

RL78/G13

クロック発生回路（クロック切り替え）CC-RL

R01AN2831JJ0100

Rev. 1.00

2015.05.28

要旨

本アプリケーションノートでは、RL78/G13 のクロック発生回路の使用方法について説明します。

スイッチが押される毎にクロック発生回路のクロック切り替えを行います。CPU/周辺ハードウェア・クロック (fCLK) として、高速オンチップ・オシレータ・クロック (32MHz)、X1 発振クロック (20MHz)、XT1 発振クロック (32.768kHz) を使用します。

対象デバイス

RL78/G13 (40 ピン以上)

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	3
2. 動作確認条件	5
3. 関連アプリケーションノート	5
4. ハードウェア説明	6
4.1 ハードウェア構成例	6
4.2 使用端子一覧	7
5. ソフトウェア説明	8
5.1 動作概要	8
5.2 ファイル構成	9
5.3 オプション・バイトの設定一覧	10
5.4 定数一覧	11
5.5 変数一覧	12
5.6 関数一覧	12
5.7 関数仕様	13
5.8 フローチャート	20
5.8.1 初期化設定関数	21
5.8.2 システム関数	22
5.8.3 入出力ポートの設定	23
5.8.4 CPU クロックの初期設定	24
5.8.5 TAU0 の設定	28
5.8.6 12 ビット・インターバル・タイマの設定	29
5.8.7 外部割り込み入力の設定	30
5.8.8 メイン処理	31
5.8.9 INTP0 外部割り込み動作開始設定	32
5.8.10 TAU0 チャンネル0 の動作開始設定	33
5.8.11 クロック切り替え	34
5.8.12 HOCO クロックから X1 発振クロック切り替え処理	35
5.8.13 X1 発振クロックから XT1 発振クロック発振切り替え処理	36
5.8.14 XT1 発振クロックから HOCO クロック切り替え処理	37
5.8.15 クロック・ステータス取得	38
5.8.16 X1 発振クロック・ステータス取得	39
5.8.17 XT1 発振クロック・ステータス取得	40
5.8.18 HOCO クロック・ステータス取得	41
5.8.19 クロック停止	42
5.8.20 TAU0 チャンネル0 パラメータ取得	43
5.8.21 TAU0 チャンネル0 リスタート	44
5.8.22 TAU0 チャンネル0 のインターバル変更	45
5.8.23 TAU0 チャンネル0 の動作停止設定	46
5.8.24 TAU0 チャンネル0 インターバル・タイマ割り込み	47
5.8.25 INTP0 外部割り込み	48
5.8.26 1ms 単位のウエイト	49
6. サンプルコード	50
7. 参考ドキュメント	50

1. 仕様

ターゲットボードのスイッチ押下を検出すると、以下の順に動作クロックを切り替えます。

- ① 高速オンチップ・オシレータ・クロック（32MHz） → X1 発振クロック（20MHz）
 - ② X1 発振クロック（20MHz） → XT1 発振クロック（32.768kHz）
 - ③ XT1 発振クロック（32.768kHz） → 高速オンチップ・オシレータ・クロック（32MHz）
- 以降、①～③の処理を繰り返します。

動作状態に応じて以下の処理を実行します。

- ・高速オンチップ・オシレータ・クロック（HOCO クロック）動作時：X1 発振クロック停止
- ・X1 発振クロック動作時：HOCO クロック停止
- ・XT1 発振クロック動作時：X1 発振クロック停止、HOCO クロック停止

なお、XT1 発振クロックは常時発振します。

また、動作クロックに応じて、ターゲットボードの LED 点滅周期を以下のように変更します。動作中のクロックが目視で判断できます。

HOCO クロック（32MHz）時の LED 点滅周期	: 0.5 秒
X1 発振クロック（20MHz）時の LED 点滅周期	: 1 秒
XT1 発振クロック（32.768kHz）時の LED 点滅周期	: 2 秒

表 1.1に使用する周辺機能と用途 を、図 1.1にクロック切り替えの動作概要を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
クロック発生回路	クロック発振と動作クロック切り替え
外部割り込み入力（INTP0）	スイッチ押下を検出
タイマ・アレイ・ユニット 0 チャンネル 0	LED 点滅周期のタイミング生成
12 ビット・インターバル・タイマ	チャタリング対策のウェイト時間生成
P62	LED 出力

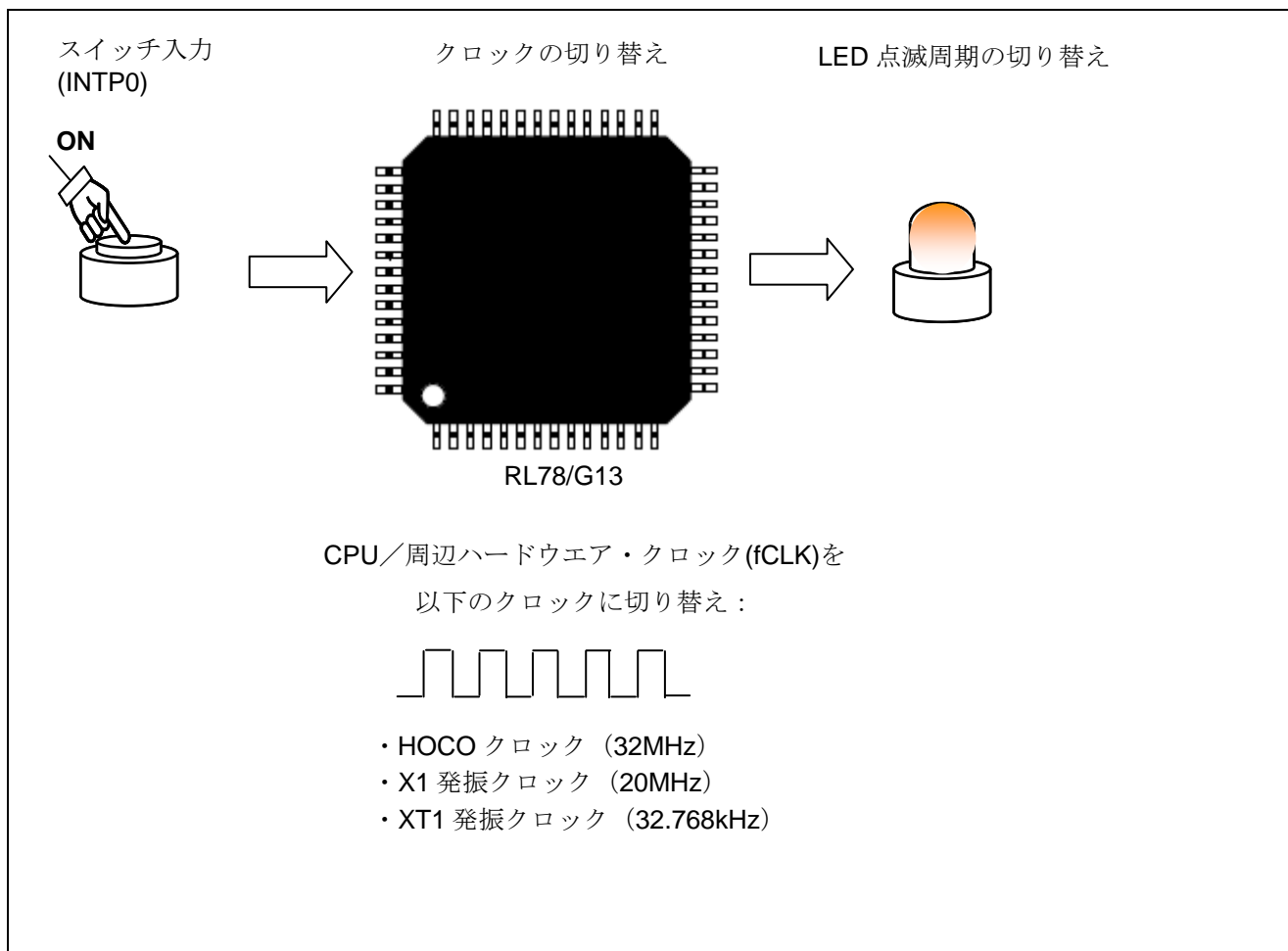


図 1.1 クロック切り替えの動作概要

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、下記の条件で動作を確認しています。

表 2.1 動作確認条件

項目	内容
使用マイコン	RL78/G13 (R5F100LEA)
動作周波数	<ul style="list-style-type: none"> ● CPU/周辺ハードウェア・クロック : ターゲットボードのスイッチ押下により切り替え HOCO クロック選択時 : 32MHz X1 発振クロック選択時 : 20MHz XT1 発振クロック選択時 : 32.768kHz
動作電圧	5.0V (2.9V~5.5V で動作可能) LVD 動作 (V_{LVD}) : リセット・モード 2.81V (2.76V~2.87V)
統合開発環境 (CS+)	ルネサス エレクトロニクス製 CS+ V3.01.00
C コンパイラ (CS+)	ルネサス エレクトロニクス製 CC-RL V1.01.00
統合開発環境 (e ² studio)	ルネサス エレクトロニクス製 e ² studio V4.0.0.26
C コンパイラ (e ² studio)	ルネサス エレクトロニクス製 CC-RL V1.01.00
使用ボード	RL78/G13 ターゲット・ボード (QB-R5F100LE-TB)

