

## R78/G23

### CPU クロックの切り替えとスタンバイ設定

---

#### 要旨

本アプリケーションノートでは、RL78/G23 の CPU クロックの切り替えとスタンバイ設定 (動作モードの切り替え) を説明します。

本アプリケーションでは、スイッチ入力により、CPU クロックと動作モードの切り替えを行います。5 つの LED 点灯を制御し、CPU クロックの状態と動作モードを表します。

#### 動作確認デバイス

RL78/G23

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

## 目次

1. 仕様	5
1.1 動作概要	5
1.2 動作詳細	8
1.3 CPU クロックの切り替え	12
1.3.1 高速オンチップ・オシレータ・クロックから中速オンチップ・オシレータ・クロック への切り替え	13
1.3.2 高速オンチップ・オシレータ・クロックから低速オンチップ・オシレータ・クロックへの切り替え	14
1.3.3 高速オンチップ・オシレータ・クロックから高速システム・クロックへの切り替え	15
1.3.4 高速オンチップ・オシレータ・クロックからサブシステム・クロックへの切り替え	17
1.3.5 中速オンチップ・オシレータ・クロックから高速オンチップ・オシレータ・クロックへの切り替え	19
1.3.6 中速オンチップ・オシレータ・クロックから低速オンチップ・オシレータ・クロックへの切り替え	20
1.3.7 中速オンチップ・オシレータ・クロックから高速システム・クロックへの切り替え	21
1.3.8 中速オンチップ・オシレータ・クロックからサブシステム・クロックへの切り替え	23
1.3.9 低速オンチップ・オシレータ・クロックから高速オンチップ・オシレータ・クロックへの切り替え	25
1.3.10 低速オンチップ・オシレータ・クロックから中速オンチップ・オシレータ・クロックへの切り替え	26
1.3.11 低速オンチップ・オシレータ・クロックから高速システム・クロックへの切り替え	27
1.3.12 高速システム・クロックから高速オンチップ・オシレータ・クロックへの切り替え	29
1.3.13 高速システム・クロックから中速オンチップ・オシレータ・クロックへの切り替え	30
1.3.14 高速システム・クロックから低速オンチップ・オシレータ・クロックへの切り替え	31
1.3.15 高速システム・クロックからサブシステム・クロックへの切り替え	32
1.3.16 サブシステム・クロックから高速オンチップ・オシレータ・クロックへの切り替え	34
1.3.17 サブシステム・クロックから中速オンチップ・オシレータ・クロックへの切り替え	35
1.3.18 サブシステム・クロックから高速システム・クロックへの切り替え	36
2. 動作確認条件	38
3. ハードウェア説明	39
3.1 ハードウェア構成例	39
3.2 使用端子一覧	39
4. ソフトウェア説明	40
4.1 オプション・バイトの設定一覧	40
4.2 定数一覧	40
4.3 変数一覧	41
4.4 関数 (サブルーチン) 一覧	42
4.5 関数 (サブルーチン) 仕様	44
4.6 フローチャート	56
4.6.1 メイン処理	57
4.6.2 状態移行 AtoB	59
4.6.3 状態移行 BtoE	59
4.6.4 状態移行 EtoO	60

4.6.5	状態移行 OtoE .....	60
4.6.6	状態移行 EtoB.....	61
4.6.7	状態移行 BtoD .....	62
4.6.8	状態移行 DtoE .....	63
4.6.9	状態移行 EtoD .....	64
4.6.10	状態移行 DtoM.....	65
4.6.11	状態移行 MtoD.....	65
4.6.12	状態移行 DtoN .....	66
4.6.13	状態移行 NtoD .....	66
4.6.14	状態移行 DtoB .....	67
4.6.15	状態移行 BtoG .....	68
4.6.16	状態移行 GtoB .....	68
4.6.17	状態移行 BtoH .....	69
4.6.18	状態移行 HtoB .....	69
4.6.19	状態移行 BtoI.....	70
4.6.20	状態移行 ItoB.....	70
4.6.21	状態移行 BtoC .....	71
4.6.22	状態移行 CtoD .....	72
4.6.23	状態移行 DtoF.....	73
4.6.24	状態移行 FtoD.....	74
4.6.25	状態移行 DtoC .....	75
4.6.26	状態移行 CtoJ.....	76
4.6.27	状態移行 JtoC.....	76
4.6.28	状態移行 CtoK .....	77
4.6.29	状態移行 KtoC .....	77
4.6.30	状態移行 CtoL.....	78
4.6.31	状態移行 LtoC.....	78
4.6.32	状態移行 CtoE .....	79
4.6.33	状態移行 EtoC .....	80
4.6.34	状態移行 CtoF.....	81
4.6.35	状態移行 FtoC.....	82
4.6.36	状態移行 CtoB .....	83
4.6.37	状態移行 BtoF.....	84
4.6.38	状態移行 FtoP.....	85
4.6.39	状態移行 PtoF.....	85
4.6.40	状態移行 FtoB.....	86
4.6.41	状態移行 BtoQ .....	87
4.6.42	状態移行 QtoB .....	87
4.6.43	状態終了処理 .....	88
4.6.44	割込み待ち (NOP 繰り返し).....	88
4.6.45	割込み待ち (HALT) .....	89
4.6.46	割込み待ち (STOP) .....	90
4.6.47	割込み待ち (SNOOZE).....	91
4.6.48	高速オンチップ・オシレータ・クロック動作開始 .....	92
4.6.49	中速オンチップ・オシレータ・クロック動作開始 .....	92
4.6.50	低速オンチップ・オシレータ・クロック動作開始 .....	93
4.6.51	高速システム・クロック動作開始 .....	93

4.6.52	サブシステム・クロック動作開始 .....	94
4.6.53	高速オンチップ・オシレータ・クロック停止 .....	94
4.6.54	中速オンチップ・オシレータ・クロック停止 .....	95
4.6.55	低速オンチップ・オシレータ・クロック停止 .....	95
4.6.56	高速システム・クロック停止 .....	96
4.6.57	サブシステム・クロック停止 .....	96
4.6.58	クロック切り替え設定 .....	97
4.6.59	高速オンチップ・オシレータ・クロック起動モード設定 .....	99
4.6.60	ウェイト処理 (OSTC レジスタ) .....	100
4.6.61	メイン初期化処理 (ユーザ定義箇所) .....	100
4.6.62	割込みコントローラ初期化処理 (ユーザ定義箇所) .....	101
4.6.63	INTP0 割込み発生時処理 .....	101
4.6.64	INTP0 割込みフラグクリア .....	102
4.6.65	INTP0 割込みフラグ確認 .....	102
4.6.66	32 ビット・インターバル・タイマコンペア値設定 .....	103
4.6.67	ウェイト処理 (32 ビット・インターバル・タイマ) .....	104
4.6.68	LED 点灯/消灯制御 .....	105
4.6.69	ポート出力制御 .....	106
5.	サンプルコード .....	107
6.	参考ドキュメント .....	107
	改訂記録 .....	108

## 1. 仕様

### 1.1 動作概要

本アプリケーションノートでは、スイッチ入力によって図 1-1に沿ったCPU クロックと動作モードの切り替えを行います。

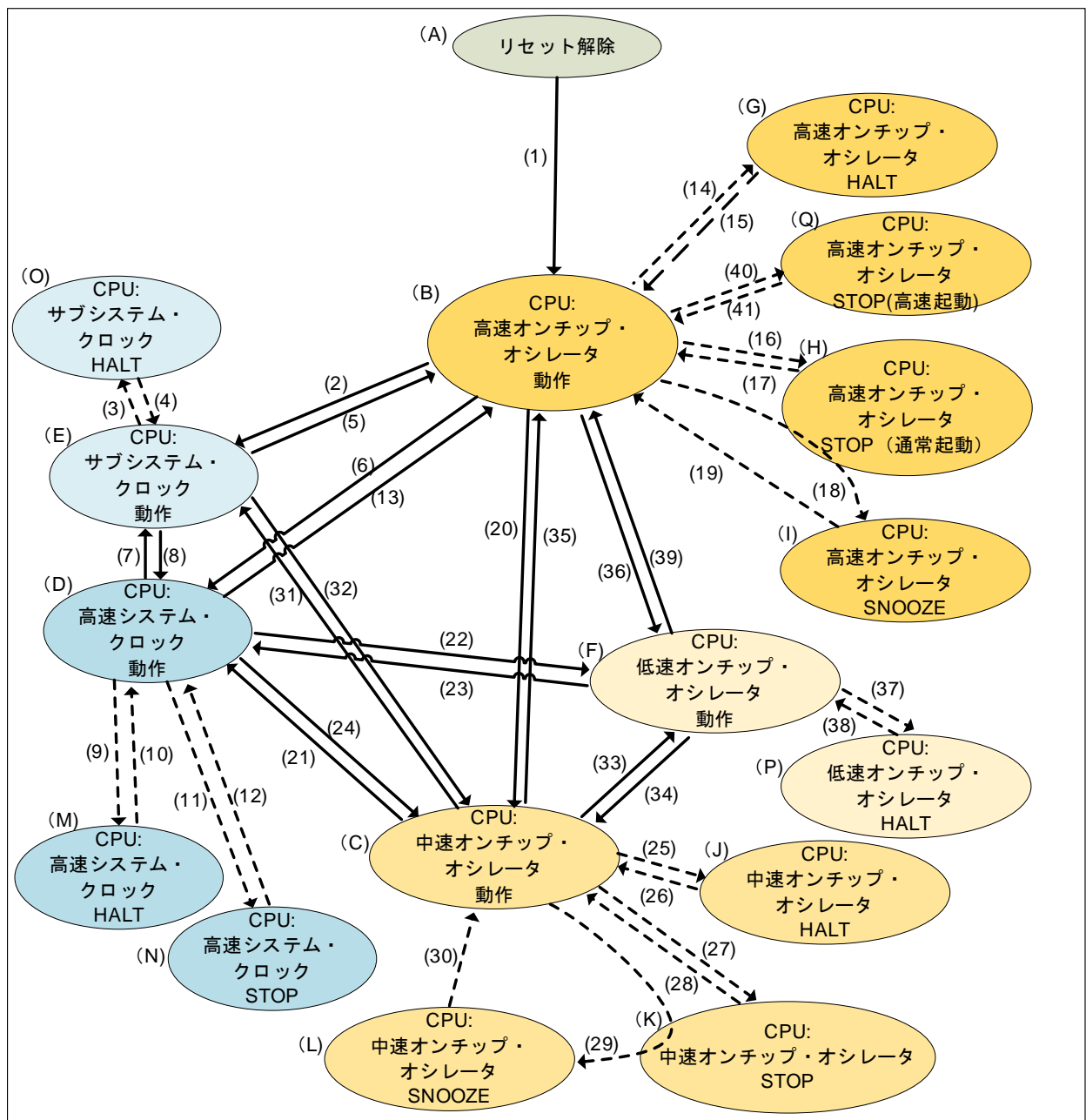
また、CPU クロックと動作モードの状態に合わせ、5つのLED 点灯を制御します。

使用する周辺機能と用途、動作モード状態移行図、動作モードとLED 表示の関係をそれぞれ表 1-1、図 1-1、表 1-2に示します。

表 1-1 使用する周辺機能と用途

周辺機能	用途
ポート出力	P03、P02、P43、P42、P77 端子に接続された LED (LED1 - LED5) の点灯制御
外部割り込み	スイッチ入力 (SW1) による端子入力エッジ検出の割り込み (INTP0)
32 ビット・インターバル・タイマ	32 ビット・インターバル・タイマのインターバル信号検出の割り込み (INTITL)
A/D コンバータ	P22 / ANI2 端子のアナログ信号入力レベルを変換。SNOOZE モードの確認に使用する
リアルタイム・クロック	定周期割り込み (INTRTC) を、A/D コンバータのハードウェア・トリガとして使用する

図 1-1 動作モード状態移行図



注意 実線はCPUクロックの状態遷移、破線はCPU動作モードの状態遷移を表します。

表 1-2 動作モードと LED 表示の関係

CPU/周辺ハードウェア・クロック (fCLK)	動作モード	LED 表示				
		LED1	LED2	LED3	LED4	LED5
高速オンチップ・オシレータ・ クロック (f <sub>IH</sub> )	通常動作モード	ON	ON	OFF	OFF	ON
	HALT モード	OFF	ON	OFF	OFF	ON
	SNOOZE モード	ON	OFF	OFF	OFF	ON
	STOP モード (通常起動)	OFF	OFF	OFF	OFF	ON
	STOP モード (高速起動)	OFF	OFF	OFF	ON	ON
中速オンチップ・オシレータ・ クロック (f <sub>IM</sub> )	通常動作モード	ON	ON	OFF	ON	OFF
	HALT モード	OFF	ON	OFF	ON	OFF
	SNOOZE モード	ON	OFF	OFF	ON	OFF
	STOP モード	OFF	OFF	OFF	ON	OFF
低速オンチップ・オシレータ・ クロック (f <sub>IL</sub> )	通常動作モード	ON	ON	OFF	ON	ON
	HALT モード	OFF	ON	OFF	ON	ON
高速システム・クロック (f <sub>MX</sub> )	通常動作モード	ON	ON	ON	OFF	ON
	HALT モード	OFF	ON	ON	OFF	ON
	STOP モード	OFF	OFF	ON	OFF	ON
サブ・クロック (f <sub>SX</sub> )	通常動作モード	ON	ON	ON	ON	OFF
	HALT モード	OFF	ON	ON	ON	OFF

注意 端子に流れる電流が 8mA 以下となるようにしてください。端子に流せる電流については、RL78/G23 ユーザーズマニュアルの電気的特性をご参照ください。

## 1.2 動作詳細

本アプリケーションノートでは、スイッチ入力により、CPUクロックと動作モードを切り替えます。CPUクロックと動作モードの切り替えは、図 1-1 の (1) から (4) の順に行います。

(1) 入出力ポートの初期設定を行います。

<設定条件>

- P03、P02、P43、P42、P77 端子: 出力ポートに設定 (LED 点灯制御で使用します)。

(2) クロック発生回路の初期設定を行います。

<設定条件>

- フラッシュの動作モードをHS (高速メイン) モードに設定します (ユーザ・オプション・バイト (000C2H / 040C2H) で設定します)。
- 高速オンチップ・オシレータ・クロックの周波数を32MHzに設定します。
- 中速オンチップ・オシレータ・クロックの周波数を4MHzに設定します。
- サブシステム・クロック端子の動作モードをXT1 発振モードに設定します。
- XT1 発振回路の発振モードを低消費発振1 (デフォルト) に設定します (接続する発振子に最適な発振モードを選択します)。
- 高速システム・クロック端子の動作モードをX1 発振モードに設定します。
- 周波数を20MHzに設定します。
- 発振安定時間を $2^{18}/f_x$ に設定します。
- CPU/周辺ハードウェア・クロック ( $f_{CLK}$ ) にメイン・システム・クロック ( $f_{MAIN}$ ) を選択します。

(3) 外部割り込み処理の初期設定を行います。

<設定条件>

- INTPO 端子の有効エッジを立ち上がりエッジに設定し、スイッチ入力を有効にします。
- 優先順位をレベル3に設定します。

(4) インターバル・タイマの初期設定を行います。

<設定条件>

- 32 ビット・カウンタ・モードに設定します。
- 動作クロックを  $f_{SXP}$  に設定します。
- クロック・ソースを  $f_{ITL0}$  に設定します。
- インターバル時間を 10ms に設定します。
- INTITL 割り込み発生を許可します。
- 優先順位をレベル3に設定します。



(5) リアルタイム・クロックの初期設定を行います。

<設定条件>

- クロック・ソースをサブシステム・クロック ( $f_{sXR}$ ) に設定します。
- 12 時間制に設定します。
- RTC1HZ 端子の出力を不許可に設定します。
- INTRTC 割り込みを許可します。
- 割り込み周期を 0.5 秒に設定します。
- 優先順位をレベル 3 に設定します。

(6) A/D コンバータの初期設定を行います。

<設定条件>

- コンパレータ動作を停止します。
- 10 ビット分解能を設定します。
- VREF (+) に VDD、VREF (-) に VSS を設定します。
- ハードウェア・トリガ・ウエイト・モードに設定します。
- ハードウェア・トリガをリアルタイム・クロック割り込み信号 (INTRTC) に設定します。
- ワンショット・セレクト・モードに設定します。
- A/D チャンネルは ANI2 を設定します。
- 変換時間モードを低電圧 1 に設定します。
- 変換時間を  $2816/f_{CLK}$  に設定します。
- $ADLL \leq ADCR_n \leq ADUL$  で割り込み要求信号 (INTAD) を発生に設定します。
- 変換結果上限値レジスタに 255、変換結果下限値レジスタに 240 を設定します。
- A/D 変換完了割り込み (INTAD) を許可します。
- 優先順位をレベル 3 に設定します。

(7) スイッチ押下による P137 / INTP0 端子の立ち上がりエッジを検出する毎に、CPU クロックと動作モードを切り替えます。ただし、SNOOZE モードからの復帰 (遷移 (19)、(30)) のみ、AD 変換完了割り込みによってこの状態へと移ります。

表 1-3、表 1-4 にスイッチ押下後の CPU クロックと動作モード、LED 表示を示します。

表 1-3 スイッチ押下後のLED 表示 (1/2)

	CPU クロック	動作モード	LED1	LED2	LED3	LED4	LED5
(1)	高速オンチップ・オシレータ・クロック	CPU 動作モード	ON	ON	OFF	OFF	ON
(2)	サブシステム・クロック	CPU 動作モード	ON	ON	ON	ON	OFF
(3)	サブシステム・クロック	HALT モード	OFF	ON	ON	ON	OFF
(4)	サブシステム・クロック	CPU 動作モード	ON	ON	ON	ON	OFF
(5)	高速オンチップ・オシレータ・クロック	CPU 動作モード	ON	ON	OFF	OFF	ON
(6)	高速システム・クロック	CPU 動作モード	ON	ON	ON	OFF	ON
(7)	サブシステム・クロック	CPU 動作モード	ON	ON	ON	ON	OFF
(8)	高速システム・クロック	CPU 動作モード	ON	ON	ON	OFF	ON
(9)	高速システム・クロック	HALT モード	OFF	ON	ON	OFF	ON
(10)	高速システム・クロック	CPU 動作モード	ON	ON	ON	OFF	ON
(11)	高速システム・クロック	STOP モード	OFF	OFF	ON	OFF	ON
(12)	高速システム・クロック	CPU 動作モード	ON	ON	ON	OFF	ON
(13)	高速オンチップ・オシレータ・クロック	CPU 動作モード	ON	ON	OFF	OFF	ON
(14)	高速オンチップ・オシレータ・クロック	HALT モード	OFF	ON	OFF	OFF	ON
(15)	高速オンチップ・オシレータ・クロック	CPU 動作モード	ON	ON	OFF	OFF	ON
(16)	高速オンチップ・オシレータ・クロック	STOP モード (通常起動)	OFF	OFF	OFF	OFF	ON
(17)	高速オンチップ・オシレータ・クロック	CPU 動作モード	ON	ON	OFF	OFF	ON
(18)	高速オンチップ・オシレータ・クロック	SNOOZE モード	ON	OFF	OFF	OFF	ON
(19)	高速オンチップ・オシレータ・クロック	CPU 動作モード	ON	ON	OFF	OFF	ON
(20)	中速オンチップ・オシレータ・クロック	CPU 動作モード	ON	ON	OFF	ON	OFF
(21)	高速システム・クロック	CPU 動作モード	ON	ON	ON	OFF	ON
(22)	低速オンチップ・オシレータ・クロック	CPU 動作モード	ON	ON	OFF	ON	ON
(23)	高速システム・クロック	CPU 動作モード	ON	ON	ON	OFF	ON
(24)	中速オンチップ・オシレータ・クロック	CPU 動作モード	ON	ON	OFF	ON	OFF

表 1-4 スイッチ押下後のLED表示 (2/2)

	CPUクロック	動作モード	LED1	LED2	LED3	LED4	LED5
(25)	中速オンチップ・オシレータ・クロック	HALTモード	OFF	ON	OFF	ON	OFF
(26)	中速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	ON	OFF
(27)	中速オンチップ・オシレータ・クロック	STOPモード	OFF	OFF	OFF	ON	OFF
(28)	中速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	ON	OFF
(29)	中速オンチップ・オシレータ・クロック	SNOOZEモード	ON	OFF	OFF	ON	OFF
(30)	中速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	ON	OFF
(31)	サブシステム・クロック	CPU動作モード	ON	ON	ON	ON	OFF
(32)	中速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	ON	OFF
(33)	低速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	ON	ON
(34)	中速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	ON	OFF
(35)	高速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	OFF	ON
(36)	低速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	ON	ON
(37)	低速オンチップ・オシレータ・クロック	HALTモード	OFF	ON	OFF	ON	ON
(38)	低速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	ON	ON
(39)	高速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	OFF	ON
(40)	高速オンチップ・オシレータ・クロック	STOPモード (高速起動)	OFF	OFF	OFF	ON	ON
(41)	高速オンチップ・オシレータ・クロック	CPU動作モード	ON	ON	OFF	OFF	ON

上記1から41までのCPUクロックと動作モード切り替え後、スイッチ押下によるP137 / INTP0端子の立ち下がりエッジを検出すると、LEDをすべて点灯し、HALTモードへ移行します (スタンバイ復帰はRESET入力のみです)。

注意 デバイス使用上の注意事項については、RL78/G23 ユーザーズマニュアル を参照してください。

### 1.3 CPU クロックの切り替え

次の CPU クロックの切り替えについて、特殊機能レジスタ (SFR) の設定手順を説明します。

- ・ 高速オンチップ・オシレータ・クロックから中速オンチップ・オシレータ・クロックへの切り替え
- ・ 高速オンチップ・オシレータ・クロックから低速オンチップ・オシレータ・クロックへの切り替え
- ・ 高速オンチップ・オシレータ・クロックから高速システム・クロックへの切り替え
- ・ 高速オンチップ・オシレータ・クロックからサブシステム・クロックへの切り替え
- ・ 中速オンチップ・オシレータ・クロックから高速オンチップ・オシレータ・クロックへの切り替え
- ・ 中速オンチップ・オシレータ・クロックから低速オンチップ・オシレータ・クロックへの切り替え
- ・ 中速オンチップ・オシレータ・クロックから高速システム・クロックへの切り替え
- ・ 中速オンチップ・オシレータ・クロックからサブシステム・クロックへの切り替え
- ・ 低速オンチップ・オシレータ・クロックから高速オンチップ・オシレータ・クロックへの切り替え
- ・ 低速オンチップ・オシレータ・クロックから中速オンチップ・オシレータ・クロックへの切り替え
- ・ 低速オンチップ・オシレータ・クロックから高速システム・クロックへの切り替え
- ・ 高速システム・クロックから高速オンチップ・オシレータ・クロックへの切り替え
- ・ 高速システム・クロックから中速オンチップ・オシレータ・クロックへの切り替え
- ・ 高速システム・クロックから低速オンチップ・オシレータ・クロックへの切り替え
- ・ 高速システム・クロックからサブシステム・クロックへの切り替え
- ・ サブシステム・クロックから高速オンチップ・オシレータ・クロックへの切り替え
- ・ サブシステム・クロックから中速オンチップ・オシレータ・クロックへの切り替え
- ・ サブシステム・クロックから高速システム・クロックへの切り替え

### 1.3.1 高速オンチップ・オシレータ・クロックから中速オンチップ・オシレータ・クロック への切り替え

CPU クロックを高速オンチップ・オシレータ・クロックから中速オンチップ・オシレータ・クロックに変更する場合、クロック動作ステータス制御レジスタ (CSC) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で中速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。

メイン・オンチップ・オシレータ・クロックのステータスが中速オンチップ・オシレータ・クロックに切り替わったのを確認し、高速オンチップ・オシレータを停止します。

- ① CSC レジスタの MIOEN ビットをセット (1) して中速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	1	x	0	0	0	0	1	0

- ② 中速オンチップ・オシレータの発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。

- ③ CKC レジスタの MCM0 ビットをクリア (0)、MCM1 ビットをセット (1) して中速オンチップ・オシレータ・クロックをメイン・オンチップ・オシレータ・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	0	0	0	0	0	1

- ④ CKC レジスタの MCS ビットが 0、MCS1 ビットが 1 になったことを確認して、HIOSTOP をセット (1) し、高速オンチップ・オシレータを停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	1	x	0	0	0	0	1	1

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

### 1.3.2 高速オンチップ・オシレータ・クロックから低速オンチップ・オシレータ・クロックへの切り替え

CPU クロックを高速オンチップ・オシレータ・クロックから低速オンチップ・オシレータ・クロックに変更する場合、サブシステム・クロック選択レジスタ (CKSEL) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で低速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。CPU/周辺ハードウェア・クロックのステータスがサブシステム・クロックに切り替わったのを確認し、高速オンチップ・オシレータを停止します。

- ① CKC レジスタの CLS ビットが 0 であることを確認して、CKSEL レジスタの SELLOSC ビットをセット (1) して低速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	0	0	SELLOSC
	0	0	0	0	0	0	0	1

- ② 低速オンチップ・オシレータの発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。

- ③ CKC レジスタの CSS ビットをセット (1) してサブシステム・クロックを CPU/周辺ハードウェア・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、元の値を保持するようにします。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	1	0	0	0	0	0	0

- ④ CKC レジスタの CLS ビットが 1 になったことを確認し、HIOSTOP をセット (1) し、高速オンチップ・オシレータを停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	1	1	0	0	0	0	0	1

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

## 1.3.3 高速オンチップ・オシレータ・クロックから高速システム・クロックへの切り替え

CPU クロックを高速オンチップ・オシレータ・クロックから高速システム・クロックに変更する場合、クロック動作モード制御レジスタ (CMC)、発振安定時間選択レジスタ (OSTS)、クロック動作ステータス制御レジスタ (CSC) で発振回路の設定と発振開始を行います。次に、発振安定時間カウンタ状態レジスタ (OSTC) で発振の安定待ちを行います。

発振安定時間経過後、システム・クロック制御レジスタ (CKC) で高速システム・クロックを  $f_{CLK}$  に設定します。

メイン・システム・クロックのステータスが高速システム・クロックに切り替わったのを確認し、高速オンチップ・オシレータを停止します。

CMC レジスタの OSCSEL ビットをセット (1) し、 $f_x > 10\text{MHz}$  の場合は AMPH ビットをセット (1) して X1 発振回路を動作させます。30~36 ピン製品では XTSEL ビットをクリア (0) してください。外部クロックを使用される場合は EXCLK ビットと OSCSEL ビットをセット (1) してください。なお、このレジスタはリセット後、8 ビット・メモリ操作命令で一度しか書き込むことができません。

## 【X1 発振モードの場合】

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	XTSEL	AMPHS1	AMPHS0	AMPH
	0	1	x	x	x	x	x	0/1

AMPH ビット : X1 発振クロックが 10 MHz 以下の場合は 0 を設定してください。

## 【外部クロック入力モードの場合】

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	XTSEL	AMPHS1	AMPHS0	AMPH
	1	1	x	x	x	x	x	x

- ① OSTS レジスタで X1 発振回路の発振安定時間を選択します。外部クロックでは設定不要です。

例) 10MHz の発振子で 102 $\mu\text{s}$  以上までウェイトする場合は、以下の値に設定してください。

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0
	0	0	0	0	0	0	1	0

- ② CSC レジスタの MSTOP ビットをクリア (0) して X1 発振回路の発振を開始します。外部クロック使用時は、MSTOP ビットをクリア (0) する前に外部クロック信号を入力してください。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	0	x	0	0	0	0	0	0

- ③ OSTC レジスタで X1 発振回路の発振安定待ちを行います。外部クロックでは発振安定待ちは不要です。

例) 10MHz の発振子で 102 $\mu\text{s}$  以上までウェイトする場合は、以下の値になるまでウェイトしてください。

	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
	1	1	1	0	0	0	0	0

- ④ CKC レジスタの MCM0 ビットをセット (1) して高速システム・クロックをメイン・システム・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	0	1	0	0	0	0

- ⑤ CKC レジスタの MCS ビットが 1 になったことを確認して、HIOSTOP をセット (1) し、高速オンチップ・オシレータを停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	0	x	0	0	0	0	0	1

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット



## 1.3.4 高速オンチップ・オシレータ・クロックからサブシステム・クロックへの切り替え

CPU クロックを高速オンチップ・オシレータ・クロックからサブシステム・クロックに変更する場合、サブシステム・クロック供給モード制御レジスタ (OSMC)、クロック動作モード制御レジスタ (CMC)、クロック動作ステータス制御レジスタ (CSC) で発振回路の設定と発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) でサブシステム・クロックを  $f_{CLK}$  に設定します。CPU/周辺ハードウェア・クロックのステータスがサブシステム・クロックに切り替わったのを確認し、高速オンチップ・オシレータを停止します。

- ① 本アプリケーションノートでは、サブシステム・クロック用発振子の発振安定時間を 32 ビット・インターバル・タイマでカウントします。32 ビット・インターバル・タイマのカウント・クロックに低速オンチップ・オシレータ・クロックを使用するため、WUTMMCK0 ビットをセット (1) します。また、STOP モード時または HALT モード時 (サブシステム・クロックで CPU 動作中の場合) に周辺機能をサブシステム・クロックで動作させる場合は RTCLPC ビットをクリア (0) します。なお、STOP 命令を実行する場合には、HIPREC ビットがセット (1) されていることを確認してください。

	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	-	-	0	HIPREC
	0	0	0	1	x	x	0	x

- ② CMC レジスタの OSCSELS ビットをセット (1) して XT 発振回路を動作させます。30~36 ピン製品では XTSEL ビットをセット (1) してください。低消費発振 1 (デフォルト) で動作させる場合は AMPHS1 ビットと AMPHS0 ビットをクリア (0) します。また、外部クロックを使用される場合は EXCLKS ビットと OSCSELS ビットをセット (1) してください。なお、このレジスタはリセット後、8 ビット・メモリ操作命令で一度しか書き込むことができません。

【XT1 発振モードの場合】

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	XTSEL	AMPHS1	AMPHS0	AMPH
	x	x	0	1		0/1	0/1	x

【外部クロック入力モードの場合】

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	XTSEL	AMPHS1	AMPHS0	AMPH
	x	x	1	1		x	x	x

- ③ CSC レジスタの XTSTOP ビットをクリア (0) して XT1 発振回路の発振を開始します。外部クロック使用時は、XTSTOP ビットをクリア (0) する前に外部クロック信号を入力してください。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	1	0	0	0	0	0	0	0

- ④ サブシステム・クロック用発振子の発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。外部クロックでは発振安定待ちは不要です。

- ⑤ CKC レジスタの CSS ビットをセット (1) してサブシステム・クロックを CPU/周辺ハードウェア・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、元の値を保持するようにします。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	1	0	0	0	0	0	0

