』を参照ください。

#### 3.1 通常起動モード

通常起動モードへ遷移する手順を示します。

エナジーハーベスト(EHC)起動モード設定(SYSTEM\_CFG\_EHC\_MODE)が無効の場合は、EHC 起動処理を行わず、通常起動を行います。

通常起動時は、リセット後にシステム初期化と RAM 初期化を行った後、main 関数へジャンプします。

システム初期化では、ボード依存となる端子の初期設定と、コンフィグレーション定義ファイルに従った動作周波数および電力制御モードの設定を行います。

コンフィグレーション定義ファイルのデフォルト設定は、EHC 起動モード無効に設定されています。

#### 3.2 EHC 起動モード

EHC 起動モードへ遷移する手順を示します。

エナジーハーベスト(EHC)起動モード設定(SYSTEM\_CFG\_EHC\_MODE)が有効の場合は、EHC 起動処理を行います。

EHC 起動時は、リセット後に EHC 機能を用いて二次電池の充電を行います。

二次電池の充電が完了した後は、通常起動と同様に、システム初期化と RAM 初期化を行った後、main 関数へジャンプします。

注. R\_CORE の EHC 起動処理はオプション機能となります。

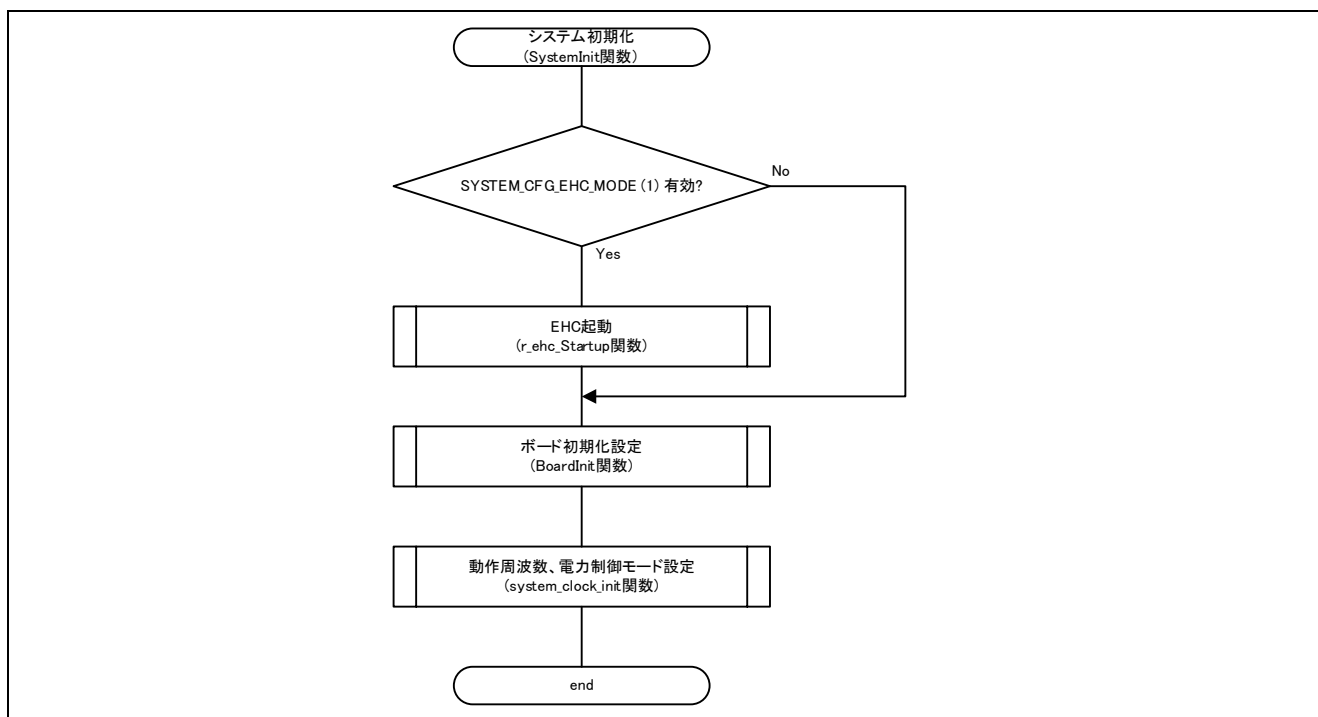


図 3-1 SystemInit 関数による起動モード遷移

## 4. ソフトウェアユニット詳細情報

### 4.1 コンフィグレーション

R\_CORE は、ユーザが定義可能なコンフィグレーションを `r_core_cfg.h` ファイルに用意します。

各初期設定の一覧を表 4-1 に示します。各設定については、表内の章番号を参照してください。

表 4-1 R\_CORE コンフィグレーションの初期設定一覧

章番号	コンフィグレーション	設定内容	初期値
4.1.1	SYSTEM_CFG_MOSC_ENABLE	メインクロック(MOSC)の動作/停止を設定	0
4.1.2	SYSTEM_CFG_MOSC_FREQUENCY_HZ	メインクロック(MOSC)の動作周波数を設定	32000000
4.1.3	SYSTEM_CFG_MOSC_DRIVE	メインクロック発振器駆動能力を設定	7
4.1.4	SYSTEM_CFG_MOSC_CLOCK_SOURCE	メインクロック発振器の発振源を設定	0
4.1.5	SYSTEM_CFG_MOSC_LOW_POWER_ENABLE	メインクロック発振器低消費発振機能を設定	0
4.1.6	SYSTEM_CFG_MOSC_WAIT_TIME	メインクロック発振器安定待ち時間を設定	5
4.1.7	SYSTEM_CFG_HOCO_ENABLE	高速オンチップオシレータ(HOCO)の動作/停止を設定	0
4.1.8	SYSTEM_CFG_HOCO_FREQUENCY	高速オンチップオシレータ(HOCO)の発振周波数を設定	0
4.1.9	SYSTEM_CFG_FLL_ENABLE	FLL の有効/無効を設定	0
4.1.10	SYSTEM_CFG_MOCO_ENABLE	中速オンチップオシレータ(MOCO)の動作/停止を設定	1
4.1.11	SYSTEM_CFG_LOCO_ENABLE	低速オンチップオシレータ(LOCO)の動作/停止を設定	1
4.1.12	SYSTEM_CFG_SOSC_ENABLE	サブクロック(SOSC)の動作/停止を設定	0
4.1.13	SYSTEM_CFG_SOSC_DRIVE	サブクロック発振器駆動能力を設定	0
4.1.14	SYSTEM_CFG_SOSC_NF_STOP	サブクロック発振器ノイズフィルタの動作/停止を設定	0
4.1.15	SYSTEM_CFG_CLCOK_SOURCE	システムクロックのクロックソースを設定	1
4.1.16	SYSTEM_CFG_ICLK_PCKA_DIV	システムクロック(ICLK)と周辺モジュールクロック A(PCLKA)の分周値を設定	0
4.1.17	SYSTEM_CFG_PCKB_DIV	周辺モジュールクロック B(PCLKB)を設定	0
4.1.18	SYSTEM_CFG_POWER_CONTROL_MODE	電力制御モードを設定	1
4.1.19	SYSTEM_CFG_EHC_MODE	EHC 起動処理の有効/無効を設定	0
4.1.20	SYSTEM_CFG_EHC_LVD1LVL	EHC 起動時の電圧検出 1 レベルを設定	7
4.1.21	SYSTEM_CFG_EHC_LVDBATLVL	EHC 起動時の電圧検出 BAT レベルを設定	0
4.1.22	SYSTEM_CFG_OFS0	オプション機能選択レジスタ 0(OFS0)の値を設定	0xFFFFFFFF
4.1.23	SYSTEM_CFG_OFS1	オプション機能選択レジスタ 1(OFS1)の値を設定	0xFFFFFFFF
4.1.24	SYSTEM_CFG_SECMPU_PC0_START	コードフラッシュ、または SRAM のセキュリティフェッチ領域 0(開始アドレス)を設定	0xFFFFFFFF
4.1.24	SYSTEM_CFG_SECMPU_PC0_END	コードフラッシュ、または SRAM のセキュリティフェッチ領域 0(終了アドレス)を設定	0xFFFFFFFF
4.1.24	SYSTEM_CFG_SECMPU_PC1_START	コードフラッシュ、または SRAM のセキュリティフェッチ領域 1(開始アドレス)を設定	0xFFFFFFFF
4.1.24	SYSTEM_CFG_SECMPU_PC1_END	コードフラッシュ、または SRAM のセキュリティフェッチ領域 1(終了アドレス)を設定	0xFFFFFFFF
4.1.24	SYSTEM_CFG_SECMPU_REGION0_START	コードフラッシュのセキュリティプログラムやデータの開始アドレスを設定	0xFFFFFFFF
4.1.24	SYSTEM_CFG_SECMPU_REGION0_END	コードフラッシュのセキュリティプログラムやデータの終了アドレスを設定	0xFFFFFFFF



4.1.24	SYSTEM_CFG_SECMPU_REGION1_START	SRAM のセキュリティプログラムやデータの開始アドレスを設定	0xFFFFFFFF
4.1.24	SYSTEM_CFG_SECMPU_REGION1_END	SRAM のセキュリティプログラムやデータの終了アドレスを設定	0xFFFFFFFF
4.1.24	SYSTEM_CFG_SECMPU_REGION2_START	セキュリティ IP のセキュリティプログラムやデータの開始アドレスを設定	0xFFFFFFFF
4.1.24	SYSTEM_CFG_SECMPU_REGION2_END	セキュリティ IP のセキュリティプログラムやデータの終了アドレスを設定	0xFFFFFFFF
4.1.24	SYSTEM_CFG_SECMPU_CONTROL_SETTING	セキュリティ有効/無効領域を設定	0xFFFF
4.1.25	SYSTEM_CFG_ID_CODE_PROTECTION_1	OCD/シリアルプログラマ ID (0~31 ビット) を設定	0xFFFFFFFF
4.1.25	SYSTEM_CFG_ID_CODE_PROTECTION_2	OCD/シリアルプログラマ ID (32~63 ビット) を設定	0xFFFFFFFF
4.1.25	SYSTEM_CFG_ID_CODE_PROTECTION_3	OCD/シリアルプログラマ ID (64~95 ビット) を設定	0xFFFFFFFF
4.1.25	SYSTEM_CFG_ID_CODE_PROTECTION_4	OCD/シリアルプログラマ ID (96~127 ビット) を設定	0xFFFFFFFF
4.1.26	SYSTEM_CFG_AWS	アクセスウィンドウ開始/終了のブロックアドレス、スタートアップ領域の選択および選択機能の保護を設定	0xFFFFFFFF

なお、コンフィグレーションの定義が初期設定状態では、以下の表 4-2 起動時の動作クロック、動作モードとなります。コンフィグレーション設定で設定可能な電力制御モードについては図 4-1 を参照ください。

表 4-2 起動時の動作クロック、動作モード

システム クロック ソース	システム クロック 分周 (ICLK, PCLKA)	周辺 クロック 分周 (PCLKB)	電源供給 モード	電力制御 モード	メイン クロック	サブ クロック	HOCO	LOCO
MOCO (2MHz)	1 分周 (2MHz)	1 分周 (2MHz)	ALLPWON	Normal High-Speed	停止	停止	停止	動作

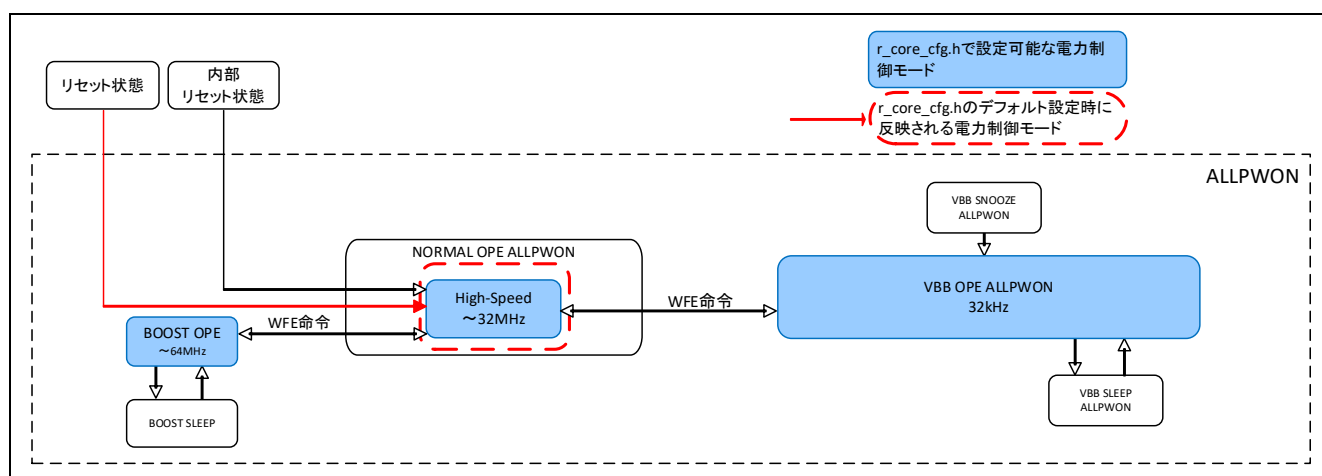


図 4-1 コンフィグレーションファイル(r\_core\_cfg.h)の設定で設定可能な電力制御モード

## 4.1.1 メインクロック発振器停止(SYSTEM\_CFG\_MOSC\_ENABLE)

R\_CORE におけるメインクロック(MOSC)の動作/停止を設定します。

名称 : SYSTEM\_CFG\_MOSC\_ENABLE

表 4-3 SYSTEM\_CFG\_MOSC\_ENABLE の設定

設定値	内容
0 (初期値)	クロックソースに選択していない時にメインクロック(MOSC)を停止(無効)にします。
1	クロックソースに選択していない時にメインクロック(MOSC)を動作(有効)にします。

## 4.1.2 メインクロック動作周波数設定(SYSTEM\_CFG\_MOSC\_FREQUENCY\_HZ)

R\_CORE におけるメインクロック(MOSC)の周波数を設定します。

名称 : SYSTEM\_CFG\_MOSC\_FREQUENCY\_HZ

表 4-4 SYSTEM\_CFG\_MOSC\_FREQUENCY\_HZ の設定

設定値	内容
32000000 (初期値)	メインクロック(MOSC)が動作(有効)時の周波数を設定します。

## 4.1.3 メインクロック発振器駆動能力切り替え(SYSTEM\_CFG\_MOSC\_DRIVE)

R\_CORE におけるメインクロック(MOSC)の使用する発振子に応じた最適な駆動能力を設定します。

このコンフィグレーションの設定値は、MOMCR.MODRV ビットに設定される値となります。

名称 : SYSTEM\_CFG\_MOSC\_DRIVE

表 4-5 SYSTEM\_CFG\_MOSC\_DRIVE の設定

設定値	内容(MOMCR.MODRV ビットの設定内容)
0	設定禁止
1	発振電流小
:	:
7 (初期値)	発振電流大

#### 4.1.4 メインクロック発振器切り替え(SYSTEM\_CFG\_MOSC\_CLOCK\_SOURCE)

R\_COREにおけるメインクロック発振器の発振源を設定します。

このコンフィグレーションの設定値は、MOMCR.MOSEL ビットに設定される値となります。

名称：SYSTEM\_CFG\_MOSC\_CLOCK\_SOURCE

表 4-6 SYSTEM\_CFG\_MOSC\_CLOCK\_SOURCE の設定

設定値	内容(MOMCR.MOSEL ビットの設定内容)
0 (初期値)	発振子
1	外部クロック入力

#### 4.1.5 メインクロック発振器低消費発振機能有効化 (SYSTEM\_CFG\_MOSC\_LOW\_POWER\_ENABLE)

R\_COREにおけるメインクロック発振器の低消費発振機能を設定します。

このコンフィグレーションの設定値は、MOMCR.OSCLPEN ビットに設定される値となります。

名称：SYSTEM\_CFG\_MOSC\_LOW\_POWER\_ENABLE

表 4-7 SYSTEM\_CFG\_MOSC\_LOW\_POWER\_ENABLE の設定

設定値	内容(MOMCR.OSCLPEN ビットの設定内容)
0 (初期値)	無効
1	有効

#### 4.1.6 メインクロック発振器安定待ち時間設定(SYSTEM\_CFG\_MOSC\_WAIT\_TIME)

R\_COREにおけるメインクロック発振器の発振安定待ち時間を設定します。

このコンフィグレーションの設定値は、MOSCWTCR.MSTS ビットに設定される値となります。

名称：SYSTEM\_CFG\_MOSC\_WAIT\_TIME

表 4-8 SYSTEM\_CFG\_MOSC\_WAIT\_TIME の設定

設定値	内容(MOSCWTCR.MSTS ビットの設定内容)
0	設定禁止
1	2 サイクル
2	5 サイクル
3	13 サイクル
4	29 サイクル
5 (初期値)	61 サイクル
6	125 サイクル
7	253 サイクル
8	509 サイクル
9	1021 サイクル

#### 4.1.7 HOCO 停止(SYSTEM\_CFG\_HOCO\_ENABLE)

R\_CORE における高速オンチップオシレータ(HOCO)の動作/停止を設定します。

名称 : SYSTEM\_CFG\_HOCO\_ENABLE

表 4-9 SYSTEM\_CFG\_HOCO\_ENABLE の設定

設定値	内容
0 (初期値)	クロックソースに選択していない時に HOCO を停止(無効)にします。
1	クロックソースに選択していない時に HOCO を動作(有効)にします。

#### 4.1.8 HOCO 発振周波数設定(SYSTEM\_CFG\_HOCO\_FREQUENCY)

R\_CORE における高速オンチップオシレータ(HOCO)の発振周波数を設定します。

このコンフィグレーションの設定値は、HOCOMCR.HCFRQ ビットに設定される値となります。

名称 : SYSTEM\_CFG\_HOCO\_FREQUENCY

表 4-10 SYSTEM\_CFG\_HOCO\_FREQUENCY の設定

設定値	内容(HOCOMCR.HCFRQ ビットの設定内容)
0	HOCO の発振周波数を 24MHz に設定します。
1 (初期値)	HOCO の発振周波数を 32MHz に設定します。
2	HOCO の発振周波数を 48MHz に設定します。
3	HOCO の発振周波数を 64MHz に設定します。

#### 4.1.9 FLL 機能有効化(SYSTEM\_CFG\_FLL\_ENABLE)

R\_CORE における HOCO の FLL 機能を設定します。

名称 : SYSTEM\_CFG\_FLL\_ENABLE

表 4-11 SYSTEM\_CFG\_FLL\_ENABLE の設定

設定値	内容(HOCOMCR.HCFRQ ビットの設定内容)
0 (初期値)	FLL 機能を無効にします。
1	FLL 機能を有効にします。

## 4.1.10 MOCO 停止(SYSTEM\_CFG\_MOCO\_ENABLE)

R\_CORE における中速オンチップオシレータ(MOCO)の動作/停止を設定します。

名称 : SYSTEM\_CFG\_MOCO\_ENABLE

表 4-12 SYSTEM\_CFG\_MOCO\_ENABLE の設定

設定値	内容
0	クロックソースに選択していない時に MOCO を停止(無効)にします。
1 (初期値)	クロックソースに選択していない時に MOCO の停止を行いません(有効)。

## 4.1.11 LOCO 停止(SYSTEM\_CFG\_LOCO\_ENABLE)

R\_CORE における低速オンチップオシレータ(LOCO)の動作/停止を設定します。

名称 : SYSTEM\_CFG\_LOCO\_ENABLE

表 4-13 SYSTEM\_CFG\_LOCO\_ENABLE の設定

設定値	内容
0	クロックソースに選択していない時に LOCO を停止(無効)にします。
1 (初期値)	クロックソースに選択していない時に LOCO を動作(有効)にします。

## 4.1.12 サブクロック発振器停止(SYSTEM\_CFG\_SOSC\_ENABLE)

R\_CORE におけるサブクロック(SOSC)の動作/停止を設定します。

名称 : SYSTEM\_CFG\_SOSC\_ENABLE

表 4-14 SYSTEM\_CFG\_SOSC\_ENABLE の設定

設定値	内容
0 (初期値)	クロックソースに選択していない時にサブクロック(SOSC)を停止(無効)にします。
1	クロックソースに選択していない時にサブクロック(SOSC)を動作(有効)にします。

## 4.1.13 サブクロック発振器駆動能力切り替え(SYSTEM\_CFG\_SOSC\_DRIVE)

R\_CORE におけるサブクロック(SOSC)の使用する発振子に応じた最適な駆動能力を設定します。

このコンフィグレーションの設定値は、SOMCR.SODRV ビットに設定される値となります。

名称 : SYSTEM\_CFG\_SOSC\_DRIVE

表 4-15 SYSTEM\_CFG\_SOSC\_DRIVE の設定

設定値	内容(SOMCR.SODRV ビットの設定内容)
0 (初期値)	サブクロック(SOSC)を標準 CL で駆動します。
1	サブクロック(SOSC)を低 CL6 で駆動します。
2	サブクロック(SOSC)を低 CL4 で駆動します。
3	サブクロック(SOSC)を低 CL7 で駆動します。