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

RL78/G13 初期設定 (R01AN2575J) アプリケーションノート

4. ハードウェア説明

4.1 ハードウェア構成例

図 4.1 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。

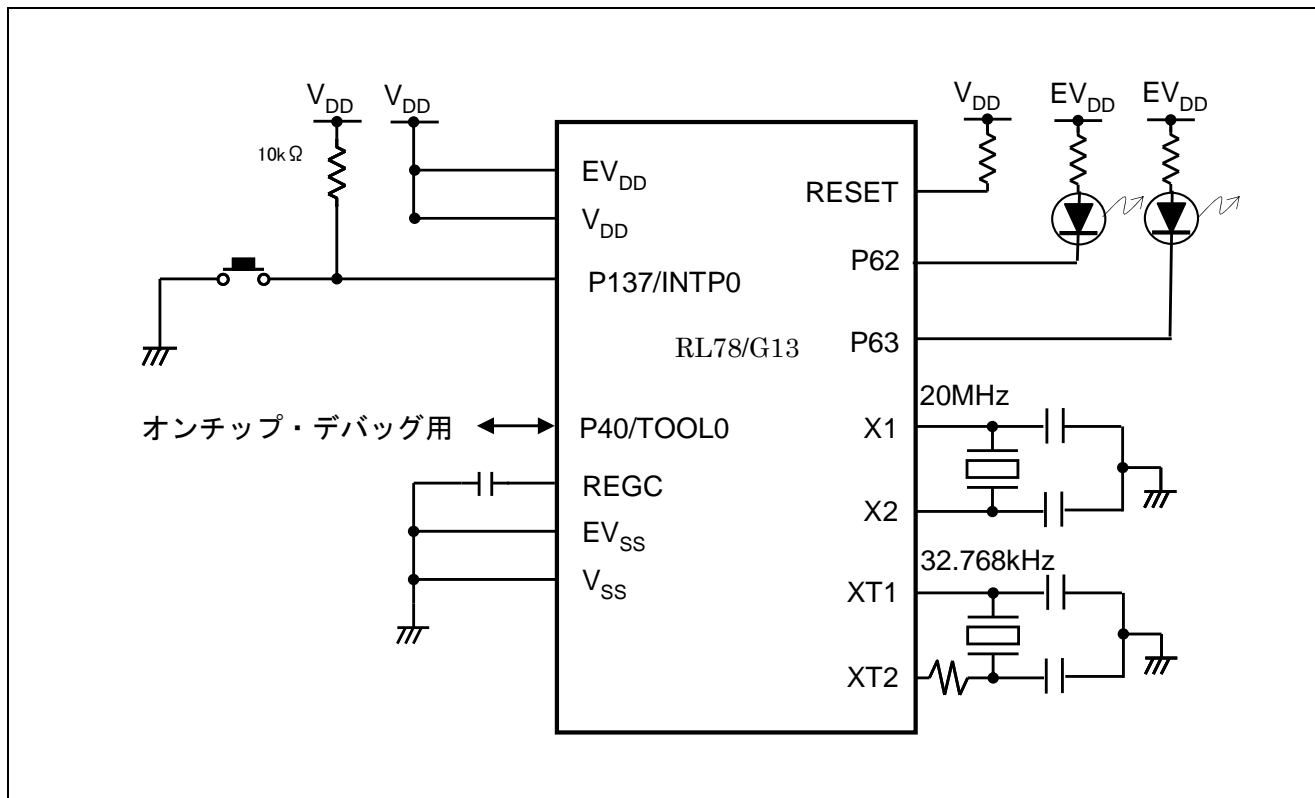


図 4.1 ハードウェア構成

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 VDD 又は VSS に接続して下さい）。

- 2 EVSS で始まる名前の端子がある場合には VSS に、EVDD で始まる名前の端子がある場合には VDD にそれぞれ接続してください。
- 3 VDD は LVD にて設定したリセット解除電圧 (V_{LVD}) 以上にしてください。
- 4 P63 に接続している LED は常時消灯します。

4.2 使用端子一覧

表 4.1に使用端子と機能を示します。

表 4.1 使用端子と機能

端子名	入出力	内容
P137/INTP0	入力	スイッチ入力
P62	出力	LED 出力

5. ソフトウェア説明

5.1 動作概要

本サンプルコードでは、ターゲットボードのスイッチ押下が検出される毎に、動作クロックの切り替えを行います。

(1) 初期設定

入出力ポート、クロック発生回路、タイマ・アレイ・ユニット 0 (TAU0)、12 ビット・インターバル・タイマ、外部割り込み入力のハードウェア初期設定を行います。初期設定終了後に割り込み処理を許可します。

LED は動作クロックに応じた TAU0 のインターバル割り込み周期で点滅します。

(2) スwitchの状態を取得

スイッチの状態を取得します。スイッチ押下が検出された場合は動作クロックの切り替えを行います。INTP0 外部割り込み発生でスイッチの状態を確認します。スイッチ押下が検出されない場合は HALT モードに遷移します。

(3) クロックの切り替え

スイッチ押下回数により、CPU/周辺ハードウェア・クロック (fCLK) を切り替えます。

以下のように CPU/周辺ハードウェア・クロック (fCLK) が切り替わります。

- ① HOCO クロック (32MHz) → X1 発振クロック (20MHz)
- ② X1 発振クロック (20MHz) → XT1 発振クロック (32.768kHz)
- ③ XT1 発振クロック (32.768kHz) → HOCO クロック (32MHz)

以降、①～③の処理を繰り返します。

(4) クロック・ステータス取得

クロック・ステータスを取得します。クロック・ステータスが変更されている場合は、動作状態に応じて以下の処理を実行します。

- ・高速オンチップ・オシレータ・クロック (HOCO クロック) 動作時 : X1 発振クロック停止
- ・X1 発振クロック動作時 : HOCO クロック停止
- ・XT1 発振クロック動作時 : X1 発振クロック停止、HOCO クロック停止

なお、XT1 発振クロックは常時発振します。

(5) LED 点滅周期変更

TAU0 の割り込みインターバルを CPU/周辺ハードウェア・クロック (fCLK) に応じて変更します。

HOCO クロック (32MHz) 時の LED 点滅周期	: 0.5 秒
X1 発振クロック (20MHz) 時の LED 点滅周期	: 1 秒
XT1 発振クロック (32.768kHz) 時の LED 点滅周期	: 2 秒

(6) HALT モードへ遷移

HALT モードに遷移します。TAU0 のインターバル割り込み、スイッチの外部割り込みで HALT モードから復帰します。HALT モードから復帰後は(2)の処理を行います。以降は(2)～(6)の処理を繰り返します。

5.2 ファイル構成

表 5.1 にサンプルコードで使用するファイルを示します。なお、統合開発環境で自動生成されるファイルは除きます。

表 5.1 ファイル構成

ファイル名	概要	備考
r_cg_cgc.c	クロック発生回路モジュール	CPU クロックの初期設定
r_cg_cgc.h	クロック発生回路モジュールの外部参照用ヘッダファイル	
r_cgc_user.c	クロック発生回路 サンプルコード独自の処理	追加関数： R_CGC_ChangeClock、 R_CGC_HOCOToX1、 R_CGC_X1ToXT1、 R_CGC_XT1ToHOCO、 R_CGC_GetClockStatus、 R_CGC_Get_X1_Status R_CGC_Get_XT1_Status、 R_CGC_Get_HOCO_Status、 R_CGC_StopClock
r_cg_intc.c	外部割り込み入力モジュール	
r_cg_intc.h	外部割り込み入力モジュールの外部参照用ヘッダファイル	
r_cg_intc_user.c	外部割り込み入力モジュール INTP0 外部割り込み	
r_cg_it.c	12 ビット・インターバル・タイマモジュール	
r_cg_it.h	12 ビット・インターバル・タイマモジュールの外部参照用ヘッダファイル	
r_cg_it_user.c	12 ビット・インターバル・タイマモジュール サンプルコード独自の処理	追加関数： R_IT_Wait_ms
r_main.c	メイン処理	
r_cg_macrodrive r.h	共通ヘッダファイル	型定義、 エラーステータスのマクロ定義
r_cg_userdefine. h	サンプルコード独自のマクロ定義	
r_cg_port.c	ポート機能モジュール	入出力ポートの設定
r_cg_port.h	ポート機能モジュールの外部参照用ヘッダファイル	
r_systeminit.c	システムモジュール	初期化設定、システム関数
r_cg_timer.c	タイマモジュール	
r_cg_timer.h	タイマモジュールの外部参照用ヘッダファイル	
r_cg_timer_user. c	タイマモジュール サンプルコード独自の処理 TAU0 チャンネル 0 割り込み	追加関数： R_TAU0_Channel0_GetParameter、 R_TAU0_Channel0_Restart、 R_TAU0_Channel0_ChangeInterval、 r_tau0_channel0_interrupt

5.3 オプション・バイトの設定一覧

表 5.2にオプション・バイト設定一覧を示します。

表 5.2 オプション・バイト設定一覧

アドレス	設定値	内容
000C0H/010C0H	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/010C1H	01111111B	LVD リセット・モード 2.81V (2.76V~2.87V)
000C2H/010C2H	11101000B	HS モード、HOCO クロック : 32MHz
000C3H/010C3H	10000100B	オンチップ・デバッグ許可 オンチップ・デバッグ・セキュリティ ID 認証失敗時にフラッシュ・メモリのデータを消去する

5.4 定数一覧

表 5.3にサンプルコードで使用する定数を示します。

表 5.3 サンプルコードで使用する定数

定数名	設定値	内容
HOCO_NEXT_STATUS_X1	1	クロック・ステータス : HOCO クロックにて動作中で、次は X1 発振クロック
X1_NEXT_STATUS_XT1	2	クロック・ステータス : X1 発振クロックにて動作中で、次は XT1 発振クロック
XT1_NEXT_STATUS_HOCO	3	クロック・ステータス : XT1 発振クロックにて動作中で、次は HOCO クロック
X1_STATUS	1	現在のステータスは X1 発振クロック動作
XT1_STATUS	2	現在のステータスは XT1 発振クロック動作
HOCO_STATUS	3	現在のステータスは HOCO クロック動作
CHATTERING_WAIT	10	チャタリング対策のウエイト時間 10ms
HOCO_LED_SETTING_CHANNEL_PRESCALER	9	HOCO クロック選択時の TAU0 チャンネル 0 動作クロック分周値
HOCO_LED_SETTING_CHANNEL_COUNT	15625	HOCO クロック選択時の TAU0 チャンネル 0 カウント値
X1_LED_SETTING_CHANNEL_PRESCALER	9	X1 発振クロック選択時の TAU0 チャンネル 0 動作クロック分周値
X1_LED_SETTING_CHANNEL_COUNT	19531	X1 発振クロック選択時の TAU0 チャンネル 0 カウント値
XT1_LED_SETTING_CHANNEL_PRESCALER	9	XT1 発振クロック選択時の TAU0 チャンネル 0 動作クロック分周値
XT1_LED_SETTING_CHANNEL_COUNT	64	XT1 発振クロック選択時の TAU0 チャンネル 0 カウント値
SWITCH_OFF	0	スイッチ押下なし
SWITCH_ON	1	スイッチ押下あり
SWITCH_ON_PORT_LEVEL	0	スイッチ ON 時の入力ポートレベル
CLOCK_NOT_OSCILLATING	0	クロックが発振していない
CLOCK_OSCILLATING	1	クロックが発振している

5.5 変数一覧

表 5.4にグローバル変数を示します。

表 5.4 グローバル変数

Type	Variable Name	Contents	Function Used
uint8_t	g_ClockStatus	クロック・ステータス	main() r_intc0_interrupt
uint8_t	g_TAU0_Channel0_Clkdiv	TAU0 チャンネル0 動作クロック分周値	main() R_TAU0_Channel0_GetParameter()
uint16_t	g_TAU0_Channel0_Count	TAU0 チャンネル0 カウンタ値	main() R_TAU0_Channel0_GetParameter()
uint8_t	g_SwitchStatus	スイッチ・ステータス	main() r_intc0_interrupt