- ⑥ CKC レジスタの CLS ビットが 1 になったことを確認して、HIOSTOP をセット (1) し、高速オンチップ・オシレータを停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	0	0	0	0	0	0	1

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

### 1.3.5 中速オンチップ・オシレータ・クロックから高速オンチップ・オシレータ・クロックへの切り替え

CPU クロックを中速オンチップ・オシレータ・クロックから高速オンチップ・オシレータ・クロックに変更する場合、クロック動作ステータス制御レジスタ (CSC) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で高速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。

メイン・オンチップ・オシレータ・クロックのステータスが高速オンチップ・オシレータ・クロックに切り替わったのを確認し、中速オンチップ・オシレータを停止します。

- ① CSC レジスタの HIOSTOP ビットをクリア (0) して高速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	x	0	0	0	0	1	0

- ② 高速オンチップ・オシレータの発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。

- ③ CKC レジスタの MCM0 ビットと MCM1 ビットをクリア (0) して高速オンチップ・オシレータ・クロックをメイン・オンチップ・オシレータ・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	0	0	0	0	1	0

- ④ CKC レジスタの MCS ビットと MCS1 ビットが 0 になったことを確認して、MIOEN をクリア (0) し、中速オンチップ・オシレータを停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	x	0	0	0	0	0	0

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

### 1.3.6 中速オンチップ・オシレータ・クロックから低速オンチップ・オシレータ・クロックへの切り替え

CPU クロックを中速オンチップ・オシレータ・クロックから低速オンチップ・オシレータ・クロックに変更する場合、サブシステム・クロック選択レジスタ (CKSEL) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で低速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。CPU/周辺ハードウェア・クロックのステータスがサブシステム・クロックに切り替わったのを確認し、中速オンチップ・オシレータを停止します。

- ① CKC レジスタの CLS ビットが 0 であることを確認して、CKSEL レジスタの SELLOSC ビットをセット (1) して低速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	0	0	SELLOSC
	0	0	0	0	0	0	0	1

- ② 低速オンチップ・オシレータの発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。

- ③ CKC レジスタの CSS ビットをセット (1) してサブシステム・クロックを CPU/周辺ハードウェア・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、元の値を保持するようにします。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	1	0	0	0	0	1	1

- ④ CKC レジスタの CLS ビットが 1 になったことを確認して、MIOEN をクリア (0) し、中速オンチップ・オシレータを停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	x	0	0	0	0	0	1

#### レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

## 1.3.7 中速オンチップ・オシレータ・クロックから高速システム・クロックへの切り替え

CPU クロックを中速オンチップ・オシレータ・クロックから高速システム・クロックに変更する場合、クロック動作モード制御レジスタ (CMC)、発振安定時間選択レジスタ (OSTS)、クロック動作ステータス制御レジスタ (CSC) で発振回路の設定と発振開始を行います。次に、発振安定時間カウンタ状態レジスタ (OSTC) で発振の安定待ちを行います。

発振安定時間経過後、システム・クロック制御レジスタ (CKC) で高速システム・クロックを  $f_{CLK}$  に設定します。

メイン・システム・クロックのステータスが高速システム・クロックに切り替わったのを確認し、中速オンチップ・オシレータを停止します。

- ① CMC レジスタの OSCSEL ビットをセット (1) し、 $f_x > 10\text{MHz}$  の場合は AMPH ビットをセット (1) して X1 発振回路を動作させます。30~36 ピン製品では XTSEL ビットをクリア (0) してください。外部クロックを使用される場合は EXCLK ビットと OSCSEL ビットをセット (1) してください。なお、このレジスタはリセット後、8 ビット・メモリ操作命令で一度しか書き込むことができません。

## 【X1 発振モードの場合】

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	XTSEL	AMPHS1	AMPHS0	AMPH
	0	1	x	x	x	x	x	0/1

AMPH ビット : X1 発振クロックが 10 MHz 以下の場合は 0 を設定してください。

## 【外部クロック入力モードの場合】

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	XTSEL	AMPHS1	AMPHS0	AMPH
	1	1	x	x	x	x	x	x

- ② OSTS レジスタで X1 発振回路の発振安定時間を選択します。外部クロックでは設定不要です。

例) 10MHz の発振子で 102 $\mu\text{s}$  以上までウェイトする場合は、以下の値に設定してください。

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0
	0	0	0	0	0	0	1	0

- ③ CSC レジスタの MSTOP ビットをクリア (0) して X1 発振回路の発振を開始します。外部クロック使用時は、MSTOP ビットをクリア (0) する前に外部クロック信号を入力してください。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	0	x	0	0	0	0	1	1

- ④ OSTC レジスタで X1 発振回路の発振安定待ちを行います。外部クロックでは発振安定待ちは不要です。

例) 10MHz の発振子で 102 $\mu$ s 以上までウエイトする場合は、以下の値になるまでウエイトしてください。

	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
	1	1	1	0	0	0	0	0

- ⑤ CKC レジスタの MCM0 ビットをセット (1) して高速システム・クロックをメイン・システム・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	0	1	0	0	1	1

- ⑥ CKC レジスタの MCS ビットが 1 になったことを確認して、MIOEN をクリア (0) し、中速オンチップ・オシレータを停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	0	x	0	0	0	0	0	1

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

## 1.3.8 中速オンチップ・オシレータ・クロックからサブシステム・クロックへの切り替え

CPU クロックを中速オンチップ・オシレータ・クロックからサブシステム・クロックに変更する場合、サブシステム・クロック供給モード制御レジスタ (OSMC)、クロック動作モード制御レジスタ (CMC)、クロック動作ステータス制御レジスタ (CSC) で発振回路の設定と発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) でサブシステム・クロックを  $f_{CLK}$  に設定します。CPU/周辺ハードウェア・クロックのステータスがサブシステム・クロックに切り替わったのを確認し、中速オンチップ・オシレータを停止します。

- ① 本アプリケーションノートでは、サブシステム・クロック用発振子の発振安定時間を 32 ビット・インターバル・タイマでカウントします。32 ビット・インターバル・タイマのカウント・クロックに低速オンチップ・オシレータ・クロックを使用するため、WUTMMCK0 ビットをセット (1) します。また、STOP モード時または HALT モード時 (サブシステム・クロックで CPU 動作中の場合) に、周辺機能をサブシステム・クロックで動作させる場合は、RTCLPC ビットをクリア (0) します。なお、STOP 命令を実行する場合には、HIPREC ビットがセット (1) されていることを確認してください。

	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	-	-	0	HIPREC
	0	0	0	1	x	x	0	x

- ② CMC レジスタの OSCSELS ビットをセット (1) して XT 発振回路を動作させます。30~36 ピン製品では XTSEL ビットをセット (1) してください。低消費発振 1 (デフォルト) で動作させる場合は AMPHS1 ビットと、AMPHS0 ビットをクリア (0) します。また、外部クロックを使用される場合は EXCLKS ビットと OSCSELS ビットをセット (1) してください。なお、このレジスタはリセット後、8 ビット・メモリ操作命令で一度しか書き込むことができません。

【XT1 発振モードの場合】

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	XTSEL	AMPHS1	AMPHS0	AMPH
	x	x	0	1	x	0/1	0/1	x

【外部クロック入力モードの場合】

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	XTSEL	AMPHS1	AMPHS0	AMPH
	x	x	1	1	x	x	x	x

- ③ CSC レジスタの XTSTOP ビットをクリア (0) して XT1 発振回路の発振を開始します。外部クロック使用時は、XTSTOP ビットをクリア (0) する前に外部クロック信号を入力してください。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	0	0	0	0	0	1	0

- ④ サブシステム・クロック用発振子の発振が安定するまでソフトウェアでウエイトします。ウエイト時間 (発振安定時間) はタイマ機能などを用いてカウントします。外部クロックでは発振安定待ちは不要です。

- ⑤ CKC レジスタの CSS ビットをセット (1) してサブシステム・クロックを CPU/周辺ハードウェア・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、元の値を保持するようにします。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	1	0	0	0	0	1	1

- ⑥ CKC レジスタの CLS ビットが 1 になったことを確認して、MIOEN をクリア (0) し、中速オンチップ・オシレータを停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	0	0	0	0	0	0	0

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット



### 1.3.9 低速オンチップ・オシレータ・クロックから高速オンチップ・オシレータ・クロックへの切り替え

CPU クロックを低速オンチップ・オシレータ・クロックから高速オンチップ・オシレータ・クロックに変更する場合、クロック動作ステータス制御レジスタ (CSC) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で高速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。

CPU/周辺ハードウェア・クロックのステータスがメイン・システム・クロックに切り替わったのを確認し、低速オンチップ・オシレータを停止します。

- ① CSC レジスタの HIOSTOP ビットをクリア (0) して高速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	x	0	0	0	0	0	0

- ② 高速オンチップ・オシレータの発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。CKC レジスタの MCM0 ビットと MCM1 ビットをクリア (0) して高速オンチップ・オシレータ・クロックをメイン・オンチップ・オシレータ・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	1	0	x	0	0	0	x	0

- ③ CKC レジスタの CLS ビットが 0 であることを確認して、サブシステム・クロック選択レジスタ (CKSEL) の SELLOSC をクリア (0) し、低速オンチップ・オシレータを停止します。本アプリケーションノートでは、サブシステム・クロック供給モード制御レジスタ (OSMC) の WUTMMCK0 ビットが 1 となっている為、低速オンチップ・オシレータは停止しません。

	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	0	0	SELLOSC
	0	0	0	0	0	0	0	0

注意 CPU/周辺ハードウェア・クロックがサブシステム・クロックで動作中にMCM0ビット及びMCM1ビットの値を変更することは禁止です。

#### レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

### 1.3.10 低速オンチップ・オシレータ・クロックから中速オンチップ・オシレータ・クロックへの切り替え

CPU クロックを低速オンチップ・オシレータ・クロックから中速オンチップ・オシレータ・クロックに変更する場合、クロック動作ステータス制御レジスタ (CSC) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で中速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。

CPU/周辺ハードウェア・クロックのステータスがメイン・システム・クロックに切り替わったのを確認し、低速オンチップ・オシレータを停止します。

- ① CSC レジスタの MIOEN ビットをセット (1) して中速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	1	1	0	0	0	0	1	1

- ② 中速オンチップ・オシレータの発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。

- ③ CKC レジスタの MCM0 ビットをクリア (0)、MCM1 ビットをセット (1) して中速オンチップ・オシレータ・クロックをメイン・オンチップ・オシレータ・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	1	0	x	0	0	0	x	1

- ④ CKC レジスタの CLS ビットが 0 であることを確認して、サブシステム・クロック選択レジスタ (CKSEL) の SELLOSC をクリア (0) し、低速オンチップ・オシレータを停止します。本アプリケーションノートでは、サブシステム・クロック供給モード制御レジスタ (OSMC) の WUTMMCK0 ビットが 1 となっている為、低速オンチップ・オシレータは停止しません。

	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	0	0	SELLOSC
	0	0	0	0	0	0	0	0

注意 CPU/周辺ハードウェア・クロックがサブシステム・クロックで動作中に MCM0 ビット及び MCM1 ビットの値を変更することは禁止です。

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

## 1.3.11 低速オンチップ・オシレータ・クロックから高速システム・クロックへの切り替え

CPU クロックを低速オンチップ・オシレータ・クロックから高速システム・クロックに変更する場合、クロック動作モード制御レジスタ (CMC)、発振安定時間選択レジスタ (OSTS)、クロック動作ステータス制御レジスタ (CSC) で発振回路の設定と発振開始を行います。次に、発振安定時間カウンタ状態レジスタ (OSTC) で発振の安定待ちを行います。

発振安定時間経過後、システム・クロック制御レジスタ (CKC) で高速システム・クロックを  $f_{CLK}$  に設定します。

CPU/周辺ハードウェア・クロックのステータスがメイン・システム・クロックに切り替わったのを確認し、低速オンチップ・オシレータを停止します。

- ① CMC レジスタの OSCSEL ビットをセット (1) し、 $f_x > 10\text{MHz}$  の場合は AMPH ビットをセット (1) して X1 発振回路を動作させます。30~36 ピン製品では XTSEL ビットをクリア (0) してください。外部クロックを使用される場合は EXCLK ビットと OSCSEL ビットをセット (1) してください。なお、このレジスタはリセット後、8 ビット・メモリ操作命令で一度しか書き込むことができません。

【X1 発振モードの場合】

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	XTSEL	AMPHS1	AMPHS0	AMPH
	0	1	x	x	x	x	x	0/1

AMPH ビット : X1 発振クロックが 10 MHz 以下の場合は 0 を設定してください。

【外部クロック入力モードの場合】

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	XTSEL	AMPHS1	AMPHS0	AMPH
	1	1	x	x	x	x	x	x

- ② OSTS レジスタで X1 発振回路の発振安定時間を選択します。外部クロックでは設定不要です。

例) 10MHz の発振子で 102 $\mu\text{s}$  以上までウェイトする場合は、以下の値に設定してください。

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0
	0	0	0	0	0	0	1	0

- ③ CSC レジスタの MSTOP ビットをクリア (0) して X1 発振回路の発振を開始します。外部クロック使用時は、MSTOP ビットをクリア (0) する前に外部クロック信号を入力してください。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	0	x	0	0	0	0	0	1

- ④ OSTC レジスタで X1 発振回路の発振安定待ちを行います。外部クロックでは発振安定待ちは不要です。

例) 10MHz の発振子で 102 $\mu\text{s}$  以上までウェイトする場合は、以下の値になるまでウェイトしてください。

	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
	1	1	1	0	0	0	0	0

- ⑤ CKC レジスタの MCM0 ビットをセット (1) して高速システム・クロックをメイン・システム・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	1	0	×	1	0	0	×	×

- ⑥ CKC レジスタの CLS ビットが 0 であることを確認して、サブシステム・クロック選択レジスタ (CKSEL) の SELLOSC をクリア (0) し、低速オンチップ・オシレータを停止します。本アプリケーションノートでは、サブシステム・クロック供給モード制御レジスタ (OSMC) の WUTMMCK0 ビットが 1 となっている為、低速オンチップ・オシレータは停止しません。

	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	0	0	SELLOSC
	0	0	0	0	0	0	0	0

注意 CPU/周辺ハードウェア・クロックがサブシステム・クロックで動作中にMCM0ビット及びMCM1ビットの値を変更することは禁止です。

レジスタ図の設定値

×：使用しないビット、空白：変更しないビット、－：予約ビットまたは、何も配置されていないビット

## 1.3.12 高速システム・クロックから高速オンチップ・オシレータ・クロックへの切り替え

CPU クロックを高速オンチップ・オシレータ・クロックから高速システム・クロックに変更する場合、クロック動作ステータス制御レジスタ (CSC) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で高速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。

メイン・システム・クロックのステータスがメイン・オンチップ・オシレータ・クロックに切り替わったのを確認し、X1 発振回路を停止します。

- ① CSC レジスタの HIOSTOP ビットをクリア (0) して高速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	0	x	0	0	0	0	0	0

- ② 高速オンチップ・オシレータの発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。

- ③ CKC レジスタの MCM0 ビットと MCM1 ビットをクリア (0) して高速オンチップ・オシレータ・クロックをメイン・オンチップ・オシレータ・クロックに設定します。CSS ビットが1のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは0に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	1	0	0	0	x	0

- ④ CKC レジスタの MCS ビットと MCS1 ビットが 0 になったことを確認して、MSTOP をセット (1) し、X1 発振回路を停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	1	x	0	0	0	0	0	0

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

## 1.3.13 高速システム・クロックから中速オンチップ・オシレータ・クロックへの切り替え

CPU クロックを高速システム・クロックから中速オンチップ・オシレータ・クロックに変更する場合、クロック動作ステータス制御レジスタ (CSC) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で中速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。

メイン・システム・クロックのステータスがメイン・オンチップ・オシレータ・クロックに切り替わったのを確認し、X1 発振回路を停止します。

- ① CSC レジスタの MIOEN ビットをセット (1) して中速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	0	x	0	0	0	0	1	1

- ② 中速オンチップ・オシレータの発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。

- ③ CKC レジスタの MCM0 ビットと MCM1 ビットをクリア (0) して高速オンチップ・オシレータ・クロックをメイン・オンチップ・オシレータ・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	1	0	0	0	x	0

- ④ CKC レジスタの MCS ビットと MCS1 ビットが 0 になったことを確認して、MSTOP をセット (1) し、X1 発振回路を停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	1	x	0	0	0	0	1	1

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

## 1.3.14 高速システム・クロックから低速オンチップ・オシレータ・クロックへの切り替え

CPU クロックを高速システム・クロックから低速オンチップ・オシレータ・クロックに変更する場合、サブシステム・クロック選択レジスタ (CKSEL) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で低速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。CPU/周辺ハードウェア・クロックのステータスがサブシステム・クロックに切り替わったのを確認し、X1 発振回路を停止します。

- ① CKC レジスタの CLS ビットが 0 であることを確認して、CKSEL レジスタの SELLOSC ビットをセット (1) して低速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	0	0	SELLOSC
	0	0	0	0	0	0	0	1

- ② 低速オンチップ・オシレータの発振が安定するまでソフトウェアでウエイトします。ウエイト時間はタイマ機能などを用いてカウントします。
- ③ CKC レジスタの CSS ビットをセット (1) してサブシステム・クロックを CPU/周辺ハードウェア・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、元の値を保持するようにします。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	1	1	1	0	0	×	×

- ④ CKC レジスタの CLS ビットが 1 になったことを確認して、MSTOP をセット (1) し、X1 発振回路を停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	1	1	0	0	0	0	0	1

レジスタ図の設定値

×: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

## 1.3.15 高速システム・クロックからサブシステム・クロックへの切り替え

CPU クロックを高速システム・クロックからサブシステム・クロックに変更する場合、サブシステム・クロック供給モード制御レジスタ (OSMC)、クロック動作モード制御レジスタ (CMC)、クロック動作ステータス制御レジスタ (CSC) で発振回路の設定と発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) でサブシステム・クロックを f<sub>CLK</sub> に設定します。CPU/周辺ハードウェア・クロックのステータスがサブシステム・クロックに切り替わったのを確認し、X1 発振回路を停止します。

なお、30~36 ピン製品では、X1 端子と XT1 端子、X2 端子と XT2 端子はそれぞれ兼用となります。このため、高速システム・クロックからサブシステム・クロックへの切り替えは、40~128 ピン製品のみ可能です。

- ① 本アプリケーションノートでは、サブシステム・クロック用発振子の発振安定時間を 32 ビット・インターバル・タイマでカウントします。32 ビット・インターバル・タイマのカウント・クロックに低速オンチップ・オシレータ・クロックを使用するため、WUTMMCK0 ビットをセット (1) します。また、STOP モード時または HALT モード時 (サブシステム・クロックで CPU 動作中の場合) に周辺機能をサブシステム・クロックで動作させる場合は、RTCLPC ビットをクリア (0) します。なお、STOP 命令を実行する場合には、HIPREC ビットがセット (1) されていることを確認してください。

	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	-	-	0	HIPREC
	0	0	0	1	x	x	0	x

- ② CSC レジスタの XTSTOP ビットをクリア (0) して XT1 発振回路の発振を開始します。外部クロック使用時は、XTSTOP ビットをクリア (0) する前に外部クロック信号を入力してください。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	0	0	0	0	0	0	0	1

- ③ サブシステム・クロック用発振子の発振が安定するまでソフトウェアでウエイトします。ウエイト時間 (発振安定時間) はタイマ機能などを用いてカウントします。外部クロックでは発振安定待ちは不要です。

- ④ CKC レジスタの CSS ビットをセット (1) してサブシステム・クロックを CPU/周辺ハードウェア・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、元の値を保持するようにします。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	1	1	1	0	0	x	x

- ⑤ CKC レジスタの CLS ビットが 1 になったことを確認して、MSTOP をセット (1) し、X1 発振回路を停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	1	1	0	0	0	0	0	1



## レジスタ図の設定値

x : 使用しないビット、空白 : 変更しないビット、- : 予約ビットまたは、何も配置されていないビット

## 1.3.16 サブシステム・クロックから高速オンチップ・オシレータ・クロックへの切り替え

CPU クロックをサブシステム・クロックから高速オンチップ・オシレータ・クロックに変更する場合、クロック動作ステータス制御レジスタ (CSC) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で高速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。

CPU/周辺ハードウェア・クロックのステータスがメイン・システム・クロックに切り替わったのを確認し、XT1 発振回路を停止します。

- ① CSC レジスタの HIOSTOP ビットをクリア (0) して高速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	0	0	0	0	0	1	0

- ② 高速オンチップ・オシレータの発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。CKC レジスタの MCM0 ビットと MCM1 ビットをクリア (0) して高速オンチップ・オシレータ・クロックをメイン・オンチップ・オシレータ・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	1	0	x	0	0	0	x	0

- ③ CKC レジスタの CLS ビットが 0 であることを確認して、XTSTOP をセット (1) し、XT1 発振回路を停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	1	0	0	0	0	1	0

レジスタ図の設定値

x : 使用しないビット、空白 : 変更しないビット、- : 予約ビットまたは、何も配置されていないビット

## 1.3.17 サブシステム・クロックから中速オンチップ・オシレータ・クロックへの切り替え

CPU クロックをサブシステム・クロックから中速オンチップ・オシレータ・クロックに変更する場合、クロック動作ステータス制御レジスタ (CSC) で発振開始を行います。次に、タイマ等で発振の安定待ちを行います。発振安定時間経過後、システム・クロック制御レジスタ (CKC) で中速オンチップ・オシレータ・クロックを  $f_{CLK}$  に設定します。

CPU/周辺ハードウェア・クロックのステータスがメイン・システム・クロックに切り替わったのを確認し、XT1 発振回路を停止します。

- ① CSC レジスタの MIOEN ビットをセット (1) して中速オンチップ・オシレータを動作させます。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	0	0	0	0	0	1	1

- ② 中速オンチップ・オシレータの発振が安定するまでソフトウェアでウェイトします。ウェイト時間はタイマ機能などを用いてカウントします。

- ③ CKC レジスタの MCM0 ビットをクリア (0)、MCM1 ビットをセット (1) して中速オンチップ・オシレータ・クロックをメイン・オンチップ・オシレータ・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	1	0	x	0	0	0	x	1

- ④ CKC レジスタの CLS ビットが 0 であることを確認して、XTSTOP をセット (1) し、XT1 発振回路を停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	1	0	0	0	0	1	1

レジスタ図の設定値

x: 使用しないビット、空白: 変更しないビット、-: 予約ビットまたは、何も配置されていないビット

## 1.3.18 サブシステム・クロックから高速システム・クロックへの切り替え

CPU クロックをサブシステム・クロックから高速システム・クロックに変更する場合、クロック動作モード制御レジスタ (CMC)、発振安定時間選択レジスタ (OSTS)、クロック動作ステータス制御レジスタ (CSC) で発振回路の設定と発振開始を行います。次に、発振安定時間カウンタ状態レジスタ (OSTC) で発振の安定待ちを行います。

発振安定時間経過後、システム・クロック制御レジスタ (CKC) で高速システム・クロックを fCfCLKLK に設定します。

CPU/周辺ハードウェア・クロックのステータスがメイン・システム・クロックに切り替わったのを確認し、XT1 発振回路を停止します。

なお、30~36 ピン製品では、X1 端子と XT1 端子、X2 端子と XT2 端子はそれぞれ兼用となります。このため、サブシステム・クロックから高速システム・クロックへの切り替えは、40~128 ピン製品のみ可能です。

- ① OSTS レジスタで X1 発振回路の発振安定時間を選択します。外部クロックでは設定不要です。

例) 10MHz の発振子で 102 $\mu$ s 以上までウェイトする場合は、以下の値に設定してください。

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0
	0	0	0	0	0	0	1	0

- ② CSC レジスタの MSTOP ビットをクリア (0) して X1 発振回路の発振を開始します。外部クロック使用時は、MSTOP ビットをクリア (0) する前に外部クロック信号を入力してください。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	1	0	0	0	0	0	x	x

- ③ OSTC レジスタで X1 発振回路の発振安定待ちを行います。外部クロックでは発振安定待ちは不要です。

例) 10MHz の発振子で 102 $\mu$ s 以上までウェイトする場合は、以下の値になるまでウェイトしてください。

	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
	1	1	1	0	0	0	0	0

- ④ CKC レジスタの MCM0 ビットをセット (1) して高速システム・クロックをメイン・システム・クロックに設定します。CSS ビットが 1 のとき MCM0 ビットと MCM1 ビットは変更禁止ですので、CSS ビットは 0 に設定します。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	1	1	0	0	x	x

- ⑤ CKC レジスタの CLS ビットが 0 になったことを確認して、XTSTOP をセット (1) し、XT1 発振回路を停止します。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	0	1	0	0	0	0	x	x

注意 CPU/周辺ハードウェア・クロックがサブシステム・クロックで動作中にMCM0ビット及びMCM1ビットの値を変更することは禁止です。

レジスタ図の設定値

x：使用しないビット、空白：変更しないビット、－：予約ビットまたは、何も配置されていないビット

## 2. 動作確認条件

本アプリケーションノートのサンプルコードは、表 2-1の条件で動作を確認しています。

表 2-1 動作確認条件

周辺機能	用途
使用マイコン	RL78/G23 (R7F100GLG)
使用ボード	RL78/G23-64p Fast Prototyping Board (RTK7RLG230CLG000BJ)
動作周波数	高速オンチップ・オシレータ・クロック ( $f_{IH}$ ) : 32MHz 中速オンチップ・オシレータ・クロック ( $f_{IM}$ ) : 4MHz 低速オンチップ・オシレータ・クロック ( $f_{IL}$ ) : 32.768kHz 高速システム・クロック (X1クロック ( $f_x$ )) : 20MHz サブシステム・クロック (XT1クロック ( $f_{XT}$ )) : 32.768kHz CPU/周辺ハードウェア・クロック : 32MHz、20MHz、4MHz、32.768kHz
動作電圧	5.0V (2.0V~5.5V で動作可能) LVD0 検出電圧: リセット・モード 立ち上がり時 TYP. 1.90V (1.84 V ~ 1.95 V) 立ち下がり時 TYP. 1.86V (1.80 V ~ 1.91 V)
統合開発環境 (CS+)	ルネサス エレクトロニクス製 CS+ for CC V8.09.00
C コンパイラ (CS+)	ルネサス エレクトロニクス製 CC-RL V1.12.00
統合開発環境 (e2studio)	ルネサス エレクトロニクス製 e2studio V2023-04 (23.4.0)
C コンパイラ (e2studio)	ルネサス エレクトロニクス製 CC-RL V1.12.00
統合開発環境 (IAR)	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 V4.21.2
C コンパイラ (IAR)	IAR Systems 製 IAR C/C++ Compiler for Renesas RL78 V4.21.2.2420
スマート・コンフィグ レータ (SC)	ルネサス エレクトロニクス製 V1.6.0
ボードサポートパッケージ (BSP)	ルネサス エレクトロニクス製 V1.60

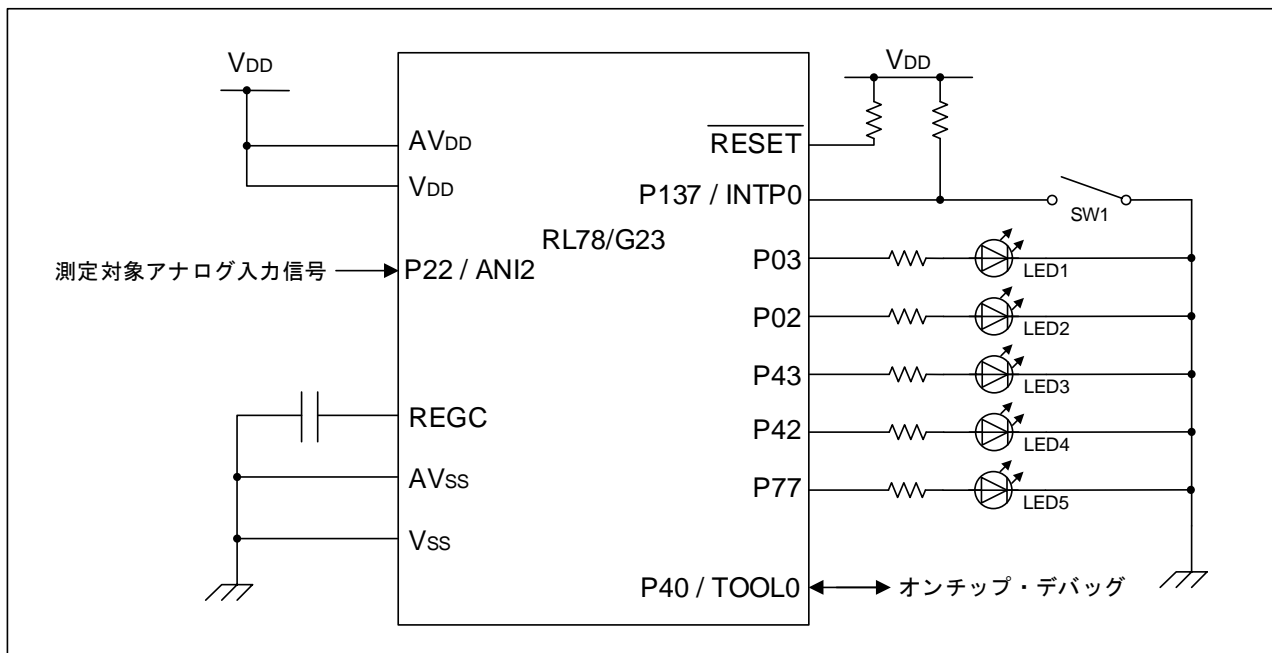
注 アプリケーションでは、CPU/周辺ハードウェア・クロックの設定を切替えています。

### 3. ハードウェア説明

#### 3.1 ハードウェア構成例

図 3-1 に本アプリケーションノートで使用するハードウェア構成例を示します。

図 3-1 ハードウェア構成



注意 1. この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください（入力専用ポートは個別に抵抗を介して  $V_{DD}$  又は  $V_{SS}$  に接続して下さい）。

注意 2.  $EV_{SS}$  で始まる名前の端子がある場合には  $V_{SS}$  に、 $EV_{DD}$  で始まる名前の端子がある場合に  $V_{DD}$  にそれぞれ接続してください。

注意 3.  $V_{DD}$  は  $LVD0$  にて設定したリセット解除電圧 ( $V_{LVD0}$ ) 以上にしてください。

#### 3.2 使用端子一覧

表 3-1 に使用端子と機能を示します。

表 3-1 使用端子と機能

端子名	入出力	内容
P137 / INTPO	入力	スイッチ (SW1) 入力ポート
P22 / ANI2	入力	A/D コンバータ アナログ入力ポート
P03	出力	LED 点灯 (LED1) 制御ポート
P02	出力	LED 点灯 (LED2) 制御ポート
P43	出力	LED 点灯 (LED3) 制御ポート
P42	出力	LED 点灯 (LED4) 制御ポート
P77	出力	LED 点灯 (LED5) 制御ポート