#### 4.1.14 サブクロック発振器ノイズフィルタ停止(SYSTEM\_CFG\_SOSC\_NF\_STOP)

R\_CORE におけるサブクロック(SOSC)発振器用ノイズフィルタの動作/停止を設定します。

このコンフィグレーションの設定値は、SOMCR.SONFSTP ビットに設定される値となります。

名称：SYSTEM\_CFG\_SOSC\_NF\_STOP

表 4-16 SYSTEM\_CFG\_SOSC\_NF\_STOP の設定

設定値	内容(SOMCR.SONFSTP ビットの設定内容)
0 (初期値)	サブクロック(SOSC)用ノイズフィルタを動作します。
1	サブクロック(SOSC)用ノイズフィルタを停止します。

#### 4.1.15 クロックソース選択(SYSTEM\_CFG\_CLOCK\_SOURCE)

システムクロックのクロックソースを設定します。

このコンフィグレーションの設定値は、SCKSCR.CKSEL ビットに設定される値となります。

名称：SYSTEM\_CFG\_CLOCK\_SOURCE

表 4-17 SYSTEM\_CFG\_CLOCK\_SOURCE の設定

設定値	内容(SCKSCR.CKSEL ビットの設定内容)
0	HOCO
1 (初期値)	MOCO
2	LOCO
3	メインクロック発振器
4	サブクロック発振器

#### 4.1.16 システムクロック(ICLK)/周辺モジュールクロック(PCLKA)選択 (SYSTEM\_CFG\_ICK\_PCKA\_DIV)

CPU、DMAC および DTC 用システムクロックと、周辺モジュールクロック A(PCLKA)の周波数を分周設定します。

このコンフィグレーションの設定値は、SCKDIVCR.ICK ビットに設定される値となります。

名称 : SYSTEM\_CFG\_ICK\_PCKA\_DIV

表 4-18 SYSTEM\_CFG\_ICK\_PCKA\_DIV の設定

設定値	内容(SCKDIVCR.ICK ビットの設定内容)
0 (初期値)	クロックソースを 1 分周
1	クロックソースを 2 分周
2	クロックソースを 4 分周
3	クロックソースを 8 分周
4	クロックソースを 16 分周
5	クロックソースを 32 分周
6	クロックソースを 64 分周

#### 4.1.17 周辺モジュールクロック B(PCLKB)選択(SYSTEM\_CFG\_PCKB\_DIV)

周辺モジュールクロック B(PCLKB)の周波数を分周設定します。

このコンフィグレーションの設定値は、SCKDIVCR.PCKB ビットに設定される値となります。

名称 : SYSTEM\_CFG\_PCKB\_DIV

表 4-19 SYSTEM\_CFG\_PCKB\_DIV の設定

設定値	内容(SCKDIVCR.PCKB ビットの設定内容)
0 (初期値)	クロックソースを 1 分周
1	クロックソースを 2 分周
2	クロックソースを 4 分周
3	クロックソースを 8 分周
4	クロックソースを 16 分周
5	クロックソースを 32 分周
6	クロックソースを 64 分周

## 4.1.18 電源状態選択(SYSTEM\_CFG\_POWER\_CONTROL\_MODE)

電力制御モードのブーストモード(BOOST)/ノーマルモード(NORMAL)/低リーク電流モード(VBB)を設定します。ノーマルモード(NORMAL)では、High-Speed モードを選択します。

名称：SYSTEM\_CFG\_POWER\_CONTROL\_MODE

表 4-20 SYSTEM\_CFG\_POWER\_CONTROL\_MODE の設定

設定値	内容
0	ブーストモード(BOOST)
1 (初期値)	ノーマルモード(Normal) High-speed モード
4	低リーク電流モード(VBB)

## 4.1.19 EHC 起動モード設定(SYSTEM\_CFG\_EHC\_MODE)

EHC 起動処理の有効/無効を選択します。

注. R\_CORE の EHC 起動処理はオプション機能となります。

名称：SYSTEM\_CFG\_EHC\_MODE

表 4-21 SYSTEM\_CFG\_EHC\_MODE の設定

設定値	内容
0	EHC 起動処理を無効にする
1	EHC 起動処理を有効にする

## 4.1.20 電圧検出 1 レベル選択(SYSTEM\_CFG\_EHC\_LVD1LVL)

EHC 起動時に電圧監視 1 が検出するレベルを選択します。Vdet1\_X の値については、UMH の電気的特性の章を参照してください。

注. R\_CORE の EHC 起動処理はオプション機能となります。

名称：SYSTEM\_CFG\_EHC\_LVD1LVL

表 4-22 SYSTEM\_CFG\_EHC\_LVD1LVL の設定

設定値	内容
0	Vdet1_0
1	Vdet1_1
2	Vdet1_3
3	Vdet1_5
4	Vdet1_7
5	Vdet1_9
6	Vdet1_B
7 (初期値)	Vdet1_D



#### 4.1.21 電圧検出 BAT レベル選択(SYSTEM\_CFG\_EHC\_LVDBATLVL)

EHC 起動時に電圧監視 BAT が検出するレベルを選択します。VdetBAT\_X の値については、UMH の電気的特性の章を参照してください。

注. R\_CORE の EHC 起動処理はオプション機能となります。

名称 : SYSTEM\_CFG\_EHC\_LVDBATLVL

表 4-23 SYSTEM\_CFG\_EHC\_LVDBATLVL の設定

設定値	内容
0 (初期値)	VdetBAT_5
1	VdetBAT_7
2	VdetBAT_8
3	VdetBAT_B
4	VdetBAT_D

#### 4.1.22 オプション設定メモリ 0(SYSTEM\_CFG\_OFS0)

オプション機能選択レジスタ(OFS0)の値を設定します。オプション機能選択レジスタ(OFS0)の詳細については、UMH を参照してください。

名称 : SYSTEM\_CFG\_OFS0

表 4-24 SYSTEM\_CFG\_OFS0 の設定

設定値	内容
0xFFFFFFFF (初期値)	オプション機能選択レジスタ 0(OFS0)に設定する値

#### 4.1.23 オプション設定メモリ 1(SYSTEM\_CFG\_OFS1)

オプション機能選択レジスタ 1(OFS1)の値を設定します。オプション機能選択レジスタ 1(OFS1)の詳細については、UMH を参照してください。

名称 : SYSTEM\_CFG\_OFS1

表 4-25 SYSTEM\_CFG\_OFS1 の設定

設定値	内容
0xFFFFFFFF (初期値)	オプション機能選択レジスタ 1(OFS1)に設定する値

## 4.1.24 オプション設定メモリ 2(SYSTEM\_CFG\_SECMPU\_xxx)

セキュリティ MPU 機能関連レジスタの値を設定します。セキュリティ MPU 機能関連レジスタの詳細については、UMH を参照してください。

名称：SYSTEM\_CFG\_SECMPU\_xxx

表 4-26 SYSTEM\_CFG\_SECMPU\_xxx の設定

コンフィグレーション	設定値	内容
SYSTEM_CFG_SECMPU_PC0_START	0xFFFFFFFF (初期値)	セキュリティ MPU プログラムカウンタ開始アドレスレジスタ 0(SECMPUPCS0)に設定する値
SYSTEM_CFG_SECMPU_PC0_END	0xFFFFFFFF (初期値)	セキュリティ MPU プログラムカウンタ終了アドレスレジスタ 0(SECMPUPCE0)に設定する値
SYSTEM_CFG_SECMPU_PC1_START	0xFFFFFFFF (初期値)	セキュリティ MPU プログラムカウンタ開始アドレスレジスタ 1(SECMPUPCS1)に設定する値
SYSTEM_CFG_SECMPU_PC1_END	0xFFFFFFFF (初期値)	セキュリティ MPU プログラムカウンタ終了アドレスレジスタ 1(SECMPUPCE1)に設定する値
SYSTEM_CFG_SECMPU_REGION0_START	0xFFFFFFFF (初期値)	セキュリティ MPU 領域 0 開始アドレスレジスタ(SECMPUS0)に設定する値
SYSTEM_CFG_SECMPU_REGION0_END	0xFFFFFFFF (初期値)	セキュリティ MPU 領域 0 終了アドレスレジスタ(SECMPUE0)に設定する値
SYSTEM_CFG_SECMPU_REGION1_START	0xFFFFFFFF (初期値)	セキュリティ MPU 領域 1 開始アドレスレジスタ(SECMPUS1)に設定する値
SYSTEM_CFG_SECMPU_REGION1_END	0xFFFFFFFF (初期値)	セキュリティ MPU 領域 1 終了アドレスレジスタ(SECMPUE1)に設定する値
SYSTEM_CFG_SECMPU_REGION2_START	0xFFFFFFFF (初期値)	セキュリティ MPU 領域 2 開始アドレスレジスタ(SECMPUS2)に設定する値
SYSTEM_CFG_SECMPU_REGION2_END	0xFFFFFFFF (初期値)	セキュリティ MPU 領域 2 終了アドレスレジスタ(SECMPUE2)に設定する値
SYSTEM_CFG_SECMPU_CONTROL_SETTING	0xFFFF (初期値)	セキュリティ MPU アクセスコントロールレジスタ(SECMPUAC)に設定する値

## 4.1.25 オプション設定メモリ 3(SYSTEM\_CFG\_ID\_CODE\_PROTECTION\_n)(n=1-4)

OCD／シリアルプログラマ ID 設定レジスタレジスタ(OSIS)の値を設定します。OCD／シリアルプログラマ ID 設定レジスタレジスタ(OSIS)の詳細については、UMH を参照してください。

名称：SYSTEM\_CFG\_ID\_CODE\_PROTECTION\_n

表 4-27 SYSTEM\_CFG\_ID\_CODE\_PROTECTION\_n の設定

コンフィグレーション	設定値	内容
SYSTEM_CFG_ID_CODE_PROTECTION_1	0xFFFFFFFF (初期値)	OSIS レジスタ(ビット 0～31)に設定する値
SYSTEM_CFG_ID_CODE_PROTECTION_2	0xFFFFFFFF (初期値)	OSIS レジスタ(ビット 32～63)に設定する値
SYSTEM_CFG_ID_CODE_PROTECTION_3	0xFFFFFFFF (初期値)	OSIS レジスタ(ビット 64～95)に設定する値
SYSTEM_CFG_ID_CODE_PROTECTION_4	0xFFFFFFFF (初期値)	OSIS レジスタ(ビット 96～127)に設定する値

## 4.1.26 オプション設定メモリ 4(SYSTEM\_CFG\_AWS)

アクセスウィンドウ設定レジスタ(AWS)の値を設定します。アクセスウィンドウ設定レジスタ(AWS)の詳細については、UMH を参照してください。

表 4-28 SYSTEM\_CFG\_AWS の設定

設定値	内容
0xFFFFFFFF (初期値)	アクセスウィンドウ設定レジスタ(AWS)に設定する値

## 4.1.27 プリプロセッサエラーチェック

『4.1 コンフィグレーション』で設定した各定義が正しく設定しているかコンパイル時に確認します。エラーとなる主な条件を表 4-29 および表 4-30 に示します。また、確認内容、および確認結果一覧については以下『4.4.1 プリプロセッサエラー一覧』を参照してください。

表 4-29 エラーとなる主な条件(クロックソース選択時)

クロック及び分周設定におけるエラーチェック条件 (クロックソース選択)		Boost mode (BOOST) $\leq 64\text{MHz}$	Normal mode (High-speed) $\leq 32\text{MHz}$	Low-Leakage - Current mode(VBB) $= 32.768\text{kHz}$
0	HOCO (24/32/48/64MHz)	該当なし	・ HOCO 設定 $\geq 48\text{MHz}$	・ HOCO 選択時
1	MOCO(2MHz)	該当なし	該当なし	・ MOCO 選択時
2	LOCO(32.768kHz)	該当なし	該当なし	該当なし
3	MOSC(8~32MHz)	該当なし	該当なし	・ MOSC 選択時
4	SOSC(32.768kHz)	該当なし	該当なし	該当なし
分周	ICLK 分周設定	該当なし	該当なし	ICLK=1 分周
	PCLKB 分周設定	ICLK $\leq$ PCLKB	ICLK $\leq$ PCLKB	ICLK $\neq$ PCLKB

表 4-30 エラーとなる主な条件(クロックソース非選択時)

クロック及び分周設定におけるエラーチェック条件 (クロックソース非選択でクロック動作選択)		Boost mode (BOOST) $\leq 64\text{MHz}$	Normal mode (High-speed) $\leq 32\text{MHz}$	Low-Leakage - Current mode (VBB) $= 32.768\text{kHz}$
0	HOCO (24/32/48/64MHz)	該当なし	・ HOCO 設定 $\geq 48\text{MHz}$	・ HOCO 選択時
1	MOCO(2MHz)	該当なし	該当なし	・ MOCO 選択時
2	LOCO(32.768kHz)	該当なし	該当なし	該当なし
3	MOSC(8~32MHz)	該当なし	該当なし	・ MOSC 選択時
4	SOSC(32.768kHz)	該当なし	該当なし	該当なし

## 4.2 定数／マクロ

### 4.2.1 オプション設定メモリ (OptionSettingMemory)

オプション設定メモリ (OptionSettingMemory)は、『4.1.22 オプション設定メモリ 0(SYSTEM\_CFG\_OFS0)』～『4.1.24 オプション設定メモリ 2(SYSTEM\_CFG\_SECMPU\_xxx)』で定義した設定を定数として定義します。

名称：static const uint32\_t OptionSettingMemory[] \_\_attribute\_\_((section(".OptionSetting")))

定義ファイル：system\_RE01\_256KB.c

### 4.2.2 シリアルプログラマ ID 設定 (OptionSettingID)

シリアルプログラマ ID 設定 (OptionSettingID)は、『4.1.25 オプション設定メモリ 3(SYSTEM\_CFG\_ID\_CODE\_PROTECTION\_n)(n=1-4)』で定義した設定を定数として定義します。

名称：static const uint32\_t OptionSettingID[] \_\_attribute\_\_((section(".OptionSettingID")))

定義ファイル：system\_RE01\_256KB.c

### 4.2.3 アクセスウィンドウ設定 (OptionSettingAWS)

アクセスウィンドウ設定 (OptionSettingAWS)は、『4.1.26 オプション設定メモリ 4(SYSTEM\_CFG\_AWS)』で定義した設定を定数として定義します。

名称：static const uint32\_t OptionSettingAWS[] \_\_attribute\_\_((section(".OptionSettingAWS")))

定義ファイル：system\_RE01\_256KB.c

#### 4.2.4 システムクロックソース定義マクロ(SYSTEM\_CLOCK\_SEL)

『4.1 コンフィグレーション』にて設定したシステムクロックソースの発振周波数を計算し本定義に反映します。

名称 : SYSTEM\_CLOCK\_SEL

定義ファイル : system\_RE01\_256KB.c

表 4-31 SYSTEM\_CLOCK\_SEL の設定値一覧

クロックソース	コンフィグで選択した設定	SYSTEM_CLOCK_SEL 設定値
HOCO(24MHz)	SYSTEM_CFG_CLOCK_SOURCE=0 SYSTEM_CFG_HOCO_FREQUENCY=0	24000000U
HOCO(32MHz)	SYSTEM_CFG_CLOCK_SOURCE=0 SYSTEM_CFG_HOCO_FREQUENCY=1	32000000U
HOCO(48MHz)	SYSTEM_CFG_CLOCK_SOURCE=0 SYSTEM_CFG_HOCO_FREQUENCY=2	48000000U
HOCO(64MHz)	SYSTEM_CFG_CLOCK_SOURCE=0 SYSTEM_CFG_HOCO_FREQUENCY=3	64000000U
MOCO	SYSTEM_CFG_CLOCK_SOURCE=1	2000000U
LOCO	SYSTEM_CFG_CLOCK_SOURCE=2	32768U
MOSC	SYSTEM_CFG_CLOCK_SOURCE=3	SYSTEM_CFG_MOSC_FREQUENCY_HZ
SOSC	SYSTEM_CFG_CLOCK_SOURCE=4	32768U

#### 4.2.5 システムクロック定義マクロ(SYSTEM\_CLOCK)

『4.2.4 システムクロックソース定義マクロ(SYSTEM\_CLOCK\_SEL)』で設定したシステムクロックソースの発振周波数と『4.1.16 システムクロック(ICLK)/周辺モジュールクロック(PCLKA)選択 (SYSTEM\_CFG\_ICK\_PCKA\_DIV)』の分周設定で計算した結果をシステムクロックとして本定義に反映します。

名称 : SYSTEM\_CLOCK

定義ファイル : system\_RE01\_256KB.c

表 4-32 SYSTEM\_CLOCK の設定値一覧

コンフィグで選択した設定	SYSTEM_CLOCK 設定値
SYSTEM_CFG_ICK_PCKA_DIV=0	SYSTEM_CLOCK_SEL / 1
SYSTEM_CFG_ICK_PCKA_DIV=1	SYSTEM_CLOCK_SEL / 2
SYSTEM_CFG_ICK_PCKA_DIV=2	SYSTEM_CLOCK_SEL / 4
SYSTEM_CFG_ICK_PCKA_DIV=3	SYSTEM_CLOCK_SEL / 8
SYSTEM_CFG_ICK_PCKA_DIV=4	SYSTEM_CLOCK_SEL / 16
SYSTEM_CFG_ICK_PCKA_DIV=5	SYSTEM_CLOCK_SEL / 32
SYSTEM_CFG_ICK_PCKA_DIV=6	SYSTEM_CLOCK_SEL / 64

## 4.2.6 システムクロックウェイトサイクル定義

定義ファイル：system\_RE01\_256KB.c

表 4-33 システムクロックウェイトサイクル

マクロ定義	設定値	備考
ICLK_1CYCLE	(1.0F / SYSTEM_CLOCK)	ICLK の 1 サイクル時間
ICLK_MOCO_1CYCLE	(0.0000005F)	MOCO の 1 サイクル時間 0.5us
MOCO_TMOCOWT_SEC	(0.000016F)	MOCO の発振安定待ち時間 16us
MOCO_RESTART_SEC	(0.0000025F)	MOCO のリスタート時間 2.5us
LOCO_TLOCOWT_SEC	(0.000130F)	LOCO の発振安定待ち時間 130us
LOCO_RESTART_SEC	(0.0001525879F)	LOCO のリスタート時間 0.1525879ms
SOSC_SUBOSCOWT_SEC	(3.0F)	サブクロックの発振安定待ち時間 3sec
SOSC_RESTART_SEC	(0.0001525879F)	サブクロックのリスタート時間 0.1525879ms
LOCO_TLOCOWT_TIME_START	$((LOCO\_TLOCOWT\_SEC / ICLK\_MOCO\_1CYCLE) / 4) + 1$	LOCO 発振安定待ちソフトウェア ループのループ回数
LOCO_RESTART_TIME_START	$((LOCO\_RESTART\_SEC / ICLK\_MOCO\_1CYCLE) / 4) + 1$	LOCO リスタート待ちソフトウェア ループのループ回数
SOSC_TSUBOSCOWT_TIME_START	$((SOSC\_SUBOSCOWT\_SEC / ICLK\_MOCO\_1CYCLE) / 4) + 1$	SOSC 発振安定待ちソフトウェア ループのループ回数
SOSC_RESTART_TIME_START	$((SOSC\_RESTART\_SEC / ICLK\_MOCO\_1CYCLE) / 4) + 1$	SOSC リスタート待ちソフトウェア ループのループ回数
MOCO_TMOCOWT_TIME_STOP	$((MOCO\_TMOCOWT\_SEC / ICLK\_1CYCLE) / 4) + 1$	MOCO 発振安定待ちソフトウェア ループのループ回数
MOCO_RESTART_TIME_STOP	$((MOCO\_RESTART\_SEC / ICLK\_1CYCLE) / 4) + 1$	MOCO リスタート待ちソフトウェア ループのループ回数
LOCO_TLOCOWT_TIME_STOP	$((LOCO\_TLOCOWT\_SEC / ICLK\_1CYCLE) / 4) + 1$	LOCO 発振安定待ちソフトウェア ループのループ回数
LOCO_RESTART_TIME_STOP	$((LOCO\_RESTART\_SEC / ICLK\_1CYCLE) / 4) + 1$	LOCO リスタート待ちソフトウェア ループのループ回数
SOSC_TSUBOSCOWT_TIME_STOP	$((SOSC\_SUBOSCOWT\_SEC / ICLK\_1CYCLE) / 4) + 1$	SOSC 発振安定待ちソフトウェア ループのループ回数
SOSC_RESTART_TIME_STOP	$((SOSC\_RESTART\_SEC / ICLK\_1CYCLE) / 4) + 1$	SOSC リスタート待ちソフトウェア ループのループ回数
LOCO_TLOCOWT_TIME_START_ALONE	$((LOCO\_TLOCOWT\_SEC / ICLK\_1CYCLE) / 4) + 1$	LOCO 発振安定待ちソフトウェア ループのループ回数
LOCO_RESTART_TIME_START_ALONE	$((LOCO\_RESTART\_SEC / ICLK\_1CYCLE) / 4) + 1$	LOCO リスタート待ちソフトウェア ループのループ回数
SOSC_TSUBOSCOWT_TIME_START_ALONE	$((SOSC\_SUBOSCOWT\_SEC / ICLK\_1CYCLE) / 4) + 1$	SOSC 発振安定待ちソフトウェア ループのループ回数
SOSC_RESTART_TIME_START_ALONE	$((SOSC\_RESTART\_SEC / ICLK\_1CYCLE) / 4) + 1$	SOSC リスタート待ちソフトウェア ループのループ回数