5.6 関数一覧

表 5.5に関数を示します。

表 5.5 関数

関数名	概要
R_INTC0_Start	INTP0 外部割り込み動作開始設定
R_TAU0_Channel0_Start	TAU0 チャンネル0 の動作開始設定
R_CGC_ChangeClock	クロック切り替え
R_CGC_HOCOToX1	HOCO クロックから X1 発振クロック切り替え処理
R_CGC_X1ToXT1	X1 発振クロックから XT1 発振クロック切り替え処理
R_CGC_XT1ToHOCO	XT1 発振クロックから HOCO クロック切り替え処理
R_CGC_GetClockStatus	クロック・ステータス取得
R_CGC_Get_X1_Status	X1 発振クロック・ステータス取得
R_CGC_Get_XT1_Status	XT1 発振クロック・ステータス取得
R_CGC_Get_HOCO_Status	HOCO クロック・ステータス取得
R_CGC_StopClock	クロック停止
R_TAU0_Channel0_GetParameter	TAU0 チャンネル0 パラメータ取得
R_TAU0_Channel0_Restart	TAU0 チャンネル0 リスタート
R_TAU0_Channel0_ChangeInterval	TAU0 チャンネル0 のインターバル変更
R_TAU0_Channel0_Stop	TAU0 チャンネル0 の動作停止設定
r_tau0_channel0_interrupt	TAU0 チャンネル0 インターバル・タイマ割り込み
r_intc0_interrupt	INTP0 外部割り込み
R_IT_Wait_ms	1ms 単位のウエイト

5.7 関数仕様

サンプルコードの関数仕様を示します。

[関数名] R_INTC0_Start

概要	INTP0 外部割り込み動作開始設定
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_intc.h" #include "r_cg_userdefine.h"
宣言	void R_INTC0_Start(void)
説明	INTP0 の割り込みマスクを解除して、割り込みを許可します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_TAU0_Channel0_Start

概要	TAU0 チャンネル 0 の動作開始設定
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
宣言	void R_TAU0_Channel0_Start(void)
説明	TAU0 チャンネル 0 の割り込みマスクを解除して、カウント動作を開始します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_CGC_ChangeClock

概要	クロック切り替え
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
宣言	void R_CGC_ChangeClock(uint8_t status)
説明	クロックを切り替えます。
引数	<ul style="list-style-type: none"> 第一引数 : staus : クロック・ステータス (1-3) 以下の定数を設定します。 HOCO_NEXT_STATUS_X1 → X1 発振クロックに切り替え X1_NEXT_STATUS_XT1 → XT1 発振クロックに切り替え XT1_NEXT_STATUS_HOCO → HOCO クロックに切り替え
リターン値	なし
備考	なし

[関数名] R_CGC_HOCOToX1

概要	HOCO クロックから X1 発振クロック切り替え処理
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_cgic.h" #include "r_cg_userdefine.h"
宣言	void R_CGC_HOCOToX1 (void)
説明	CPU/周辺ハードウェア・クロック (fCLK) を HOCO クロックから X1 発振クロックに切り替えます。
引数	なし
リターン値	なし
備考	なし

[関数名] R_CGC_X1ToXT1

概要	X1 発振クロックから XT1 発振クロック切り替え処理
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_cgic.h" #include "r_cg_userdefine.h"
宣言	void R_CGC_X1ToXT1(void)
説明	CPU/周辺ハードウェア・クロック (fCLK) を X1 発振クロックから XT1 発振クロックに切り替えます。
引数	なし
リターン値	なし
備考	なし

[関数名] R_CGC_XT1ToHOCO

概要	XT1 発振クロックから HOCO クロック切り替え処理
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_cgic.h" #include "r_cg_userdefine.h"
宣言	void R_CGC_XT1ToHOCO(void)
説明	CPU/周辺ハードウェア・クロック (fCLK) を XT1 発振クロックから HOCO クロックに切り替えます。
引数	なし
リターン値	なし
備考	なし

[関数名] R_CGC_Get_X1_Status

概要	X1 発振クロック・ステータス取得
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
宣言	uint8_t R_CGC_Get_X1_Status(void)
説明	X1 発振クロック・ステータスを取得します。
引数	なし
リターン値	<ul style="list-style-type: none"> クロックが切り替わっていない場合： CLOCK_NOT_OSCILLATING (0x00) クロックが切り替わっている場合： CLOCK_OSCILLATING (0x01)
備考	なし

[関数名] R_CGC_Get_XT1_Status

概要	XT1 発振クロック・ステータス取得
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
宣言	uint8_t R_CGC_Get_XT1_Status(void)
説明	XT1 発振クロック・ステータスを取得します。
引数	なし
リターン値	<ul style="list-style-type: none"> クロックが切り替わっていない場合： CLOCK_NOT_OSCILLATING (0x00) クロックが切り替わっている場合： CLOCK_OSCILLATING (0x01)
備考	なし

[関数名] R_CGC_Get_HOCO_Status

概要	HOCO クロック・ステータス取得
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
宣言	uint8_t R_CGC_Get_HOCO_Status(void)
説明	HOCO クロック・ステータスを取得します。
引数	なし
リターン値	<ul style="list-style-type: none"> クロックが切り替わっていない場合： CLOCK_NOT_OSCILLATING (0x00) クロックが切り替わっている場合： CLOCK_OSCILLATING (0x01)
備考	なし

[関数名] R_CGC_StopClock

概要	クロック停止
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
宣言	void R_CGC_StopClock(uint8_t status)
説明	引数で指定したクロックを停止します。
引数	<ul style="list-style-type: none"> 第一引数 : staus : 停止するクロック (1-3) 以下の定数を設定します。 X1_STATUS XT1_STATUS → HOCO を停止 HOCO_STATUS → 処理なし (切り替え前のクロックである XT1 発振クロックは常時発振のため)
リターン値	なし
備考	なし

[関数名] R_TAU0_Channel0_GetParameter

概要	TAU0 チャンネル0 パラメータ取得
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
宣言	void R_TAU0_Channel0_GetParameter(uint8_t status, uint8_t *p_clkdiv, uint16_t *p_count)
説明	<ul style="list-style-type: none"> 引数で指定したクロック・ステータスのパラメータを p_clkdiv、p_count に取得します。 p_clkdiv ← TAU0 チャンネル0 の動作クロック分周値がセットされます。 p_count ← TAU0 チャンネル0 のカウント値がセットされます。
引数	<ul style="list-style-type: none"> 第一引数 : status : クロック・ステータス (1-3) 以下の定数を設定します。 X1_STATUS → X1 発振クロックのパラメータを取得する XT1_STATUS → XT1 発振クロックのパラメータを取得する HOCO_STATUS → HOCO クロックのパラメータを取得する 第二引数 : *p_clkdiv : TAU0 チャンネル0 動作クロック分周値を格納 第三引数 : *p_count : TAU0 チャンネル0 カウント値を格納
リターン値	なし
備考	<ul style="list-style-type: none"> p_clkdiv、p_count は R_TAU0_Channel0_Restart の第一、第二引数となります。

[関数名] R_TAU0_Channel0_Restart

概要	TAU0 チャンネル0 リスタート
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
宣言	void R_TAU0_Channel0_Restart(uint8_t clkdiv, uint16_t count)
説明	<ul style="list-style-type: none"> TAU0 チャンネル0 のインターバル・タイマを一旦停止します。 指定された引数で R_TAU0_Channel0_ChangeInterval を呼び出し、TAU0 チャンネル0 のインターバルを変更します。 TAU0 チャンネル0 のインターバル・タイマを再スタートします。
引数	<ul style="list-style-type: none"> 第一引数 : clkdiv : TAU0 チャンネル0 動作クロック分周値 第二引数 : count : TAU0 チャンネル0 カウント値
リターン値	なし
備考	なし

[関数名] R_TAU0_Channel0_ChangeInterval

概要	TAU0 チャンネル0 のインターバル変更
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
宣言	void R_TAU0_Channel0_ChangeInterval(uint8_t clkdiv, uint16_t count)
説明	<ul style="list-style-type: none"> TAU0 チャンネル0 の動作クロック分周値を clkdiv に変更します。 TAU0 チャンネル0 のカウント値を count に変更します。
引数	<ul style="list-style-type: none"> 第一引数 : clkdiv : 動作クロック分周値 (0-15) 0 の場合) $fCLK / 2^0$ 15 の場合) $fCLK / 2^{15}$ fCLK : CPU/周辺ハードウェア・クロック周波数 第二引数 : count : カウント値 (0~65535)
リターン値	なし
備考	<ul style="list-style-type: none"> 第一引数 clkdiv が 15 より大きい場合は 15 として扱います。

[関数名] R_TAU0_Channel0_Stop

概要	TAU0 チャンネル0 の動作停止設定
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
宣言	void R_TAU0_Channel0_Stop(void)
説明	TAU0 チャンネル0 の割り込みをマスクして、カウント動作を停止します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_tau0_channel0_interrupt

概要	TAU0 チャネル0 インターバル・タイマ割り込み
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
宣言	static void __near r_tau0_channel0_interrupt(void)
説明	本割り込み処理に遷移する度に LED 出力(P62)を反転します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_intc0_interrup

概要	INTP0 外部割り込み
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_intc.h" #include "r_cg_userdefine.h"
宣言	static void __near r_intc0_interrup(void)
説明	<ul style="list-style-type: none"> • スイッチ・ステータスの更新 スイッチ状態を 2 回チェックします。1 回目と 2 回目のスイッチ状態確認処理の間にチャタリング対策ウエイト(10ms)を行います。 2 回連続でスイッチ押下ありを検出できない場合は g_SwitchStatus に SWITCH_OFF (0x00) を設定します。 2 回連続でスイッチ押下ありを検出できた場合に g_SwitchStatus に SWITCH_ON (0x01) を設定します。 • クロック・ステータスの更新 スイッチの押下回数により g_ClockStatus に以下の定数を設定します。 HOCO_NEXT_STATUS_X1 X1_NEXT_STATUS_XT1 XT1_NEXT_STATUS_HOCO
引数	なし
リターン値	なし
備考	なし

[関数名] R_IT_Wait_ms

概要	1ms 単位のウエイト
ヘッダ	#include "r_cg_macrodriver.h" #include "r_cg_it.h" #include "r_cg_userdefine.h"
宣言	void R_IT_Wait_ms(uint16_t wait_count)
説明	12 ビット・インターバル・タイマを使用し、wait_count * 1ms のウエイトを行います。
引数	<ul style="list-style-type: none"> • 第一引数 : wait_count : ウエイト・カウンタ
リターン値	なし
備考	<ul style="list-style-type: none"> • 本関数では 12 ビット・インターバル・タイマのベクタ割り込みを使用しません。12 ビット・インターバル・タイマをスタート後に割り込み要求フラグをポーリングし、引数で指定されたウエイトを行います。 • 第一引数 wait_count が 0 の場合はウエイトしません。