注意 本アプリケーションノートは、使用端子のみを端子処理しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください。

## 4. ソフトウェア説明

### 4.1 オプション・バイトの設定一覧

表 4-1にオプション・バイト設定を示します。

表 4-1 オプション・バイト設定

アドレス	設定値	内容
000C0H / 040C0H	11101111B	ウォッチドッグ・タイマ動作停止 (リセット解除後、カウント停止)
000C1H / 040C1H	11111110B	LVD0 検出電圧: リセット・モード 立ち上がり時 TYP. 1.90V (1.84 V ~ 1.95 V) 立ち下がり時 TYP. 1.86V (1.80 V ~ 1.91 V)
000C2H / 040C2H	11101000B	HSモード、 高速オンチップ・オシレータ・クロック : 32MHz
000C3H / 040C3H	10000100B	オンチップ・デバッグ許可

### 4.2 定数一覧

本アプリケーションノートで使用している定数を、表 4-2、表 4-3 に示します。

表 4-2 定数一覧 (1/2)

定数名	設定値	内容
R_LED_STATE_A	0b00000U	状態 (A) におけるLED点灯パターン
R_LED_STATE_B	0b10011U	状態 (B) におけるLED点灯パターン
R_LED_STATE_C	0b01011U	状態 (C) におけるLED点灯パターン
R_LED_STATE_D	0b10111U	状態 (D) におけるLED点灯パターン
R_LED_STATE_E	0b01111U	状態 (E) におけるLED点灯パターン
R_LED_STATE_F	0b11011U	状態 (F) におけるLED点灯パターン
R_LED_STATE_G	0b10010U	状態 (G) におけるLED点灯パターン
R_LED_STATE_H	0b10000U	状態 (H) におけるLED点灯パターン
R_LED_STATE_I	0b10001U	状態 (I) におけるLED点灯パターン
R_LED_STATE_J	0b01010U	状態 (J) におけるLED点灯パターン
R_LED_STATE_K	0b01000U	状態 (K) におけるLED点灯パターン
R_LED_STATE_L	0b01001U	状態 (L) におけるLED点灯パターン
R_LED_STATE_M	0b10110U	状態 (M) におけるLED点灯パターン
R_LED_STATE_N	0b10100U	状態 (N) におけるLED点灯パターン
R_LED_STATE_O	0b01110U	状態 (O) におけるLED点灯パターン
R_LED_STATE_P	0b11010U	状態 (P) におけるLED点灯パターン
R_LED_STATE_Q	0b11000U	状態 (Q) におけるLED点灯パターン
R_LED_STATE_END	0b11111U	状態遷移終了におけるLED点灯パターン



表 4-3 定数一覧(1/2)

定数名	設定値	内容
R_fCLK_fMAIN_fOCO_HIGH	0x01U	クロックモード : fCLKに高速オンチップ・オシレータ・クロックを使用する
R_fCLK_fMAIN_fOCO_MID	0x02U	クロックモード : fCLKに中速オンチップ・オシレータ・クロックを使用する
R_fCLK_fMAIN_fOCO_LOW	0x03U	クロックモード : fCLKに低速オンチップ・オシレータ・クロックを使用する
R_fCLK_fMAIN_fMX	0x08U	クロックモード : fCLKに高速システム・クロックを使用する
R_fCLK_fSUB	0x10U	クロックモード : fCLKにサブシステム・クロックを使用する
R_NORMAL_AWAKEN	0U	高速オンチップ・オシレータ・クロックを通常起動
R_NORMAL_AWAKEN	1U	高速オンチップ・オシレータ・クロックを高速起動
R_PORT_NEGATIVE_MASK	0U	ポート出力制御でビットマスクを反転して用いる
R_PORT_POSITIVE_MASK	1U	ポート出力制御でビットマスクをそのまま用いる
R_MSEC_PER_COUNT	33UL	1ミリ秒あたりの32ビット・インターバル・タイマのカウント値
R_WAIT_OSTC_COUNT	0xD0U	X1発振安定待ちで、OSTCを待つ値
R_WAIT_MSEC_HIGH_OCO	4UL	高速オンチップ・オシレータ・クロックの発振安定待ち時間 (ミリ秒)
R_WAIT_MSEC_MID_OCO	4UL	中速オンチップ・オシレータ・クロックの発振安定待ち時間 (ミリ秒)
R_WAIT_MSEC_LOW_OCO	4UL	低速オンチップ・オシレータ・クロックの発振安定待ち時間 (ミリ秒)
R_WAIT_MSEC_XT1	4UL	サブシステム・クロックの発振安定待ち時間 (ミリ秒)
R_WAIT_MSEC_CHATTERING	5UL	スイッチのチャタリング安定待ち時間 (ミリ秒)
_0_INTERRUPT_FLAG_OFF	0U	割り込みフラグがクリア状態
_1_INTERRUPT_FLAG_ON	1U	割り込みフラグがオン状態

### 4.3 変数一覧

本アプリケーションノートで使用しているグローバル変数 (複数の関数で利用される変数) を、表 4-4 に示します。

表 4-4 変数一覧

型	変数名	内容	使用している関数
uint8_t	g_intp0	INTP0 割り込みフラグ。 割り込みがあったとき _1_INTERRUPT_FLAG_ON になり、 それ以外では _0_INTERRUPT_FLAG_OFF になる。	R_Config_INTC_Create_UserInit、 r_Config_INTC_intp0_interrupt、 r_intp0_clear_flag、 r_intp0_is_flag_on

## 4.4 関数 (サブルーチン) 一覧

表 4-5、表 4-6 に関数 (サブルーチン) を示します。

表 4-5 関数一覧(1/2)

関数名	概要
r_AtoB()	状態 (A) から状態 (B) へ移行。状態遷移 (1)。
r_BtoE()	状態 (B) から状態 (E) へ移行。状態遷移 (2)。
r_EtoO()	状態 (E) から状態 (O) へ移行。状態遷移 (3)。
r_OtoE()	状態 (O) から状態 (E) へ移行。状態遷移 (4)。
r_EtoB()	状態 (E) から状態 (B) へ移行。状態遷移 (5)。
r_BtoD()	状態 (B) から状態 (D) へ移行。状態遷移 (6)。
r_DtoE()	状態 (D) から状態 (E) へ移行。状態遷移 (7)。
r_EtoD()	状態 (E) から状態 (D) へ移行。状態遷移 (8)。
r_DtoM()	状態 (D) から状態 (M) へ移行。状態遷移 (9)。
r_MtoD()	状態 (M) から状態 (D) へ移行。状態遷移 (10)。
r_DtoN()	状態 (D) から状態 (N) へ移行。状態遷移 (11)。
r_NtoD()	状態 (N) から状態 (D) へ移行。状態遷移 (12)。
r_DtoB()	状態 (D) から状態 (B) へ移行。状態遷移 (13)。
r_BtoG()	状態 (B) から状態 (G) へ移行。状態遷移 (14)。
r_GtoB()	状態 (G) から状態 (B) へ移行。状態遷移 (15)。
r_BtoH()	状態 (B) から状態 (H) へ移行。状態遷移 (16)。
r_HtoB()	状態 (H) から状態 (B) へ移行。状態遷移 (17)。
r_BtoI()	状態 (B) から状態 (I) へ移行。状態遷移 (18)。
r_ItoB()	状態 (I) から状態 (B) へ移行。状態遷移 (19)。
r_BtoC()	状態 (B) から状態 (C) へ移行。状態遷移 (20)。
r_CtoD()	状態 (C) から状態 (D) へ移行。状態遷移 (21)。
r_DtoF()	状態 (D) から状態 (F) へ移行。状態遷移 (22)。
r_FtoD()	状態 (F) から状態 (D) へ移行。状態遷移 (23)。
r_DtoC()	状態 (D) から状態 (C) へ移行。状態遷移 (24)。
r_CtoJ()	状態 (C) から状態 (J) へ移行。状態遷移 (25)。
r_JtoC()	状態 (J) から状態 (C) へ移行。状態遷移 (26)。
r_CtoK()	状態 (C) から状態 (K) へ移行。状態遷移 (27)。
r_KtoC()	状態 (K) から状態 (C) へ移行。状態遷移 (28)。
r_CtoL()	状態 (C) から状態 (L) へ移行。状態遷移 (29)。
r_LtoC()	状態 (L) から状態 (C) へ移行。状態遷移 (30)。
r_CtoE()	状態 (C) から状態 (E) へ移行。状態遷移 (31)。
r_EtoC()	状態 (E) から状態 (C) へ移行。状態遷移 (32)。
r_CtoF()	状態 (C) から状態 (F) へ移行。状態遷移 (33)。
r_FtoC()	状態 (F) から状態 (C) へ移行。状態遷移 (34)。
r_CtoB()	状態 (C) から状態 (B) へ移行。状態遷移 (35)。
r_BtoF()	状態 (B) から状態 (F) へ移行。状態遷移 (36)。
r_FtoP()	状態 (F) から状態 (P) へ移行。状態遷移 (37)。
r_PtoF()	状態 (P) から状態 (F) へ移行。状態遷移 (38)。
r_FtoB()	状態 (F) から状態 (B) へ移行。状態遷移 (39)。
r_BtoQ()	状態 (B) から状態 (Q) へ移行。状態遷移 (40)。
r_QtoB()	状態 (Q) から状態 (B) へ移行。状態遷移 (41)。

表 4-6 関数一覧(2/2)

関数名	概要
r_end()	終了状態へ移行。
r_nop_loop()	NOP 命令を繰り返し、外部割込み発生を待つ。
r_halt()	HALT 命令実行後、NOP 命令を繰り返し、外部割込み発生を待つ。
r_stop()	STOP 命令実行後、NOP 命令を繰り返し、外部割込み発生を待つ。
r_snooze()	A/D コンバータを SNOOZE モードで起動し、STOP 命令実行。 A/D 変換完了割込みで復帰した後は、NOP 命令を繰り返し、 外部割込み発生を待つ。
r_start_high_oco()	高速オンチップ・オシレータ・クロックを開始。
r_start_mid_oco()	中速オンチップ・オシレータ・クロックを開始。
r_start_low_oco()	低速オンチップ・オシレータ・クロックを開始。
r_start_x1()	高速システム・クロックを開始。
r_start_xt1()	サブシステム・クロックを開始。
r_stop_high_oco()	高速オンチップ・オシレータ・クロックを停止。
r_stop_mid_oco()	中速オンチップ・オシレータ・クロックを停止。
r_stop_low_oco()	低速オンチップ・オシレータ・クロックを停止。
r_stop_x1()	高速システム・クロックを停止。
r_stop_xt1()	サブシステム・クロックを停止。
r_set_fclk()	クロック切り替えのためのレジスタ設定を行う。
r_set_high_speed_awakening()	高速オンチップ・オシレータ・クロックの起動モードを設定する。
r_wait_OSTC()	X1 クロックの発振安定待ちを行う。
r_user_init()	全体のユーザ定義初期化処理。
R_Config_INTC_Create_UserInit()	割込みコントローラのユーザ定義初期化処理。
r_Config_INTC_intp0_interrupt()	INTP0 割込み発生時の処理。INTP0 割込みフラグをセットする。
r_intp0_clear_flag()	INTP0 割込みフラグをクリアする。
r_intp0_is_flag_on()	INTP0 割込みフラグがセットされているかを判定する。
r_ITL000_ITL001_ITL012_ITL013_ set_compare_value()	32 ビット・インターバル・タイマのコンペア値を設定する。
r_ITL000_ITL001_ITL012_ITL013_ wait_interval_timer()	32 ビット・インターバル・タイマを使用し、指定時間経過を待つ。
r_port_set_LED()	5 つの LED の点灯/消灯を設定する。
r_port_bitmask_to_port()	ビット列をもとに、ポート出力値の 0/1 を切り替える。

## 4.5 関数 (サブルーチン) 仕様

サンプルコードの関数 (サブルーチン) 仕様を示します。

### [関数名] r\_AtoB()

---

概要	(A) から (B) へ状態移行
宣言	void r_AtoB (void)
説明	高速オンチップ・オシレータ・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

### [関数名] r\_BtoE()

---

概要	(B) から (E) へ状態移行
宣言	void r_BtoE (void)
説明	高速オンチップ・オシレータ・クロック動作状態からサブシステム・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

### [関数名] r\_EtoO()

---

概要	(E) から (O) へ状態移行
宣言	void r_EtoO (void)
説明	サブシステム・クロック動作状態からサブシステム・クロックHALT状態へ切り替えます。LED 点灯制御を行った後、HALT命令を実行し、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

### [関数名] r\_OtoE()

---

概要	(O) から (E) へ状態移行
宣言	void r_OtoE (void)
説明	サブシステム・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

### [関数名] r\_EtoB()

---

概要	(E) から (B) へ状態移行
宣言	void r_EtoB (void)
説明	サブシステム・クロック動作状態から高速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_BtoD()**


---

概要	(B) から (D) へ状態移行
宣言	void r_BtoD (void)
説明	高速オンチップ・オシレータ・クロック動作状態から高速システム・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_DtoE()**


---

概要	(D) から (E) へ状態移行
宣言	void r_DtoE (void)
説明	高速システム・クロック動作状態からサブシステム・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_EtoD()**


---

概要	(E) から (D) へ状態移行
宣言	void r_EtoD (void)
説明	サブシステム・クロック動作状態から高速システム・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_DtoM()**


---

概要	(D) から (M) へ状態移行
宣言	void r_DtoM (void)
説明	高速システム・クロック動作状態から高速システム・クロックHALT状態へ切り替えます。LED 点灯制御を行った後、HALT命令を実行し、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_MtoD()**


---

概要	(M) から (D) へ状態移行
宣言	void r_MtoD (void)
説明	高速システム・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_DtoN()**


---

概要	(D) から (N) へ状態移行
宣言	void r_DtoN (void)
説明	高速システム・クロック動作状態から高速システム・クロックSTOP状態へ切り替えます。LED 点灯制御を行った後、STOP命令を実行し、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_NtoD()**


---

概要	(N) から (D) へ状態移行
宣言	void r_NtoD (void)
説明	高速システム・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_DtoB()**


---

概要	(D) から (B) へ状態移行
宣言	void r_DtoB (void)
説明	高速システム・クロック動作状態から高速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_BtoG()**


---

概要	(B) から (G) へ状態移行
宣言	void r_BtoG (void)
説明	高速オンチップ・オシレータ・クロック動作状態から、高速オンチップ・オシレータ・クロックHALT状態へ切り替えます。LED 点灯制御を行った後、HALT命令を実行し、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_GtoB()**


---

概要	(G) から (B) へ状態移行
宣言	void r_GtoB (void)
説明	高速オンチップ・オシレータ・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_BtoH()**


---

概要	(B) から (H) へ状態移行
宣言	void r_BtoH (void)
説明	高速オンチップ・オシレータ・クロック動作状態から高速オンチップ・オシレータ・クロックSTOP状態へ切り替えます。LED 点灯制御を行った後、STOP命令を実行し、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_HtoB()**


---

概要	(H) から (B) へ状態移行
宣言	void r_HtoB (void)
説明	高速オンチップ・オシレータ・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_Btol()**


---

概要	(B) から (I) へ状態移行
宣言	void r_Btol (void)
説明	高速オンチップ・オシレータ・クロック動作状態から高速オンチップ・オシレータ・クロックSNOOZE状態へ切り替えます。LED 点灯制御を行った後、A/DコンバータをSNOOZEモードで起動してSTOP命令を実行し、A/D変換完了割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_ltoB()**


---

概要	(I) から (B) へ状態移行
宣言	void r_ltoB (void)
説明	高速オンチップ・オシレータ・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_BtoC()**


---

概要	(B) から (C) へ状態移行
宣言	void r_BtoC (void)
説明	高速オンチップ・オシレータ・クロック動作状態から中速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_CtoD()**


---

概要	(C) から (D) へ状態移行
宣言	void r_CtoD (void)
説明	中速オンチップ・オシレータ・クロック動作状態から高速システム・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし



**[関数名] r\_DtoF()**

概要	(D) から (F) へ状態移行
宣言	void r_DtoF (void)
説明	高速システム・クロック動作状態から低速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_FtoD()**

概要	(F) から (D) へ状態移行
宣言	void r_FtoD (void)
説明	低速オンチップ・オシレータ・クロック動作状態から高速システム・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_DtoC()**

概要	(D) から (C) へ状態移行
宣言	void r_DtoC (void)
説明	高速システム・クロック動作状態から中速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_CtoJ()**

概要	(C) から (J) へ状態移行
宣言	void r_CtoJ (void)
説明	中速オンチップ・オシレータ・クロック動作状態から中速オンチップ・オシレータ・クロックHALT状態へ切り替えます。LED 点灯制御を行った後、HALT命令を実行し、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_JtoC()**

概要	(J) から (C) へ状態移行
宣言	void r_JtoC (void)
説明	中速オンチップ・オシレータ・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし



**[関数名] r\_CtoK()**


---

概要	(C) から (K) へ状態移行
宣言	void r_CtoK (void)
説明	中速オンチップ・オシレータ・クロック動作状態から中速オンチップ・オシレータ・クロックSTOP状態へ切り替えます。LED 点灯制御を行った後、STOP命令を実行し、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_KtoC()**


---

概要	(K) から (C) へ状態移行
宣言	void r_KtoC (void)
説明	中速オンチップ・オシレータ・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_CtoL()**


---

概要	(C) から (L) へ状態移行
宣言	void r_CtoL (void)
説明	中速オンチップ・オシレータ・クロック動作状態から中速オンチップ・オシレータ・クロックSNOOZE状態へ切り替えます。LED 点灯制御を行った後、A/DコンバータをSNOOZEモードで起動してSTOP命令を実行し、A/D変換完了割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_LtoC()**


---

概要	(L) から (C) へ状態移行
宣言	void r_LtoC (void)
説明	中速オンチップ・オシレータ・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_CtoE()**


---

概要	(C) から (E) へ状態移行
宣言	void r_CtoE (void)
説明	中速オンチップ・オシレータ・クロック動作状態からサブシステム・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_EtoC()**

概要	(E) から (C) へ状態移行
宣言	void r_EtoC (void)
説明	サブシステム・クロック動作状態から中速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_CtoF()**

概要	(C) から (F) へ状態移行
宣言	void r_CtoF (void)
説明	中速オンチップ・オシレータ・クロック動作状態から低速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_FtoC()**

概要	(F) から (C) へ状態移行
宣言	void r_FtoC (void)
説明	低速オンチップ・オシレータ・クロック動作状態から中速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_CtoB()**

概要	(C) から (B) へ状態移行
宣言	void r_CtoB (void)
説明	中速オンチップ・オシレータ・クロック動作状態から高速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_BtoF()**

概要	(B) から (F) へ状態移行
宣言	void r_BtoF (void)
説明	高速オンチップ・オシレータ・クロック動作状態から低速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_FtoP()**


---

概要	(F) から (P) へ状態移行
宣言	void r_FtoP (void)
説明	低速オンチップ・オシレータ・クロック動作状態から低速オンチップ・オシレータ・クロックHALT状態へ切り替えます。LED 点灯制御を行った後、HALT命令を実行し、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_PtoF()**


---

概要	(P) から (F) へ状態移行
宣言	void r_PtoF (void)
説明	低速オンチップ・オシレータ・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_FtoB()**


---

概要	(F) から (B) へ状態移行
宣言	void r_FtoB (void)
説明	低速オンチップ・オシレータ・クロック動作状態から高速オンチップ・オシレータ・クロック動作状態へ切り替えます。クロック切り替え後、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_BtoQ()**


---

概要	(B) から (Q) へ状態移行
宣言	void r_DtoN (void)
説明	高速オンチップ・オシレータ・クロック動作状態から高速オンチップ・オシレータ・クロックSTOP状態 (高速起動) へ切り替えます。高速オンチップ・オシレータ高速起動の設定をして、LED 点灯制御を行った後、STOP命令を実行し、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_QtoB()**


---

概要	(Q) から (B) へ状態移行
宣言	void r_QtoB (void)
説明	高速オンチップ・オシレータ高速起動の解除後、高速オンチップ・オシレータ・クロック動作状態として、LED 点灯制御を行い、外部割込み待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_end()**


---

概要	状態遷移を終了
宣言	void r_end (void)
説明	遷移終了状態として、LED 点灯制御を行います。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_nop\_loop()**


---

概要	NOP命令の繰り返しによる外部割り込み待ち
宣言	void r_nop_Loop (void)
説明	NOP 命令を繰り返し実行しながら、外部割り込み発生を待ちます。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_halt()**


---

概要	HALT状態への移行と復帰待ち
宣言	void r_halt (void)
説明	HALT命令を繰り返し実行しながら、外部割り込み発生を待ちます。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_stop()**


---

概要	STOP状態への移行と復帰待ち
宣言	void r_stop (void)
説明	STOP命令を繰り返し実行しながら、外部割り込み発生を待ちます。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_snooze()**


---

概要	SNOOZE状態への移行と復帰待ち
宣言	void r_snooze (void)
説明	A/DコンバータをSNOOZEモードで起動してから、STOP命令を行い、A/D変換完了割り込み発生を待ちます。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_start\_high\_oco()**


---

概要	高速オンチップ・オシレータ・クロックの開始
宣言	void r_start_high_oco (void)
説明	高速オンチップ・オシレータ・クロックの開始設定をし、発振安定待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_start\_mid\_oco()**


---

概要	中速オンチップ・オシレータ・クロックの開始
宣言	void r_start_mid_oco (void)
説明	高速オンチップ・オシレータ・クロックの開始設定をし、発振安定待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_start\_low\_oco()**


---

概要	低速オンチップ・オシレータ・クロックの開始
宣言	void r_start_low_oco (void)
説明	低速オンチップ・オシレータ・クロックの開始設定をし、発振安定待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_start\_x1()**


---

概要	高速システム・クロックの開始
宣言	void r_start_x1 (void)
説明	高速システム・クロックの開始設定をし、発振安定待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_start\_xt1()**


---

概要	サブシステム・クロックの開始
宣言	void r_start_xt1 (void)
説明	サブシステム・クロックの開始設定をし、発振安定待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_stop\_high\_oco()**


---

概要	高速オンチップ・オシレータ・クロックの停止
宣言	void r_stop_high_oco (void)
説明	高速オンチップ・オシレータ・クロックの停止条件を確認後、停止設定をします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_stop\_mid\_oco()**


---

概要	中速オンチップ・オシレータ・クロックの停止
宣言	void r_stop_mid_oco (void)
説明	中速オンチップ・オシレータ・クロックの停止条件を確認後、停止設定をします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_stop\_low\_oco**


---

概要	低速オンチップ・オシレータ・クロックの停止
宣言	void r_stop_low_oco (void)
説明	低速オンチップ・オシレータ・クロックの停止条件を確認後、停止設定をします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_stop\_x1()**


---

概要	高速システム・クロックの停止
宣言	void r_stop_x1 (void)
説明	高速システム・クロックの停止条件を確認後、停止設定をします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_stop\_xt1()**


---

概要	サブシステム・クロックの停止
宣言	void r_stop_xt1 (void)
説明	サブシステム・クロックの停止条件を確認後、停止設定をします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_set\_fclk()**


---

概要	クロック切り替えの設定
宣言	void r_set_fclk (r_fclk_mode_t mode)
説明	CKCレジスタの設定をし、指定されたクロック動作モードへと切り替えます。
引数	r_fclk_mode_t mode :[クロック動作モード]
リターン値	なし
備考	なし

**[関数名] r\_set\_high\_speed\_awakening()**


---

概要	高速オンチップ・オシレータを高速起動するかの設定
宣言	void r_set_high_speed_awakening (r_awaken_mode_t flag)
説明	高速オンチップ・オシレータを高速起動にするかどうかを設定します。
引数	r_awaken_mode_t flag :[高速起動にするなら1、しないなら0]
リターン値	なし
備考	なし

**[関数名] r\_wait\_OSTC()**


---

概要	X1の発振安定待ち
宣言	void r_wait_OSTC (void)
説明	OSTCレジスタの値を確認し、X1の発振安定待ちをします。
引数	なし
リターン値	なし
備考	なし

**[関数名] R\_Config\_INTC\_Create\_UserInit()**


---

概要	割り込みコントローラのユーザ定義初期化処理
宣言	void R_Config_INTC_Create_UserInit (void)
説明	割り込みコントローラの初期化処理のうち、ユーザ定義箇所を実装しています。本アプリケーションノートではINTP0割り込みフラグを初期化します。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_Config\_INTC\_intp0\_interrupt()**


---

概要	INTP0割り込み発生時の処理
宣言	void __near r_Config_INTC_intp0_interrupt (void)
説明	INTP0割り込み発生時の処理を実装しています。本アプリケーションノートではINTP0割り込みフラグをセットします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_intp0\_clear\_flag()**

概要	INTP0割込みフラグのクリア
宣言	void r_intp0_clear_flag (void)
説明	INTP0割込みフラグをクリアします。
引数	なし
リターン値	なし
備考	なし

**[関数名] r\_intp0\_is\_flag\_on()**

概要	INTP0割込みフラグがセットされているかを判定
宣言	uint8_t r_intp0_is_flag_on (void)
説明	INTP0割込みフラグがセットされているかどうかを返します。
引数	なし
リターン値	1: セットされている 0: セットされていない
備考	なし

**[関数名] r\_ITL000\_ITL001\_ITL012\_ITL013\_set\_compare\_value()**

概要	32ビット・インターバル・タイマのコンペア値設定
宣言	void r_ITL000_ITL001_ITL012_ITL013_set_compare_value (uint32_t count)
説明	32ビット・インターバル・タイマのコンペア値を設定します。
引数	uint32_t count :[設定するコンペア値。下位16ビットがITLCMP00、 上位16ビットがITLCMP01に設定される]
リターン値	なし
備考	なし

**[関数名] r\_ITL000\_ITL001\_ITL012\_ITL013\_wait\_interval\_timer()**

概要	32ビット・インターバル・タイマによるウェイト
宣言	void r_ITL000_ITL001_ITL012_ITL013_wait_interval_timer (uint32_t msec)
説明	32ビット・インターバル・タイマを利用し、一定時間、ウェイトを行います。
引数	uint32_t msec :[ウェイト時間をミリ秒で指定]
リターン値	なし
備考	なし

**[関数名] r\_port\_set\_LED()**

概要	5つのLEDの点灯/消灯を制御
宣言	void r_port_set_LED (uint8_t state)
説明	ポート出力につながった5つのLEDについて、点灯/消灯の制御をします。
引数	uint8_t state :[LEDの点灯/消灯状態。下位ビットから順にLED1~5に 対応。0なら消灯し、1なら点灯する]
リターン値	なし
備考	なし

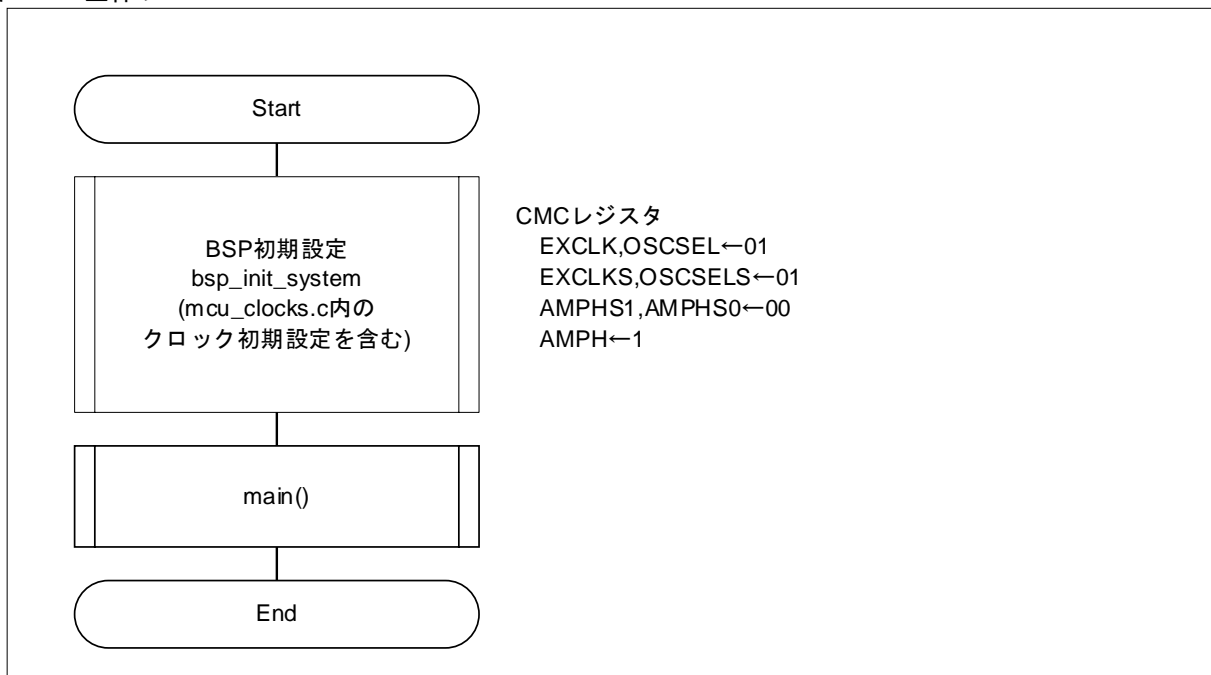
**[関数名] r\_port\_bitmask\_to\_port()**

概要	ポート出力値の0/1を設定
宣言	void r_port_bitmask_to_port (uint8_t mask, volatile unsigned char __near * port, r_config_port_mask_t flag)
説明	指定されたポートについて、出力値を0にするか1にするか設定します。
引数	uint8_t mask :[ポートに対するビットマスク] volatile unsigned char __near * port :[ポートのアドレス] r_config_port_mask_t flag :[ビットマスクをそのまま出力値に適用する か、反転して出力値に適用するか]
リターン値	なし
備考	なし

## 4.6 フローチャート

図 4-1 に本アプリケーションノートの全体フローを示します。

図 4-1 全体フロー





## 4.6.1 メイン処理

図 4-2、図 4-3 にメイン処理のフローチャートを示します。

図 4-2 メイン処理 (1/2)