4.3 関数仕様

4.3.1 system\_clock\_init 関数

表 4-34 system\_clock\_init 関数仕様

書式	void system_clock_init(void)
仕様説明	クロックソースを r_core_cfg.h の設定に従って動作周波数および電力制御モードの初期化を行います。
引数	なし
戻り値	なし
備考	—

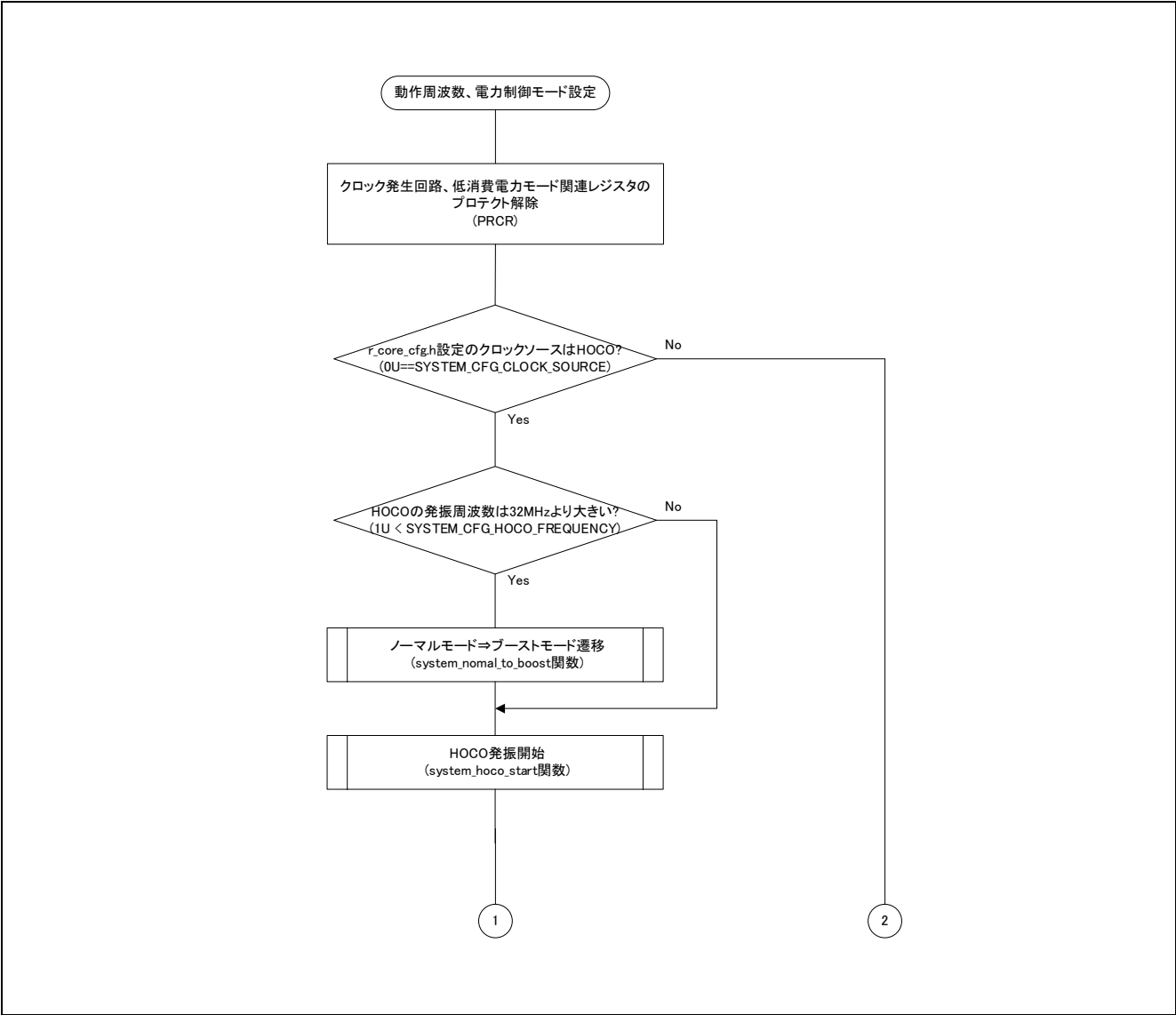


図 4-2 system\_clock\_init 関数フロー(1/5)



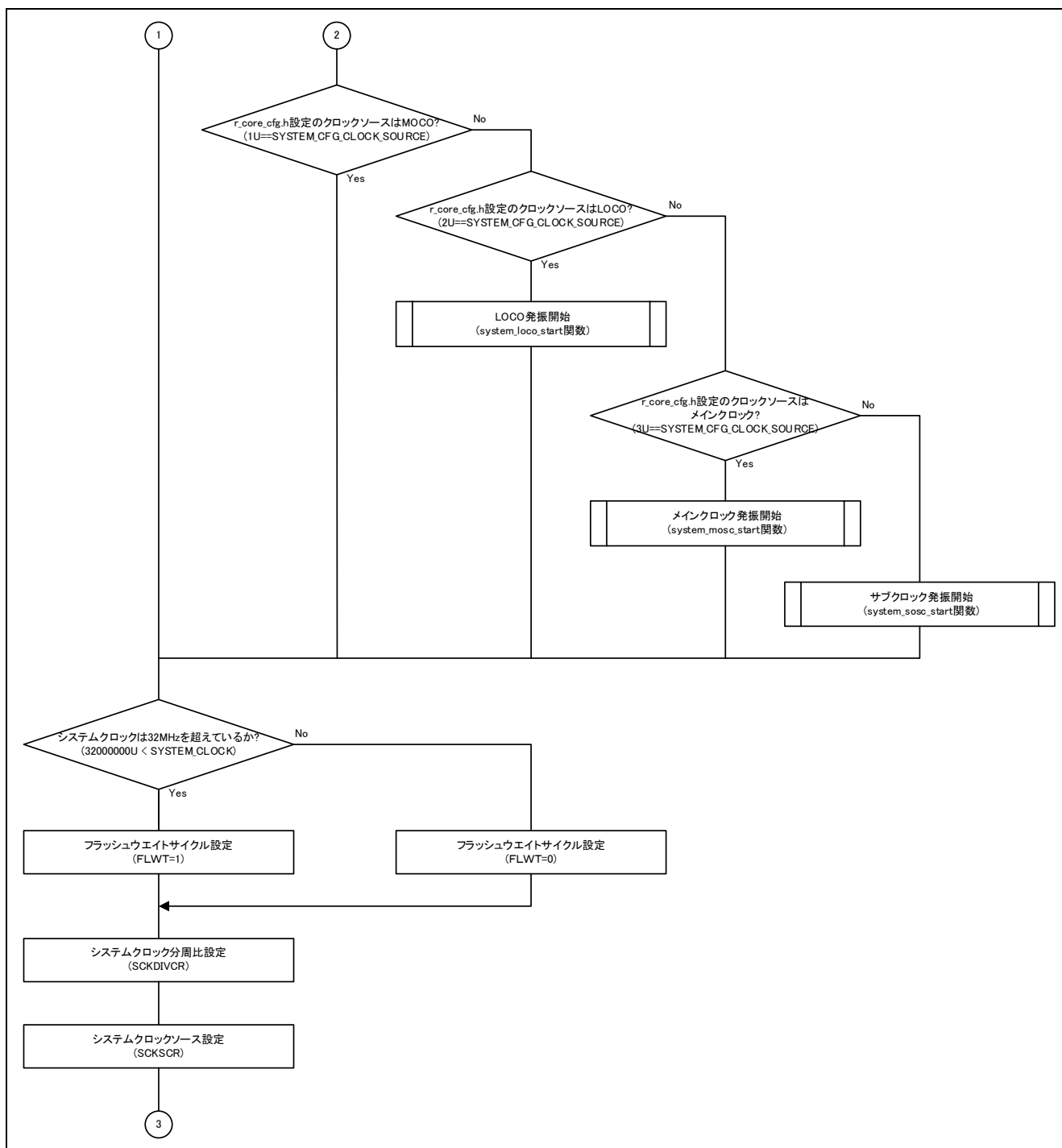


図 4-3 system\_clock\_init 関数フロー(2/5)

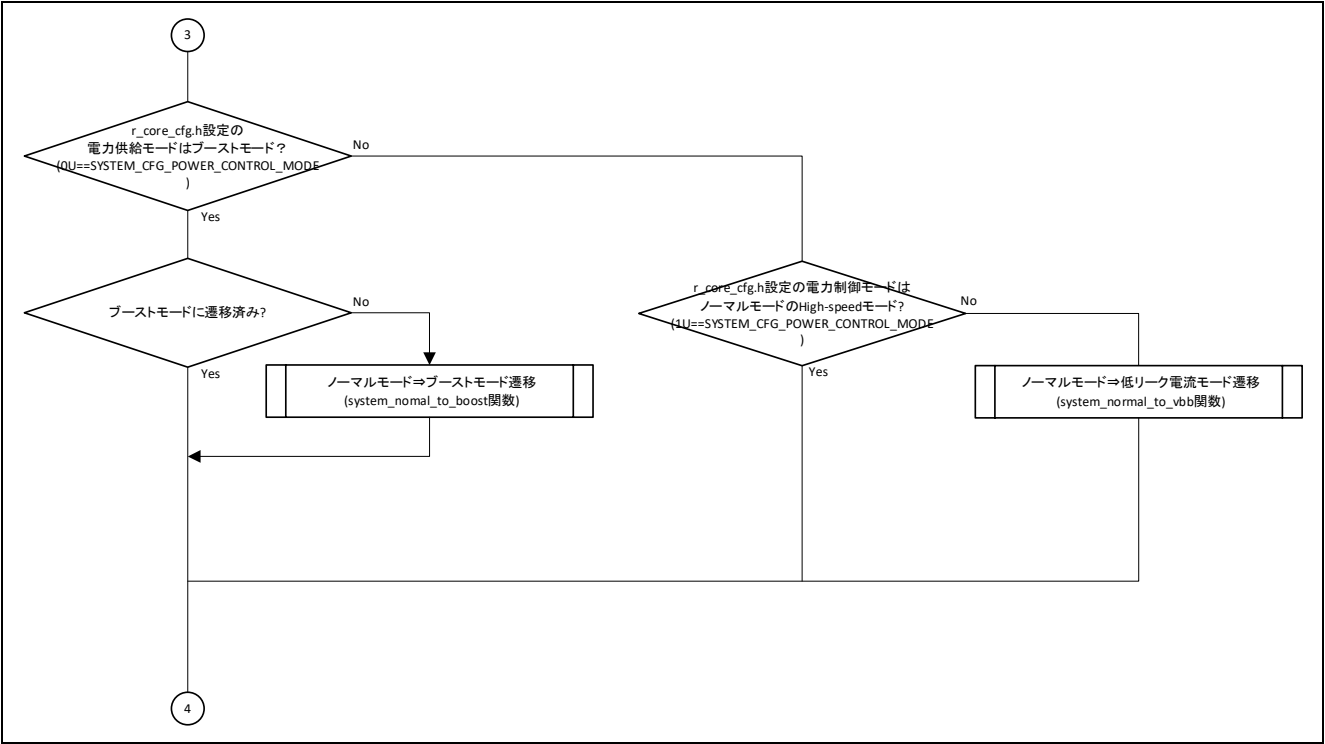


図 4-4 system\_clock\_init 関数フロー(3/5)

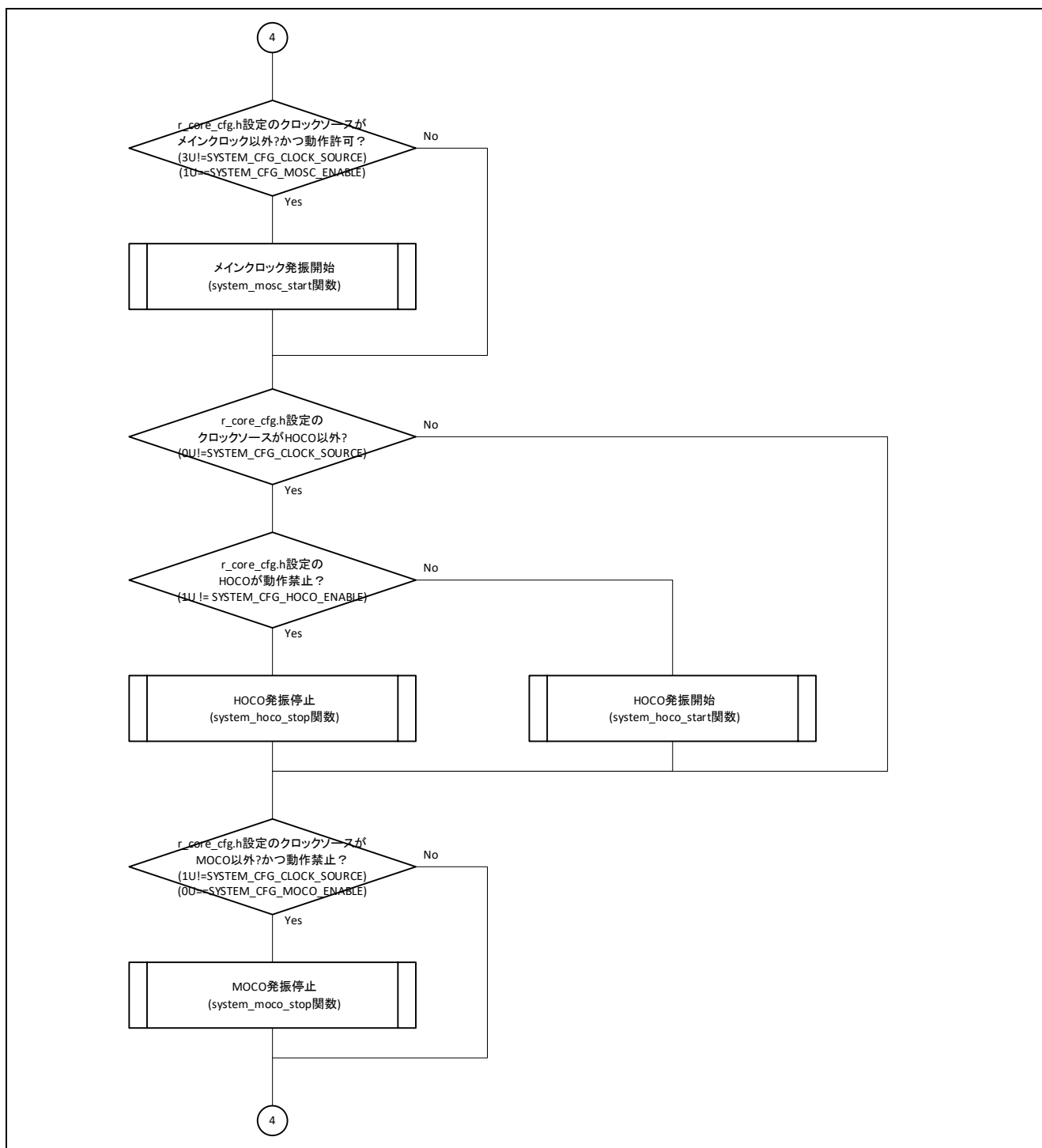


図 4-5 system\_clock\_init 関数フロー(4/5)

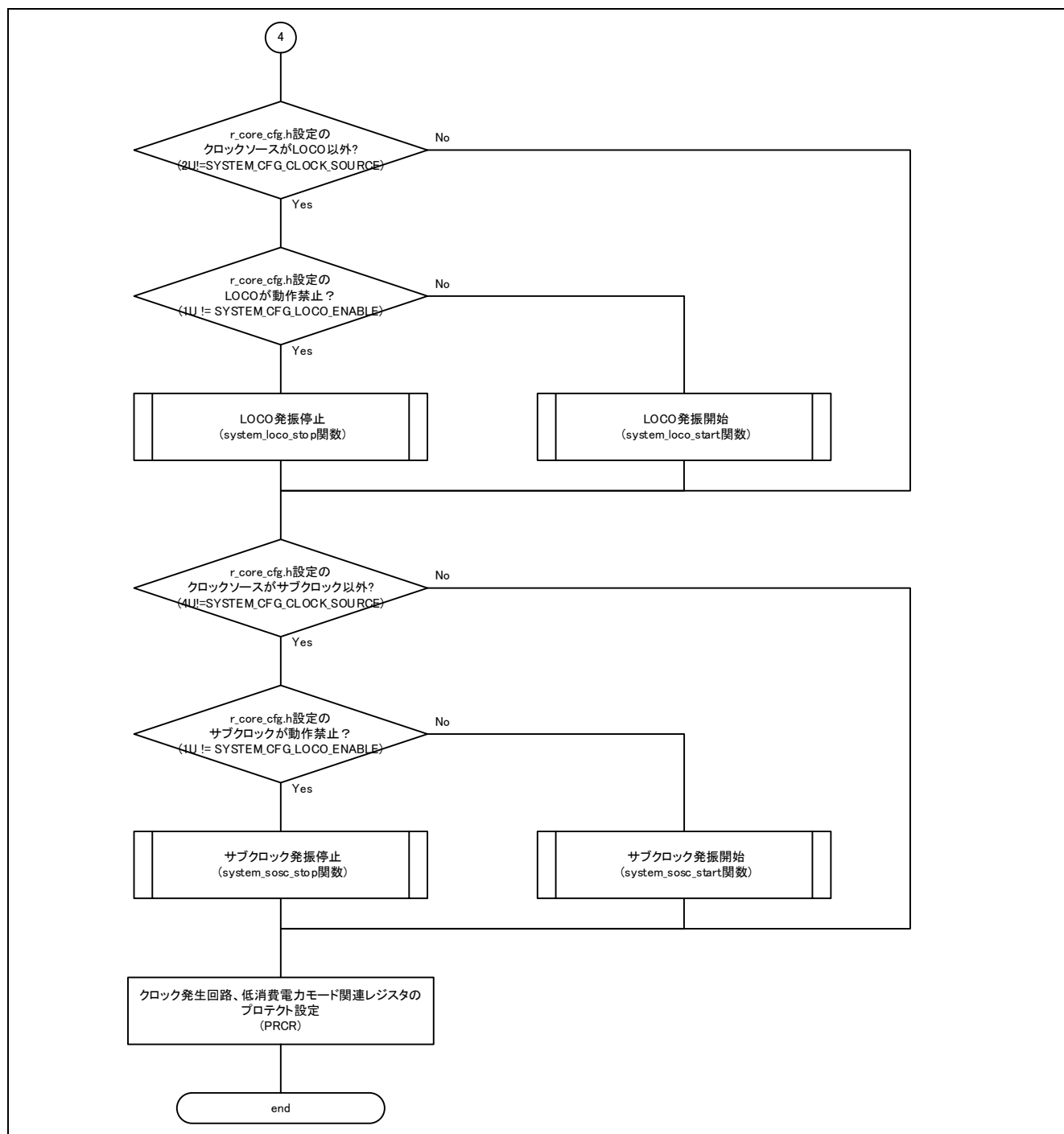


図 4-6 system\_clock\_init 関数フロー(5/5)