5.8 フローチャート

図 5.1にサンプルコードの全体フローを示します。

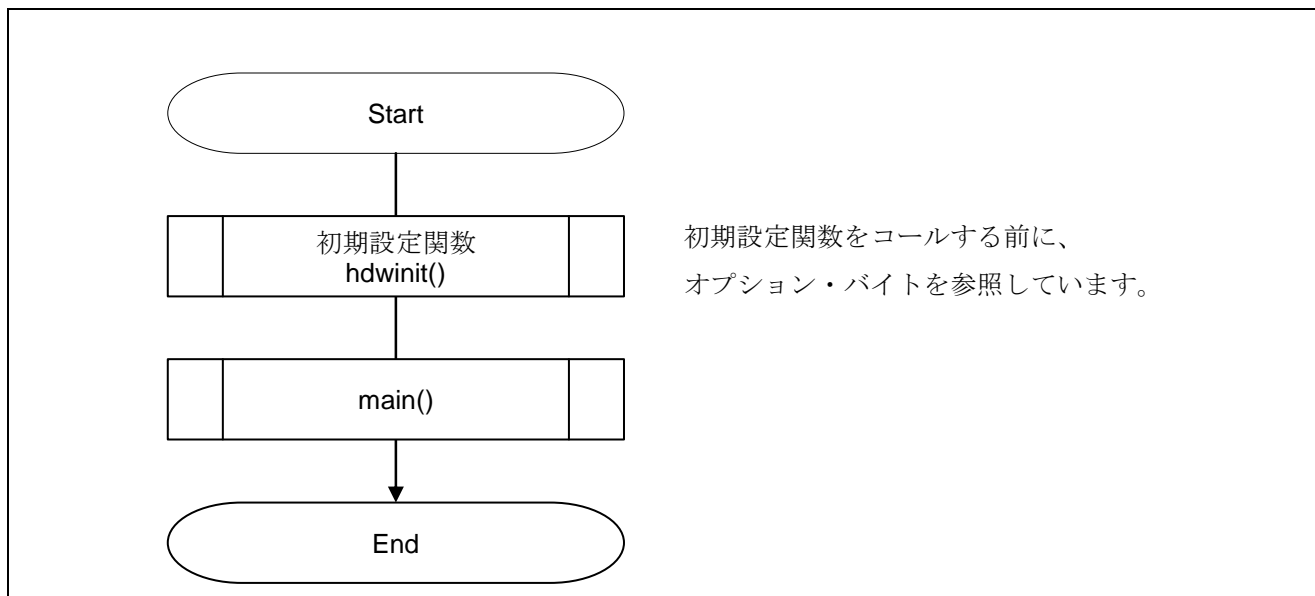


図 5.1 全体フロー

5.8.1 初期化設定関数

図 5.2に初期設定関数のフローチャートを示します。

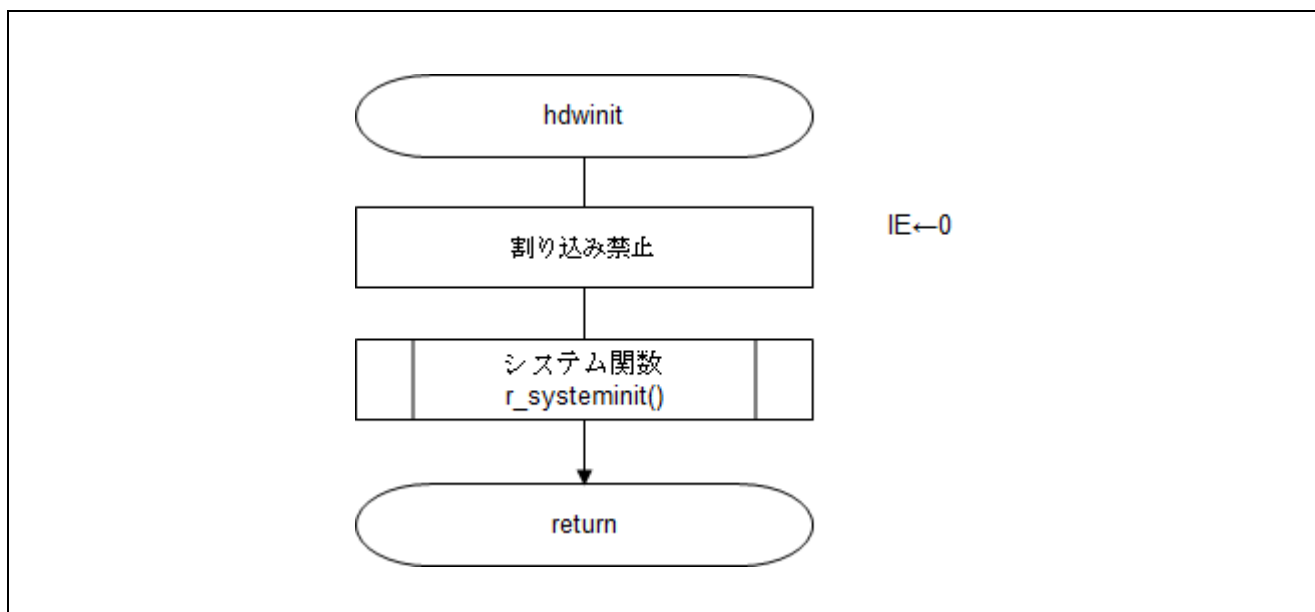


図 5.2 初期設定関数

5.8.2 システム関数

図 5.3にシステム関数のフローチャートを示します。

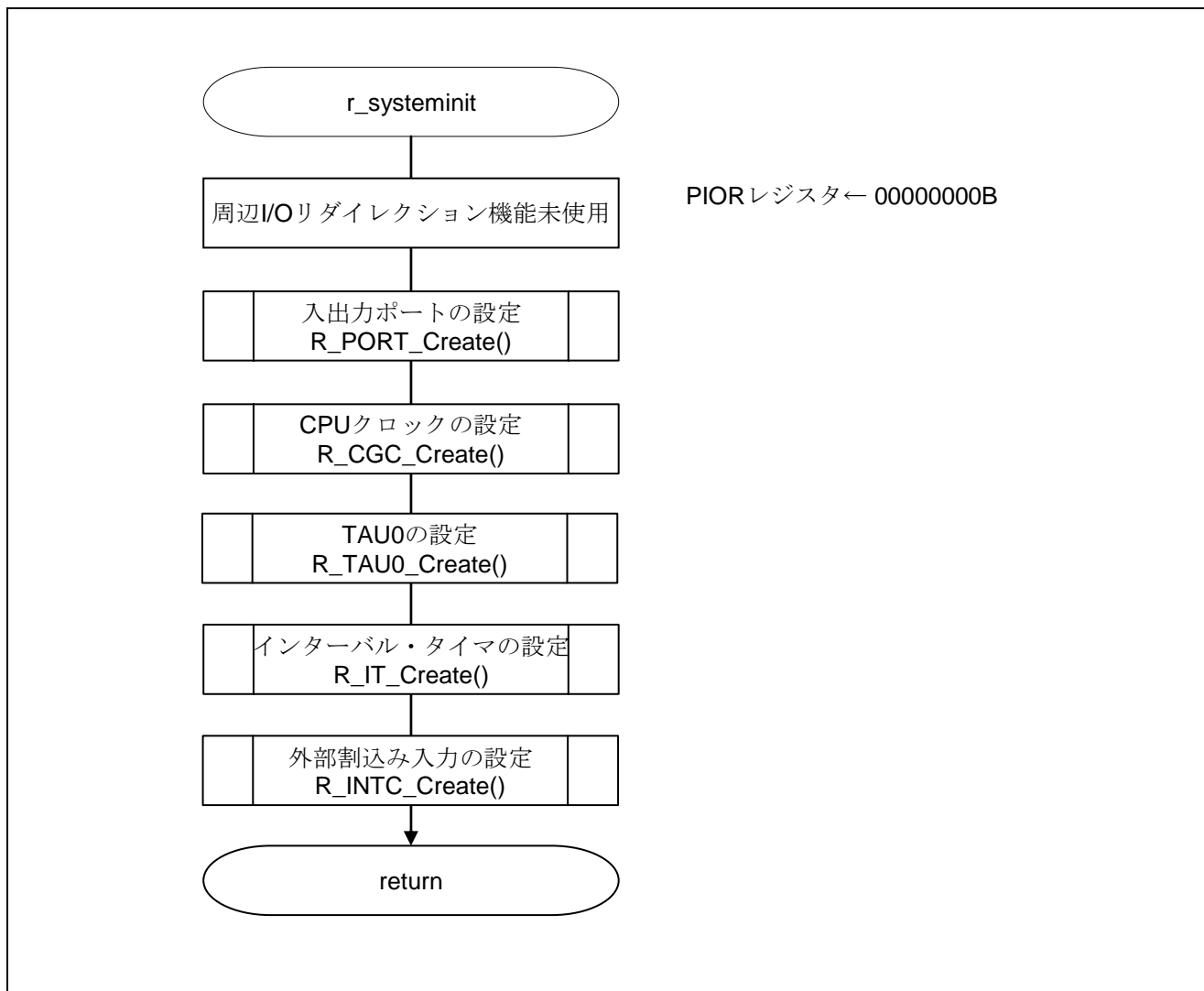


図 5.3 システム関数

5.8.3 入出力ポートの設定

図 5.4に入出力ポートの設定のフローチャートを示します。

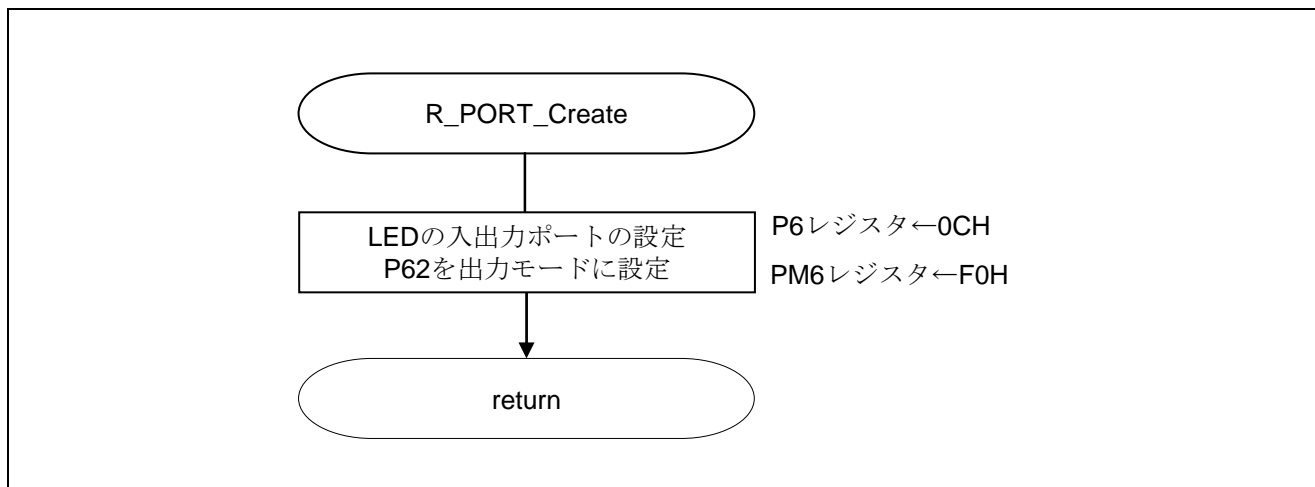


図 5.4 入出力ポートの設定

- 注意 1 未使用ポートの設定については、RL78/G13 初期設定（R01AN2575J）アプリケーションノート“フローチャート”を参照して下さい。
- 未使用のポートは、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。また、未使用の入力専用ポートは個別に抵抗を介して VDD 又は VSS に接続して下さい。
 - P63 に接続している LED は常時消灯とするので、ハイ・レベルを設定します。