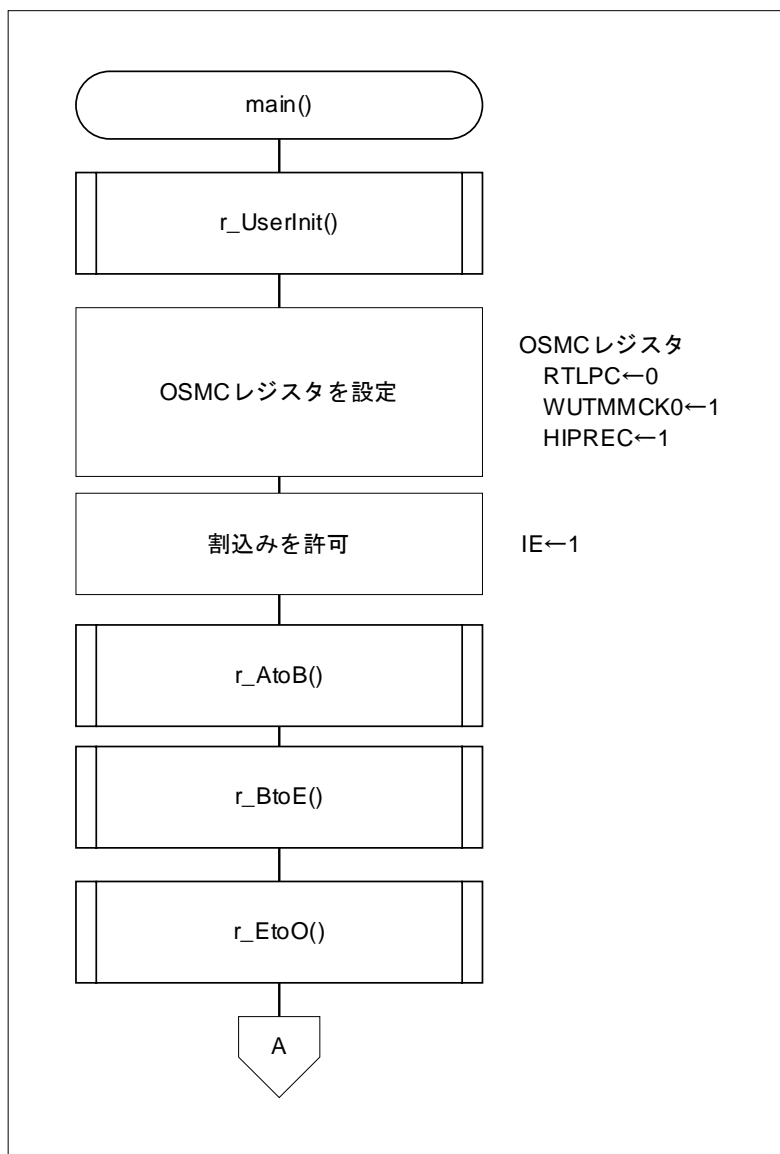
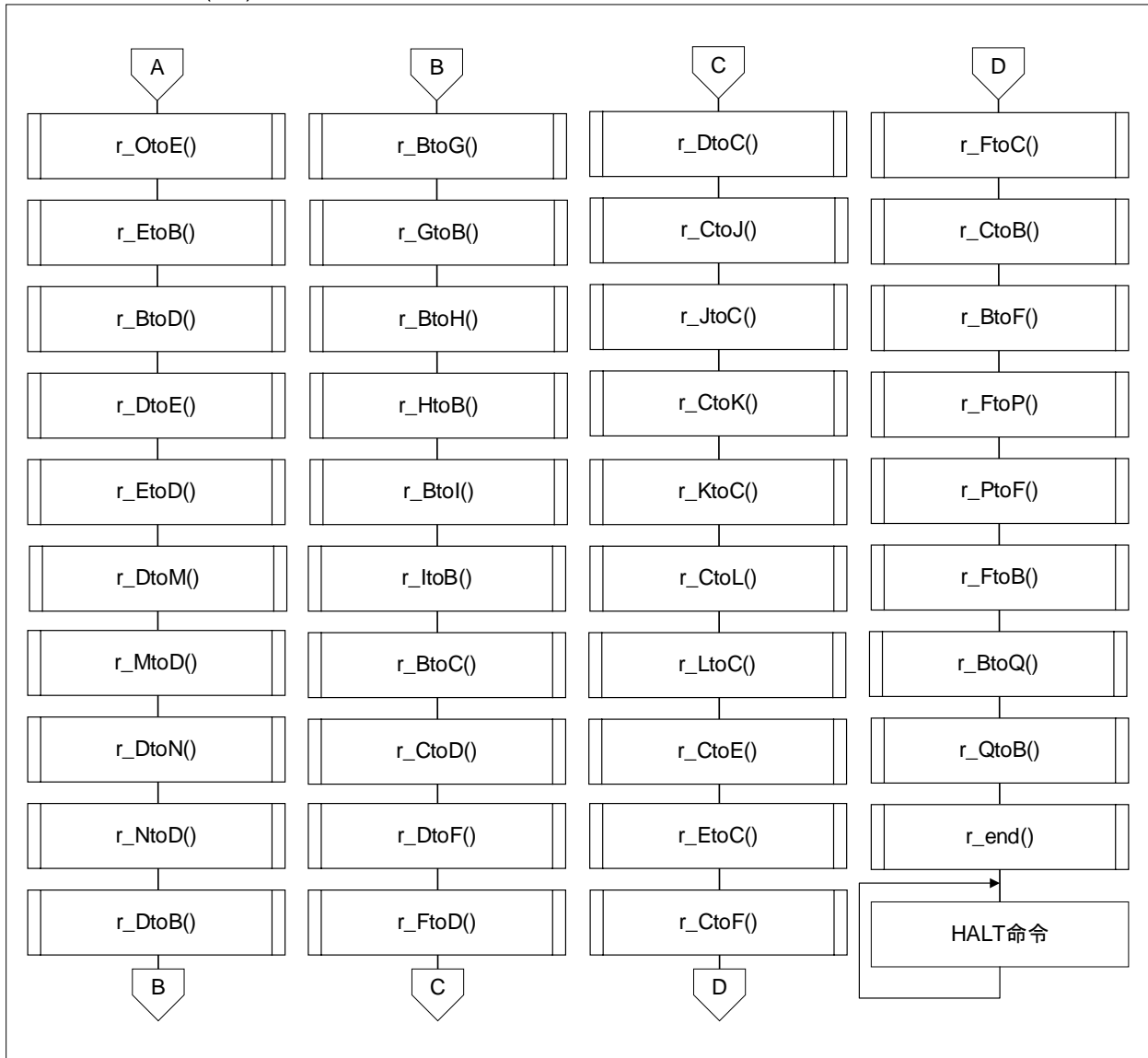


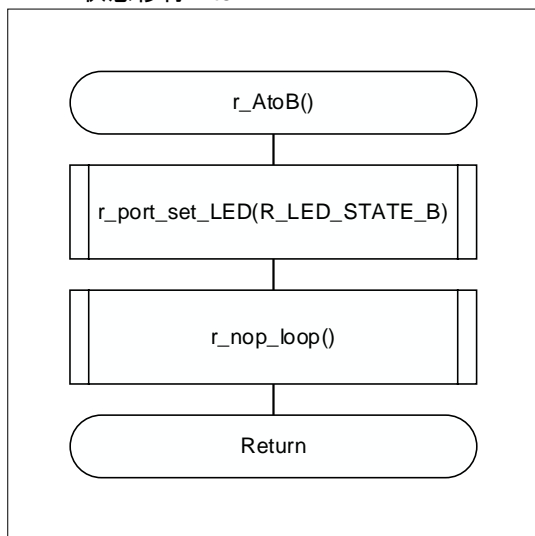
図 4-3 メイン処理 (2/2)