4.3.2 system\_mosc\_start 関数

表 4-35 system\_mosc\_start 関数仕様

書式	void system_mosc_start(void)
仕様説明	メインクロックの動作を開始します。
引数	なし
戻り値	なし
備考	—

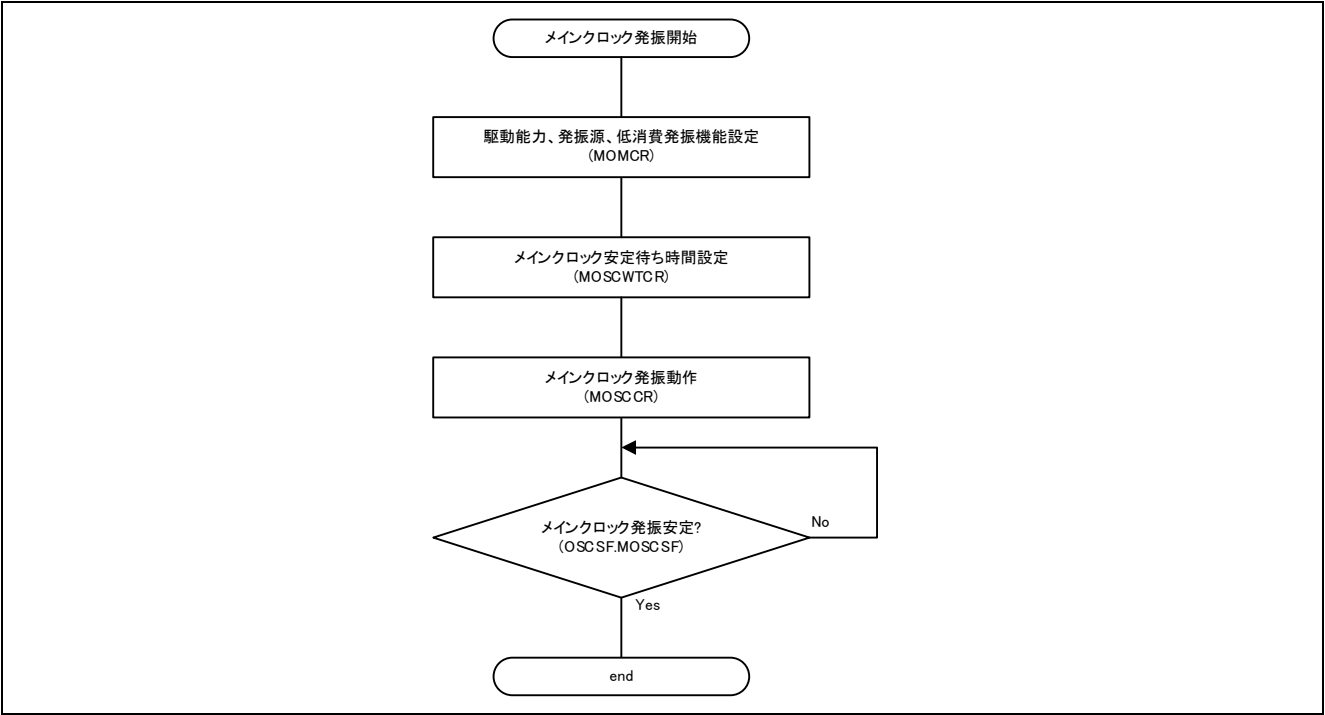


図 4-7 system\_mosc\_start 関数フロー

4.3.3 system\_mosc\_stop 関数

表 4-36 system\_mosc\_stop 関数仕様

書式	void system_mosc_stop(void)
仕様説明	メインクロックの動作を停止します。
引数	なし
戻り値	なし
備考	—

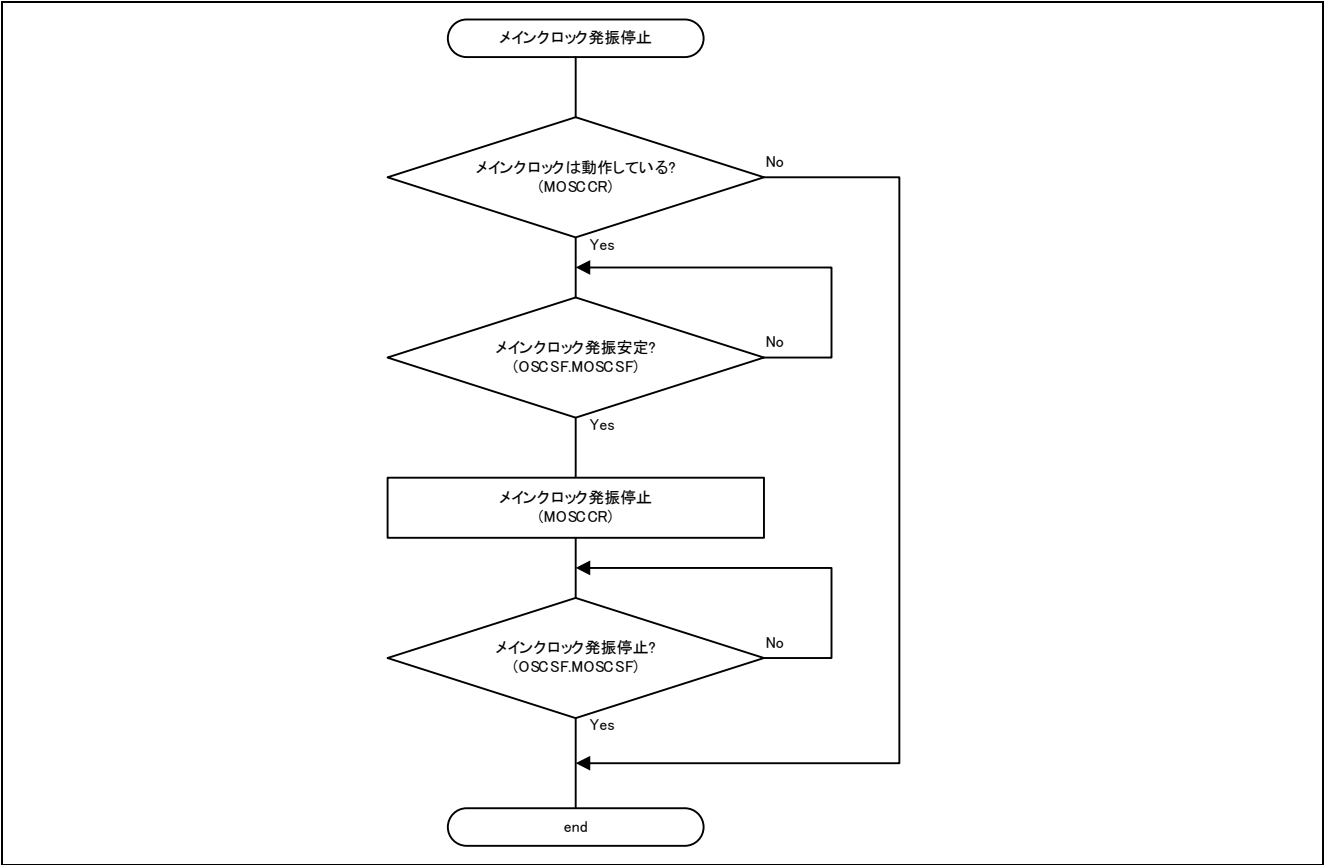


図 4-8 system\_mosc\_stop 関数フロー

#### 4.3.4 system\_hoco\_start 関数

表 4-37 system\_hoco\_start 関数仕様

書式	void system_hoco_start(void)
仕様説明	高速オンチップオシレータの動作を開始します。
引数	なし
戻り値	なし
備考	—

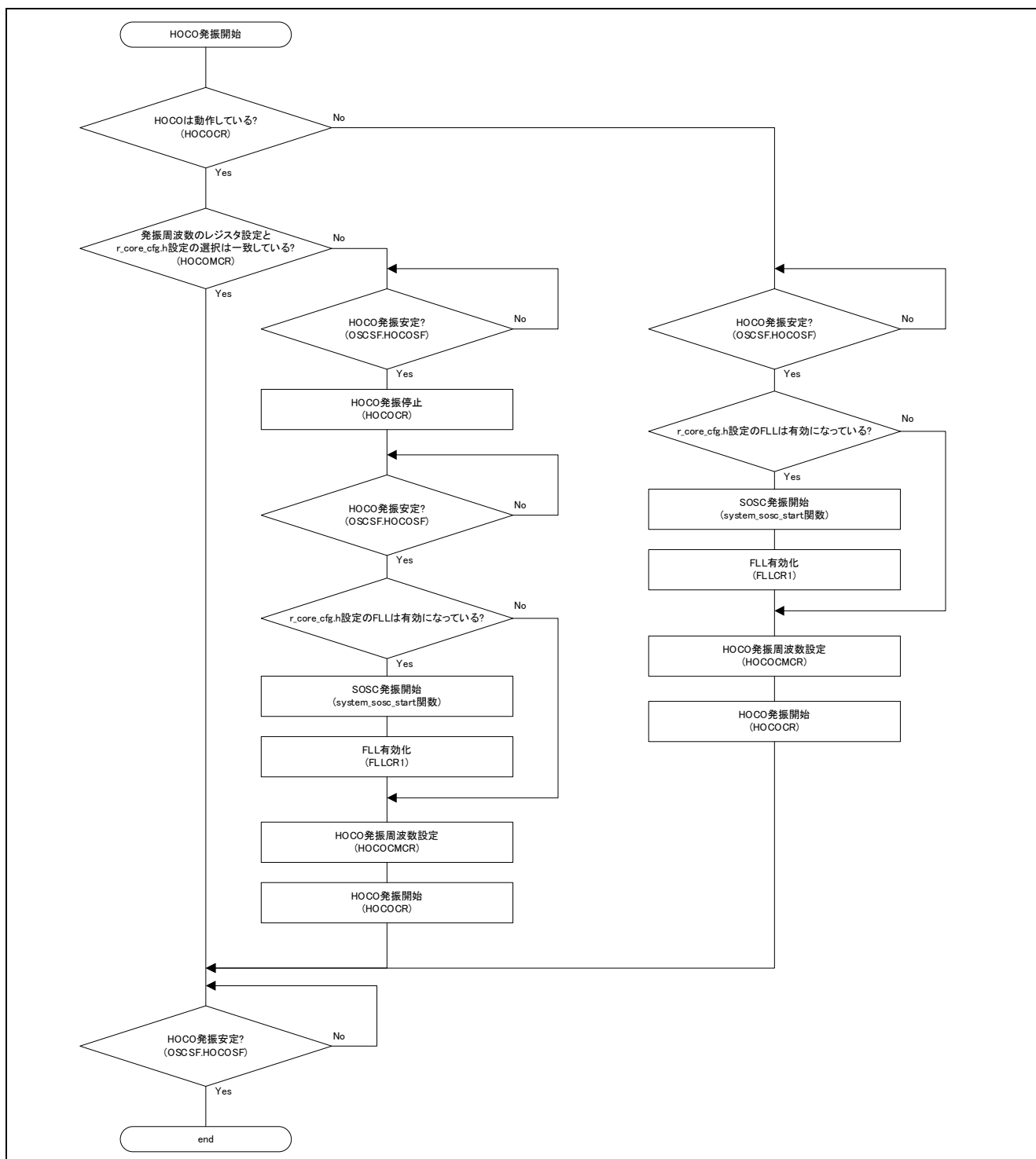


図 4-9 system\_hoco\_start 関数フロー

4.3.5 system\_hoco\_stop 関数

表 4-38 system\_hoco\_stop 関数仕様

書式	void system_hoco_stop(void)
仕様説明	高速オンチップオシレータの動作を停止します。
引数	なし
戻り値	なし
備考	—

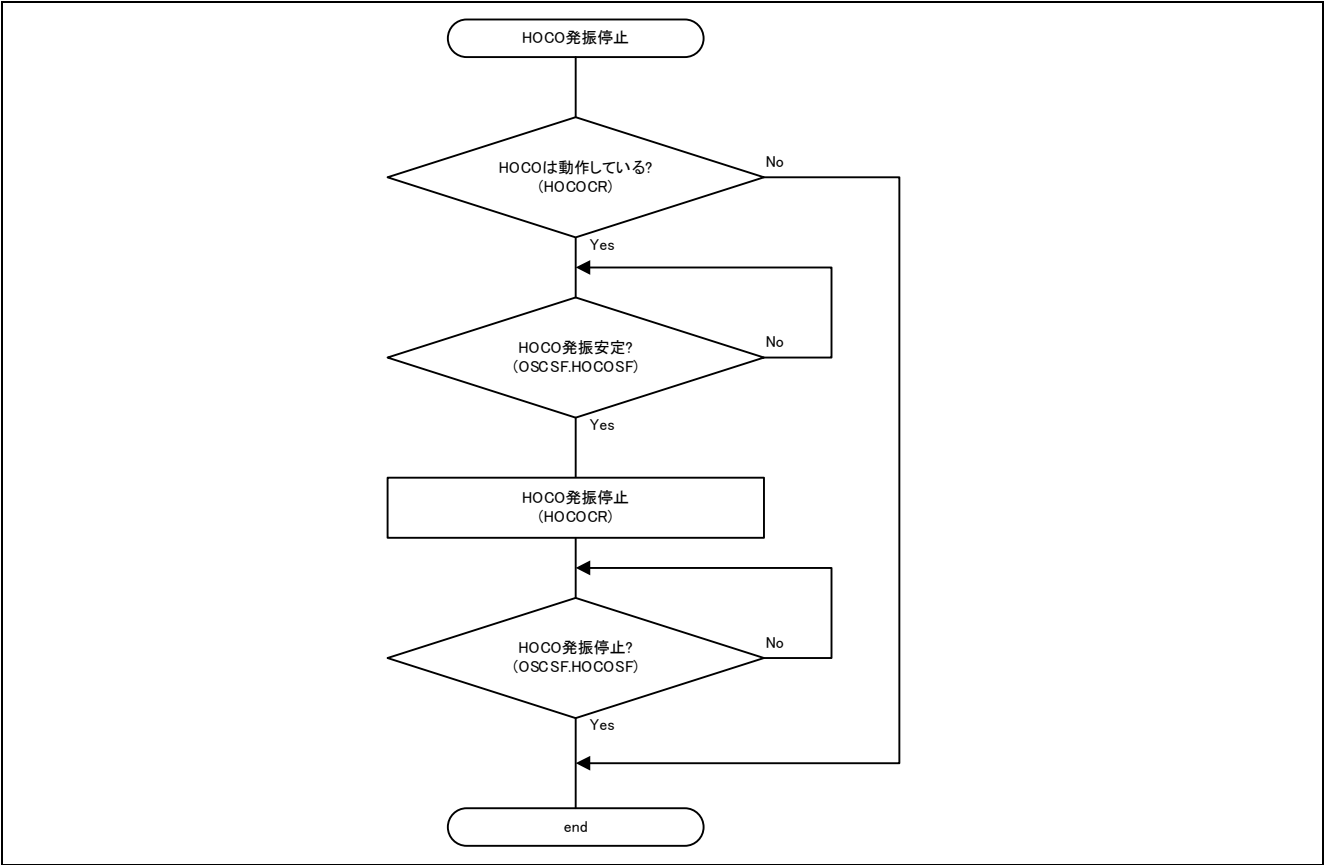


図 4-10 system\_hoco\_stop 関数フロー



4.3.6 system\_moco\_stop 関数

表 4-39 system\_moco\_stop 関数仕様

書式	void system_moco_stop(void)
仕様説明	中速オンチップオシレータの動作を停止します。
引数	なし
戻り値	なし
備考	—

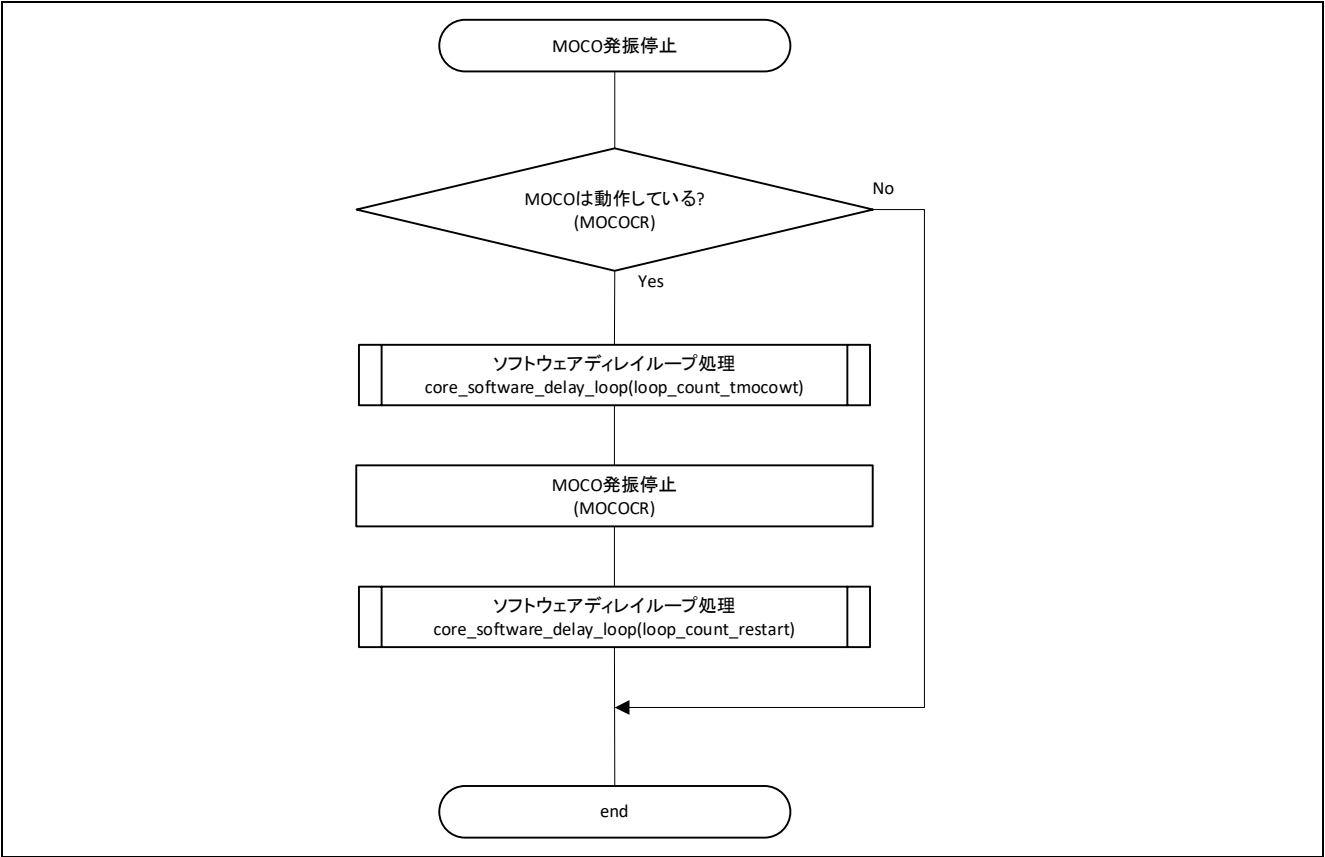


図 4-11 system\_moco\_stop 関数フロー

4.3.7 system\_loco\_start 関数

表 4-40 system\_loco\_start 関数仕様

書式	void system_loco_start(void)
仕様説明	低速オンチップオシレータの動作を開始します。
引数	なし
戻り値	なし
備考	—