5.8.4 CPU クロックの初期設定

図 5.5にCPU クロックの初期設定のフローチャートを示します。

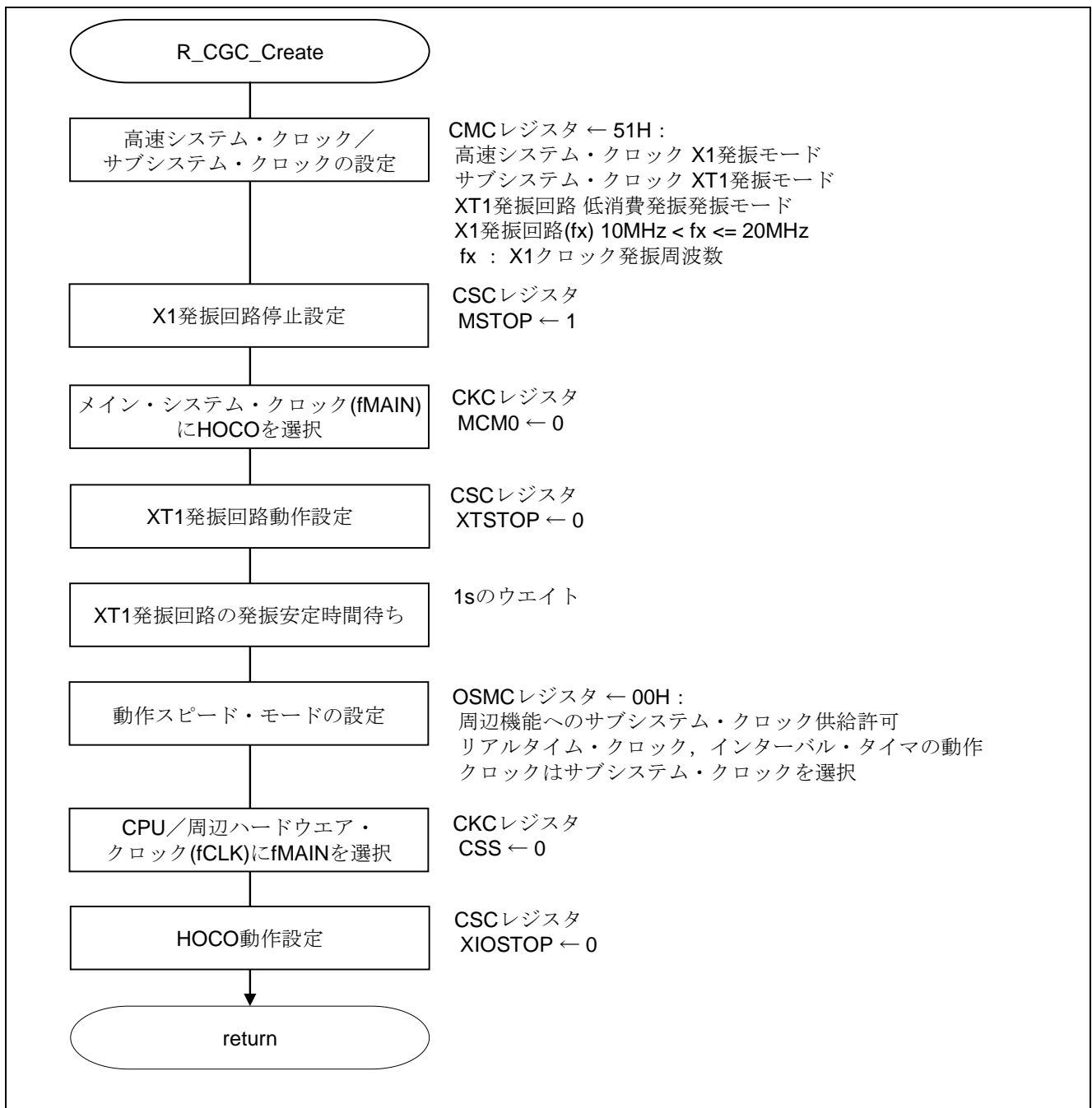


図 5.5 CPU クロックの初期設定

備考 CPU クロックの設定 (R_CGC_Create()) で、サブシステム・クロックの発振安定待ち (約 1s) の処理を行っています。

発振安定待ち時間は、r_cg_cgc.h 内の定数 CGC_SUBWAITTIME で指定しています。

注意 CPU クロックの設定 (R_CGC_Create()) については、RL78/G13 初期設定 (R01AN2575J) アプリケーションノート"フローチャート"を参照して下さい。

クロック発生回路の動作モード設定

・クロック動作制御レジスタ（CMC）

：高速システム・クロック端子の動作モードを選択、

サブ・システムクロック端子の動作モードを選択、

XT1 発振回路の発振モード選択、X1 クロック発振周波数の制御

略号：CMC

	7	6	5	4	3	2	1	0
HOCO の設定	0	1	0	1	0	0	0	1
X1 発振時の設定								
XT1 発振時の設定								

ビット 7-6

EXCLK	OSCSEL	高速システム・クロック 端子の動作モード	X1/P121 端子	X2/EXCLK/P122 端子
0	0	入力ポート・モード	入力ポート	
0	1	X1 発振モード	水晶/セラミック発振子接続	
1	0	入力ポート・モード	入力ポート	
1	1	外部クロック入力モード	入力ポート	外部クロック入力

ビット 5-4

EXCLKS	OSCSELS	サブ・システムクロック 端子の動作モード	XT1/P123 端子	XT2/EXCLKS/P124 端子
0	0	入力ポート・モード	入力ポート	
0	1	XT1 発振モード	水晶振動子接続	
1	0	入力ポート・モード	入力ポート	
1	1	外部クロック入力モード	入力ポート	外部クロック入力

ビット 2-1

AMPHS1	AMPHS0	XT1 発振回路の発振モード選択	
0	0	低消費発振（デフォルト）	発振余裕度：中
0	1	通常発振	発振余裕度：高
1	0	超低消費発振	発振余裕度：低
1	1	設定禁止	

ビット 0

AMPHS	X1 クロック発振周波数の制御
0	$1 \text{ MHz} \leq f_x \leq 10 \text{ MHz}$
1	$10 \text{ MHz} < f_x \leq 20 \text{ MHz}$

注意 レジスタ設定の詳細については、RL78/G13 ユーザーズマニュアル ハードウェア編を参照してください。

クロックの選択

・システム・クロック制御レジスタ（CKC）

: CPU/周辺ハードウェア・クロック（fCLK）のステータス、

CPU/周辺ハードウェア・クロック（fCLK）の選択、

メイン・システム・クロック（fMAIN）のステータス、

メイン・システム・クロック（fMAIN）の動作制御

略号：CKC

	7	6	5	4	3	2	1	0
	CLS	CSS	MCS	MCM0	0	0	0	0
HOCO 時の設定	0	0	0	0	0	0	0	0
X1 発振時の設定	0	0	1	1	0	0	0	0
XT1 発振時の設定	1	1	0	0	0	0	0	0

ビット7

CLS ^{注1}	CPU/周辺ハードウェア・クロック（fCLK）のステータス
0	メイン・システム・クロック（fMAIN）
1	サブ・システムクロック（fSUB）

ビット6

CSS	CPU/周辺ハードウェア・クロック（fCLK）の選択
0	メイン・システム・クロック（fMAIN）
1 ^{注2}	サブ・システムクロック（fSUB）

ビット5

MCS ^{注1}	メイン・システム・クロック（fMAIN）のステータス
0	高速オンチップ・オシレータ・クロック（fIH）
1	高速システム・クロック（fMX）

ビット4

MCM0 ^{注2}	メイン・システム・クロック（fMAIN）の動作制御
0	メイン・システム・クロック（fMAIN）に高速オンチップ・オシレータ・クロック（fIH）を選択
1	メイン・システム・クロック（fMAIN）に高速システム・クロック（fMX）を選択

注1 ビット7,5は、Read Only です。

注2 CSS = 1 を設定した状態で、MCM0 ビットの値を変更することは禁止です。

注意 レジスタ設定の詳細については、RL78/G13 ユーザーズマニュアル ハードウェア編を参照してください。

クロック動作制御

・クロック動作ステータス制御レジスタ（CSC）

：高速システム・クロックの動作制御、

サブ・システムクロックの動作制御、

高速オンチップ・オシレータ・クロックの動作制御

略号：CSC

	7	6	5	4	3	2	1	0
	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
HOCO 時の設定	1	0	0					0
X1 発振時の設定	0							1
XT1 発振時の設定	1							1

ビット 7

MSTOP	高速システム・クロックの動作制御		
	X1 発振モード時	外部クロック入力モード時	入力ポート・モード時
0	X1 発振回路動作	EXCLK 端子からの外部クロック有効	入力ポート
1	X1 発振回路停止	EXCLK 端子からの外部クロック無効	

ビット 6

XTSTOP	サブ・システムクロックの動作制御		
	XT1 発振モード時	外部クロック入力モード時	入力ポート・モード時
0	XT1 発振回路動作	EXCLKS 端子からの外部クロック有効	入力ポート
1	XT1 発振回路停止	EXCLKS 端子からの外部クロック無効	