#### 4.6.2 状態移行 AtoB

図 4-4 に状態移行 AtoB のフローチャートを示します。

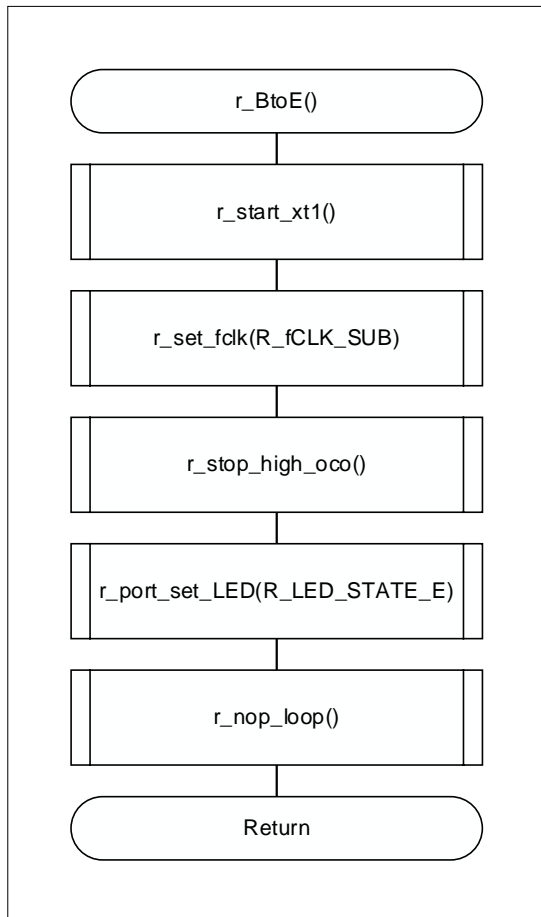
図 4-4 状態移行 AtoB



#### 4.6.3 状態移行 BtoE

図 4-5 に状態移行 BtoE のフローチャートを示します。

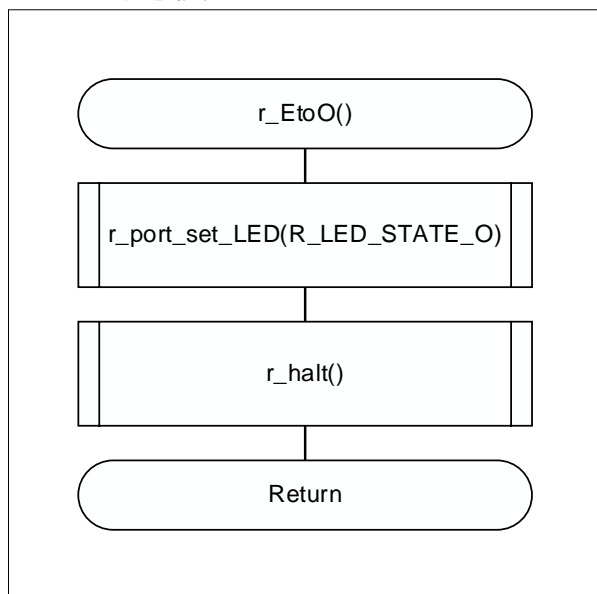
図 4-5 状態移行 BtoE



#### 4.6.4 状態移行 EtoO

図 4-6 に状態移行 EtoO のフローチャートを示します。

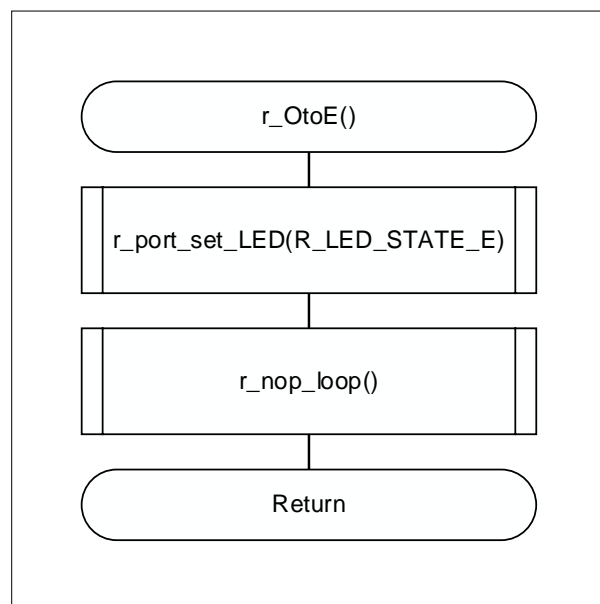
図 4-6 状態移行 EtoO



#### 4.6.5 状態移行 OtoE

図 4-7 に状態移行 OtoE のフローチャートを示します。

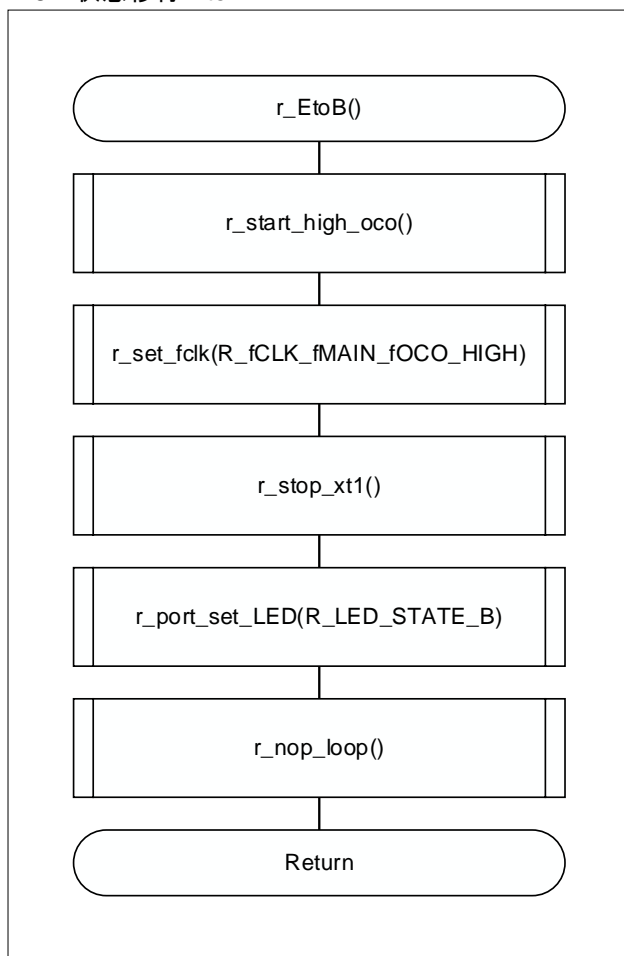
図 4-7 状態移行 OtoE



## 4.6.6 状態移行 EtoB

図 4-8 に状態移行 EtoB のフローチャートを示します。

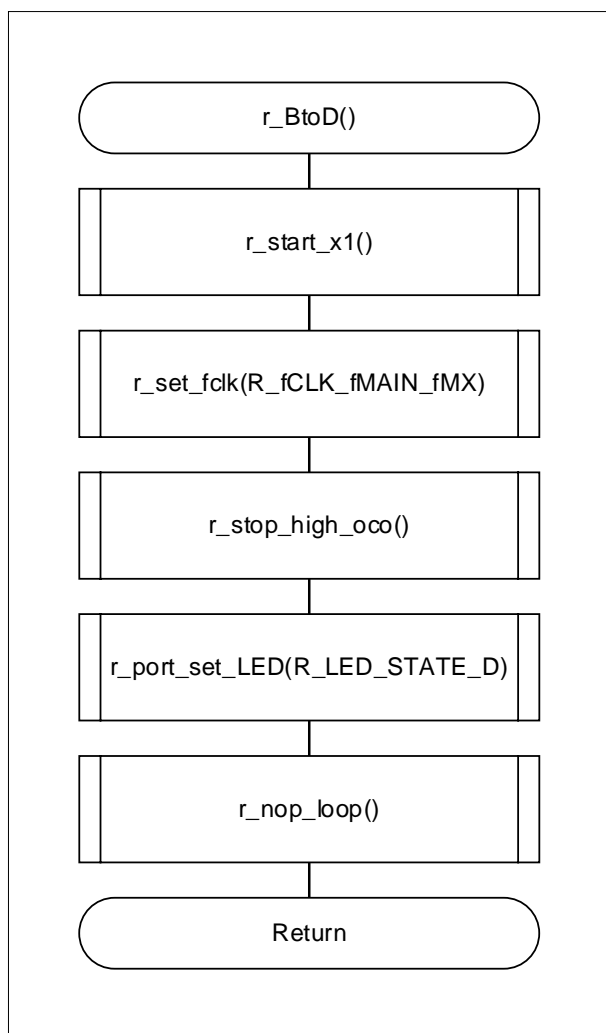
図 4-8 状態移行 EtoB



## 4.6.7 状態移行 BtoD

図 4-9 に状態移行 BtoD のフローチャートを示します。

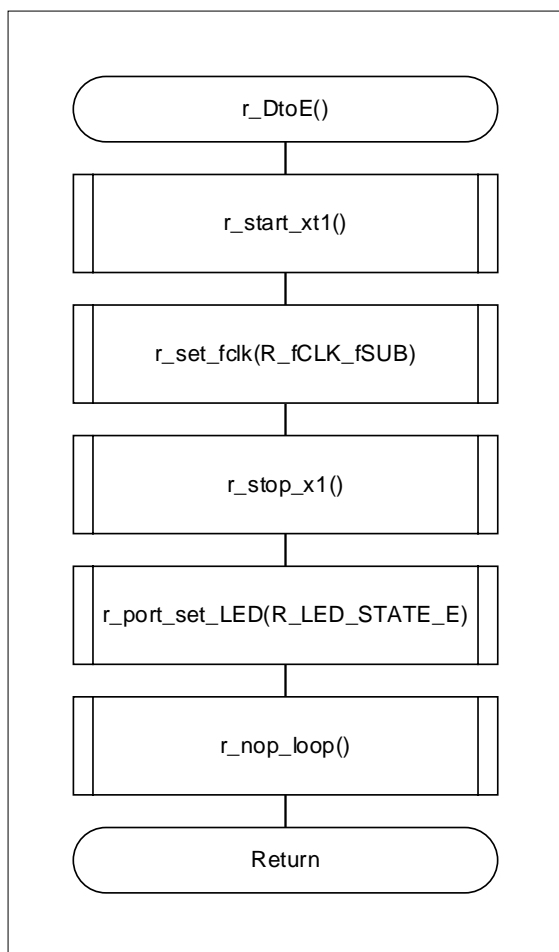
図 4-9 状態移行 BtoD



## 4.6.8 状態移行 DtoE

図 4-10 に状態移行 DtoE のフローチャートを示します。

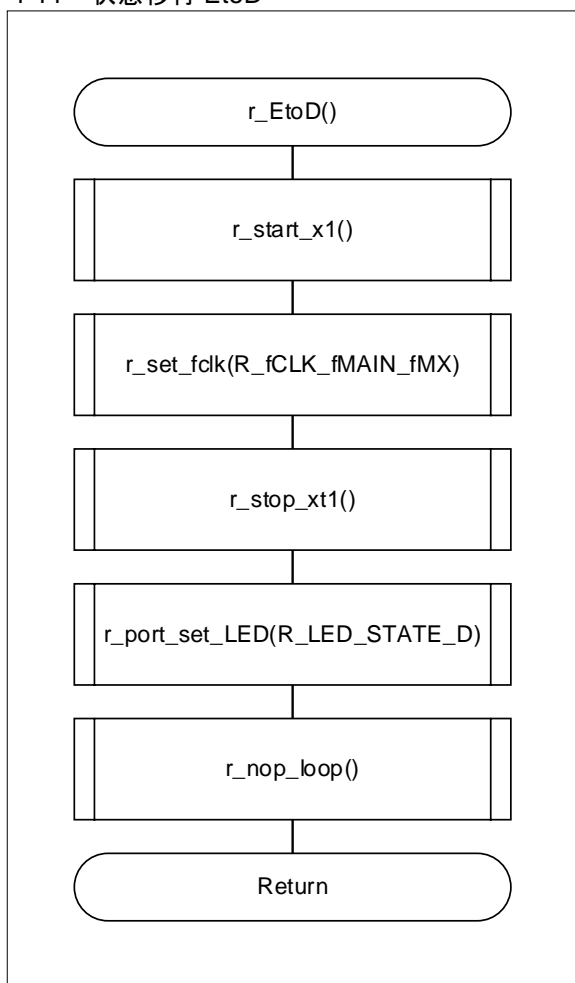
図 4-10 状態移行 DtoE



## 4.6.9 状態移行 EtoD

図 4-11 に状態移行 EtoD のフローチャートを示します。

図 4-11 状態移行 EtoD

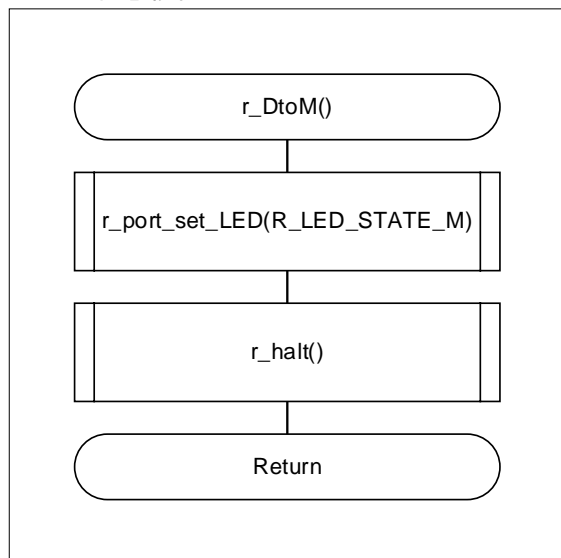




## 4.6.10 状態移行 DtoM

図 4-12 に状態移行 DtoM のフローチャートを示します。

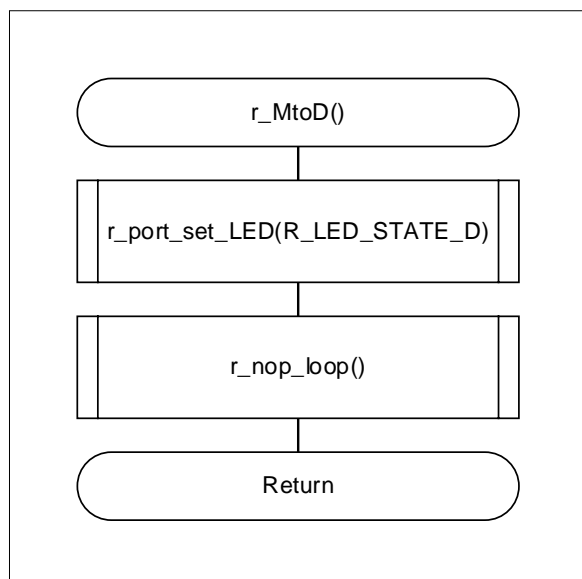
図 4-12 状態移行 DtoM



## 4.6.11 状態移行 MtoD

図 4-13 に状態移行 MtoD のフローチャートを示します。

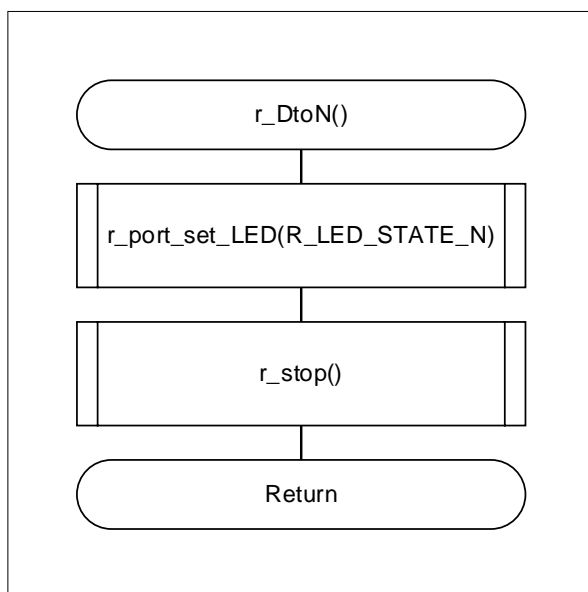
図 4-13 状態移行 MtoD



## 4.6.12 状態移行 DtoN

図 4-14 に状態移行 DtoN のフローチャートを示します。

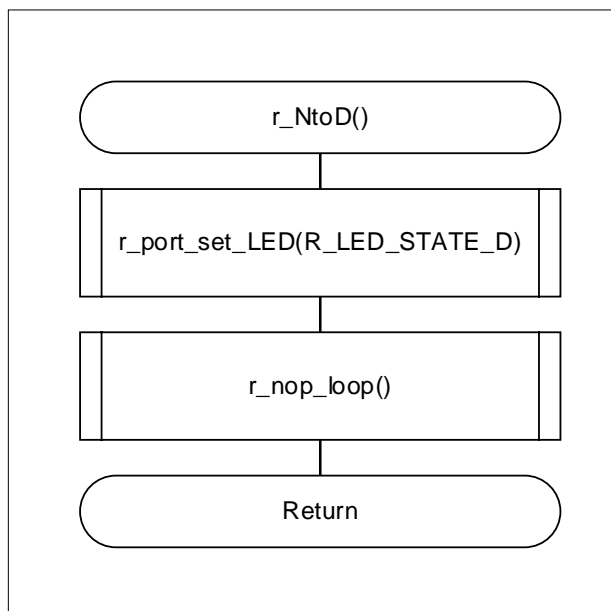
図 4-14 状態移行 DtoN



## 4.6.13 状態移行 NtoD

図 4-15 に状態移行 NtoD のフローチャートを示します。

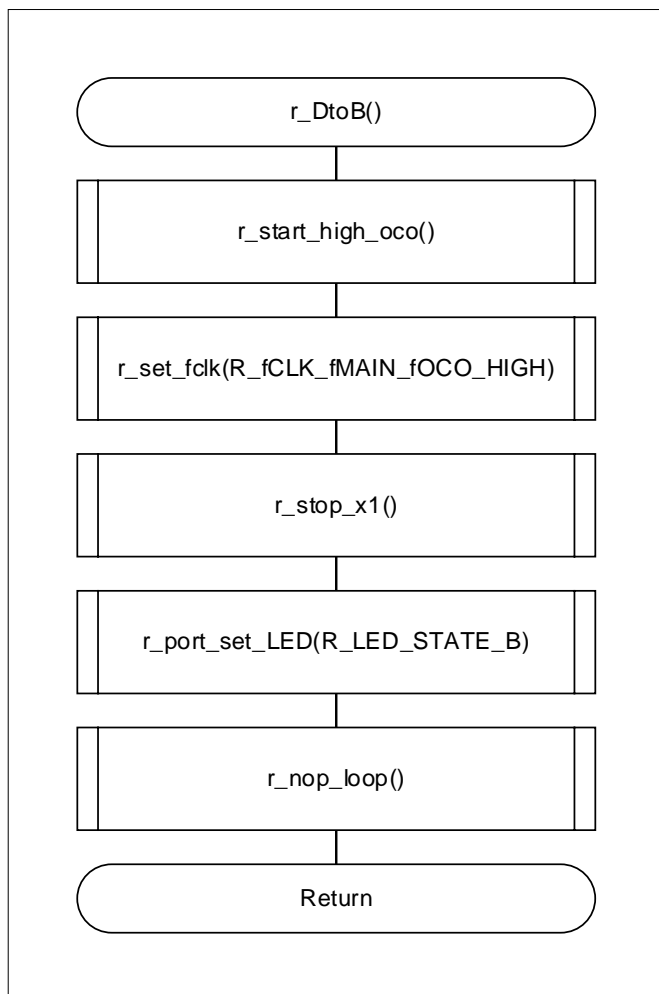
図 4-15 状態移行 NtoD



## 4.6.14 状態移行 DtoB

図 4-16 に状態移行 DtoB のフローチャートを示します。

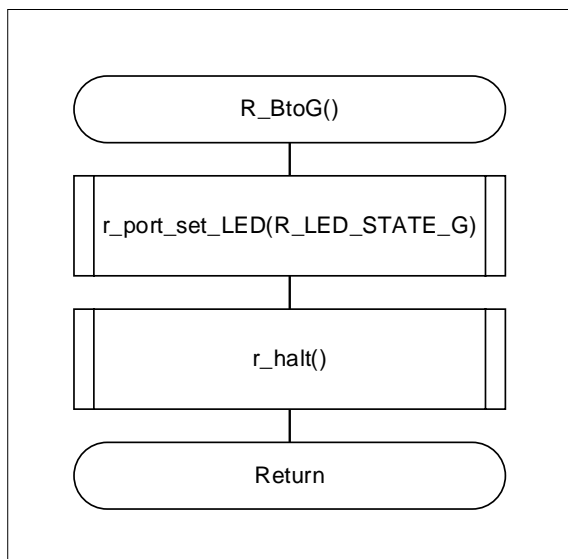
図 4-16 状態移行 DtoB



## 4.6.15 状態移行 BtoG

図 4-17 に状態移行 BtoG のフローチャートを示します。

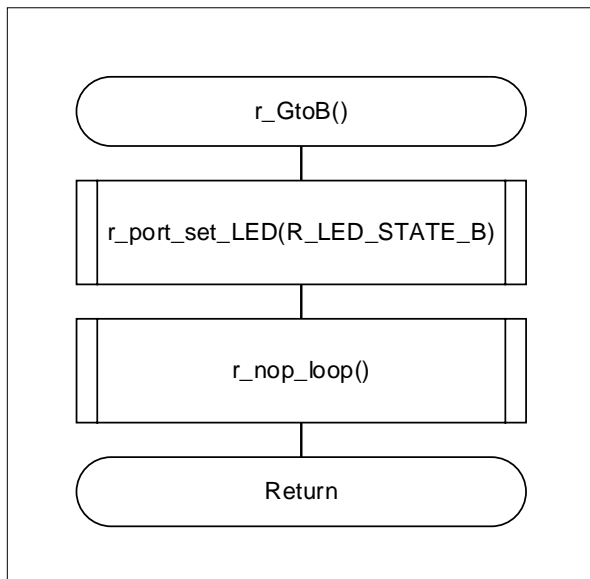
図 4-17 状態移行 BtoG



## 4.6.16 状態移行 GtoB

図 4-18 に状態移行 GtoB のフローチャートを示します。

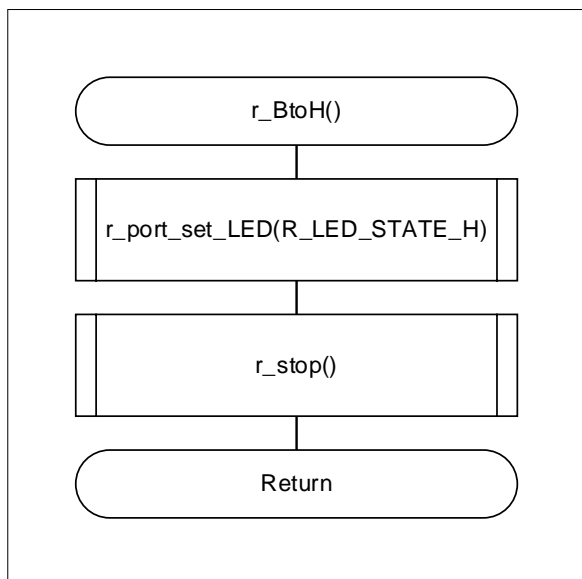
図 4-18 状態移行 GtoB



## 4.6.17 状態移行 BtoH

図 4-19 に状態移行 BtoH のフローチャートを示します。

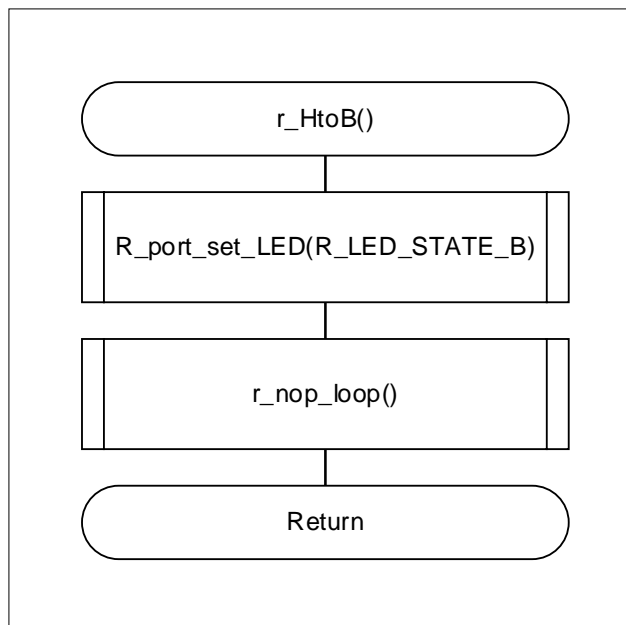
図 4-19 状態移行 BtoH



## 4.6.18 状態移行 HtoB

図 4-20 に状態移行 HtoB のフローチャートを示します。

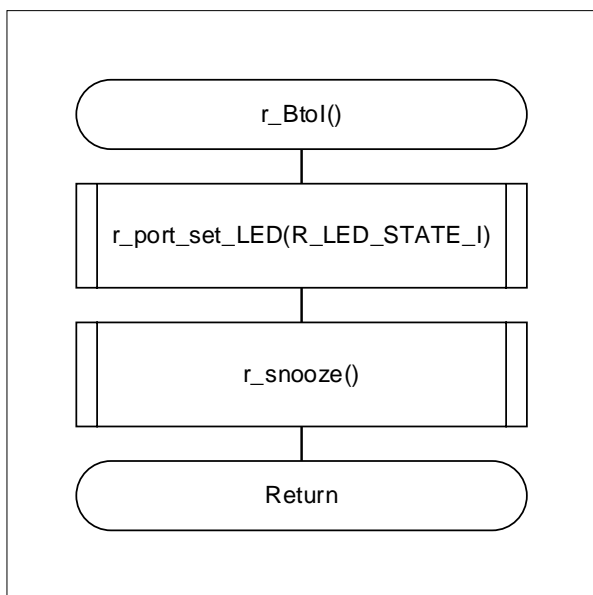
図 4-20 状態移行 HtoB



## 4.6.19 状態移行 BtoI

図 4-21 に状態移行 BtoI のフローチャートを示します。

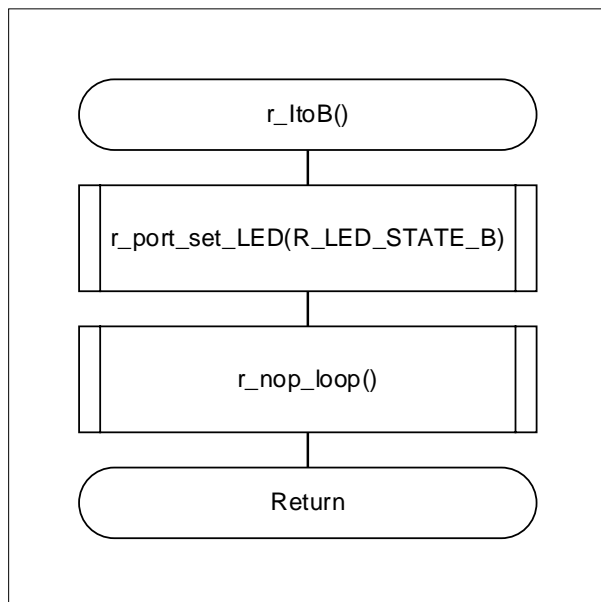
図 4-21 状態移行 BtoI



## 4.6.20 状態移行 ItoB

図 4-22 に状態移行 ItoB のフローチャートを示します。

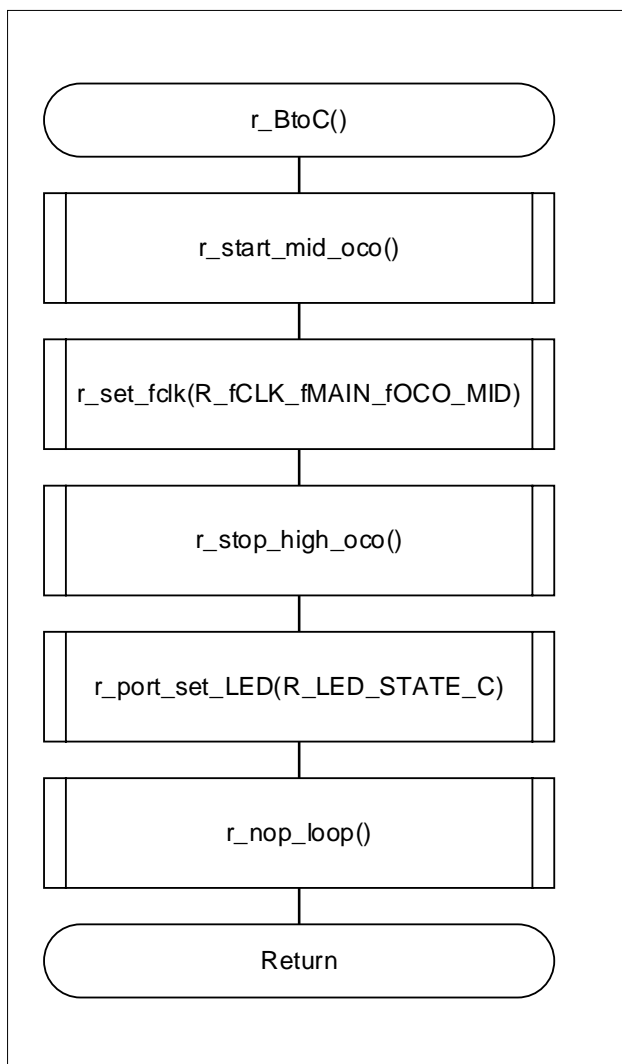
図 4-22 状態移行 ItoB



## 4.6.21 状態移行 BtoC

図 4-23 に状態移行 BtoC のフローチャートを示します。

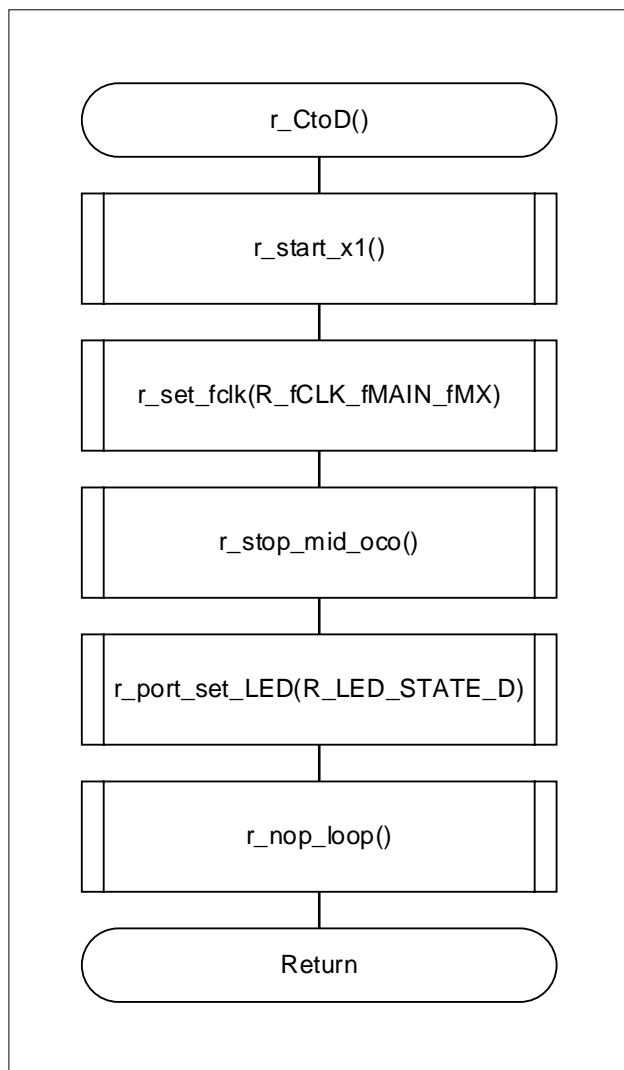
図 4-23 状態移行 BtoC



## 4.6.22 状態移行 CtoD

図 4-24 に状態移行 CtoD のフローチャートを示します。

図 4-24 状態移行 CtoD

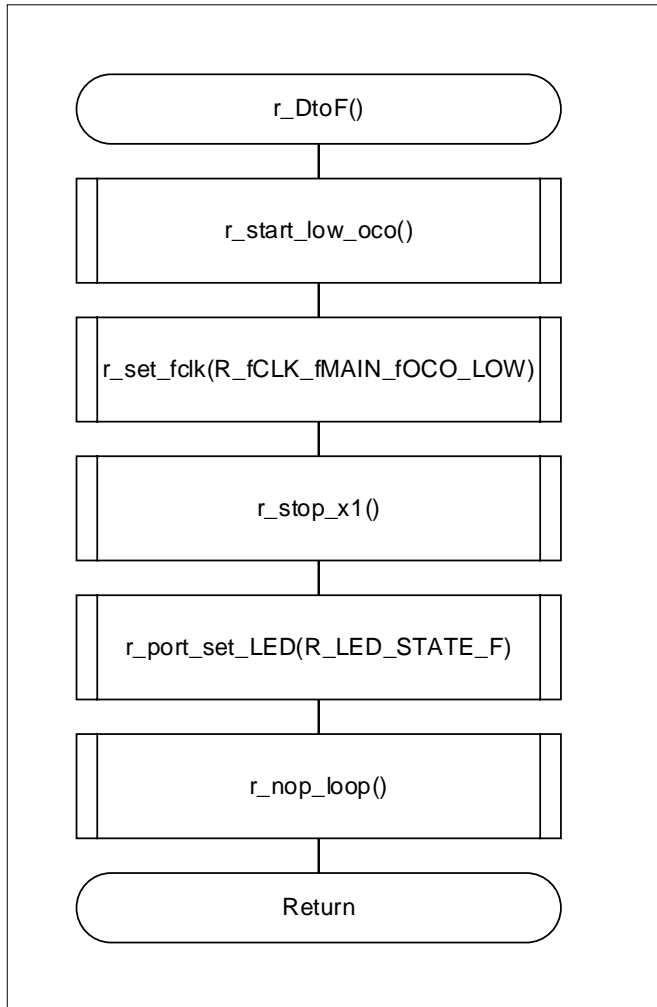




## 4.6.23 状態移行 DtoF

図 4-25 に状態移行 DtoF のフローチャートを示します。

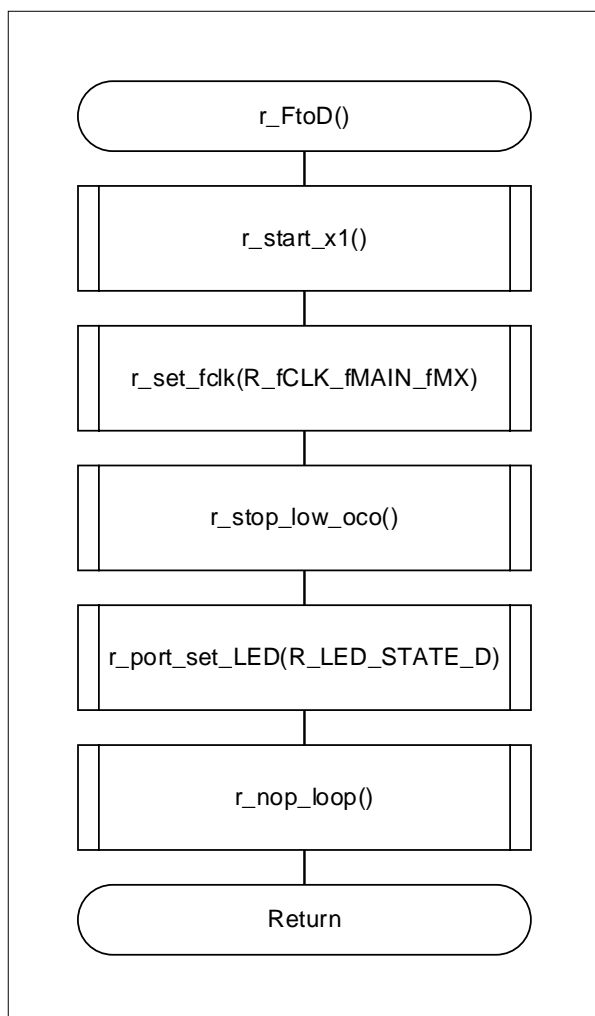
図 4-25 状態移行 DtoF



## 4.6.24 状態移行 FtoD

図 4-26 に状態移行 FtoD のフローチャートを示します。

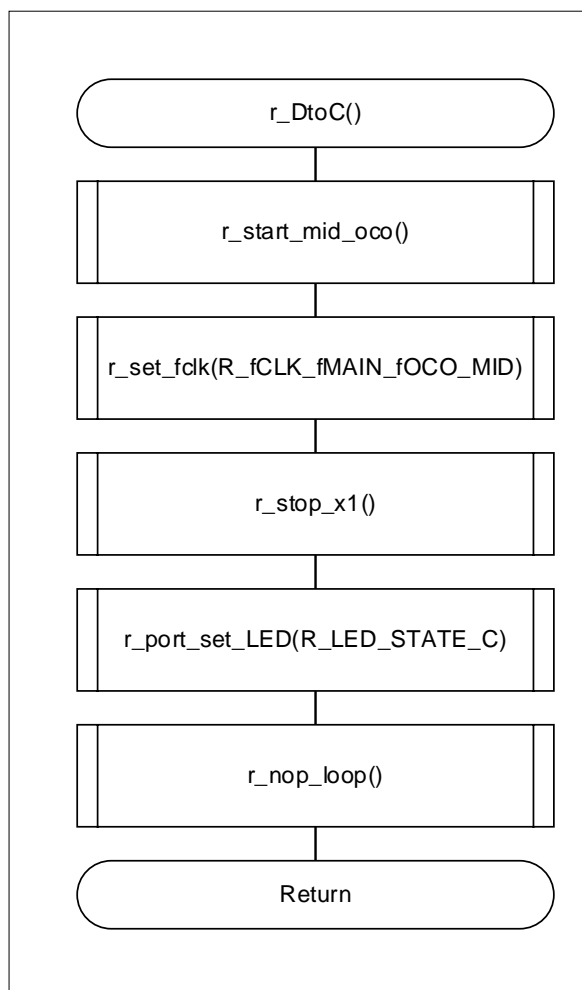