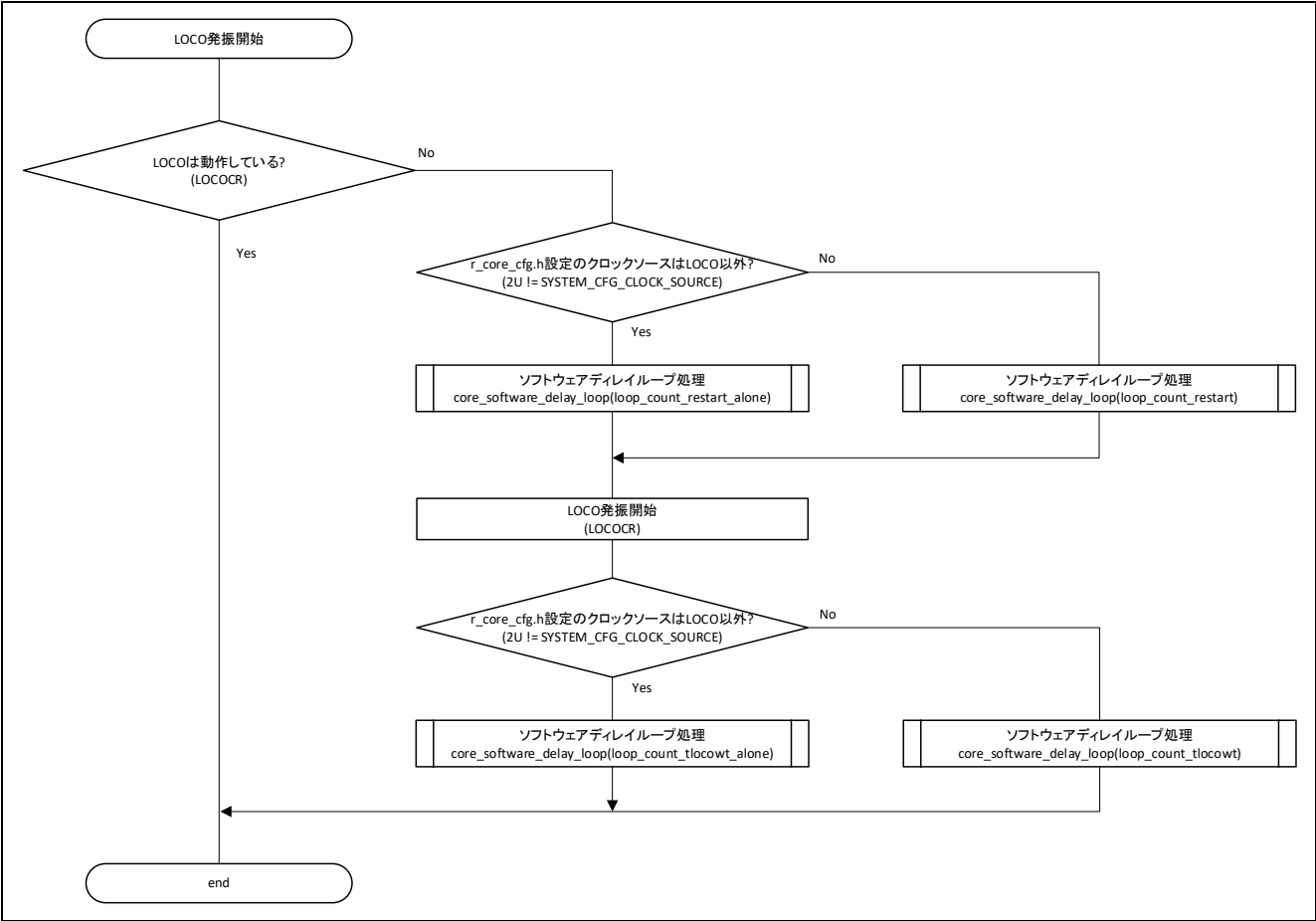


図 4-12 system\_loco\_start 関数フロー

4.3.8 system\_loco\_stop 関数

表 4-41 system\_loco\_stop 関数仕様

書式	void system_loco_stop(void)
仕様説明	低速オンチップオシレータの動作を停止します。
引数	なし
戻り値	なし
備考	—

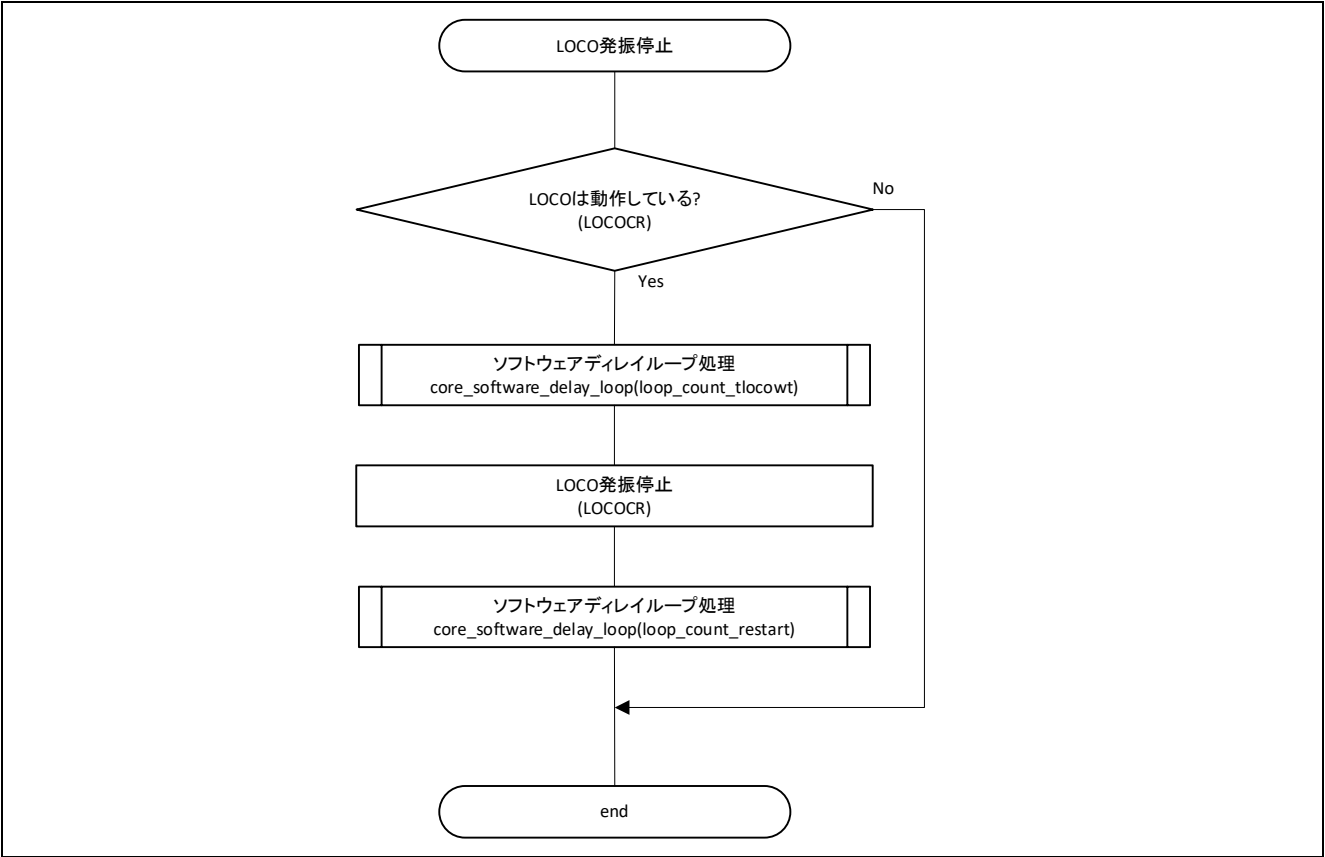


図 4-13 system\_loco\_stop 関数フロー

4.3.9 system\_sosc\_start 関数

表 4-42 system\_sosc\_start 関数仕様

書式	void system_sosc_start(void)
仕様説明	サブクロックの動作を開始します。
引数	なし
戻り値	なし
備考	—

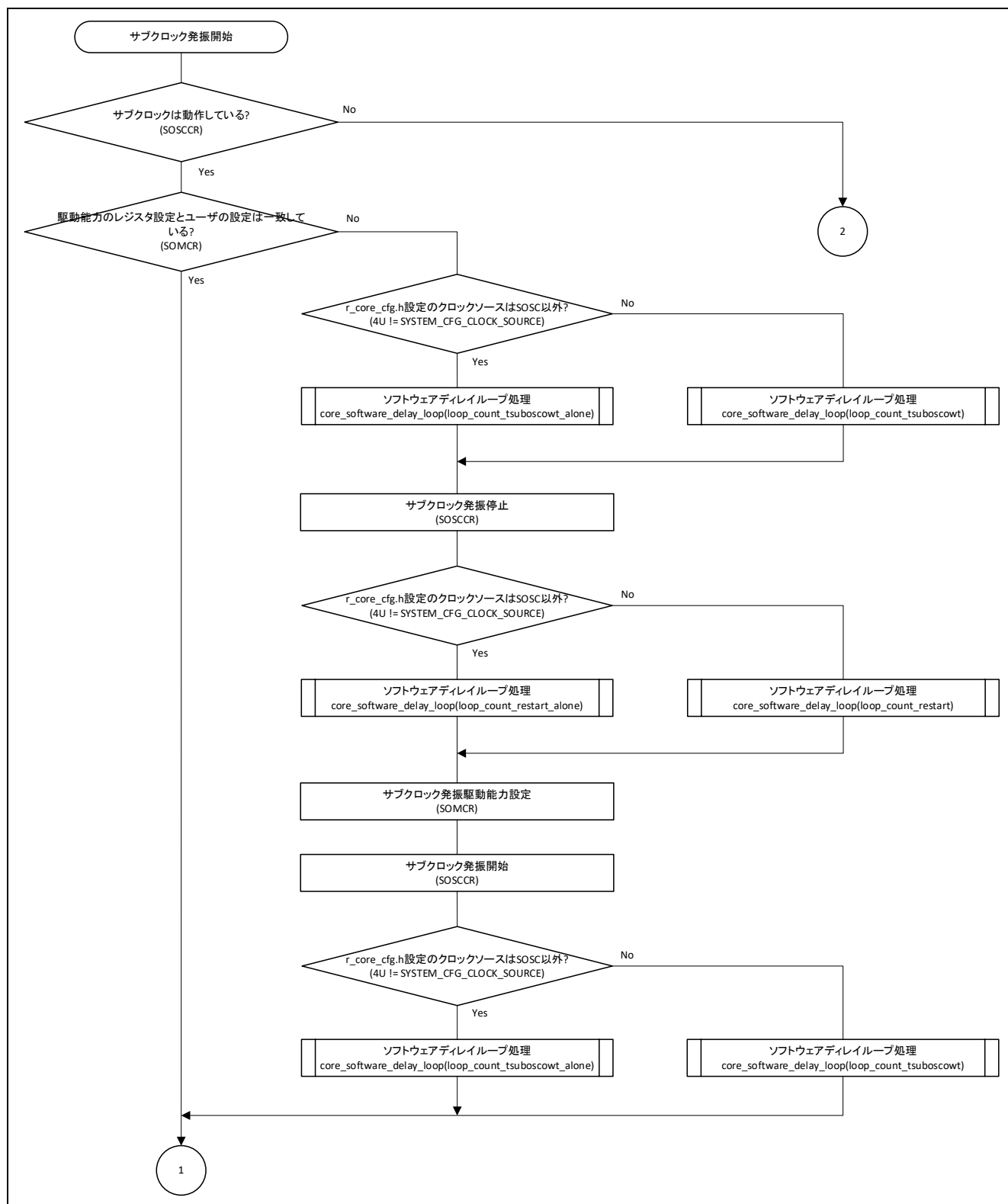


図 4-14 system\_sosc\_start 関数フロー (1/2)

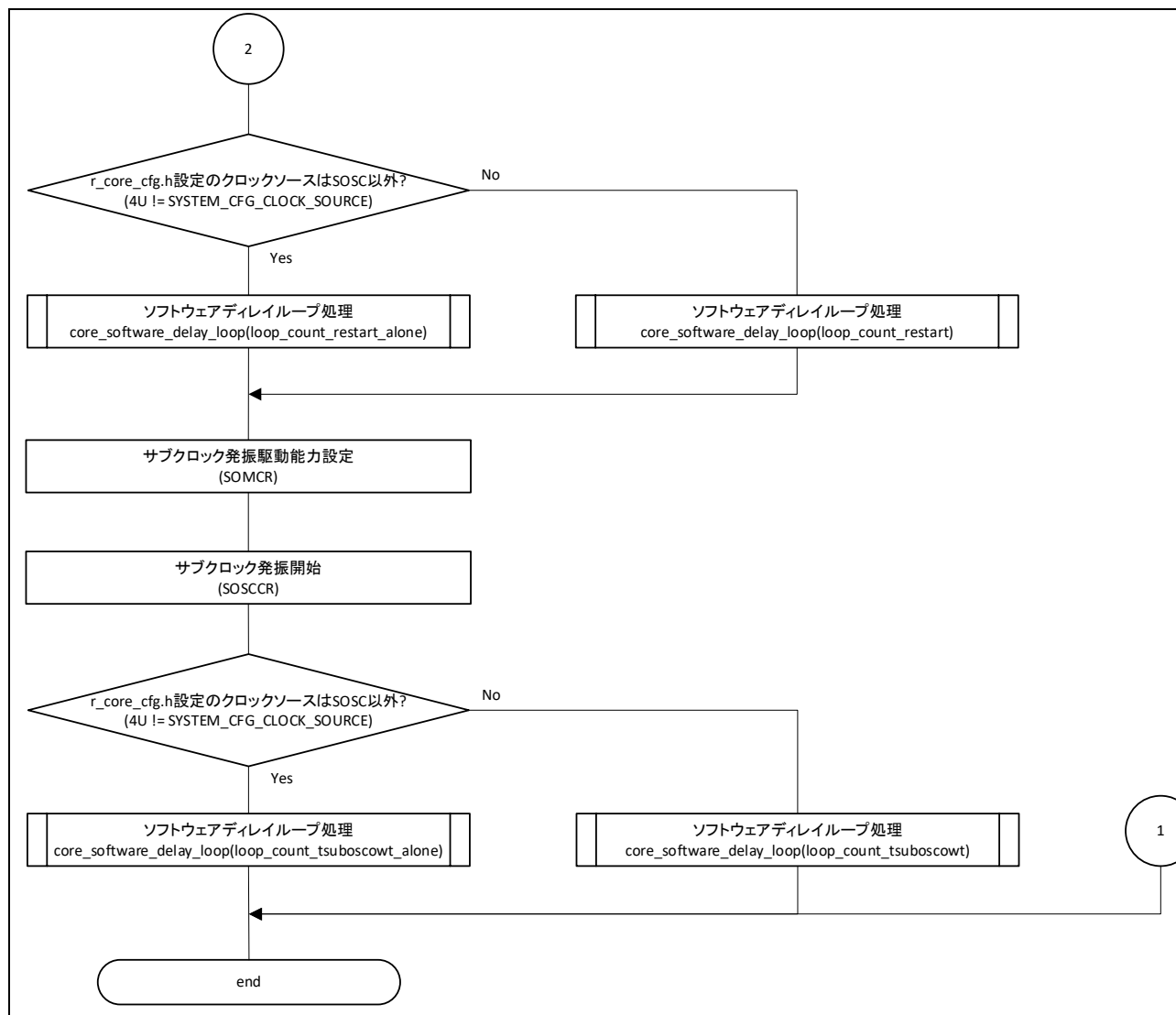


図 4-15 system\_sosc\_start 関数フロー (2/2)

4.3.10 system\_sosc\_stop 関数

表 4-43 system\_sosc\_stop 関数仕様

書式	void system_sosc_stop(void)
仕様説明	サブクロックの動作を停止します。
引数	なし
戻り値	なし
備考	—

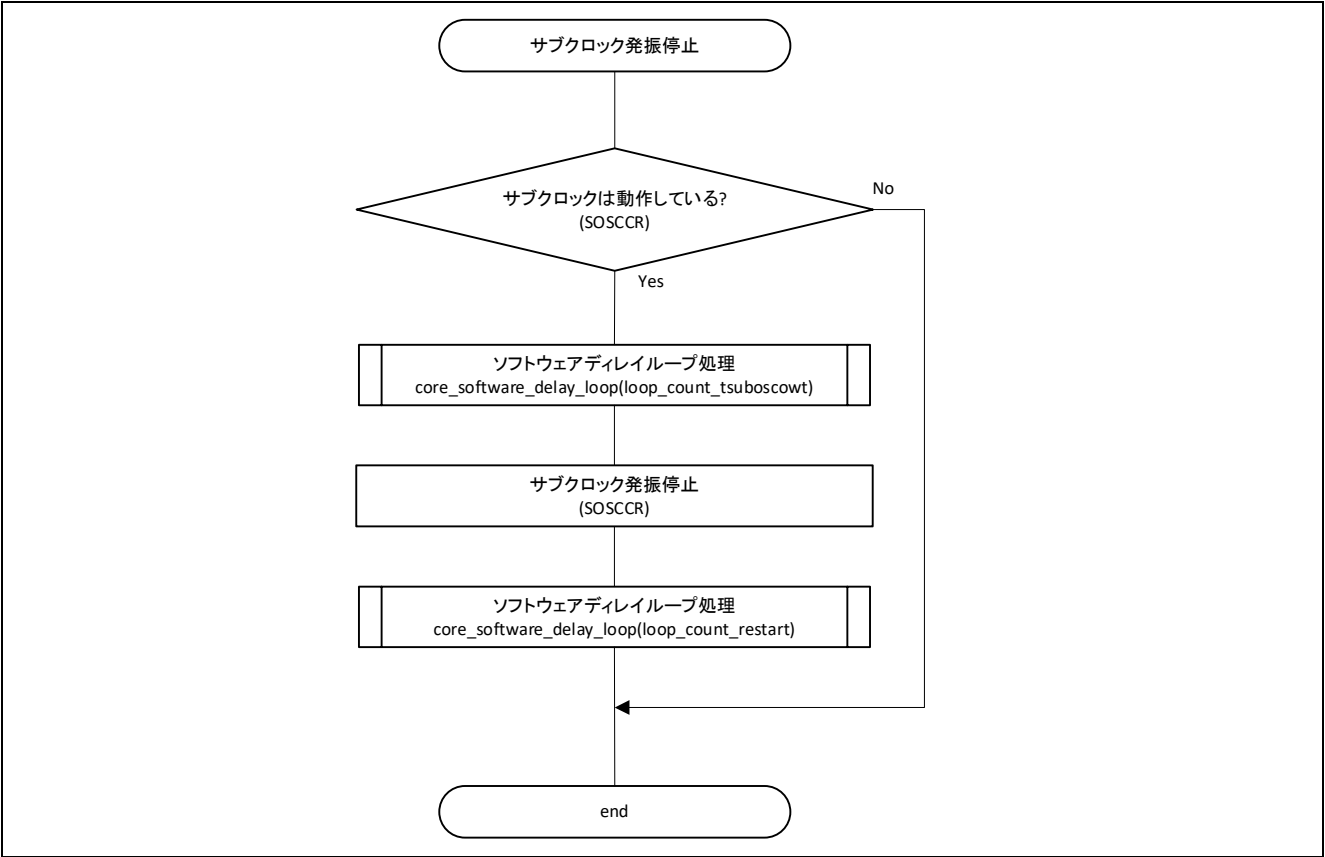


図 4-16 system\_sosc\_stop 関数フロー

## 4.3.11 system\_normal\_to\_boost 関数

表 4-44 system\_normal\_to\_boost 関数仕様

書式	void system_normal_to_boost(void)
仕様説明	電力制御モードをノーマルモード(High-speed)からブーストモードへ移行します。
引数	なし
戻り値	なし
備考	—

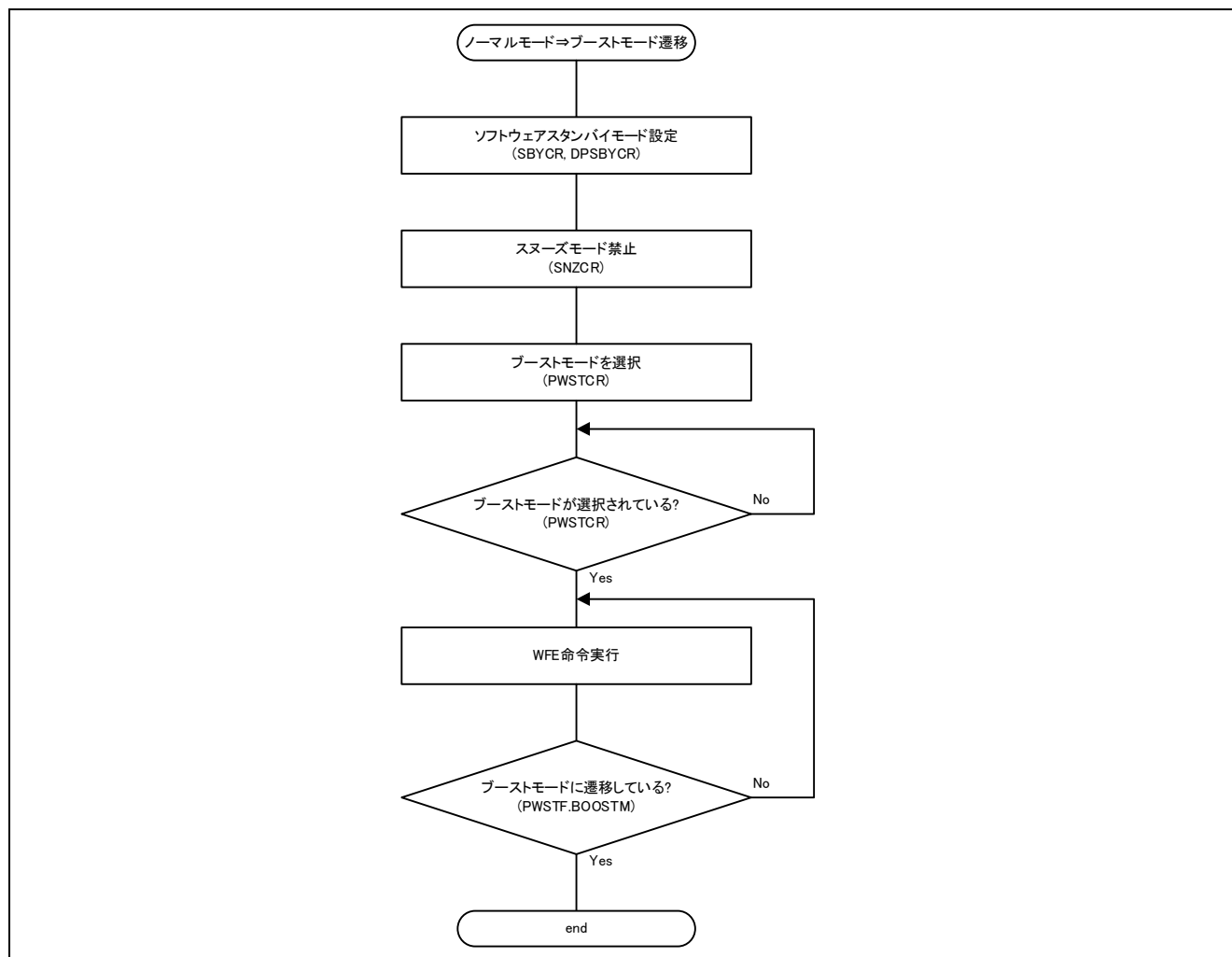


図 4-17 system\_normal\_to\_boost 関数フロー



## 4.3.12 system\_normal\_to\_vbb 関数

表 4-45 system\_normal\_to\_vbb 関数仕様

書式	void system_normal_to_vbb(void)
仕様説明	電力制御モードをノーマルモード(High-speed)から低リーク電流(VBB)モードへ移行します。
引数	なし
戻り値	なし
備考	—