ビット 0

HIOSTOP	高速オンチップ・オシレータ・クロックの動作制御
0	高速オンチップ・オシレータ動作
1	高速オンチップ・オシレータ停止

注意 レジスタ設定の詳細については、RL78/G13 ユーザーズマニュアル ハードウェア編を参照してください。

5.8.5 TAU0 の設定

図 5.6にTAU0 の設定のフローチャートを示します。

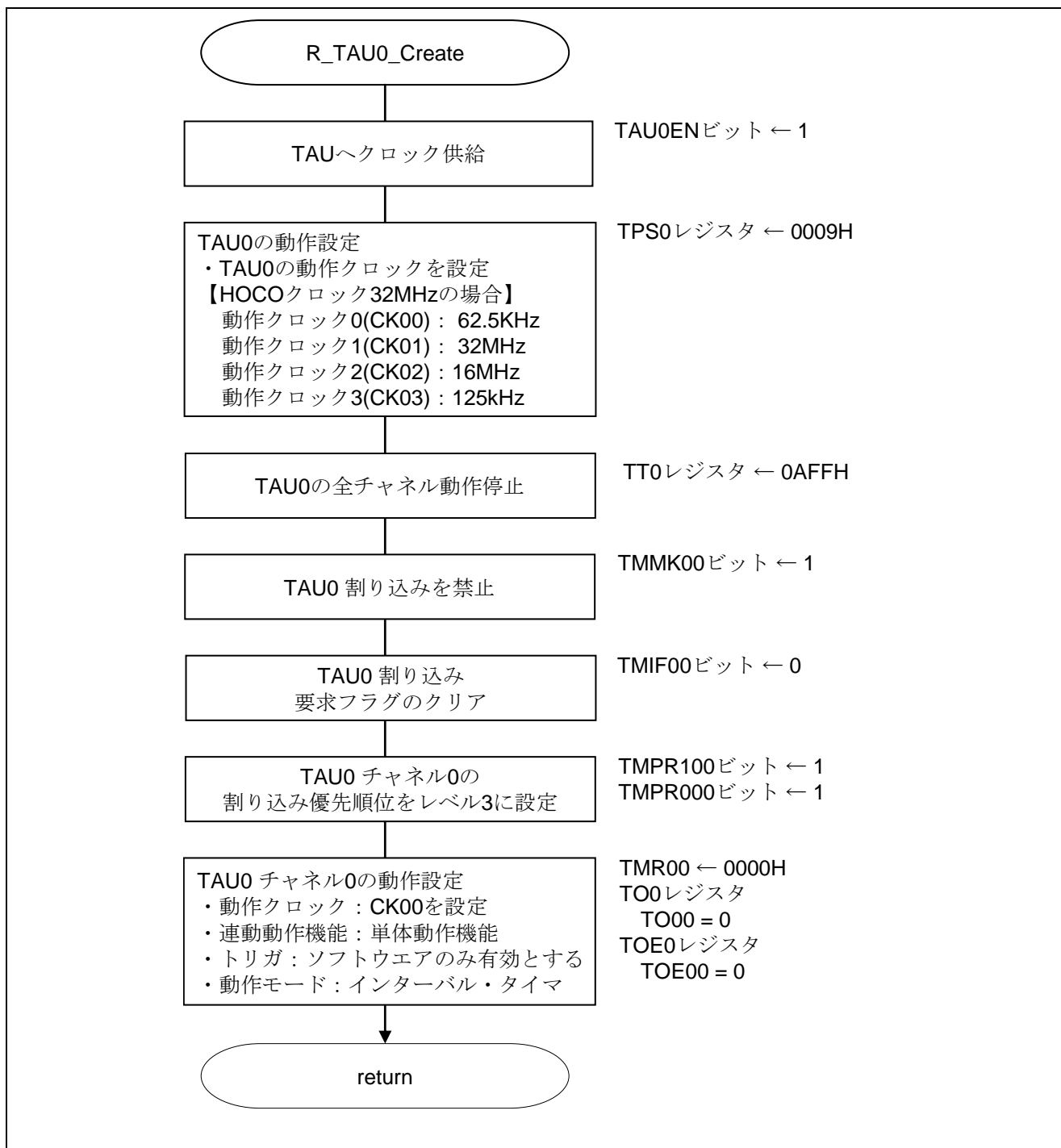


図 5.6 TAU0 の設定

5.8.6 12ビット・インターバル・タイマの設定

図 5.7に12ビット・インターバル・タイマの設定のフローチャートを示します。

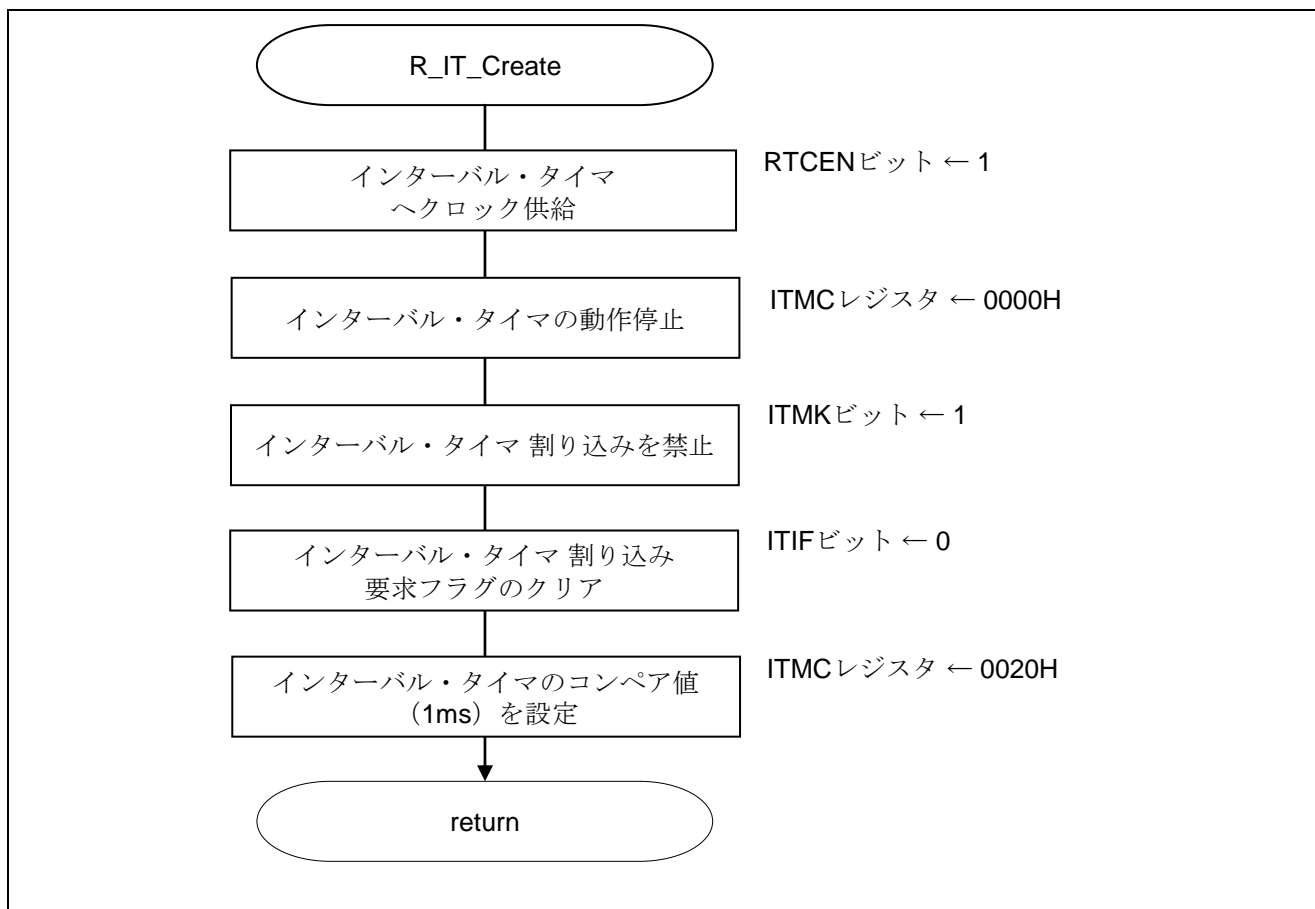


図 5.7 12ビット・インターバル・タイマの設定

5.8.7 外部割り込み入力の設定

図 5.8に外部割り込み入力の設定のフローチャートを示します。

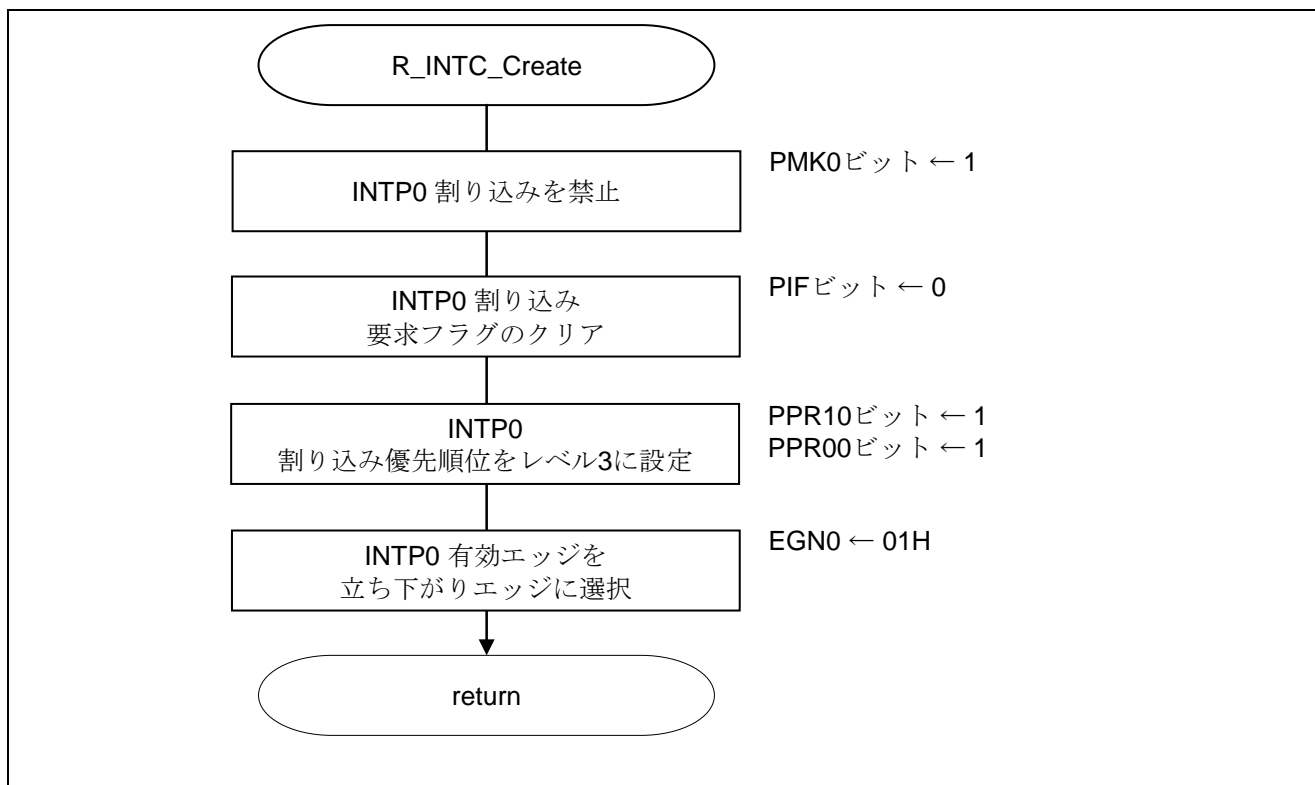


図 5.8 外部割り込み入力の設定

5.8.8 メイン処理