図 4-26 状態移行 FtoD



## 4.6.25 状態移行 DtoC

図 4-27 に状態移行 DtoC のフローチャートを示します。

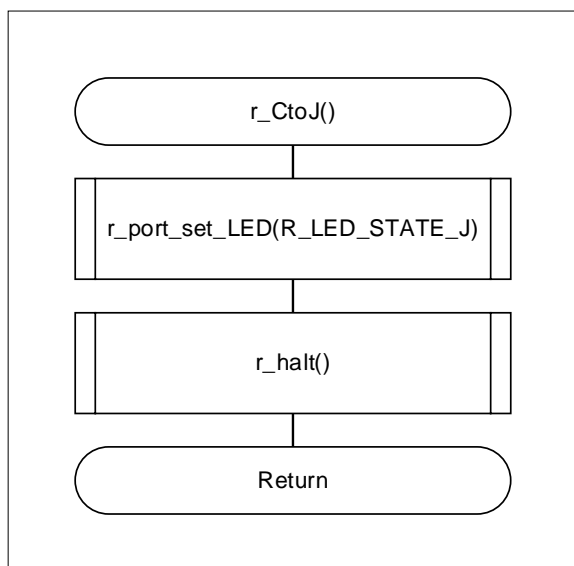
図 4-27 状態移行 DtoC



## 4.6.26 状態移行 CtoJ

図 4-28 に状態移行 CtoJ のフローチャートを示します。

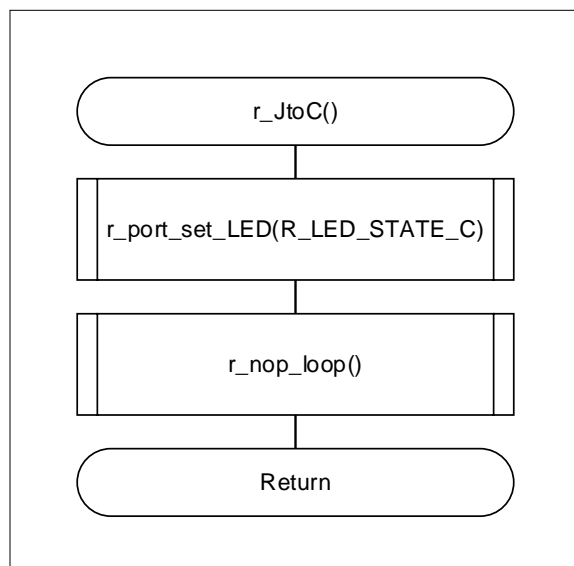
図 4-28 状態移行 CtoJ



## 4.6.27 状態移行 JtoC

図 4-29 に状態移行 JtoC のフローチャートを示します。

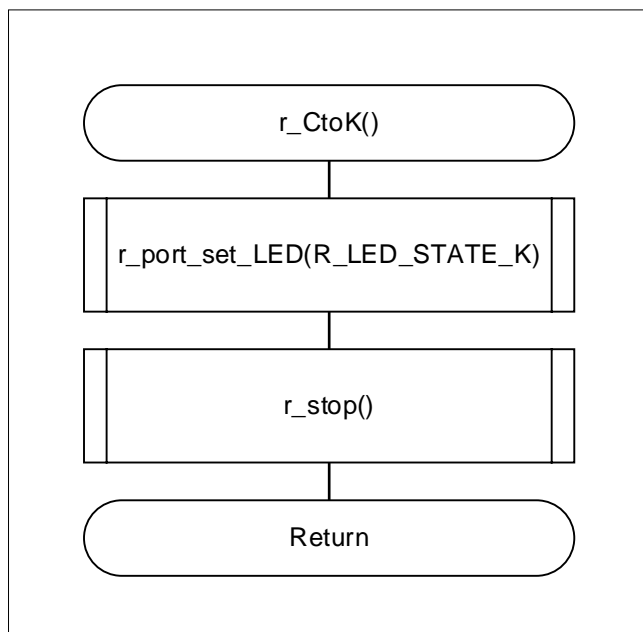
図 4-29 状態移行 JtoC



## 4.6.28 状態移行 CtoK

図 4-30 に状態移行 CtoK のフローチャートを示します。

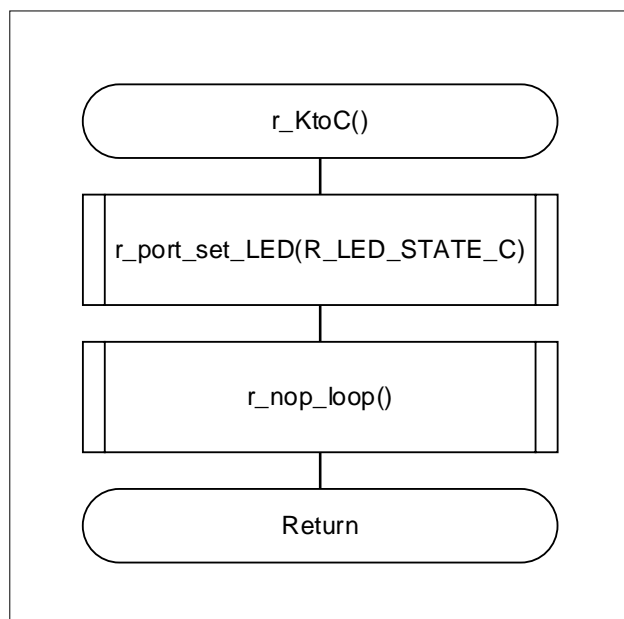
図 4-30 状態移行 CtoK



## 4.6.29 状態移行 KtoC

図 4-31 に状態移行 KtoC のフローチャートを示します。

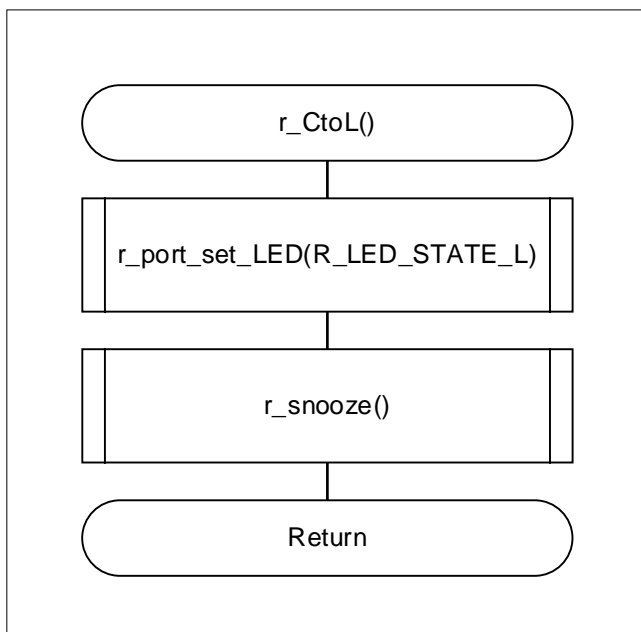
図 4-31 状態移行 KtoC



## 4.6.30 状態移行 CtoL

図 4-32 に状態移行 CtoL のフローチャートを示します。

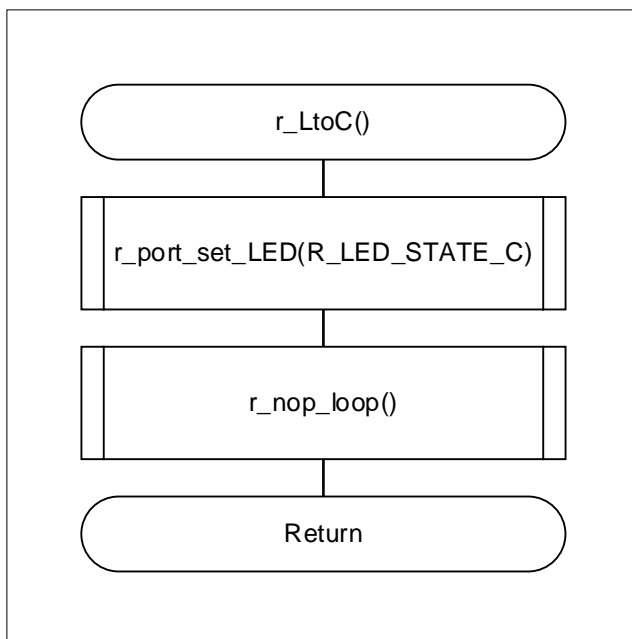
図 4-32 状態移行 CtoL



## 4.6.31 状態移行 LtoC

図 4-33 に状態移行 LtoC のフローチャートを示します。

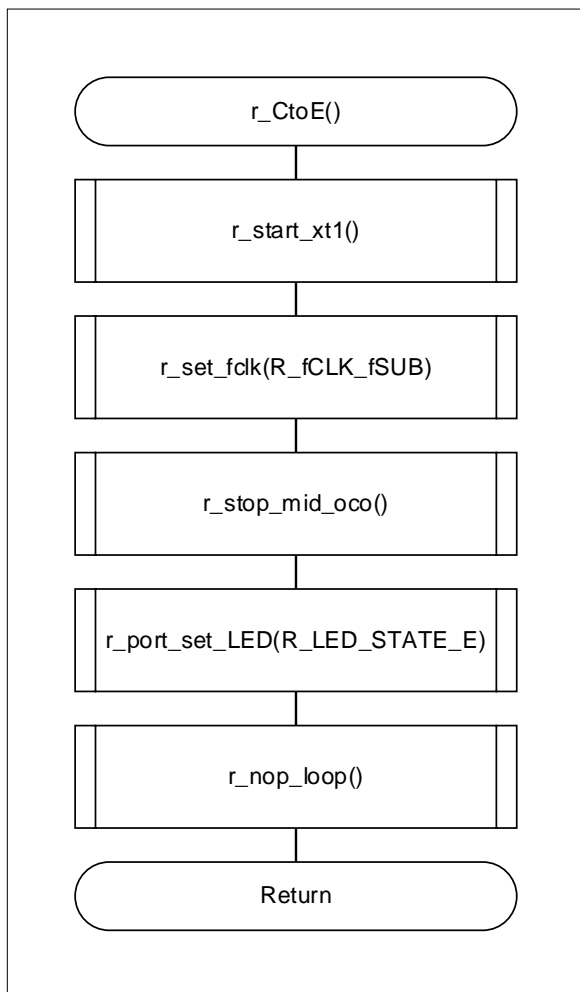
図 4-33 状態移行 LtoC



## 4.6.32 状態移行 CtoE

図 4-34 に状態移行 CtoE のフローチャートを示します。

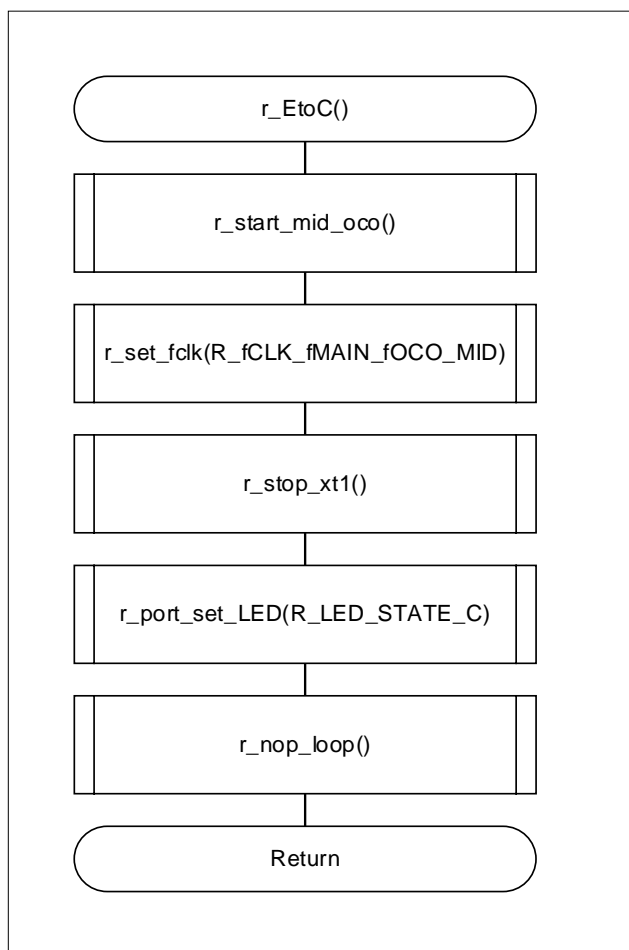
図 4-34 状態移行 CtoE



## 4.6.33 状態移行 EtoC

図 4-35 に状態移行 EtoC のフローチャートを示します。

図 4-35 状態移行 EtoC

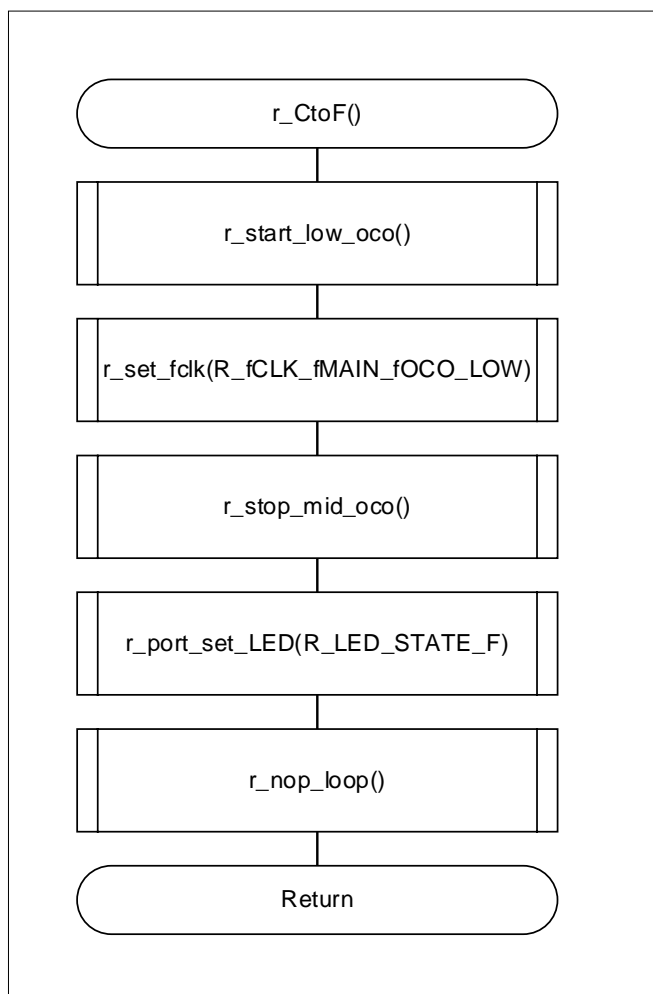




## 4.6.34 状態移行 CtoF

図 4-36 に状態移行 CtoF のフローチャートを示します。

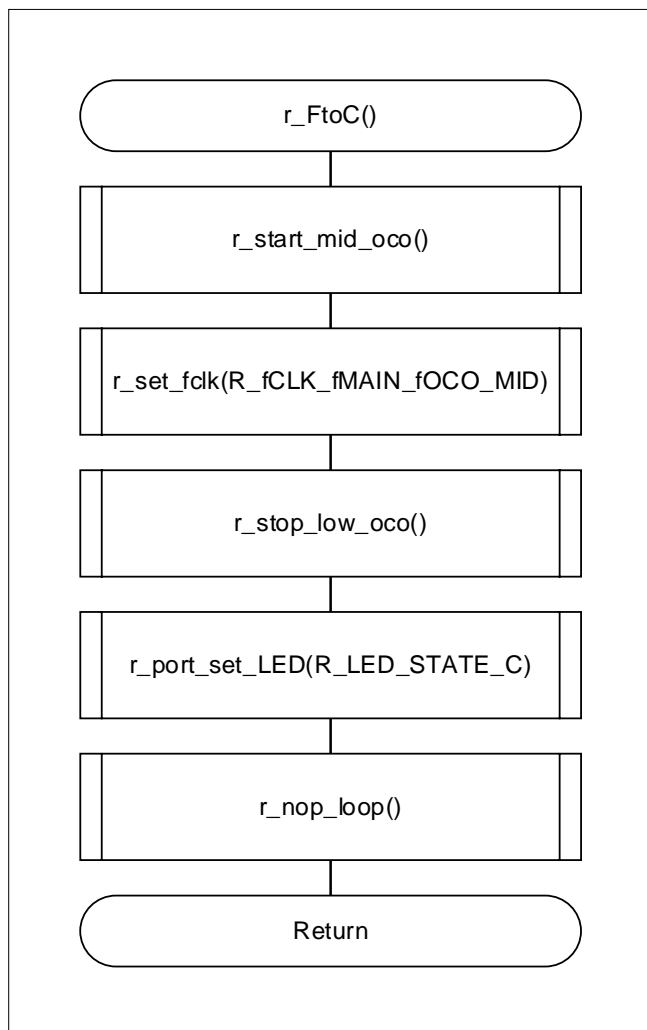
図 4-36 状態移行 CtoF



## 4.6.35 状態移行 FtoC

図 4-37 に状態移行 FtoC のフローチャートを示します。

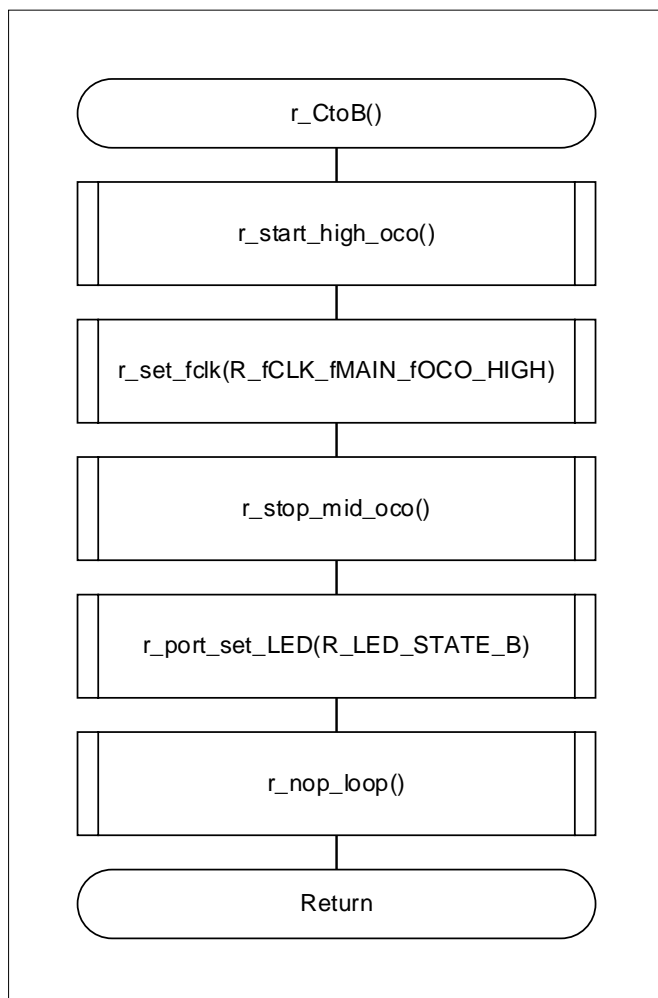
図 4-37 状態移行 FtoC



## 4.6.36 状態移行 CtoB

図 4-38 に状態移行 CtoB のフローチャートを示します。

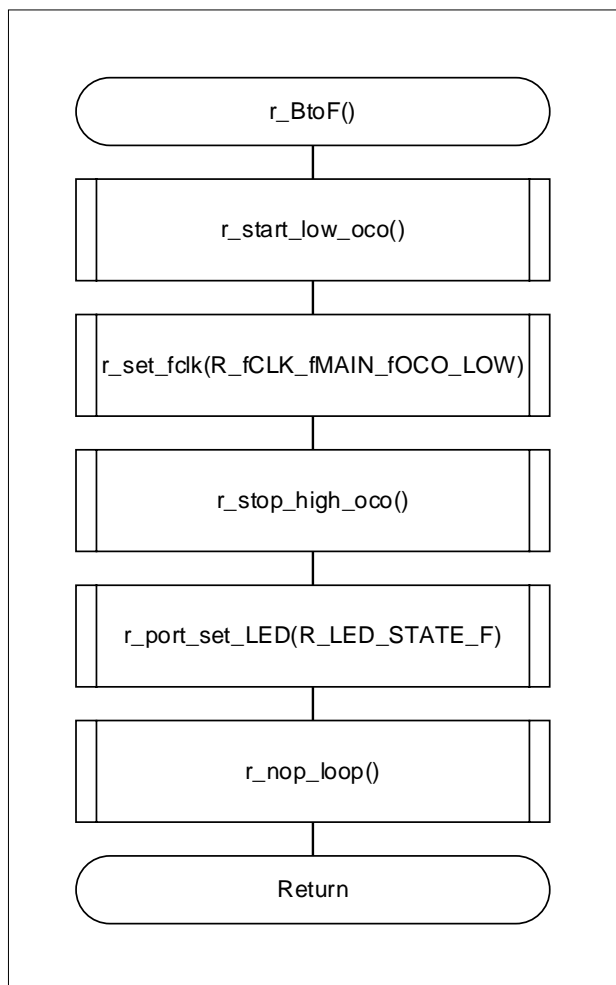
図 4-38 状態移行 CtoB



## 4.6.37 状態移行 BtoF

図 4-39 に状態移行 BtoF のフローチャートを示します。

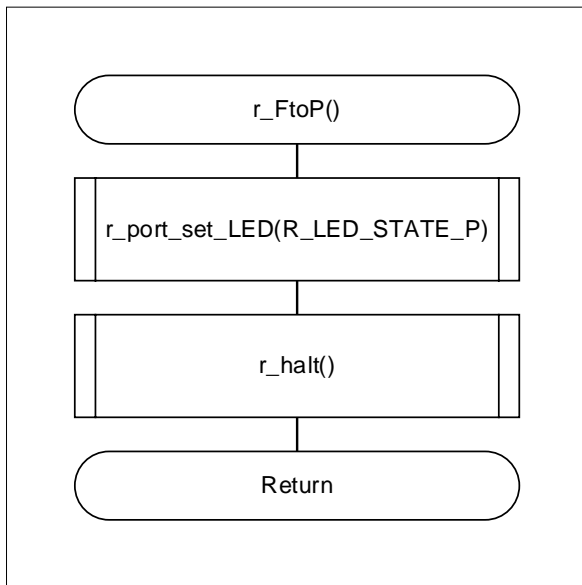
図 4-39 状態移行 BtoF



## 4.6.38 状態移行 FtoP

図 4-40 に状態移行 FtoP のフローチャートを示します。

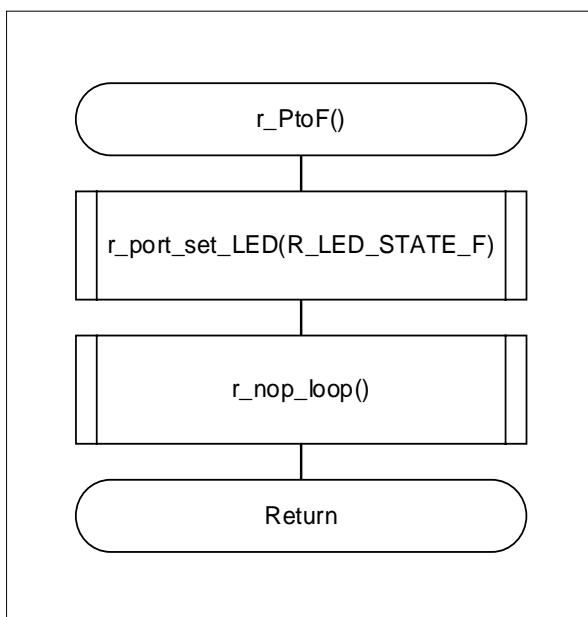
図 4-40 状態移行 FtoP



## 4.6.39 状態移行 PtoF

図 4-41 に状態移行 PtoF のフローチャートを示します。

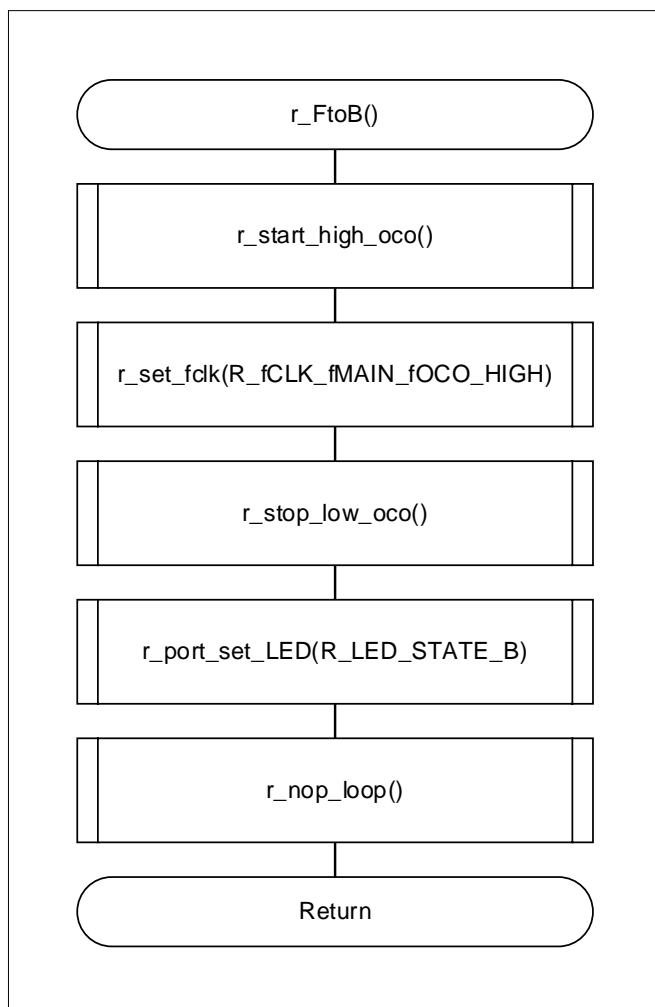
図 4-41 状態移行 PtoF



## 4.6.40 状態移行 FtoB

図 4-42 に状態移行 FtoB のフローチャートを示します。

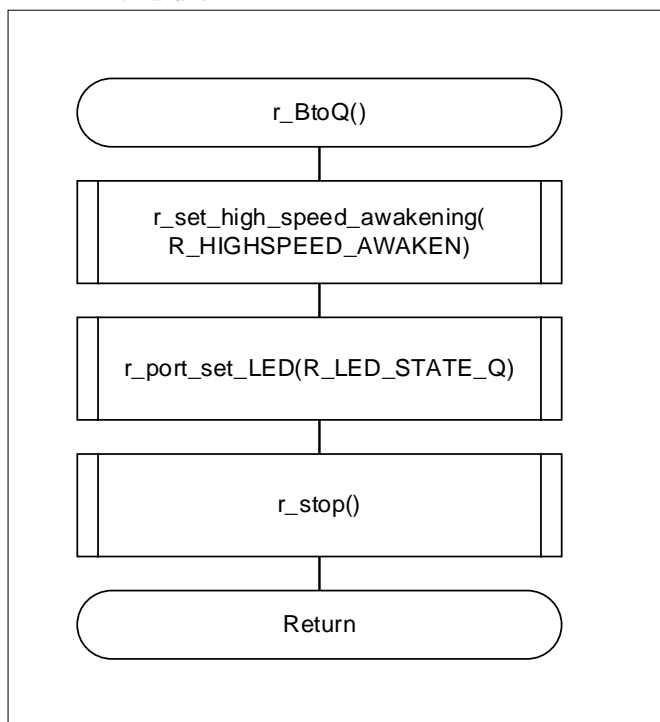
図 4-42 状態移行 FtoB



## 4.6.41 状態移行 BtoQ

図 4-43 に状態移行 BtoQ のフローチャートを示します。

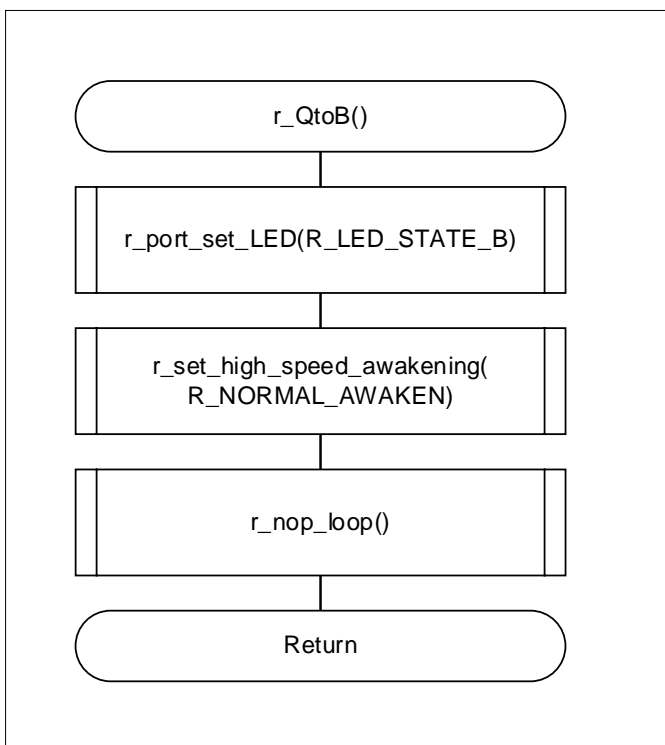
図 4-43 状態移行 BtoQ



## 4.6.42 状態移行 QtoB

図 4-44 に状態移行 QtoB のフローチャートを示します。

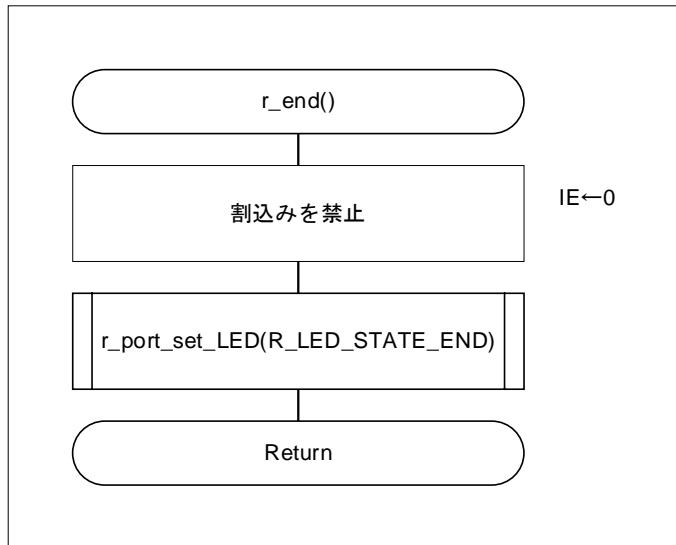
図 4-44 状態移行 QtoB



## 4.6.43 状態終了処理

図 4-45 に状態終了処理のフローチャートを示します。

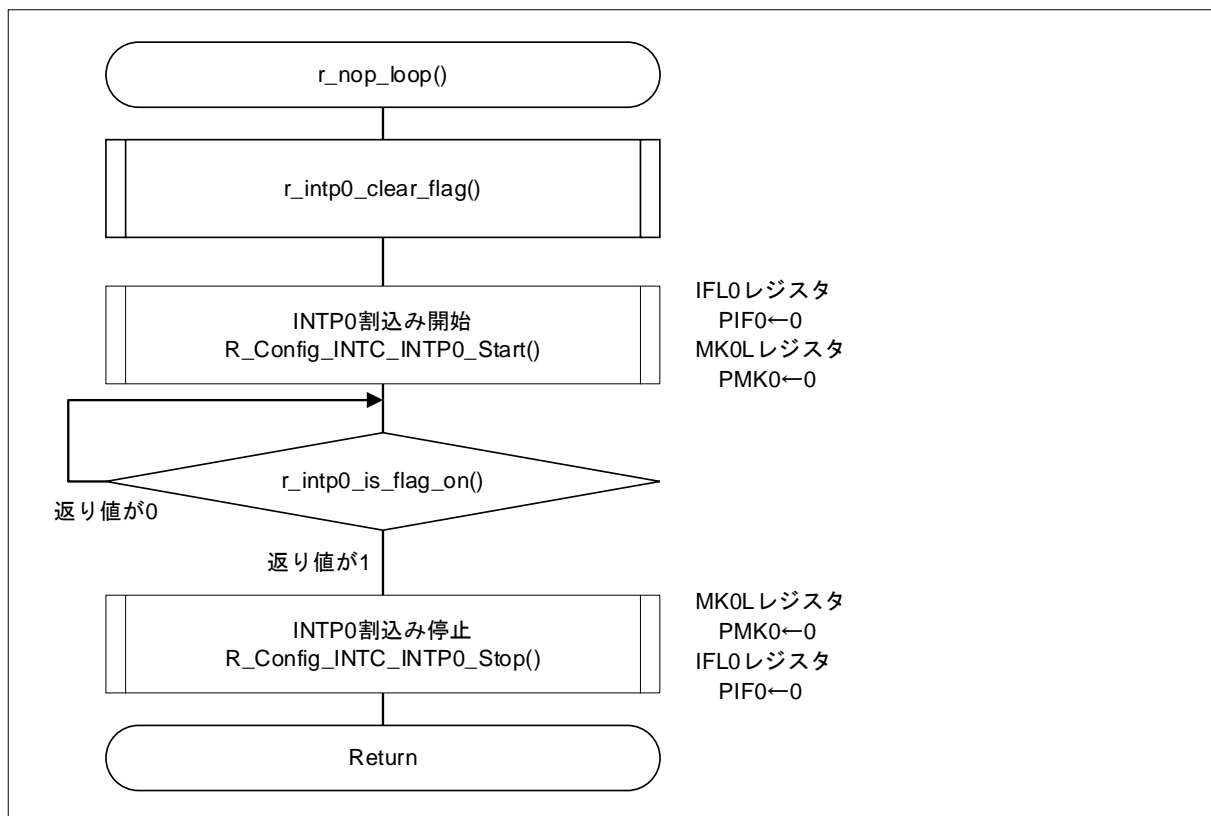
図 4-45 状態終了処理



## 4.6.44 割り込み待ち (NOP 繰り返し)

図 4-46 に割り込み待ち (NOP 繰り返し) のフローチャートを示します。

図 4-46 割り込み待ち (NOP 繰り返し)

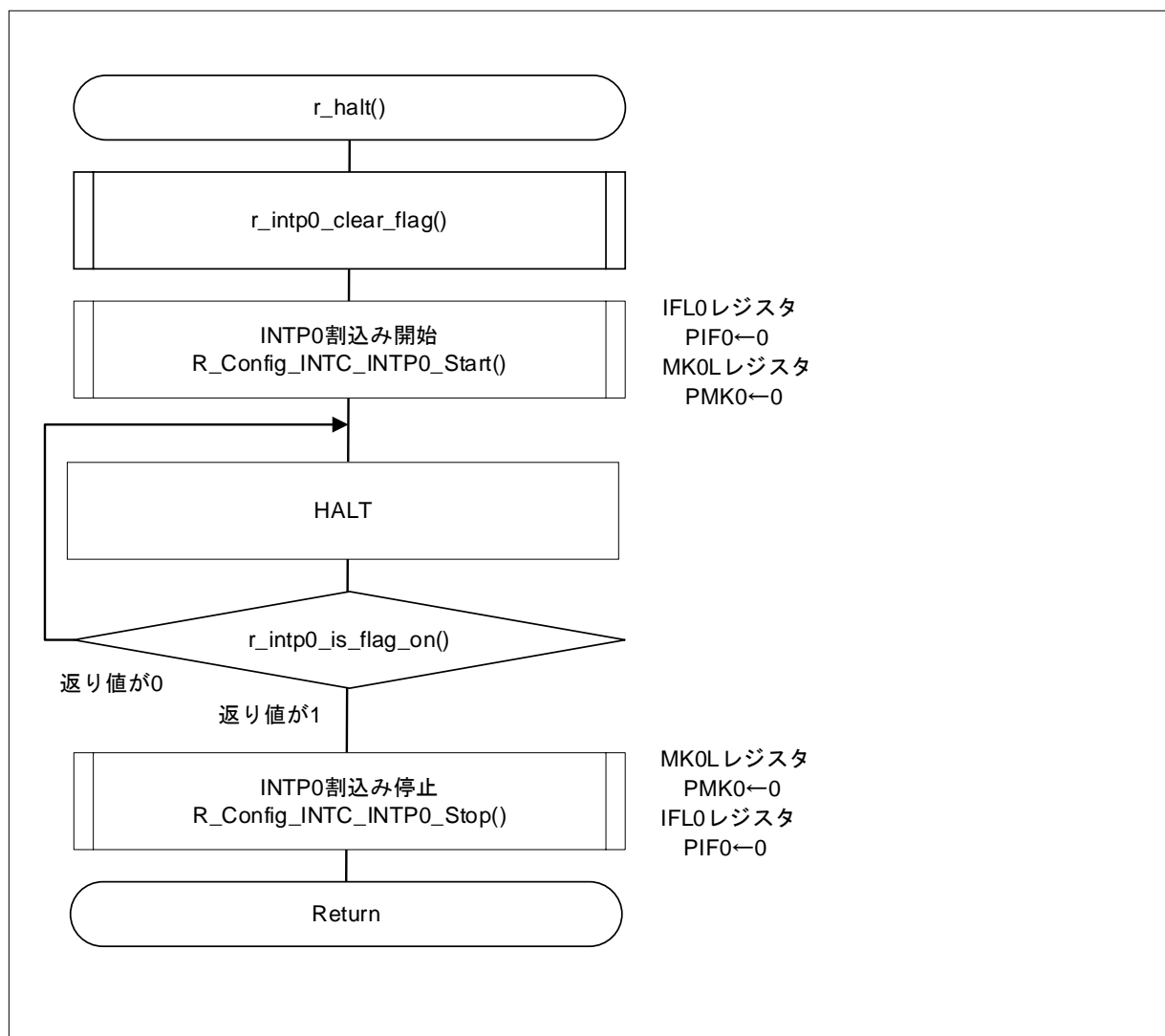




## 4.6.45 割込み待ち (HALT)

図 4-47 に割込み待ち (HALT) のフローチャートを示します。

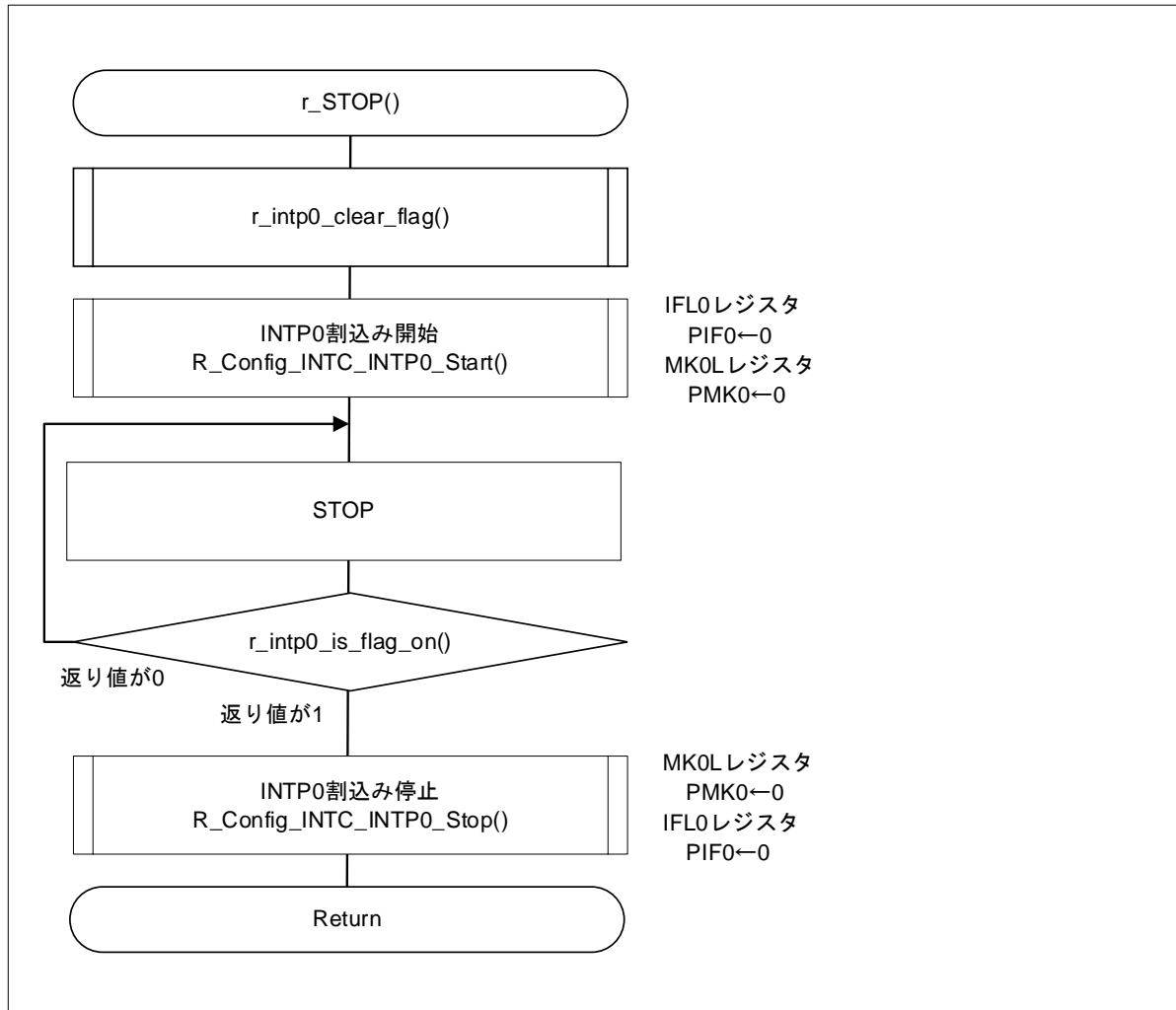
図 4-47 割込み待ち (HALT)



## 4.6.46 割込み待ち (STOP)

図 4-48 に割込み待ち (STOP) のフローチャートを示します。

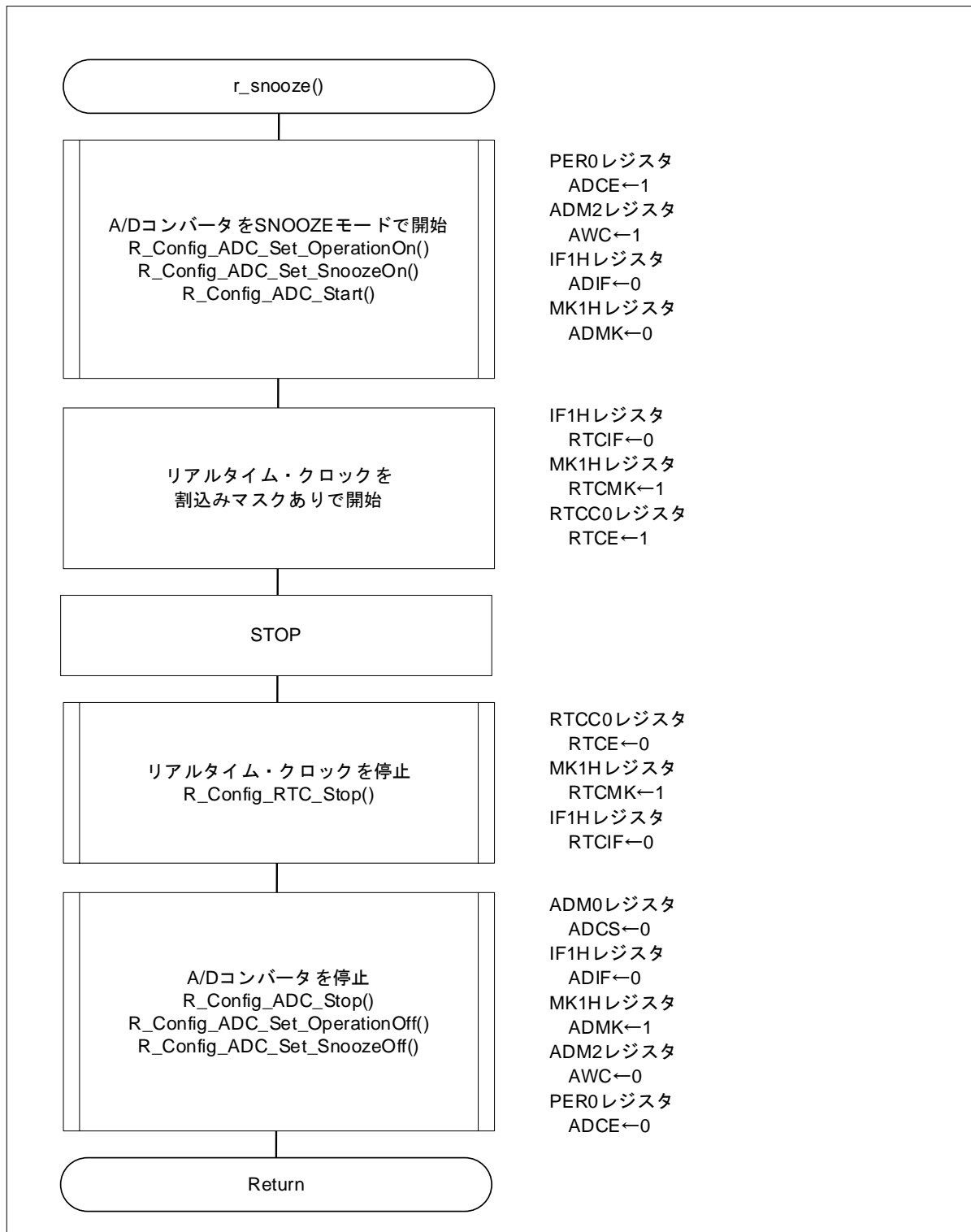
図 4-48 割込み待ち (STOP)