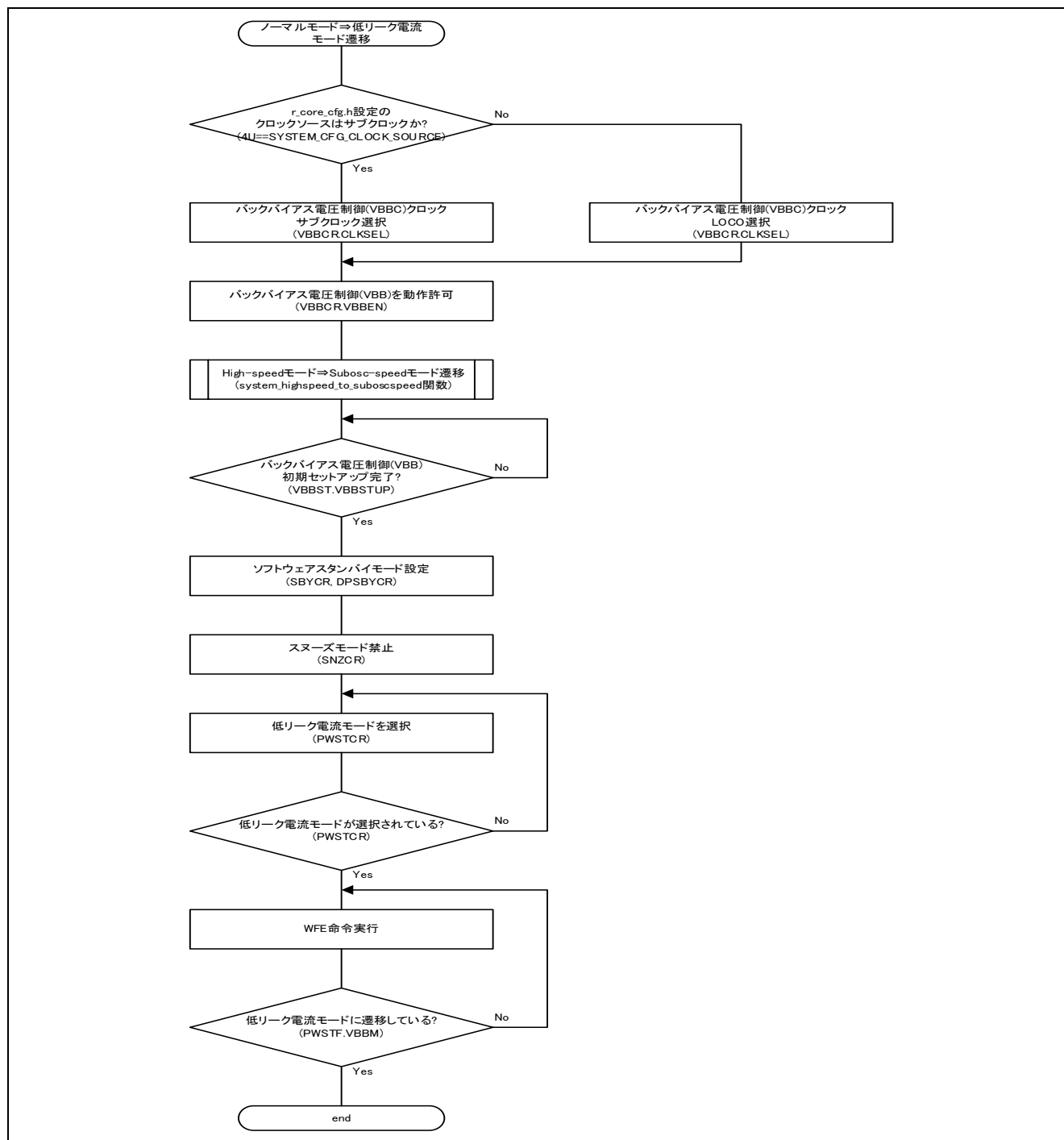


図 4-18 system\_normal\_to\_vbb 関数フロー

4.3.13    Reset\_Handler 関数

表 4-46    Reset\_Handler 関数仕様

書式	void Reset_Handler(void)
仕様説明	起動時の初期動作を行います。
引数	なし
戻り値	なし
備考	—

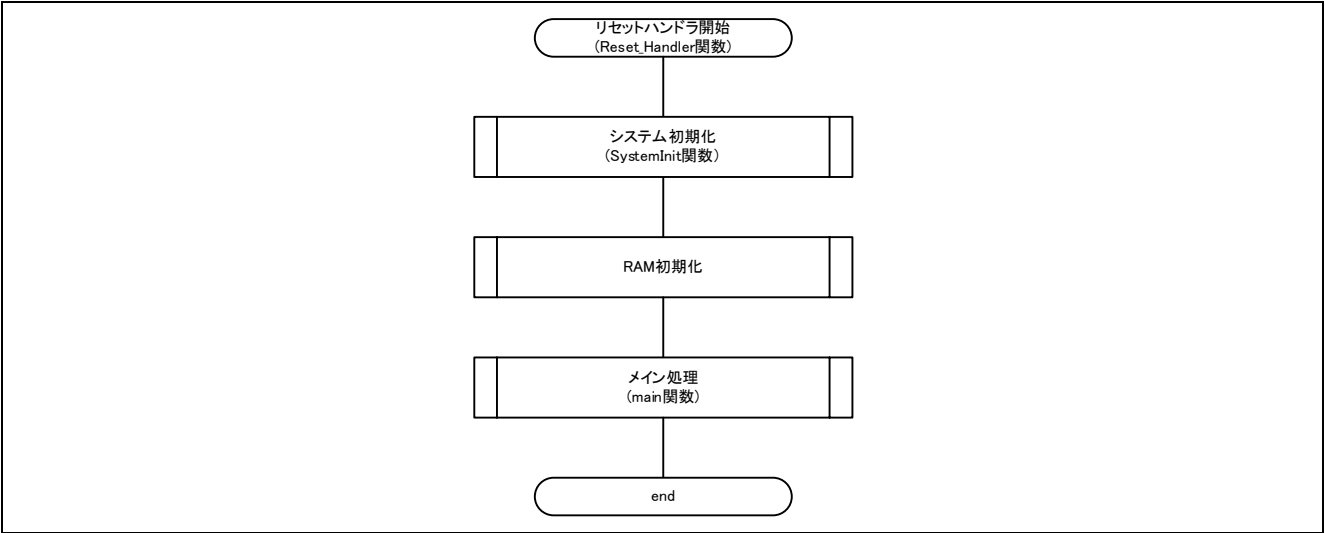


図 4-19    Reset\_Handler 関数フロー

4.3.14 SystemInit 関数

表 4-47 SystemInit 関数仕様

書式	void SystemInit(void)
仕様説明	起動時のシステム初期化を行います。
引数	なし
戻り値	なし
備考	—

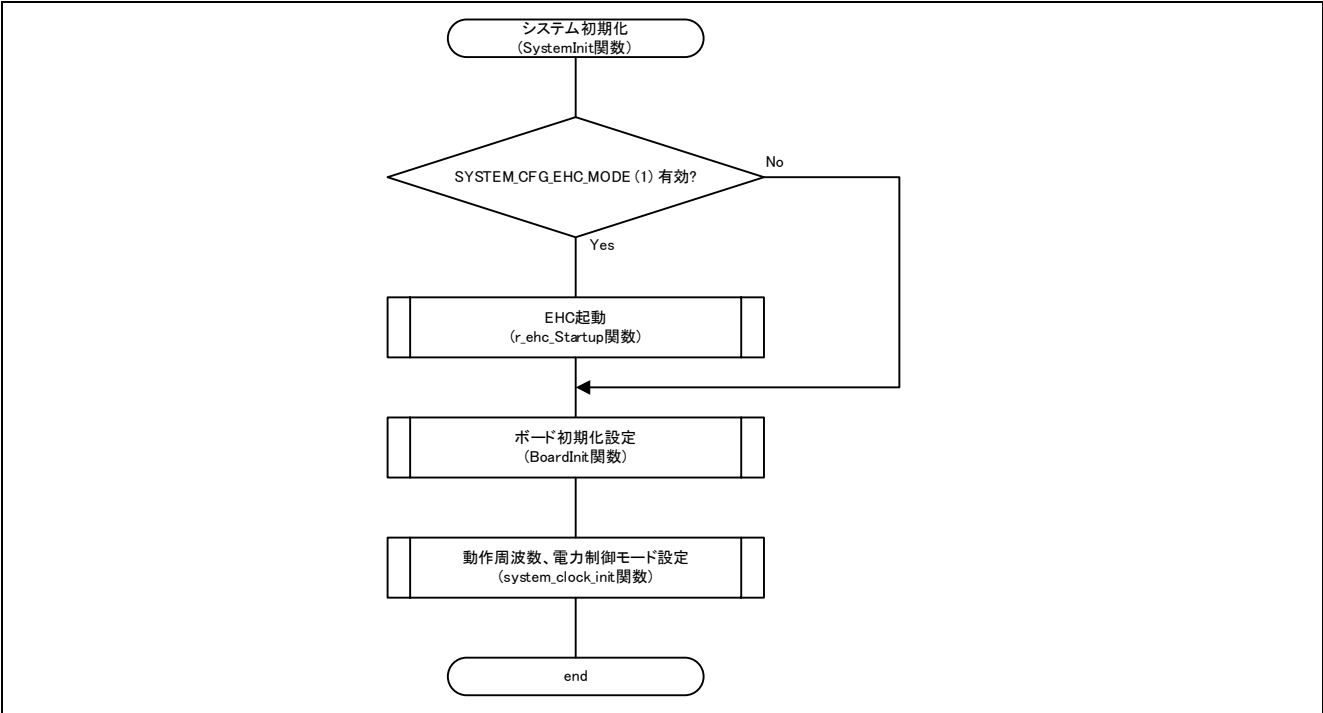


図 4-20 SystemInit 関数フロー

## 4.4 付録

## 4.4.1 プリプロセッサエラー一覧

表 4-48 プリプロセッサエラー一覧

No.	エラー内容	エラーに関わるコンフィグ
1	"ERROR - HOCO frequency can not selected Normal mode (High-speed) (r_core_cfg.h)."	・ HOCO をシステムクロックに選択して Normal mode(High-speed) で設定できないコンフィグを選択しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_POWER_CONTROL_MODE SYSTEM_CFG_CLOCK_SOURCE SYSTEM_CFG_HOCO_FREQUENCY
2		・ HOCO をシステムクロックに選択していない時に設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_HOCO_ENABLE SYSTEM_CFG_HOCO_FREQUENCY SYSTEM_CFG_POWER_CONTROL_MODE
3	"ERROR - HOCO frequency can not selected Low-Leakage-Current mode (VBB) (r_core_cfg.h)."	・ HOCO をシステムクロックに選択して Low-Leakage-Current mode (VBB) で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_POWER_CONTROL_MODE SYSTEM_CFG_CLOCK_SOURCE
4		・ HOCO をシステムクロックに選択していない時に設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_HOCO_ENABLE SYSTEM_CFG_POWER_CONTROL_MODE
5	"ERROR - HOCO frequency parameter of enable select is out of range (r_core_cfg.h)."	・ HOCO(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_HOCO_ENABLE
6	"ERROR - HOCO frequency (r_core_cfg.h)"	・ HOCO(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_HOCO_FREQUENCY
7	"ERROR - MOCO frequency can not selected Low-Leakage-Current mode (VBB) (r_core_cfg.h)"	・ MOCO をシステムクロックに選択して Low-Leakage-Current mode (VBB)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_POWER_CONTROL_MODE SYSTEM_CFG_CLOCK_SOURCE
8		・ MOCO をシステムクロックに選択していない時に設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_MOCO_ENABLE SYSTEM_CFG_POWER_CONTROL_MODE
9	"ERROR - MOCO frequency parameter of enable select is out of range (r_core_cfg.h)."	・ MOCO 停止(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。

		SYSTEM_CFG_MOCO_ENABLE
10	"ERROR - LOCO frequency division ratio (r_core_cfg.h)"	<ul style="list-style-type: none"> <li>・ LOCO をシステムクロックに選択して Low-Leakage-Current mode (VBB) で設定できないコンフィグを設定しています。</li> <li>以下のコンフィグ設定を見直してください。</li> </ul> SYSTEM_CFG_POWER_CONTROL_MODE SYSTEM_CFG_CLOCK_SOURCE SYSTEM_CFG_ICK_PCKA_DIV SYSTEM_CFG_PCKB_DIV
11	"ERROR - LOCO frequency parameter of enable select is out of range (r_core_cfg.h)."	<ul style="list-style-type: none"> <li>・ LOCO 停止(パラメータチェック)で設定できないコンフィグを設定しています。</li> <li>以下のコンフィグ設定を見直してください。</li> </ul> SYSTEM_CFG_LOCO_ENABLE
12	"ERROR - MOSC frequency out of range (r_core_cfg.h)"	<ul style="list-style-type: none"> <li>・ メインクロック(MOSC) をシステムクロックに選択して設定できないコンフィグを設定しています。</li> <li>以下のコンフィグ設定を見直してください。</li> </ul> SYSTEM_CFG_MOSC_FREQUENCY_HZ SYSTEM_CFG_CLOCK_SOURCE
13	"ERROR - MOSC frequency selected Low-Leakage-Current mode (VBB) (r_core_cfg.h)"	<ul style="list-style-type: none"> <li>・ メインクロック(MOSC) をシステムクロックに選択して設定できないコンフィグを設定しています。</li> <li>以下のコンフィグ設定を見直してください。</li> </ul> SYSTEM_CFG_POWER_CONTROL_MODE SYSTEM_CFG_CLOCK_SOURCE
14	"ERROR - MOSC frequency can not selected Low-Leakage-Current mode (VBB) (r_core_cfg.h)."	<ul style="list-style-type: none"> <li>・ メインクロック(MOSC) をシステムクロックに選択していない時に設定できないコンフィグを設定しています。</li> <li>以下のコンフィグ設定を見直してください。</li> </ul> SYSTEM_CFG_MOSC_ENABLE SYSTEM_CFG_POWER_CONTROL_MODE
15	"ERROR - MOSC frequency parameter of enable select is out of range (r_core_cfg.h)."	<ul style="list-style-type: none"> <li>・ メインクロック(MOSC)(パラメータチェック)で設定できないコンフィグを設定しています。</li> <li>以下のコンフィグ設定を見直してください。</li> </ul> SYSTEM_CFG_MOSC_ENABLE
16	"ERROR - MOSC frequency (r_core_cfg.h)"	<ul style="list-style-type: none"> <li>・ メインクロック(MOSC)(パラメータチェック)で設定できないコンフィグを設定しています。</li> <li>以下のコンフィグ設定を見直してください。</li> </ul> SYSTEM_CFG_MOSC_FREQUENCY_HZ SYSTEM_CFG_MOSC_CLOCK_SOURCE
17	"ERROR - MOSC frequency parameter of drive capability select is out of range (r_core_cfg.h)"	<ul style="list-style-type: none"> <li>・ メインクロック(MOSC)(パラメータチェック)で設定できないコンフィグを設定しています。</li> <li>以下のコンフィグ設定を見直してください。</li> </ul> SYSTEM_CFG_MOSC_DRIVE
18	"ERROR - MOSC frequency parameter of clock source select is out of range (r_core_cfg.h)."	<ul style="list-style-type: none"> <li>・ メインクロック(MOSC)(パラメータチェック)で設定できないコンフィグを設定しています。</li> <li>以下のコンフィグ設定を見直してください。</li> </ul> SYSTEM_CFG_MOSC_CLOCK_SOURCE
19	"ERROR - MOSC frequency parameter of low power enable select is out of range (r_core_cfg.h)."	<ul style="list-style-type: none"> <li>・ メインクロック(MOSC)(パラメータチェック)で設定できないコンフィグを設定しています。</li> <li>以下のコンフィグ設定を見直してください。</li> </ul> SYSTEM_CFG_MOSC_LOW_POWER_ENABLE

20	"ERROR - MOSC frequency parameter of stabilization time select is out of range (r_core_cfg.h)"	・メインクロック(MOSC)(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_MOSC_WAIT_TIME
21	"ERROR - SOSC frequency division ratio (r_core_cfg.h)"	・サブクロック(SOSC) をシステムクロックに選択して設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_POWER_CONTROL_MODE SYSTEM_CFG_CLOCK_SOURCE SYSTEM_CFG_ICK_PCKA_DIV SYSTEM_CFG_PCKB_DIV
22	"ERROR - SOSC frequency parameter of enable select is out of range (r_core_cfg.h)."	・サブクロック(SOSC)(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_SOSC_ENABLE
23	"ERROR - SOSC frequency parameter of drive capability select is out of range (r_core_cfg.h)."	・サブクロック(SOSC)(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_SOSC_DRIVE
24	"ERROR - SOSC frequency parameter of noise filter select is out of range (r_core_cfg.h)."	・サブクロック(SOSC)(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_SOSC_NF_STOP
25	"ERROR - Clock source select is out of range (r_core_cfg.h)"	・クロックソース選択(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_CLOCK_SOURCE
26	"ERROR - ICLK and PCLKA frequency division ratio select is out of range (r_core_cfg.h)"	・システムクロック(ICLK)/周辺モジュールクロック(PCLKA)選択(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_ICK_PCKA_DIV
27	"ERROR - PCLKB frequency division ratio select is out of range (r_core_cfg.h)"	・周辺モジュールクロック B(PCLKB)選択(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_PCKB_DIV
28	"ERROR - Power control mode select is out of range (r_core_cfg.h)"	・電源状態選択(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_POWER_CONTROL_MODE
29	"ERROR - EHC operating mode select is out of range (r_core_cfg.h)."	・EHC 起動モード設定(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_EHC_MODE
30	"ERROR - PCKB frequency division ratio (r_core_cfg.h)."	・周辺モジュールクロック B(PCLKB)(パラメータチェック)で設定できないコンフィグを設定しています。 以下のコンフィグ設定を見直してください。 SYSTEM_CFG_ICK_PCKA_DIV SYSTEM_CFG_PCKB_DIV

		SYSTEM_CFG_HOCO_FREQUENCY SYSTEM_CFG_CLOCK_SOURCE SYSTEM_CFG_MOSC_FREQUENCY_HZ
--	--	--

5. EHC 起動モード

- EHC 起動モードを動作させるには、以下の条件が必要になります。
- ・ オプションとなるエナジーハーベスト(EHC)起動ソースファイル(r\_ehc.c)の取り込み
  - ・ コンフィグレーション定義ファイル(r\_core\_cfg.h)での EHC 起動モード設定を有効

5.1 ファイル構成

EHC 起動モードは、RE01 256KB グループ CMSIS Driver Package のオプションに該当し、ベンダ独自ファイル格納ディレクトリ内の r\_ehc.c のファイルで構成されます。本ファイルの役割を表 5-1 に示します。RE01 256KB グループ CMSIS Driver Package における r\_ehc のファイル構成を図 5-1 に示します。内部構成については図 5-2 に示します。

表 5-1 エナジーハーベスト(EHC)起動ソースファイルの役割

ファイル名	内容
r_ehc.c	エナジーハーベスト(EHC)起動ソースファイルです。 EHC 起動モードでは、二次電池への充電判定および二次電池を使用した動作をするための初期化、電力制御モード遷移、電源供給モードの遷移を行います。 EHC 起動モードで実行する場合は、本ファイルをビルドする必要があります。