図 5.9にメイン処理のフローチャートを示します。

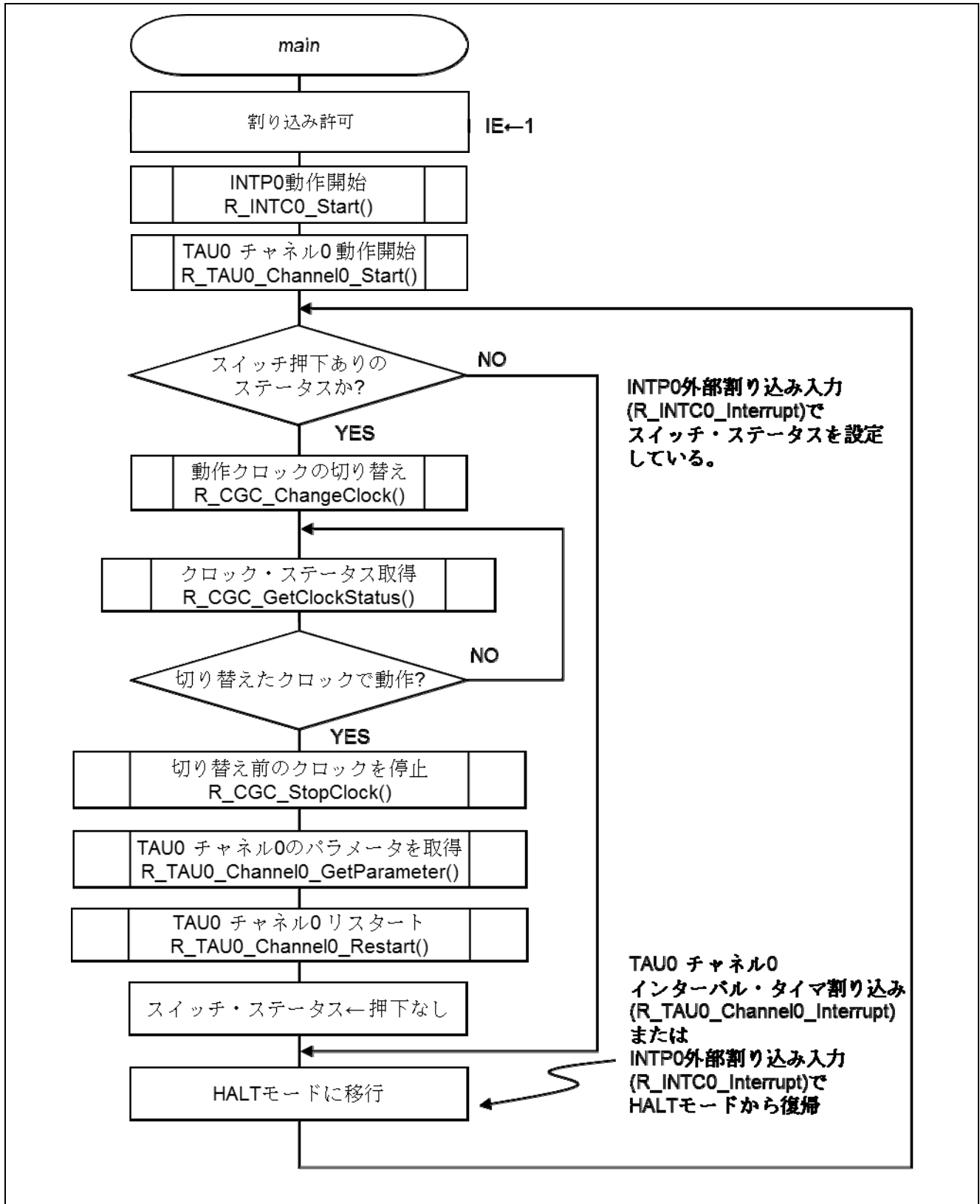


図 5.9 メイン処理

5.8.9 INTPO 外部割り込み動作開始設定

図 5.10にINTPO 外部割り込み動作開始設定のフローチャートを示します。

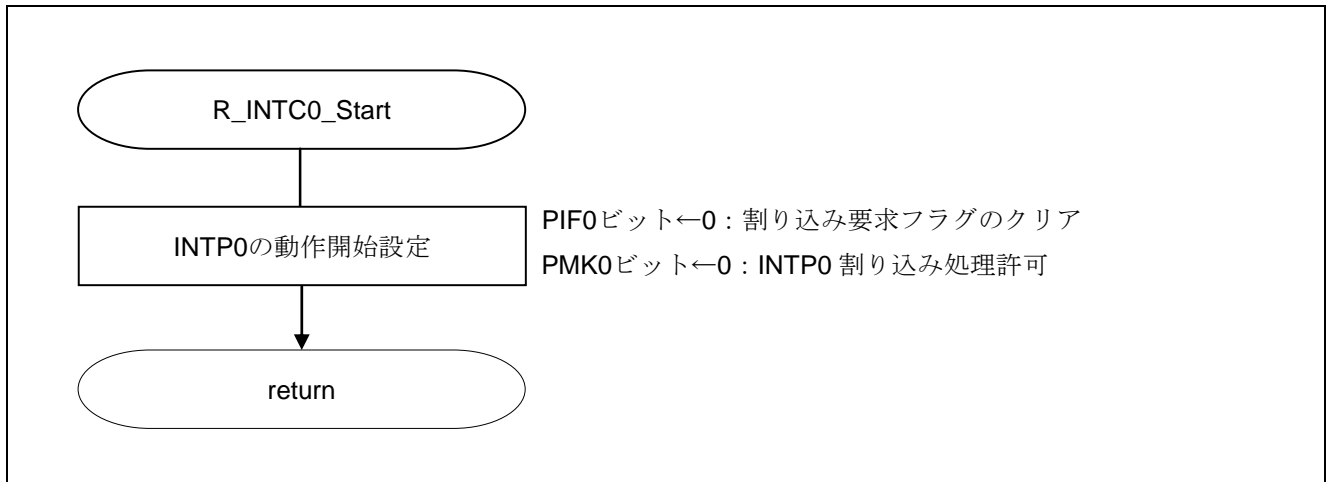


図 5.10 INTPO 外部割り込み動作開始設定

5.8.10 TAU0 チャンネル0 の動作開始設定

図 5.11にTAU0 チャンネル0 の動作開始設定のフローチャートを示します。

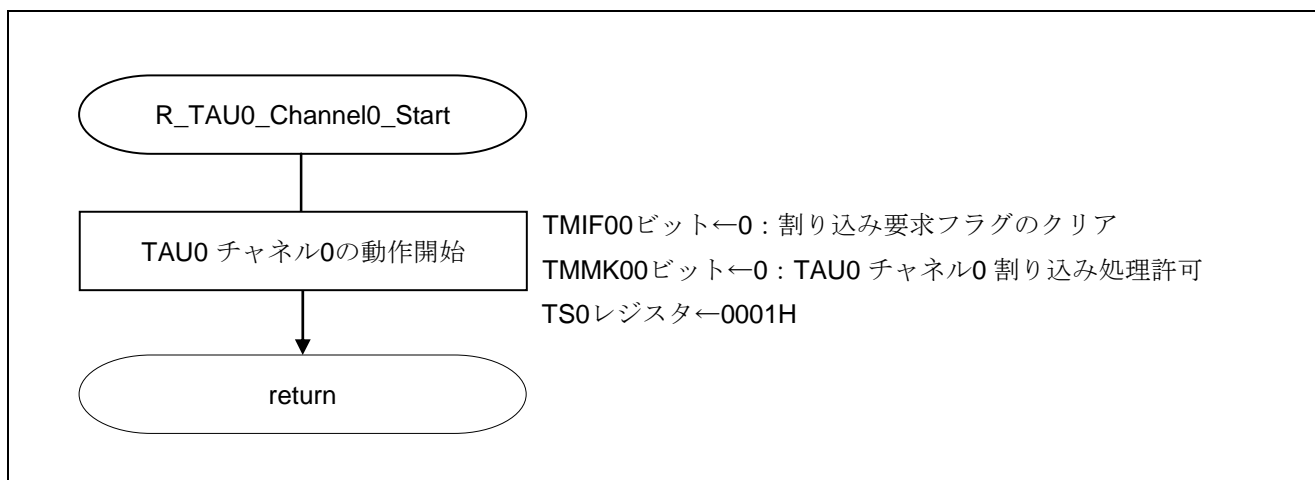


図 5.11 TAU0 チャンネル0 の動作開始設定

5.8.11 クロック切り替え

図 5.12にクロック切り替えのフローチャートを示します。

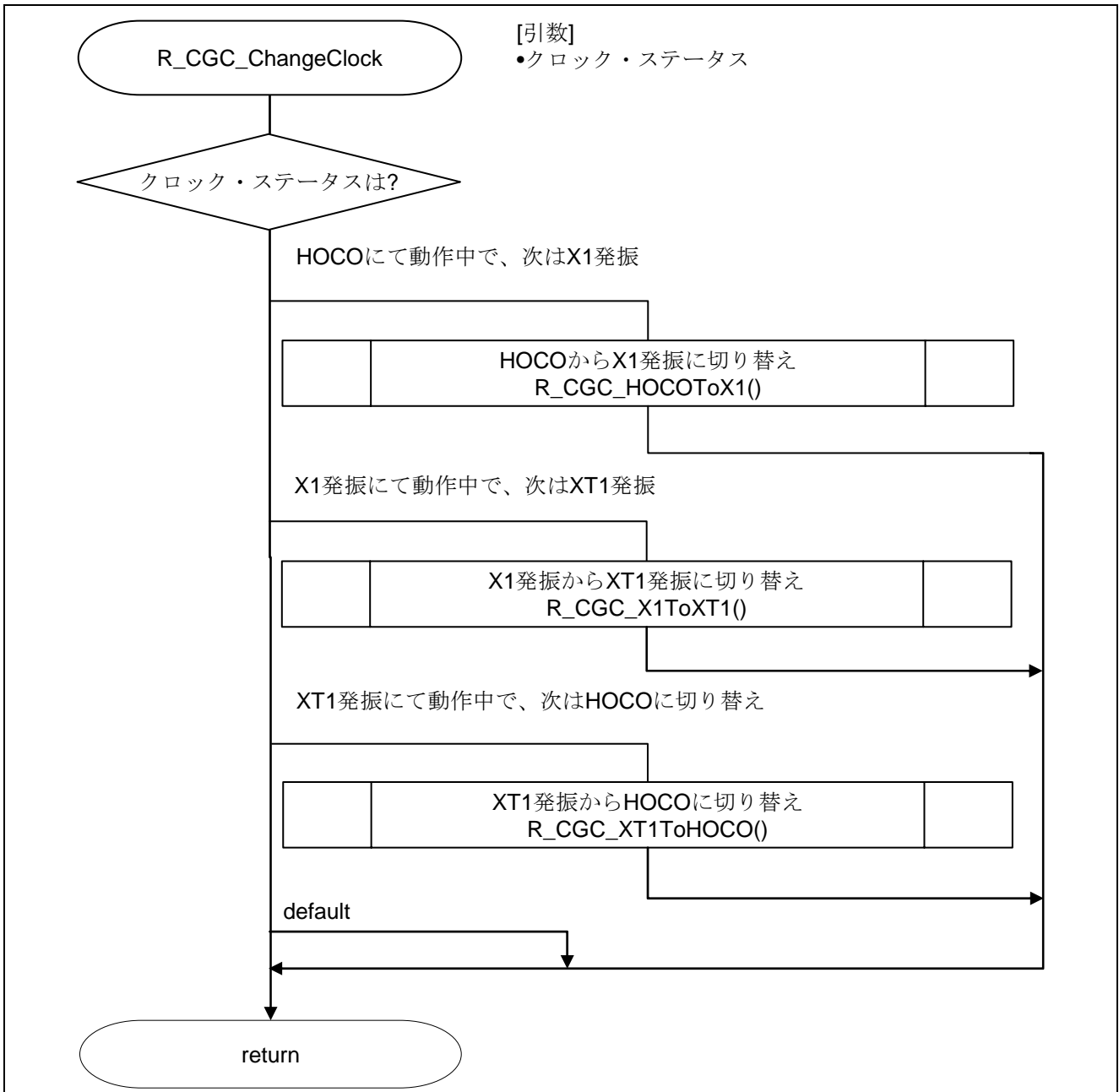


図 5.12 クロック切り替え

5.8.12 HOCO クロックから X1 発振クロック切り替え処理