## 4.6.47 割込み待ち (SNOOZE)

図 4-49 に割込み待ち (SNOOZE) のフローチャートを示します。

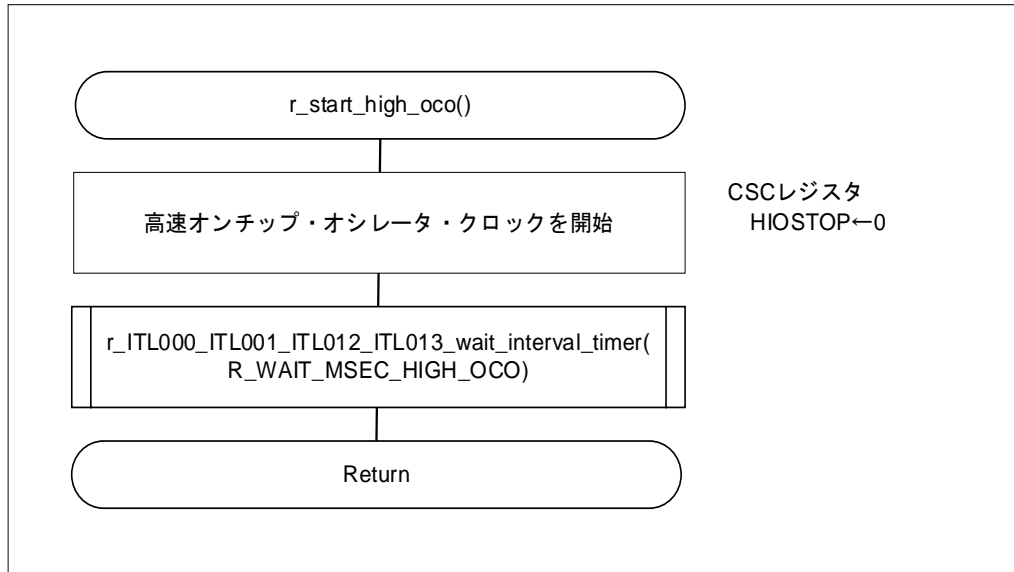
図 4-49 割込み待ち (SNOOZE)



## 4.6.48 高速オンチップ・オシレータ・クロック動作開始

図 4-50 に高速オンチップ・オシレータ・クロック動作開始のフローチャートを示します。

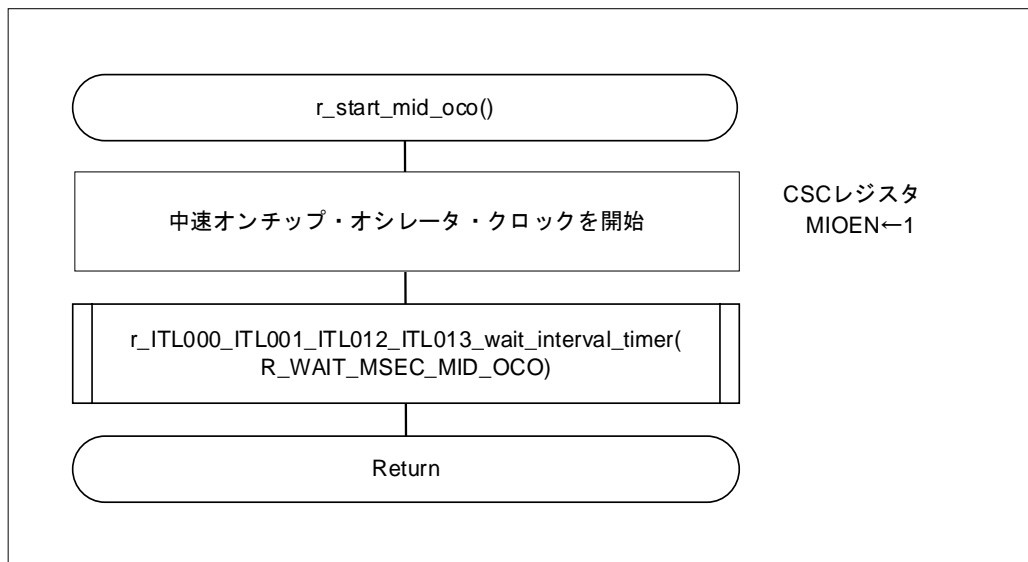
図 4-50 高速オンチップ・オシレータ・クロック動作開始



## 4.6.49 中速オンチップ・オシレータ・クロック動作開始

図 4-51 に中速オンチップ・オシレータ・クロック動作開始のフローチャートを示します。

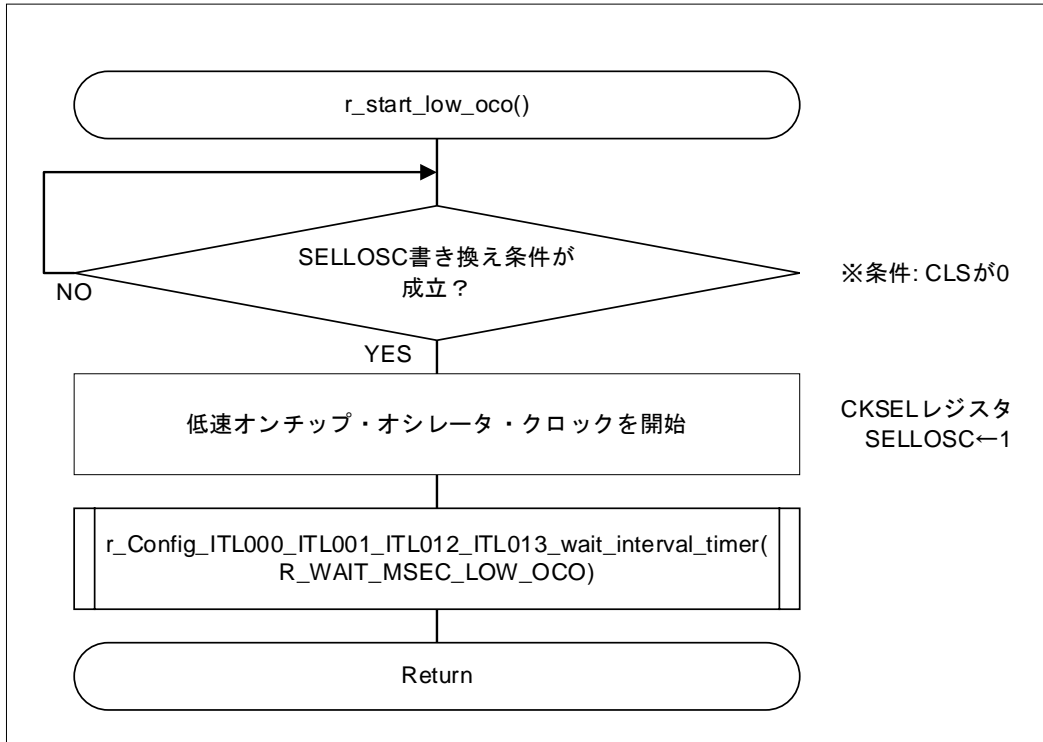
図 4-51 中速オンチップ・オシレータ・クロック動作開始



## 4.6.50 低速オンチップ・オシレータ・クロック動作開始

図 4-52 に低速オンチップ・オシレータ・クロック動作開始のフローチャートを示します。

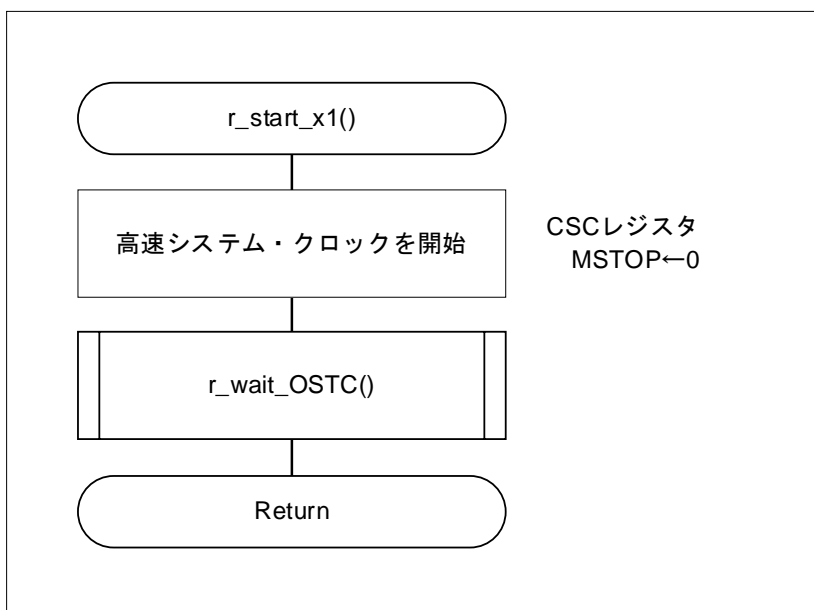
図 4-52 低速オンチップ・オシレータ・クロック動作開始



## 4.6.51 高速システム・クロック動作開始

図 4-53 に高速システム・クロック動作開始のフローチャートを示します。

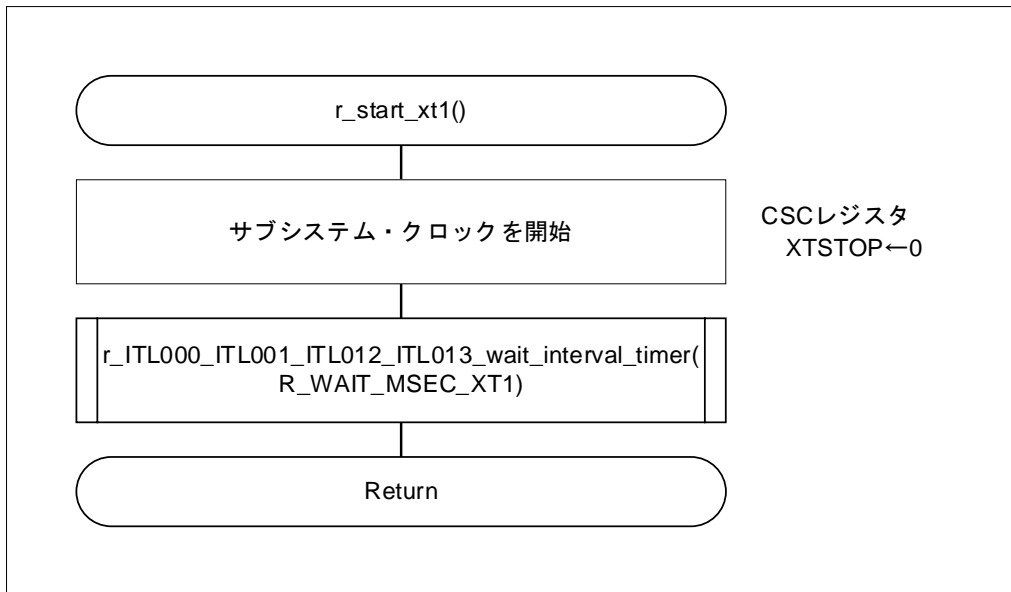
図 4-53 高速システム・クロック動作開始



## 4.6.52 サブシステム・クロック動作開始

図 4-54 にサブシステム・クロック動作開始のフローチャートを示します。

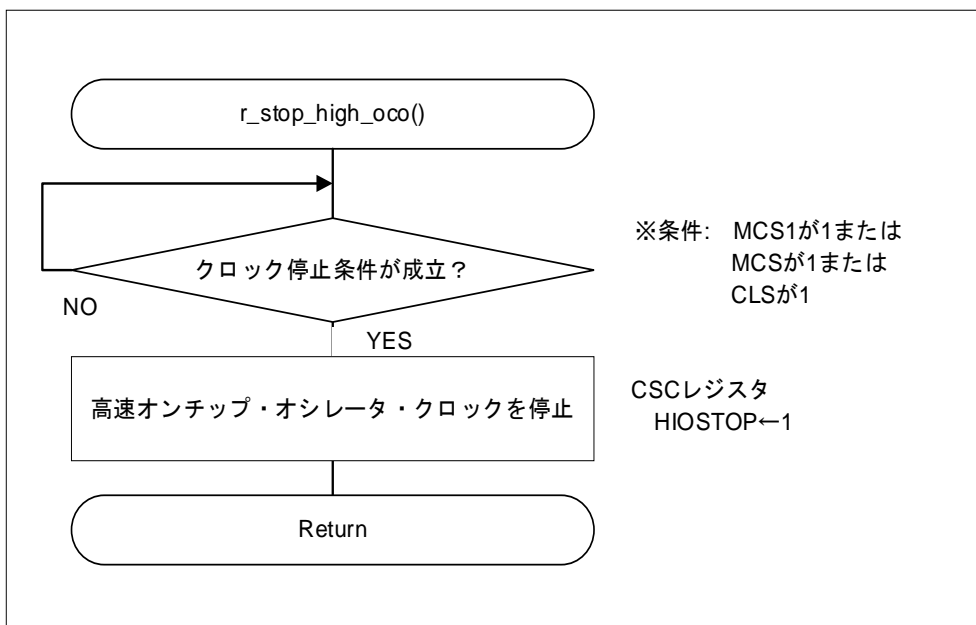
図 4-54 サブシステム・クロック動作開始



## 4.6.53 高速オンチップ・オシレータ・クロック停止

図 4-55 に高速オンチップ・オシレータ・クロック停止のフローチャートを示します。

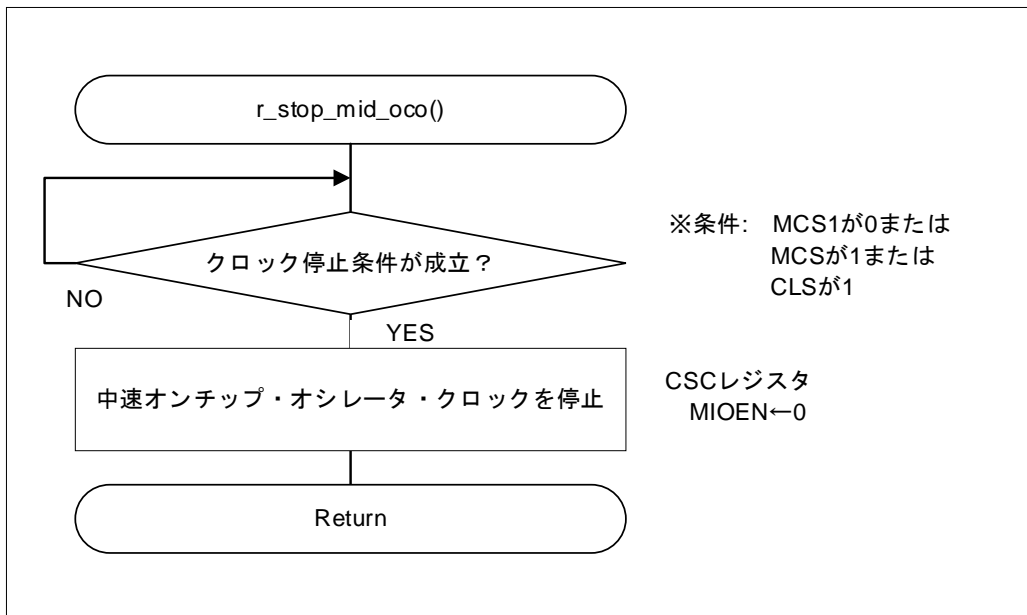
図 4-55 高速オンチップ・オシレータ・クロック停止



## 4.6.54 中速オンチップ・オシレータ・クロック停止

図 4-56 に中速オンチップ・オシレータ・クロック停止のフローチャートを示します。

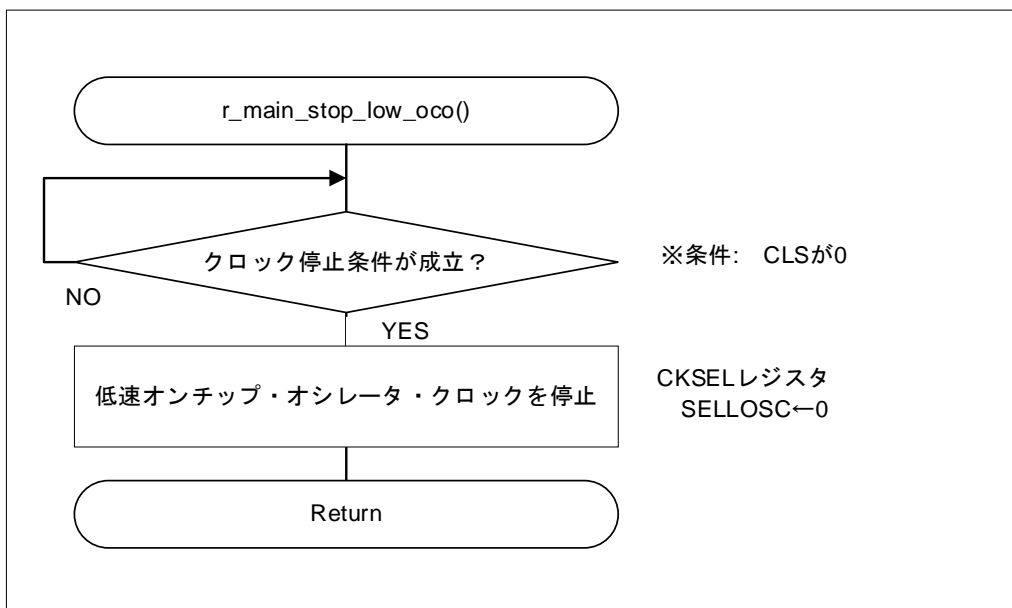
図 4-56 中速オンチップ・オシレータ・クロック停止



## 4.6.55 低速オンチップ・オシレータ・クロック停止

図 4-57 に低速オンチップ・オシレータ・クロック停止のフローチャートを示します。

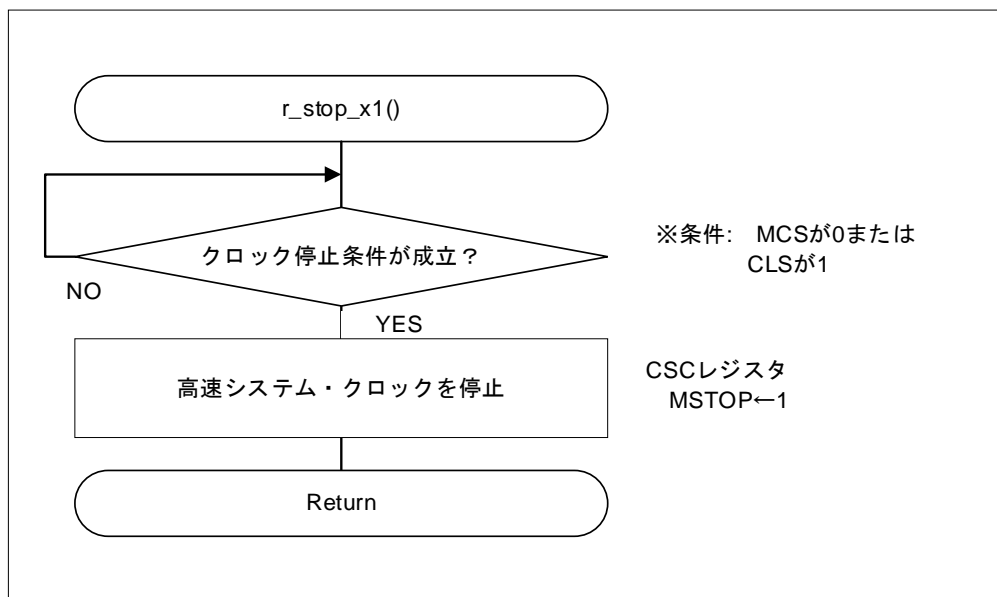
図 4-57 低速オンチップ・オシレータ・クロック停止



## 4.6.56 高速システム・クロック停止

図 4-58 に高速システム・クロック停止のフローチャートを示します。

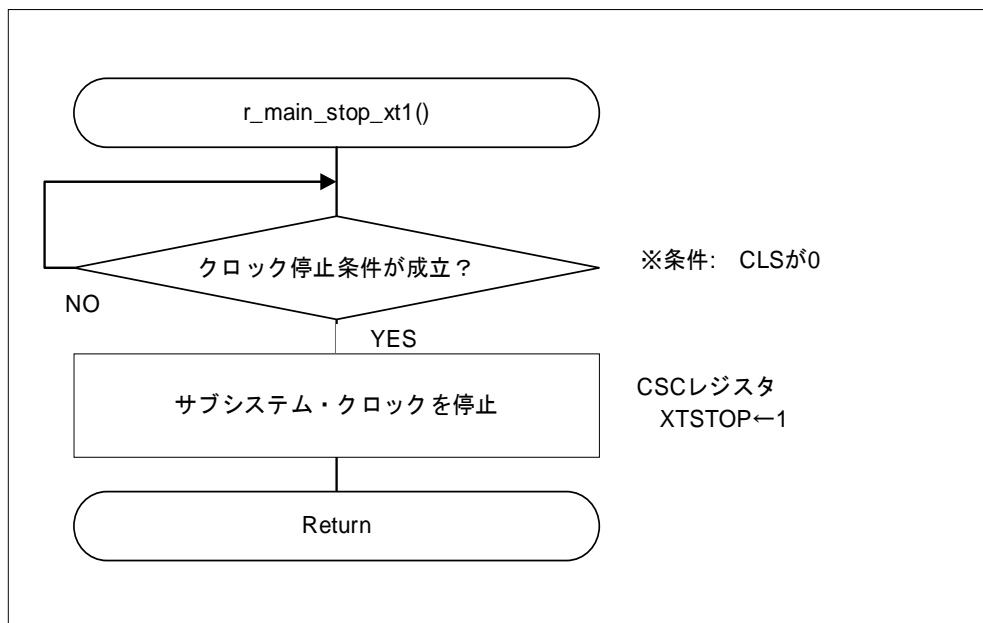
図 4-58 高速システム・クロック停止



## 4.6.57 サブシステム・クロック停止

図 4-59 にサブシステム・クロック停止のフローチャートを示します。

図 4-59 サブシステム・クロック停止





## 4.6.58 クロック切り替え設定

図 4-60、図 4-61、図 4-62、図 4-63 にクロック切り替え設定のフローチャートを示します。

図 4-60 クロック切り替え設定 (1/4)

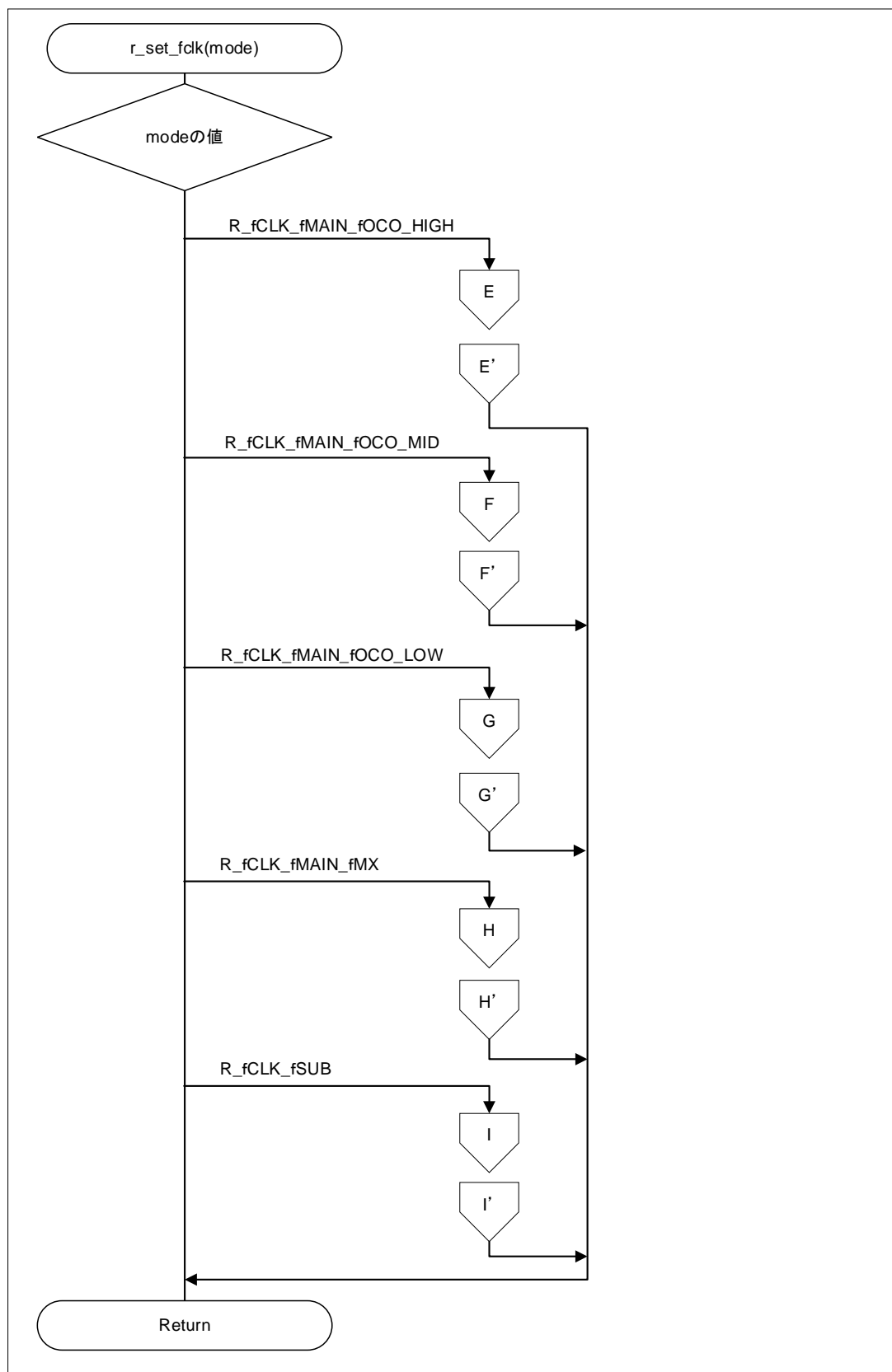


図 4-61 クロック切り替え設定 (2/4)

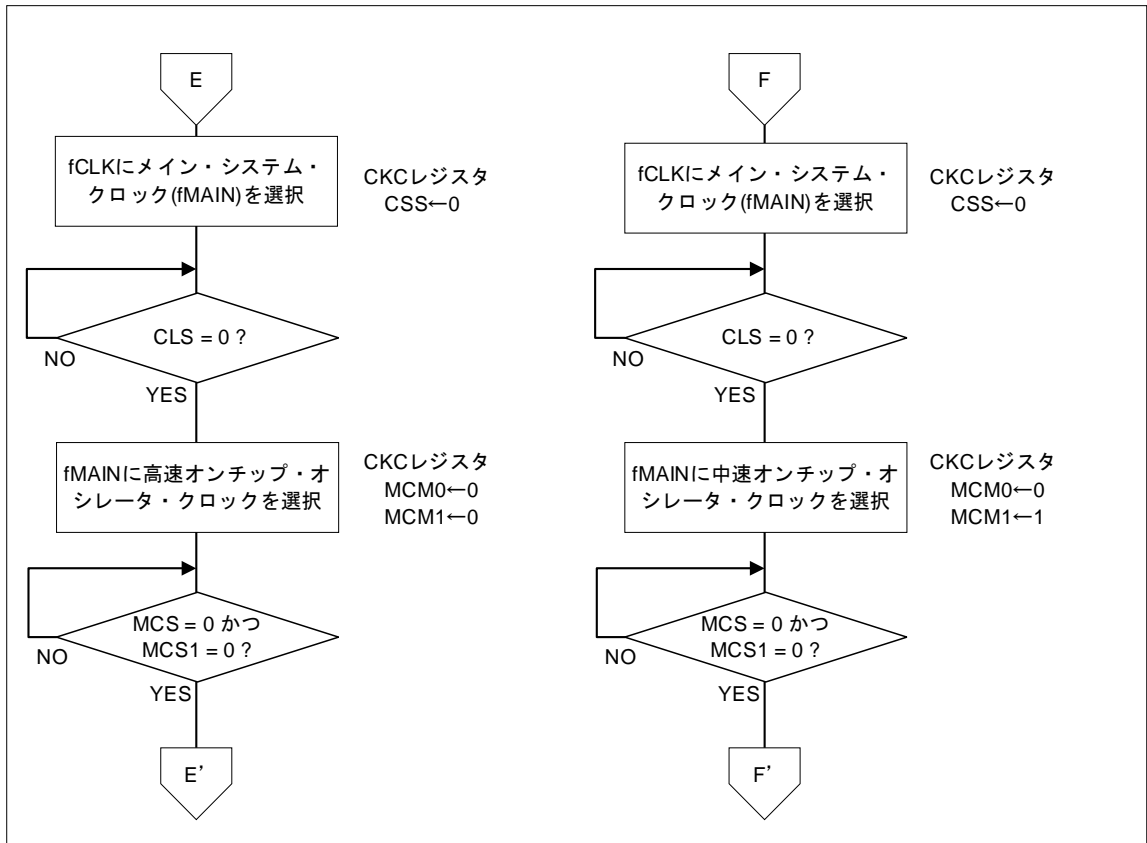


図 4-62 クロック切り替え設定 (3/4)

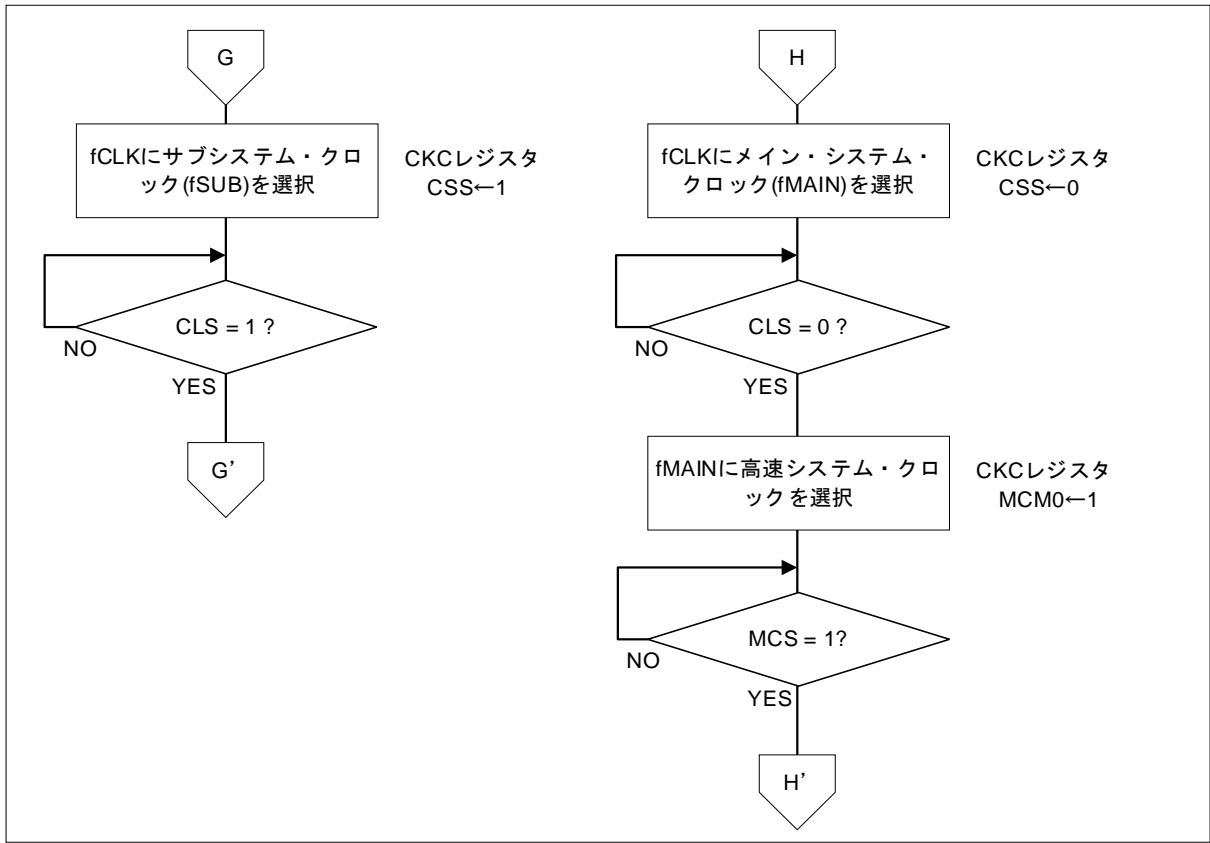
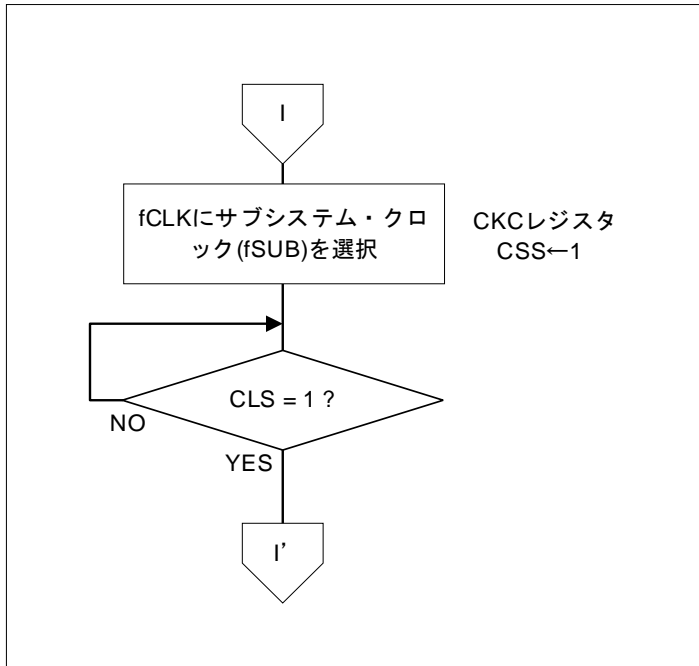