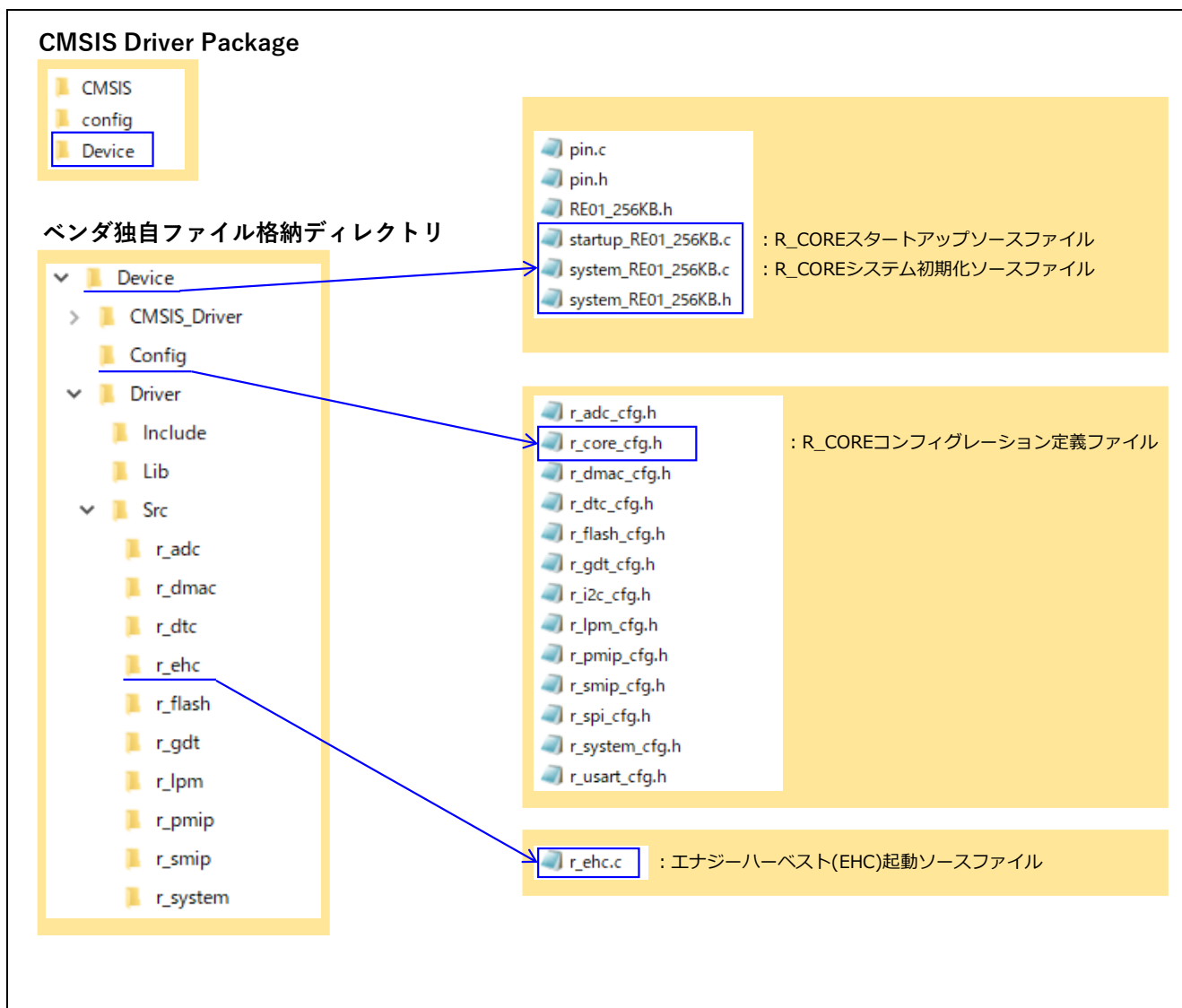


図 5-1 CMSIS Driver Package における r\_ehc のファイル構成

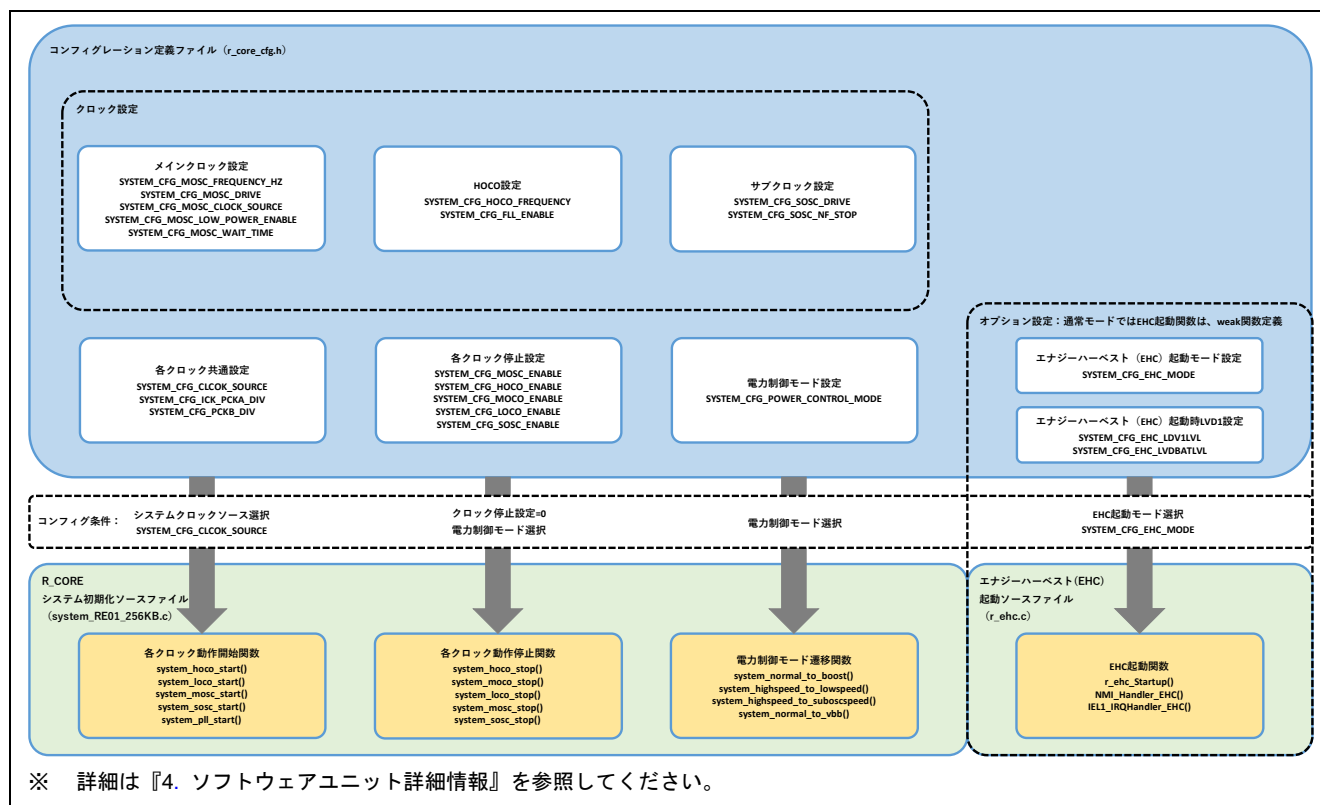


図 5-2 EHC の位置づけ

## 5.2 EHC 起動モード遷移

EHC 起動モードへ遷移する手順を示します。

エナジーハーベスト(EHC)起動モード設定(SYSTEM\_CFG\_EHC\_MODE)が有効の時に EHC 初期化を行なって起動します。起動モード設定が有効でリセットした場合、再度 EHC 起動し二次電池への再充電を行います。二次電池電圧低下による割込みが発生するまで動作し続けます。

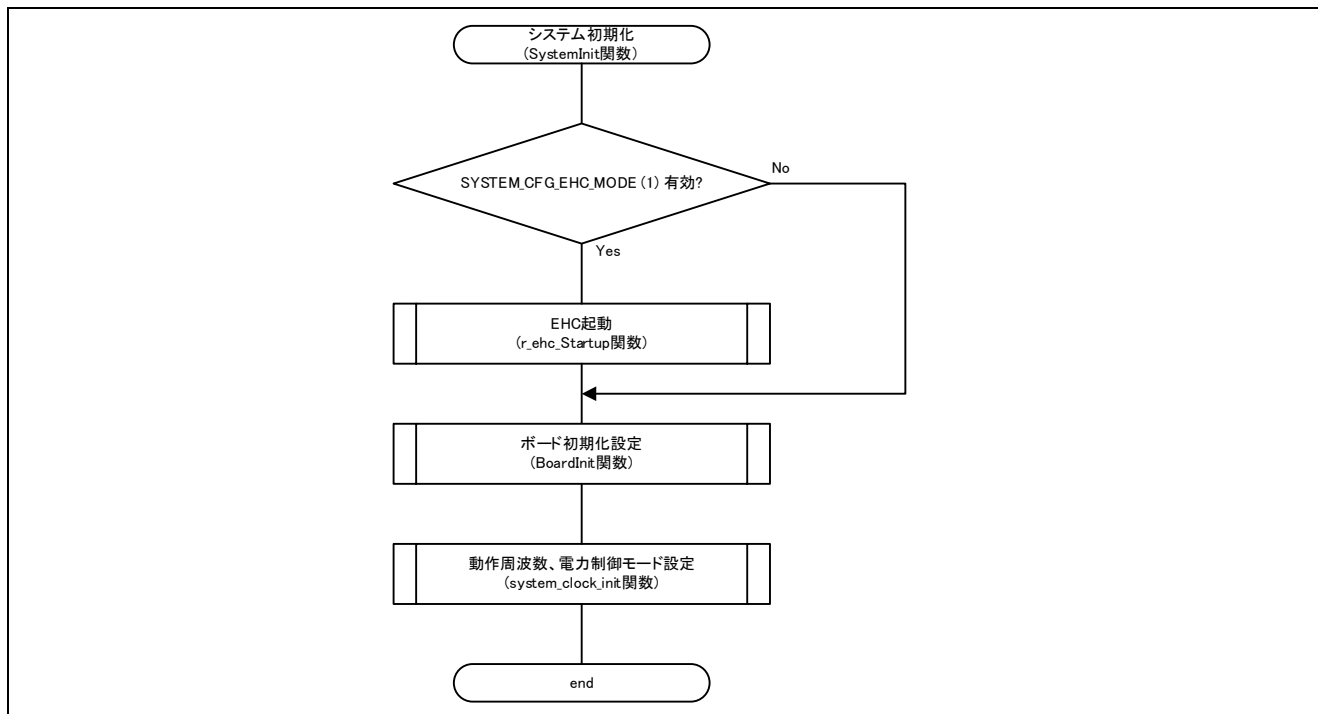


図 5-3 SystemInit 関数による起動モード遷移

### 5.3 コンフィグレーション

R\_CORE は、ユーザが定義可能なコンフィグレーションを `r_core_cfg.h` ファイルに用意します。

各初期設定の一覧を表 5-2 に示します。各設定については、表内の章番号を参照してください。

表 5-2 R\_CORE コンフィグレーションの初期設定一覧

章番号	コンフィグレーション	設定内容	初期設定
4.1.19	SYSTEM_CFG_EHC_MODE	EHC 起動処理の有効/無効を設定	1
4.1.20	SYSTEM_CFG_EHC_LVD1LVL	電圧検出 1 レベルを設定	7
4.1.21	SYSTEM_CFG_EHC_LVDBATLVL	電圧検出 BAT レベルを設定	0

### 5.4 関数仕様

EHC 起動モードでは、デバイスの消費電力を抑えるため低消費電力状態に遷移します。EHC 起動モードでの状態遷移を図 5-4 に示します。

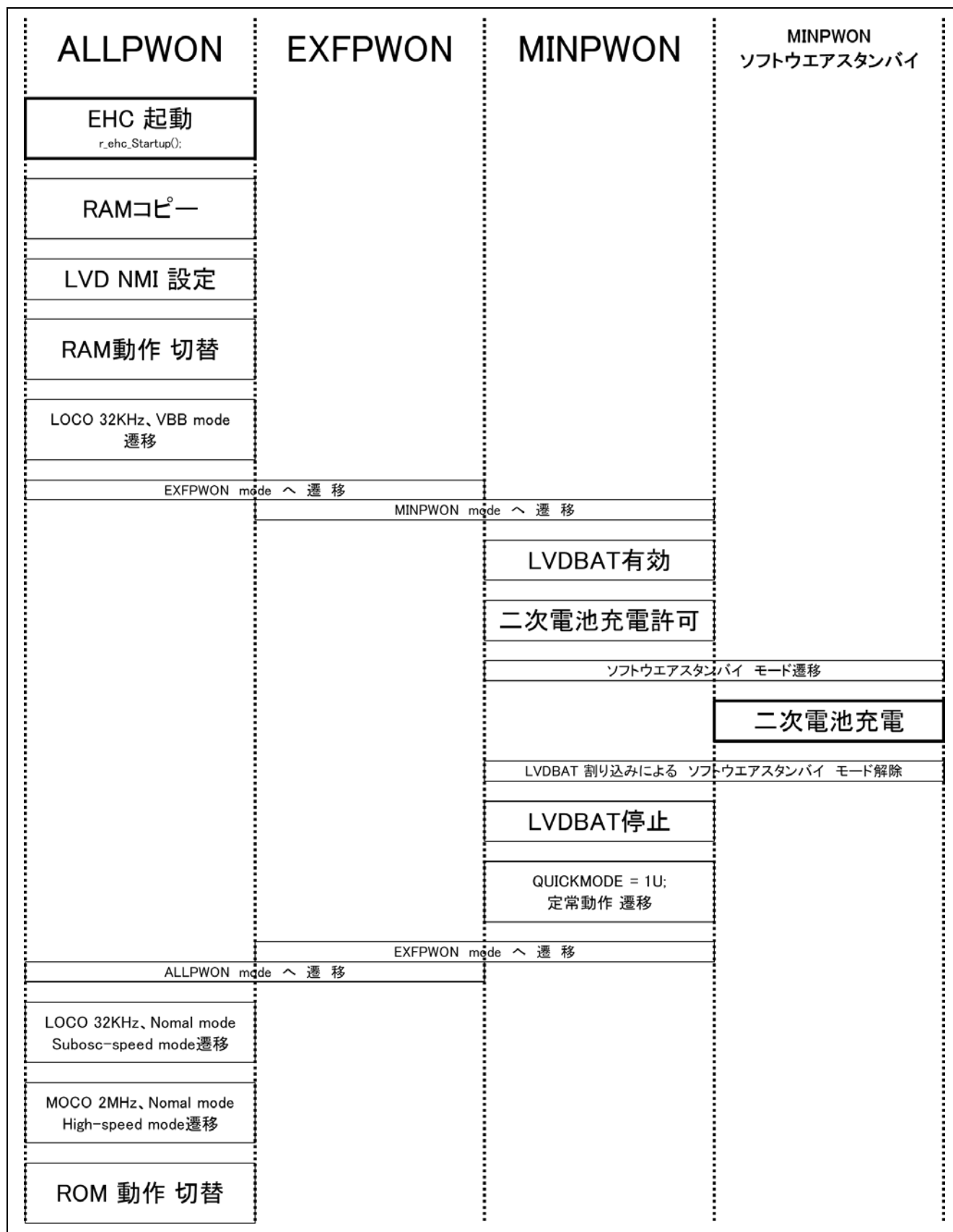


図 5-4 EHC 起動モード時の状態遷移

## 5.4.1 r\_ehc\_Startup 関数

表 5-3 r\_ehc\_Startup 関数仕様

書式	void r_ehc_Startup(void)
仕様説明	EHC 起動処理を行います。
引数	なし
戻り値	なし
備考	—

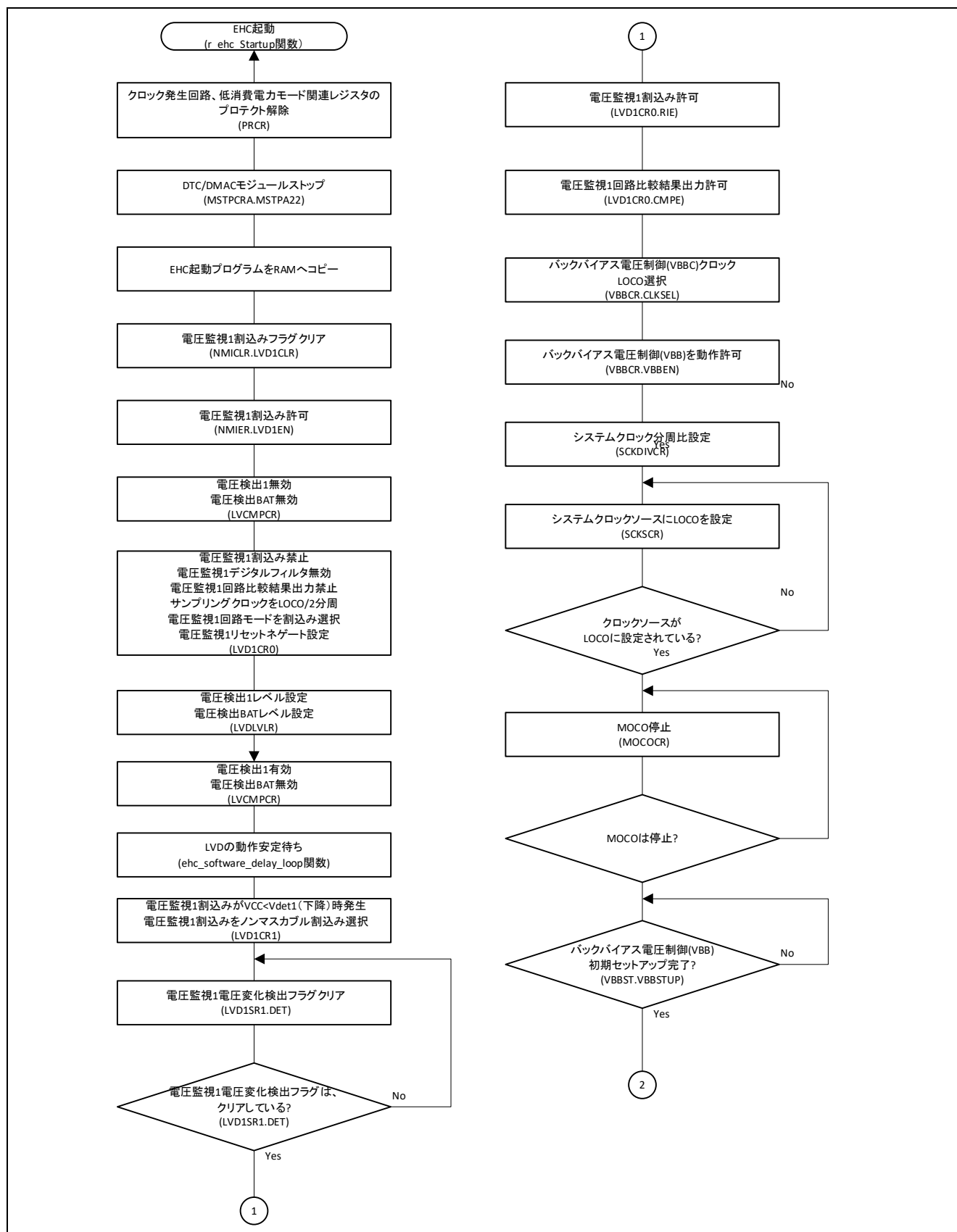


図 5-5 r\_ehc\_Startup 関数フロー(1/3)

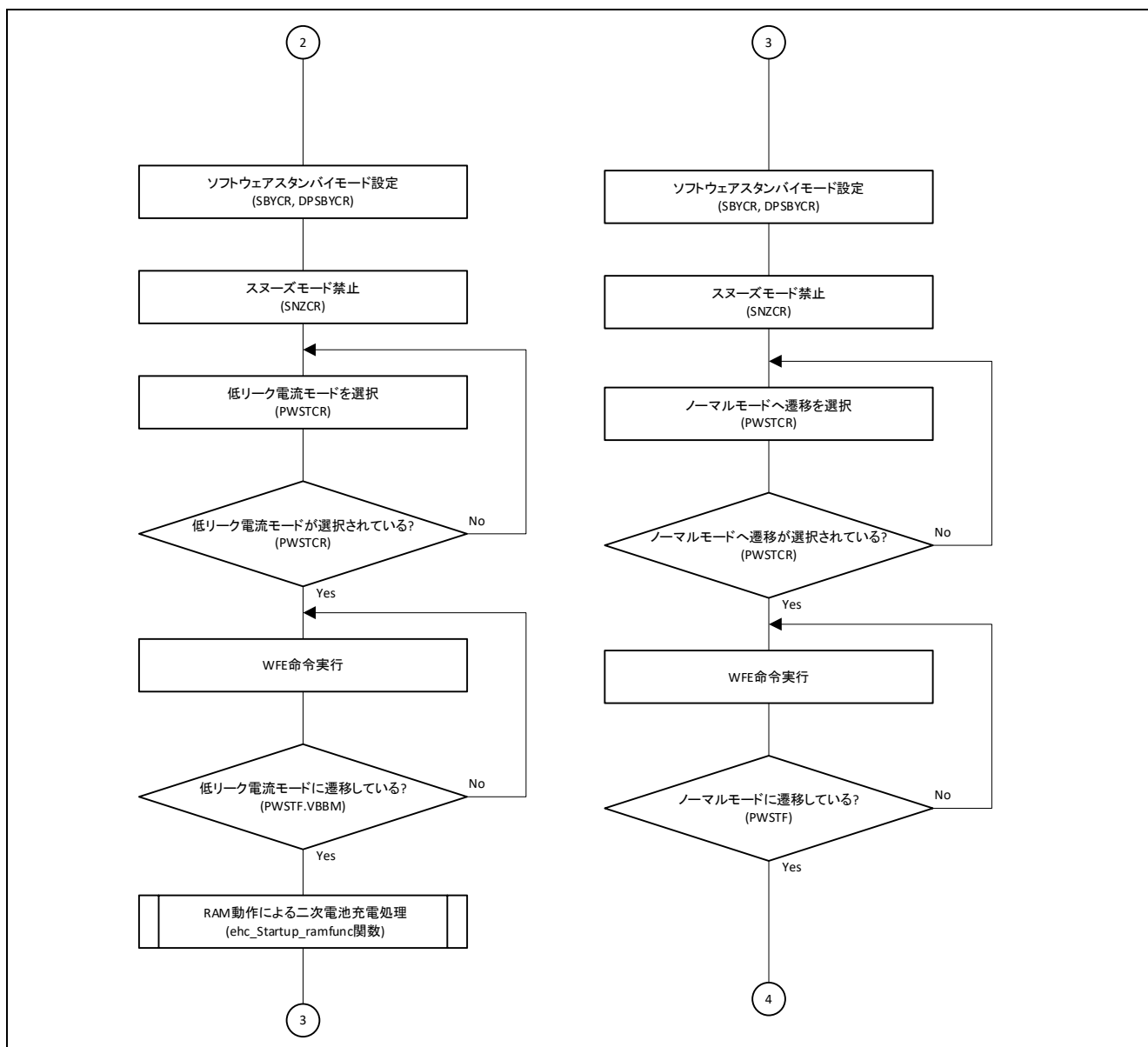


図 5-6 r\_ehc\_Startup 関数フロー(2/3)



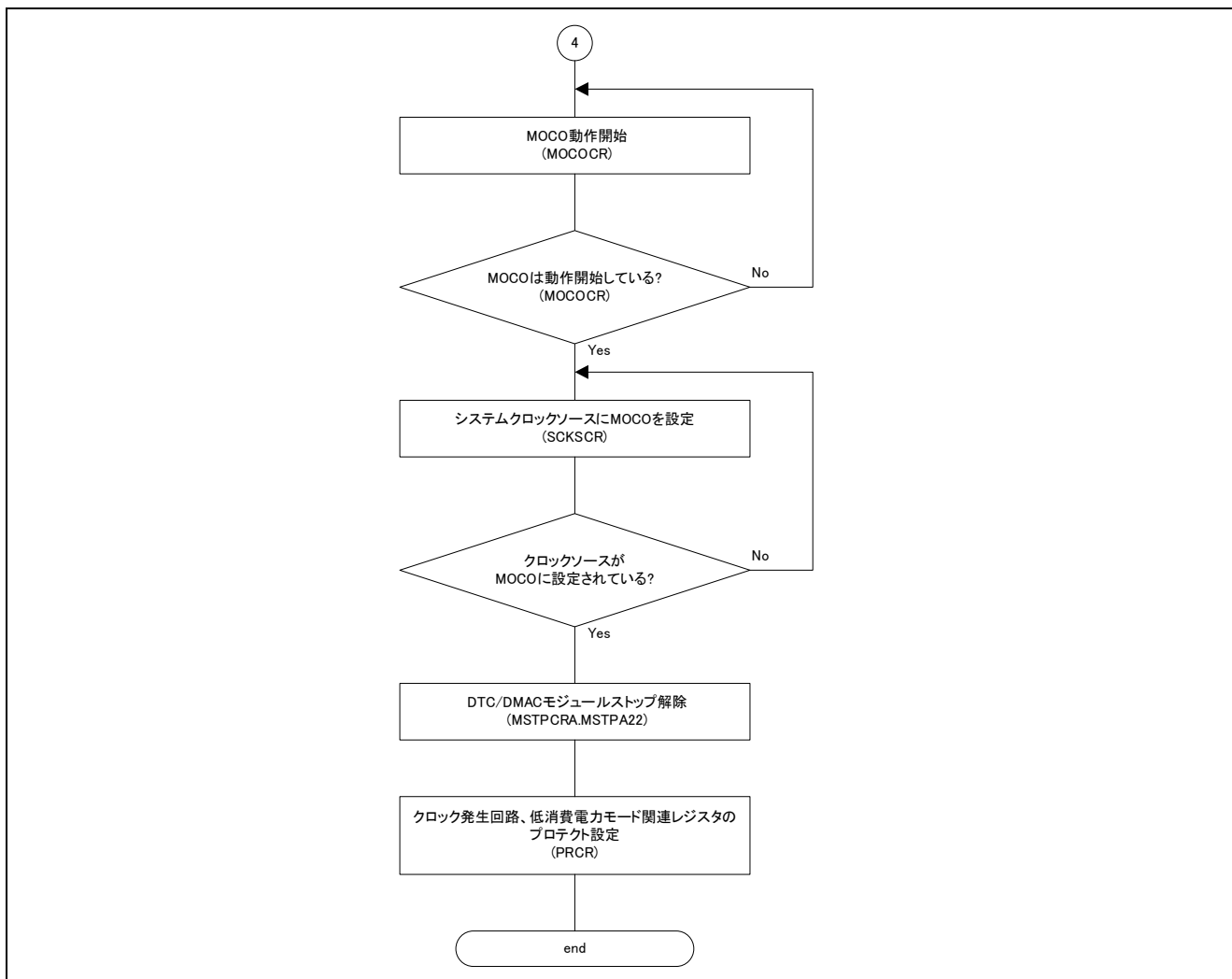


図 5-7 r\_ehc\_Startup 関数フロー(3/3)

5.4.2 ehc\_Startup\_ramfunc 関数

表 5-4 ehc\_Startup\_ramfunc 関数仕様

書式	void ehc_Startup_ramfunc(void)
仕様説明	EHC 起動時の二次電池充電動作を行います。
引数	なし
戻り値	なし
備考	—

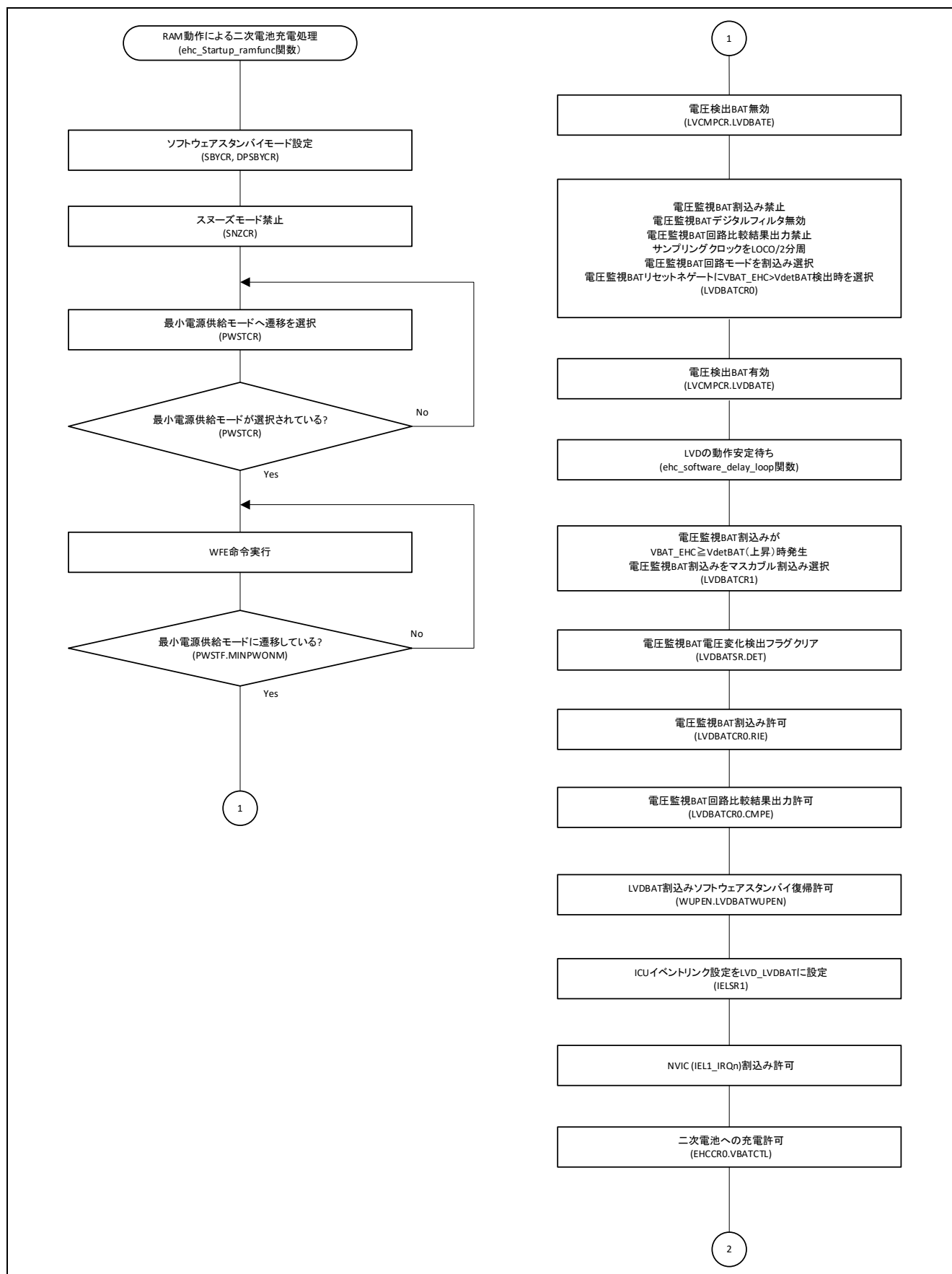


図 5-8 ehc\_Startup\_ramfunc 関数フロー(1/3)

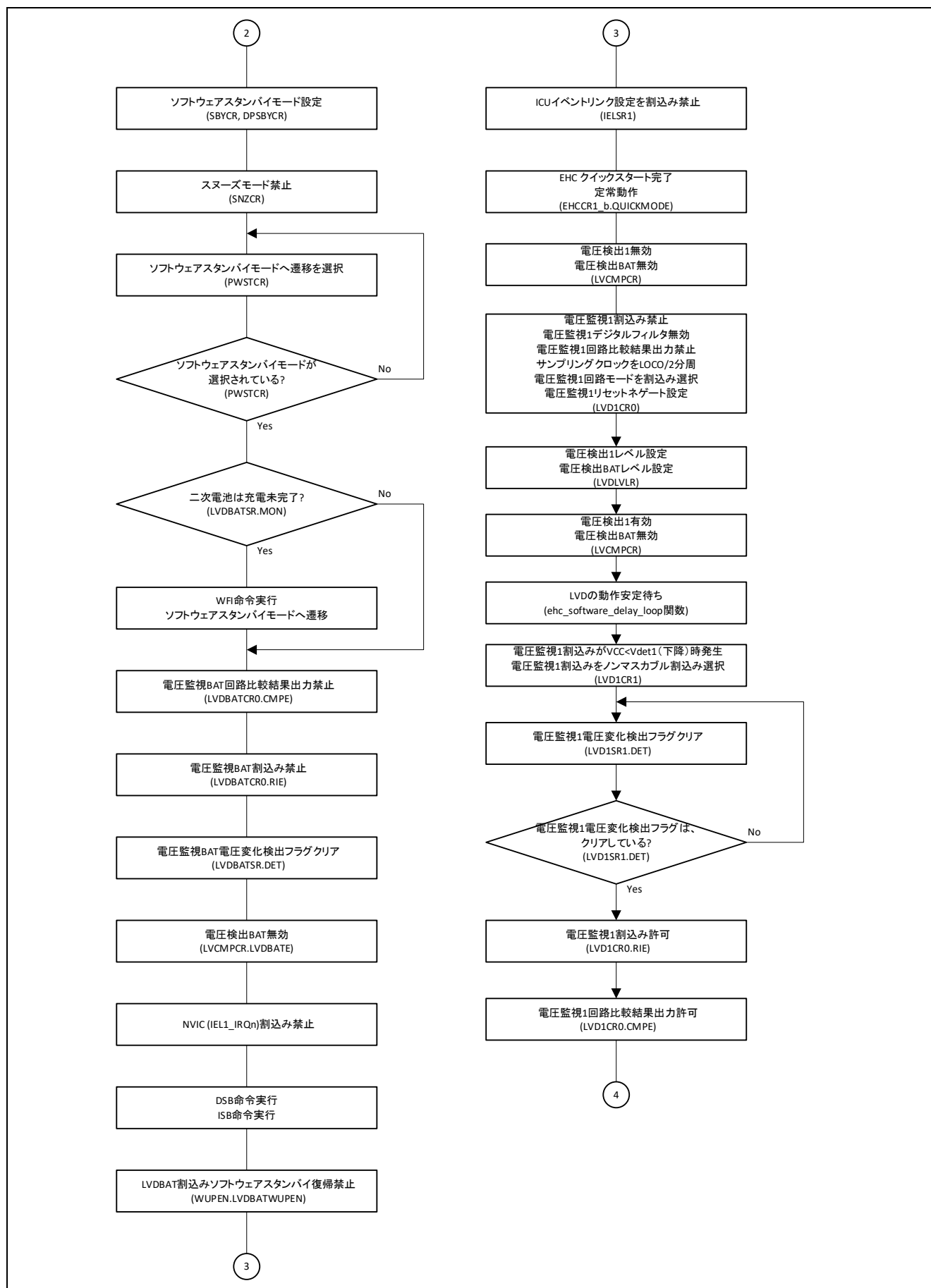


図 5-9 ehc\_Startup\_ramfunc 関数フロー(2/3)

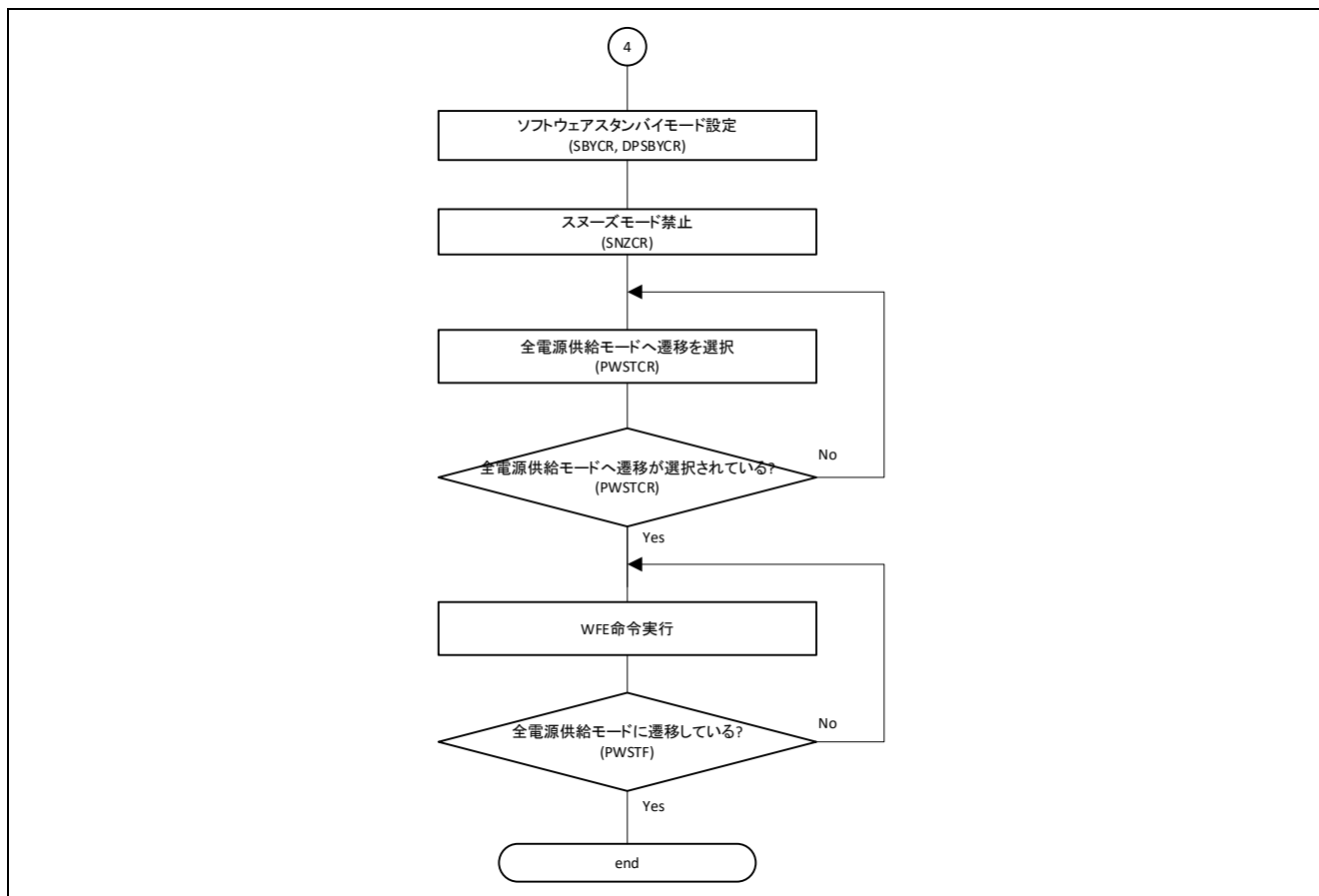


図 5-10 ehc\_Startup\_ramfunc 関数フロー(3/3)

5.4.3 NMI\_Handler\_EHC 関数

表 5-5 NMI\_Handler\_EHC 関数仕様

書式	void NMI_Handler_EHC(void)
仕様説明	電圧監視 1 割り込みが発生したとき、本関数に分岐します。 EHC 機能を初期化します。
引数	なし
戻り値	なし
備考	・ MCU 内部の電圧低下を検出すると電圧監視 1 割り込みが発生します。EHC 機能の初期化により、MCU 内部への電源供給が停止します。 ・ VCC 電圧下降時の割り込みハンドラ(関数名：NMI_Handler)はユーザが作成してください。

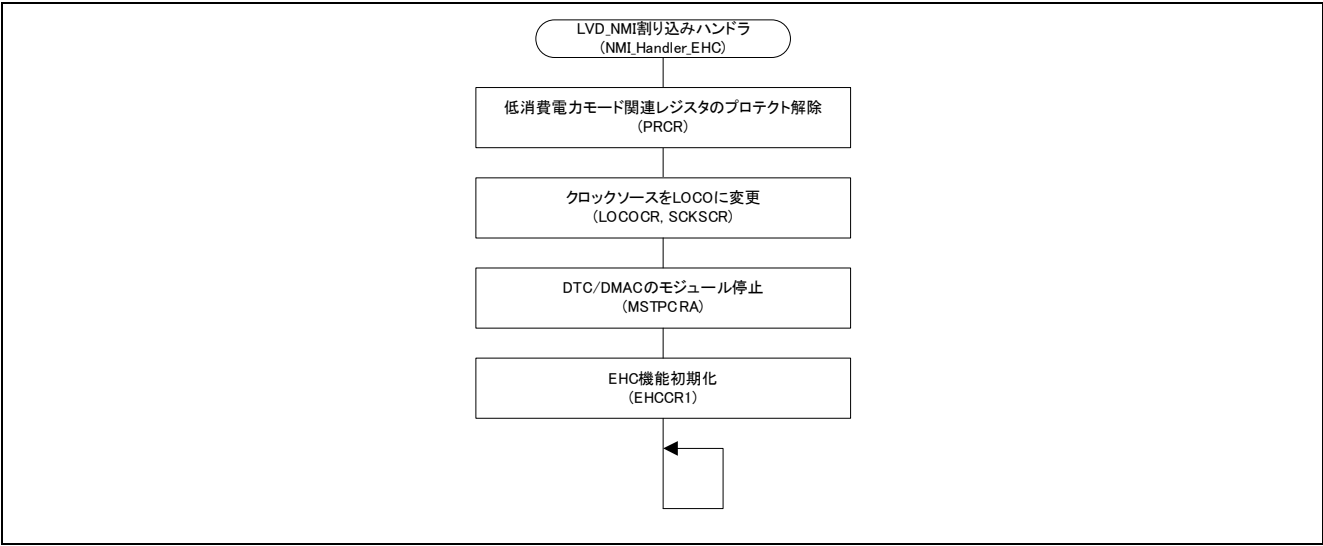


図 5-11 NMI\_Handler\_EHC 関数フロー

5.4.4 IEL1\_IRQHandler\_EHC 関数

表 5-6 IEL1\_IRQHandler\_EHC 関数仕様

書式	void IEL1_IRQHandler_EHC(void)
仕様説明	電圧監視 BAT 割り込みが発生したとき、本関数に分岐します。 割り込みステータスフラグをクリアします。
引数	なし
戻り値	なし
備考	二次電池の充電完了を検出すると電圧監視 BAT 割り込みが発生し、ソフトウェアスタンバイモードが解除されます。

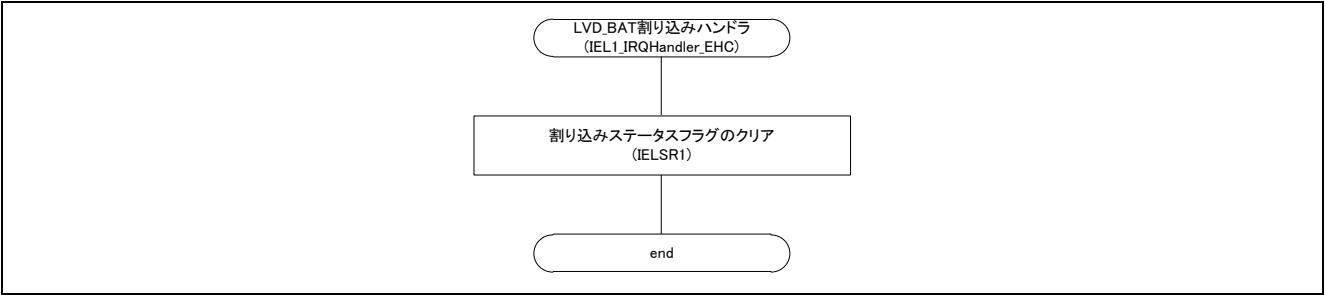


図 5-12 IEL1\_IRQHandler\_EHC 関数フロー

## ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/contact/>

すべての商標および登録商標は、それぞれの所有者に帰属します。



## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2020.07.02	－	初版

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレス(予約領域)のアクセス禁止

【注意】リザーブアドレス(予約領域)のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス(予約領域)があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子(または外部発振回路)を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子(または外部発振回路)を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違うと、内部ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、  
電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、  
融端末基幹システム、各種安全制御装置等  
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載されている内容または当社製品についてご不明点がございましたら、当社の営業担当までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。  
注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

Re 4.0-1 2017.11)



ルネサス エレクトロニクス株式会社

営業お問合せ窓口

<http://www.renesas.com>

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

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア） ■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口： <https://www.renesas.com/contact/>