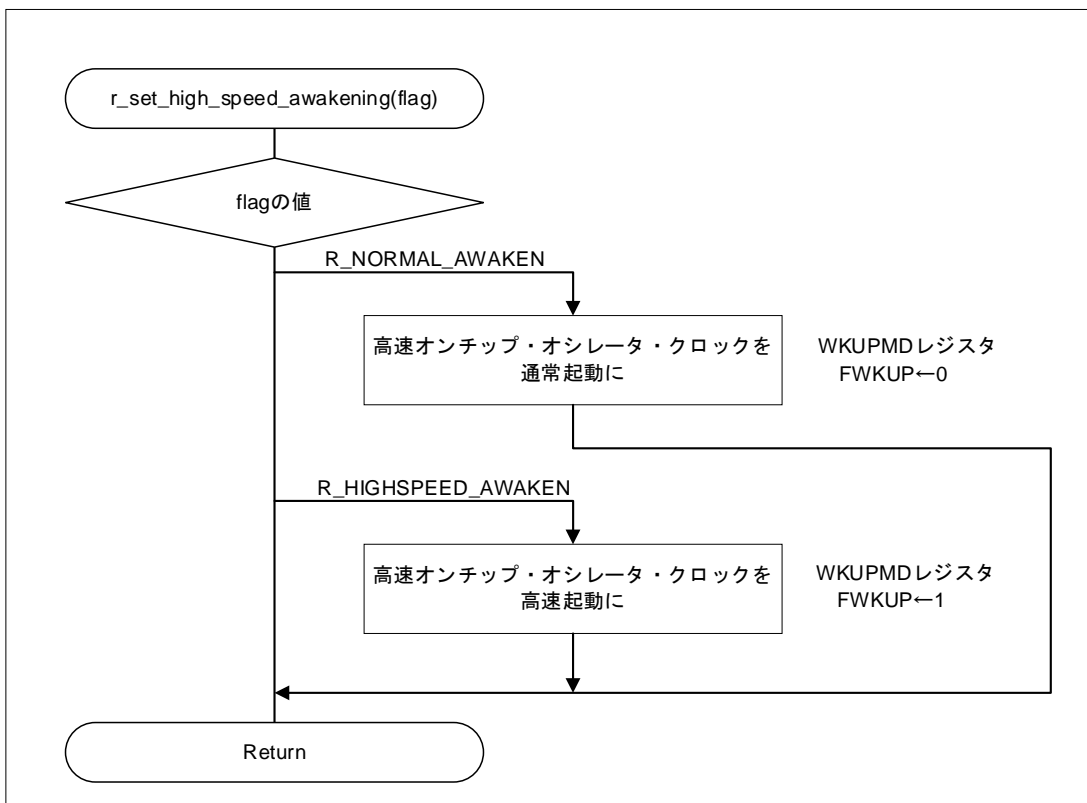
図 4-63 クロック切り替え設定 (4/4)



## 4.6.59 高速オンチップ・オシレータ・クロック起動モード設定

図 4-64 に高速オンチップ・オシレータ・クロック起動モード設定のフローチャートを示します。

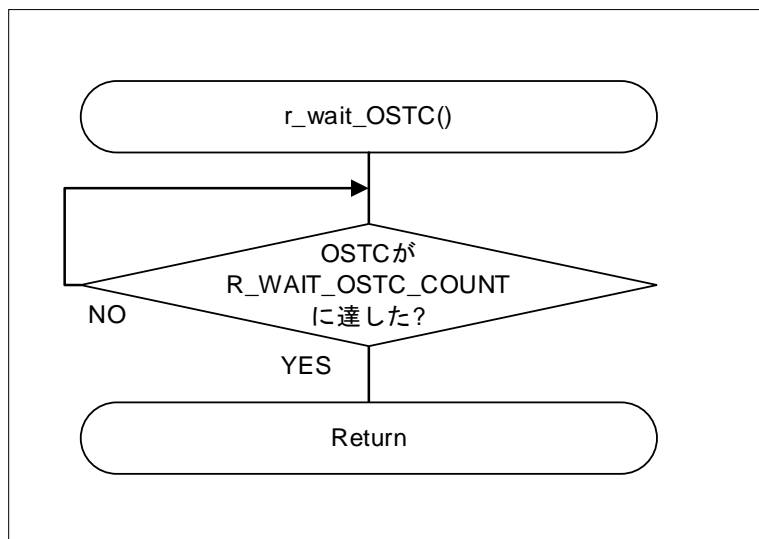
図 4-64 高速オンチップ・オシレータ・クロック起動モード設定



## 4.6.60 ウェイト処理 (OSTC レジスタ)

図 4-65 にウェイト処理 (OSTC レジスタ) のフローチャートを示します。

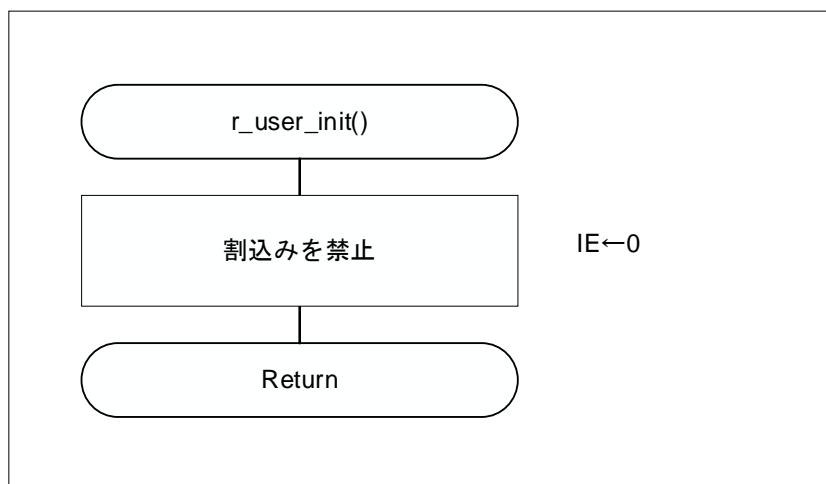
図 4-65 ウェイト処理 (OSTC レジスタ)



## 4.6.61 メイン初期化処理 (ユーザ定義箇所)

図 4-66 にメイン初期化処理 (ユーザ定義箇所) のフローチャートを示します。

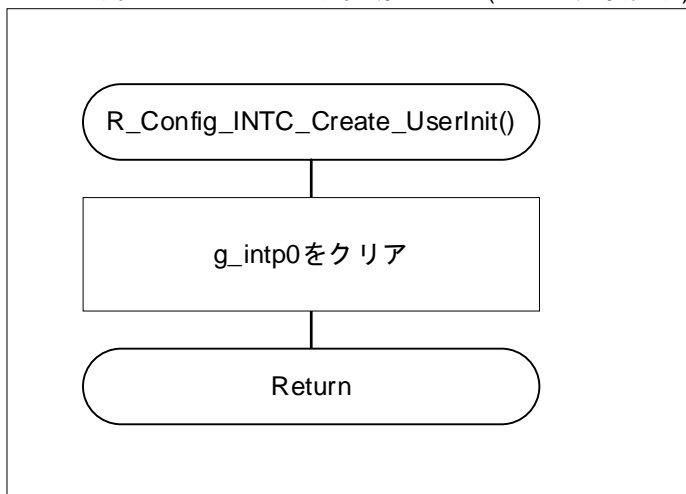
図 4-66 メイン初期化処理 (ユーザ定義箇所)



## 4.6.62 割込みコントローラ初期化処理 (ユーザ定義箇所)

図 4-67 に割込みコントローラ初期化処理 (ユーザ定義箇所) のフローチャートを示します。

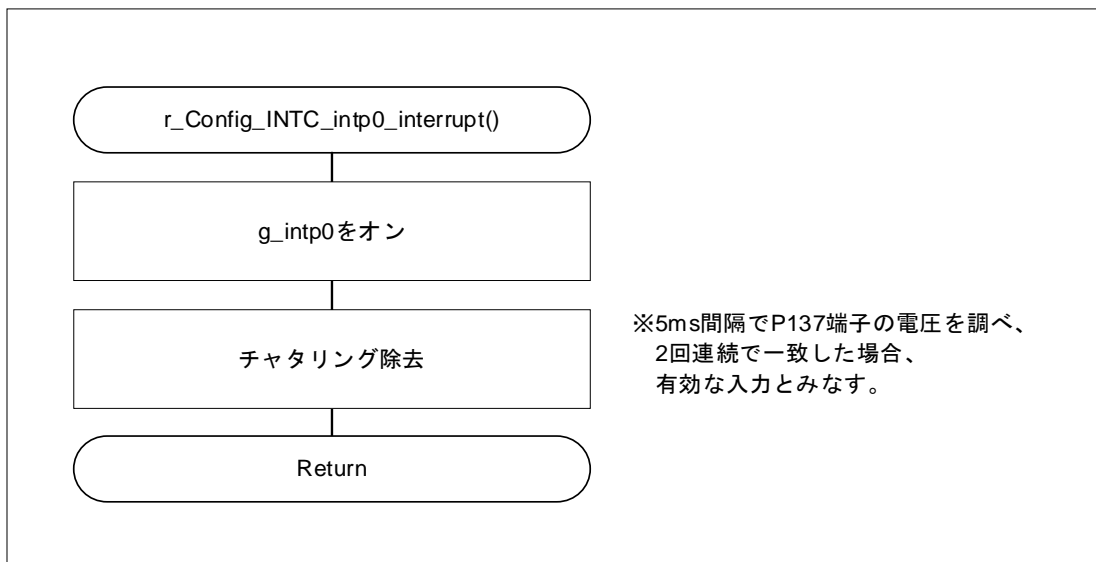
図 4-67 割込みコントローラ初期化処理 (ユーザ定義箇所)



## 4.6.63 INTP0 割込み発生時処理

図 4-68 に INTP0 割込み発生時処理のフローチャートを示します。

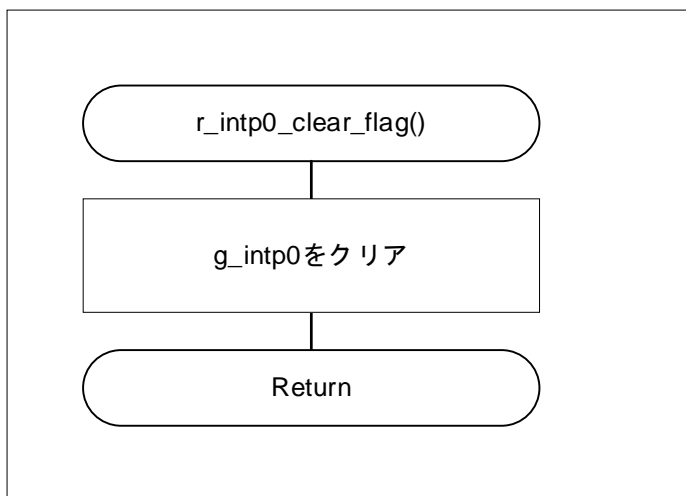
図 4-68 INTP0 割込み発生時処理



## 4.6.64 INTP0 割込みフラグクリア

図 4-69 に INTP0 割込みフラグクリアのフローチャートを示します。

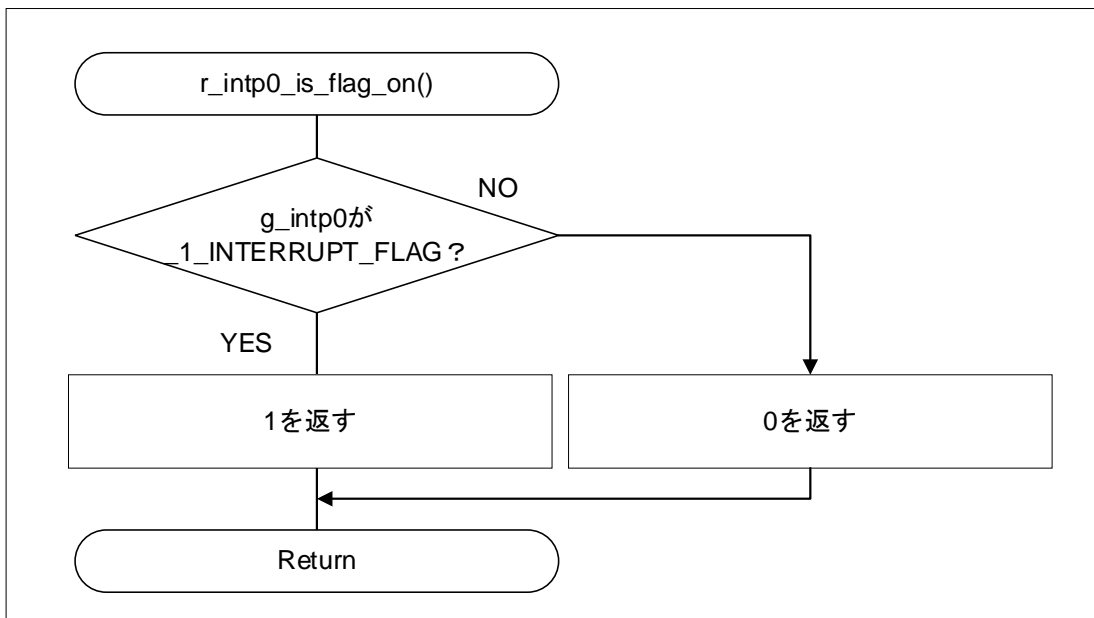
図 4-69 INTP0 割込みフラグクリア



## 4.6.65 INTP0 割込みフラグ確認

図 4-70 に INTP0 割込みフラグ確認のフローチャートを示します。

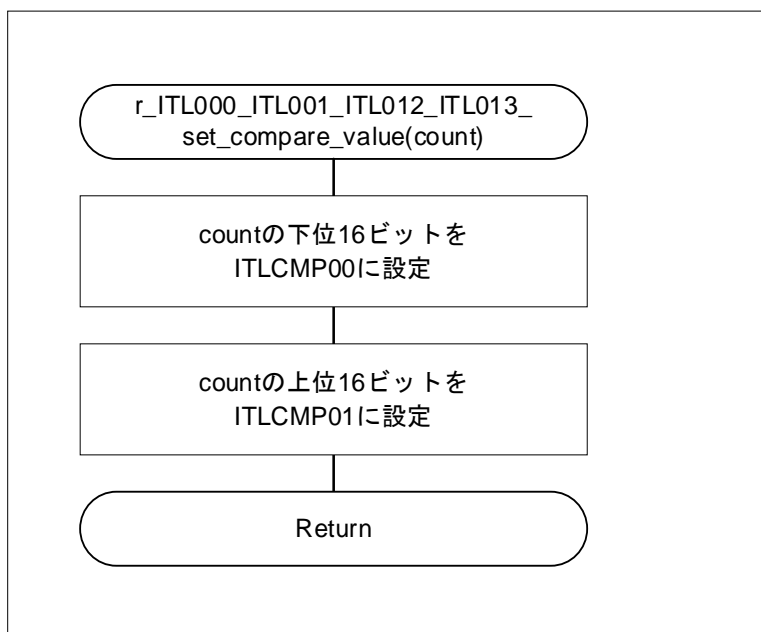
図 4-70 INTP0 割込みフラグ確認



## 4.6.66 32 ビット・インターバル・タイマコンペア値設定

図 4-71 に 32 ビット・インターバル・タイマコンペア値設定のフローチャートを示します。

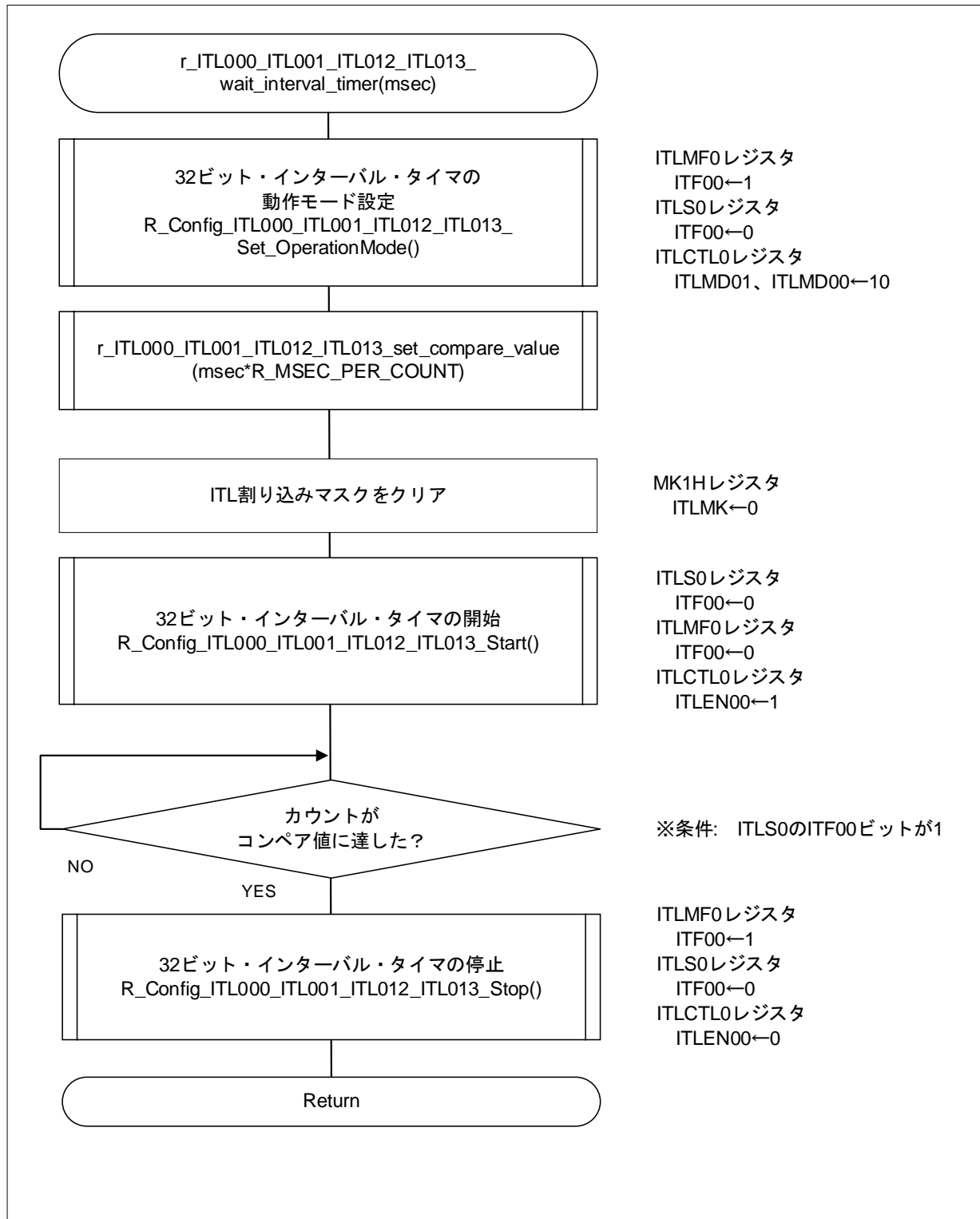
図 4-71 32 ビット・インターバル・タイマコンペア値設定



## 4.6.67 ウェイト処理 (32 ビット・インターバル・タイマ)

図 4-72 にウェイト処理 (32 ビット・インターバル・タイマ) のフローチャートを示します。

図 4-72 ウェイト処理 (32 ビット・インターバル・タイマ)

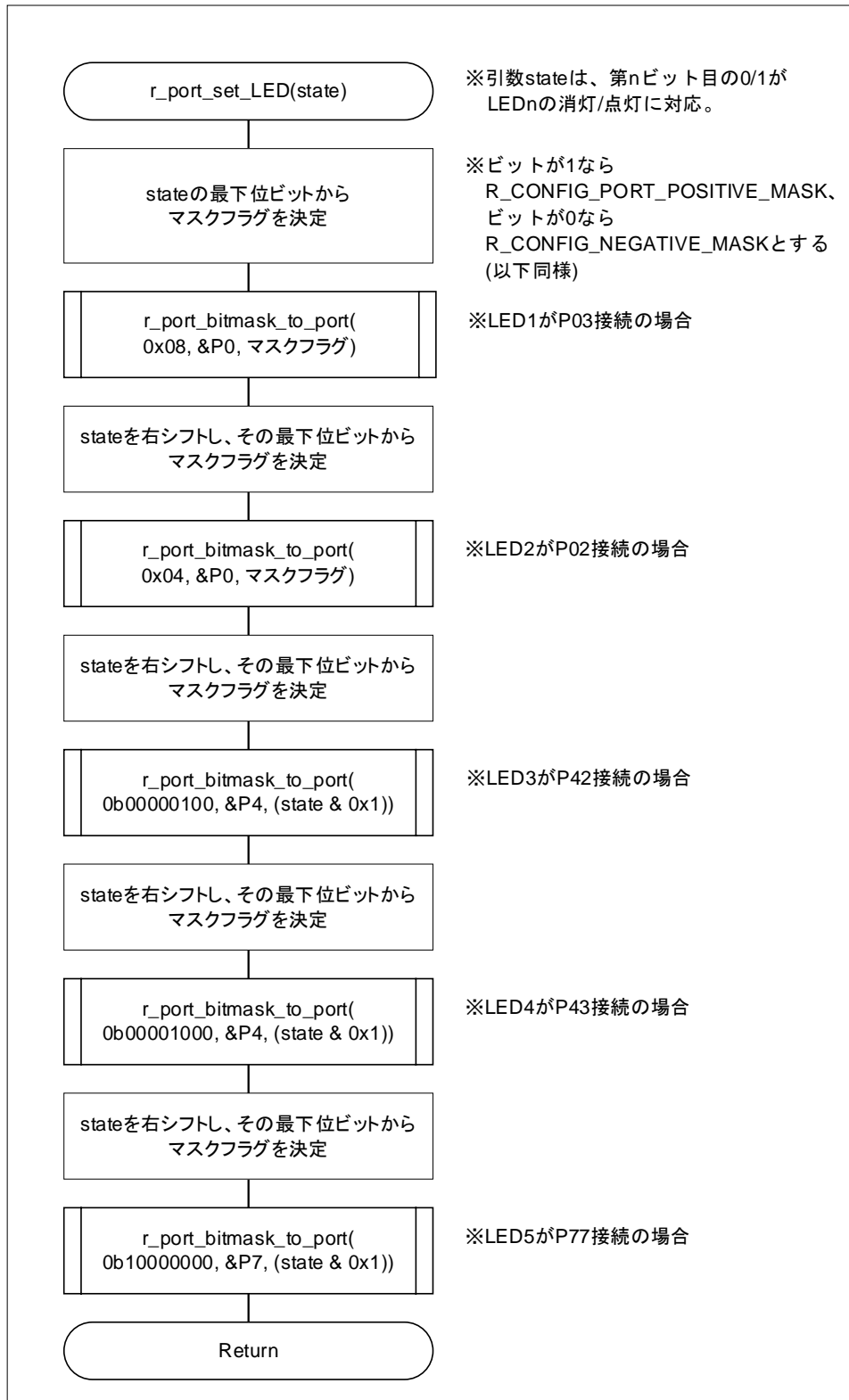




## 4.6.68 LED 点灯/消灯制御

図 4-73 に LED 点灯/消灯制御のフローチャートを示します。

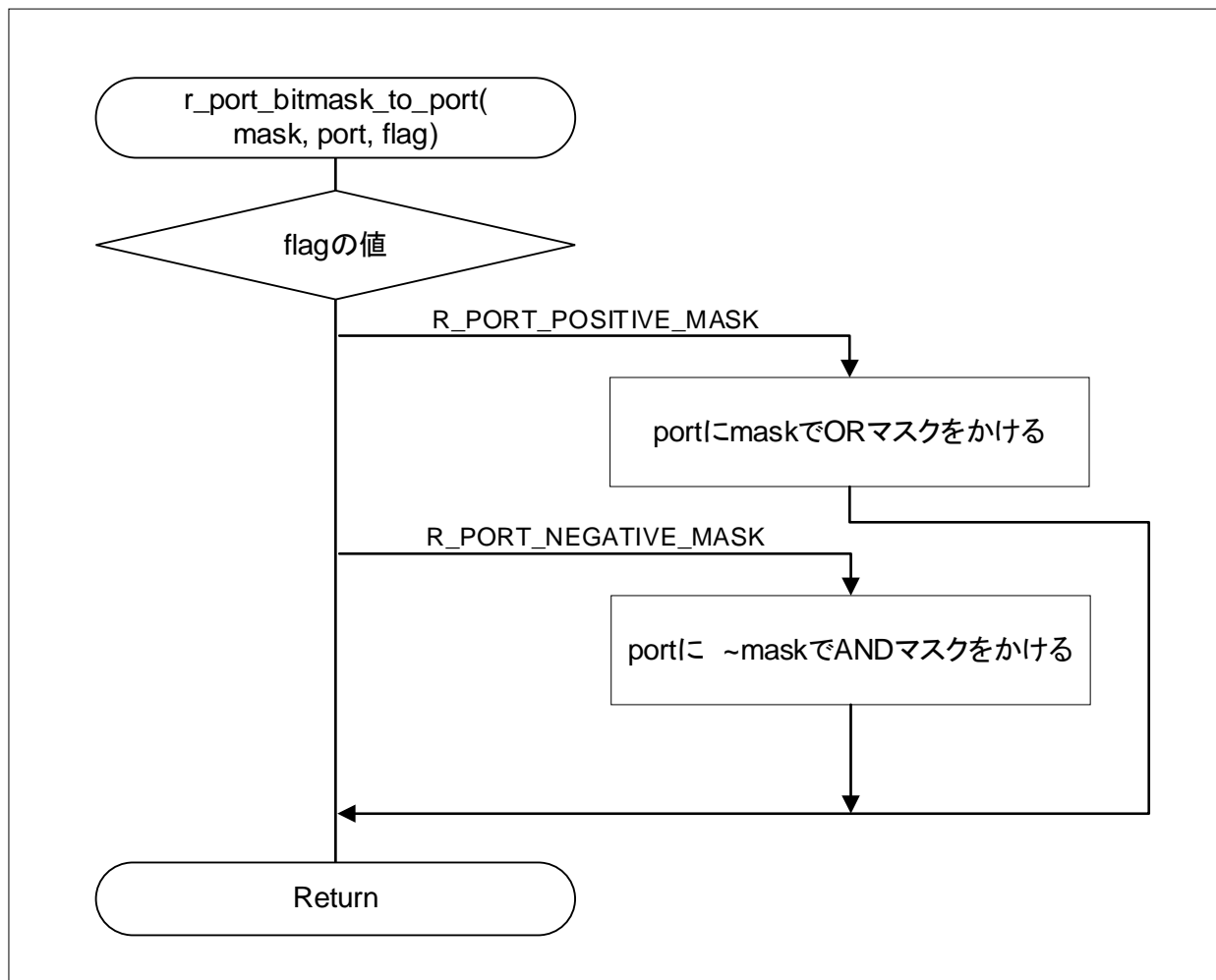
図 4-73 LED 点灯/消灯制御



## 4.6.69 ポート出力制御

図 4-74 にポート出力制御のフローチャートを示します。

図 4-74 ポート出力制御



## 5. サンプルコード

サンプルコードは、ルネサスエレクトロニクスホームページから入手してください。

## 6. 参考ドキュメント

RL78/G23 ユーザーズマニュアルハードウェア編 (R01UH0896J)

RL78 ファミリユーザーズマニュアルソフトウェア編 (R01US0015J)

(最新版をルネサスエレクトロニクスホームページから入手してください。)

テクニカルアップデート

(最新の情報をルネサスエレクトロニクスホームページから入手してください。)

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	2021.04.13	—	初版発行
1.01	2021.07.12	38	動作確認条件を更新
1.02	2023.10.6	38	動作確認条件を更新

## 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. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

### 2. 電源投入時の処置

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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

### 4. 未使用端子の処理

未使用端子は、「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。

### 5. クロックについて

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。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

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、将来の拡張機能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。

### 8. 製品間の相違について

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えば、フラッシュメモリ、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。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、お客様の責任において、お客様の機器・システムを設計ください。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まれます。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、変更、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、変更、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。
6. 当社は、当社製品の品質水準を「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、家電、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）、交通管制（信号）、大規模通信機器、金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も、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. 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いたします。
13. 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。
14. 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な点がございましたら、当社の営業担当者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。

(Rev.5.0-1 2020.10)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレストシア）

[www.renesas.com](http://www.renesas.com)

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。