図 5.13にHOCO クロックから X1 発振クロック切り替え処理のフローチャートを示します。

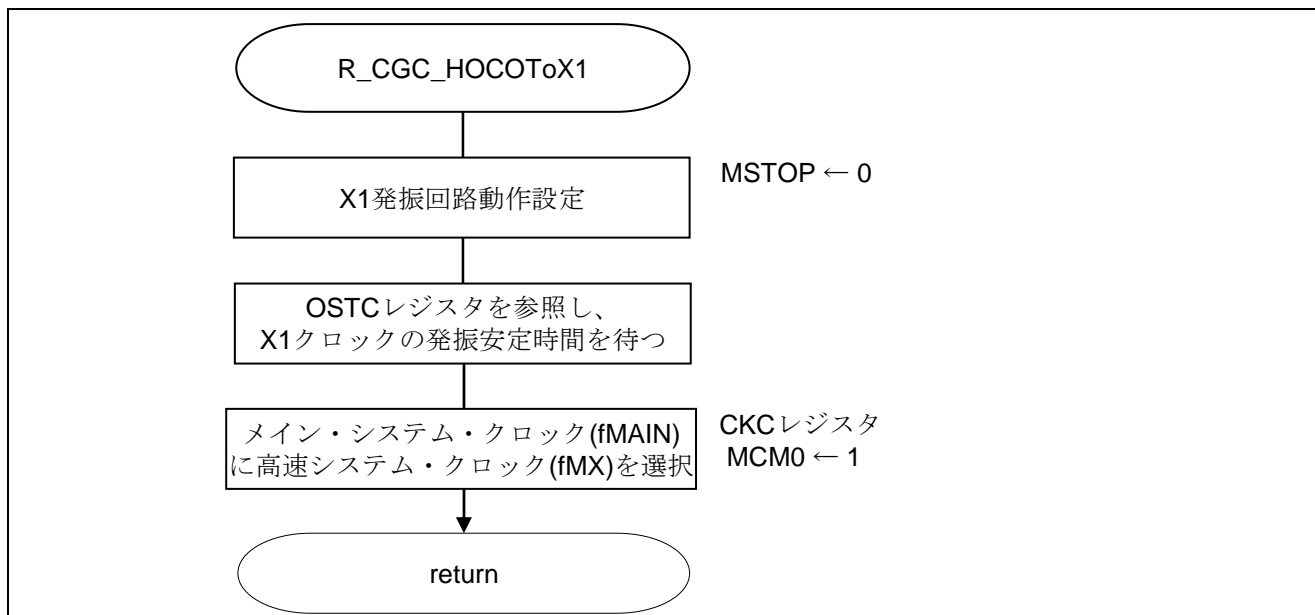


図 5.13 HOCO クロックから X1 発振クロック切り替え処理

5.8.13 X1 発振クロックから XT1 発振クロック発振切り替え処理

図 5.14にX1 発振クロックから XT1 発振クロック切り替え処理のフローチャートを示します。

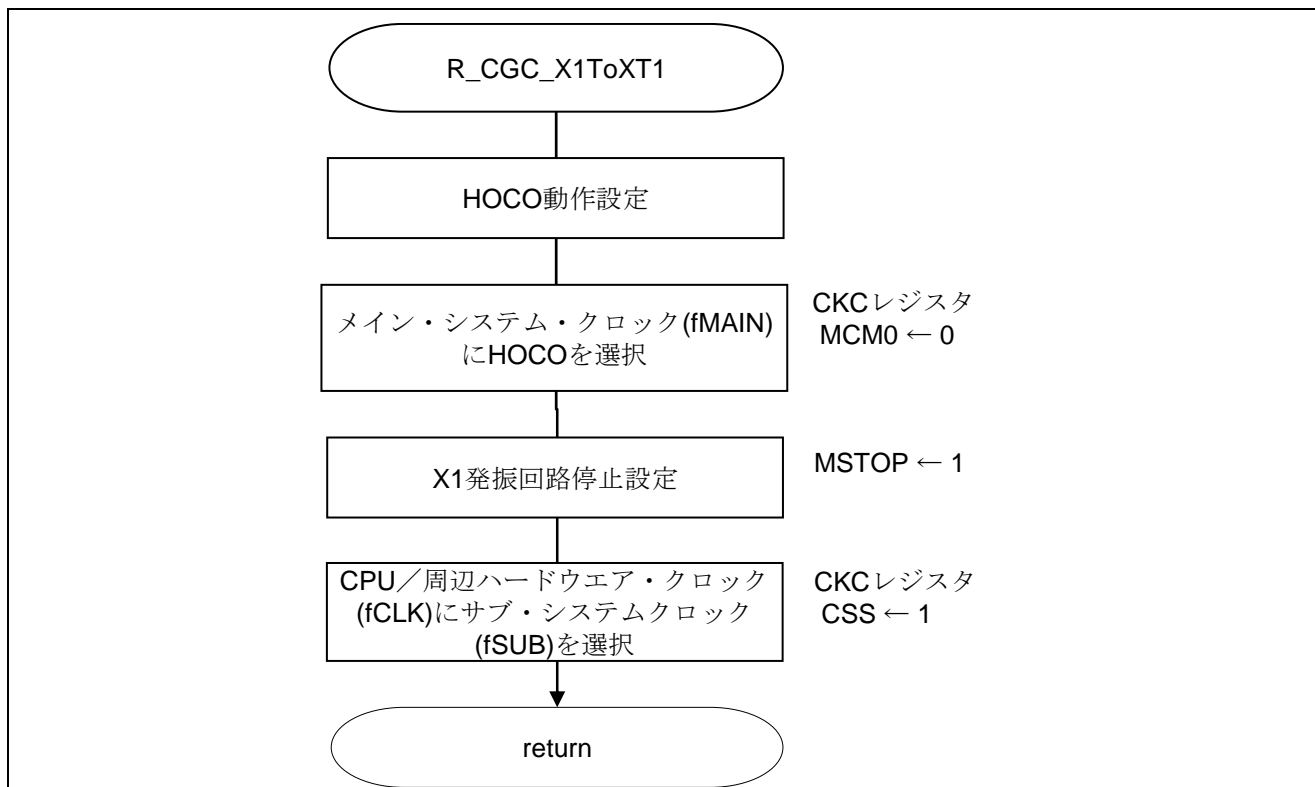


図 5.14 X1 発振クロックから XT1 発振クロック切り替え処理

5.8.14 XT1 発振クロックから HOCO クロック切り替え処理

図 5.15にXT1 発振クロックから HOCO クロック切り替え処理のフローチャートを示します。

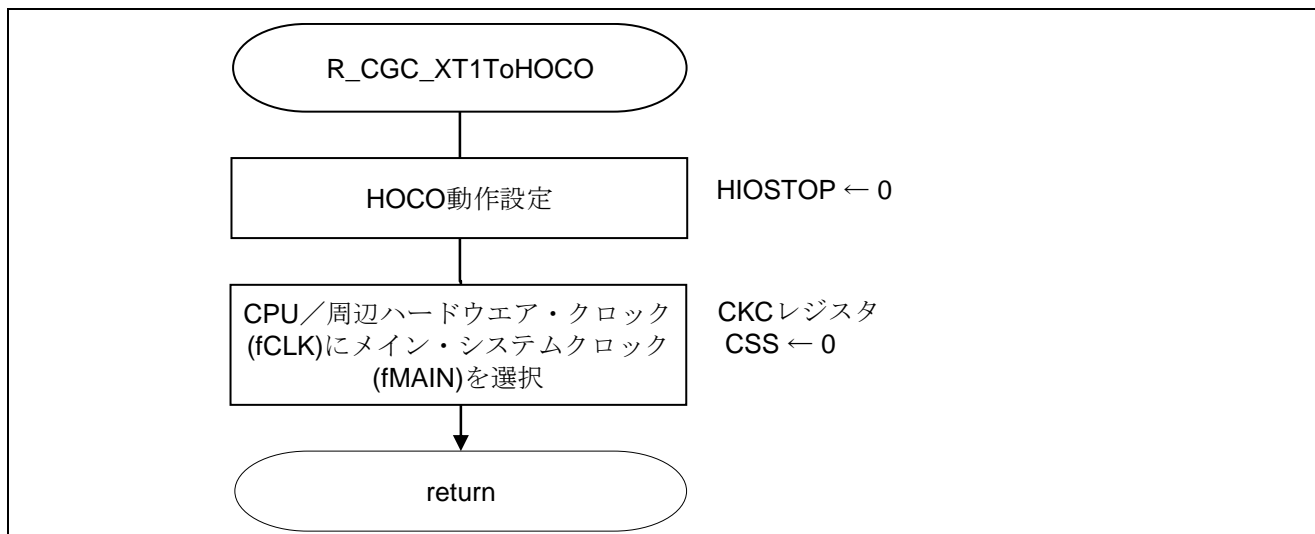


図 5.15 XT1 発振クロックから HOCO クロック切り替え処理

5.8.15 クロック・ステータス取得

図 5.16にクロック・ステータス取得のフローチャートを示します。

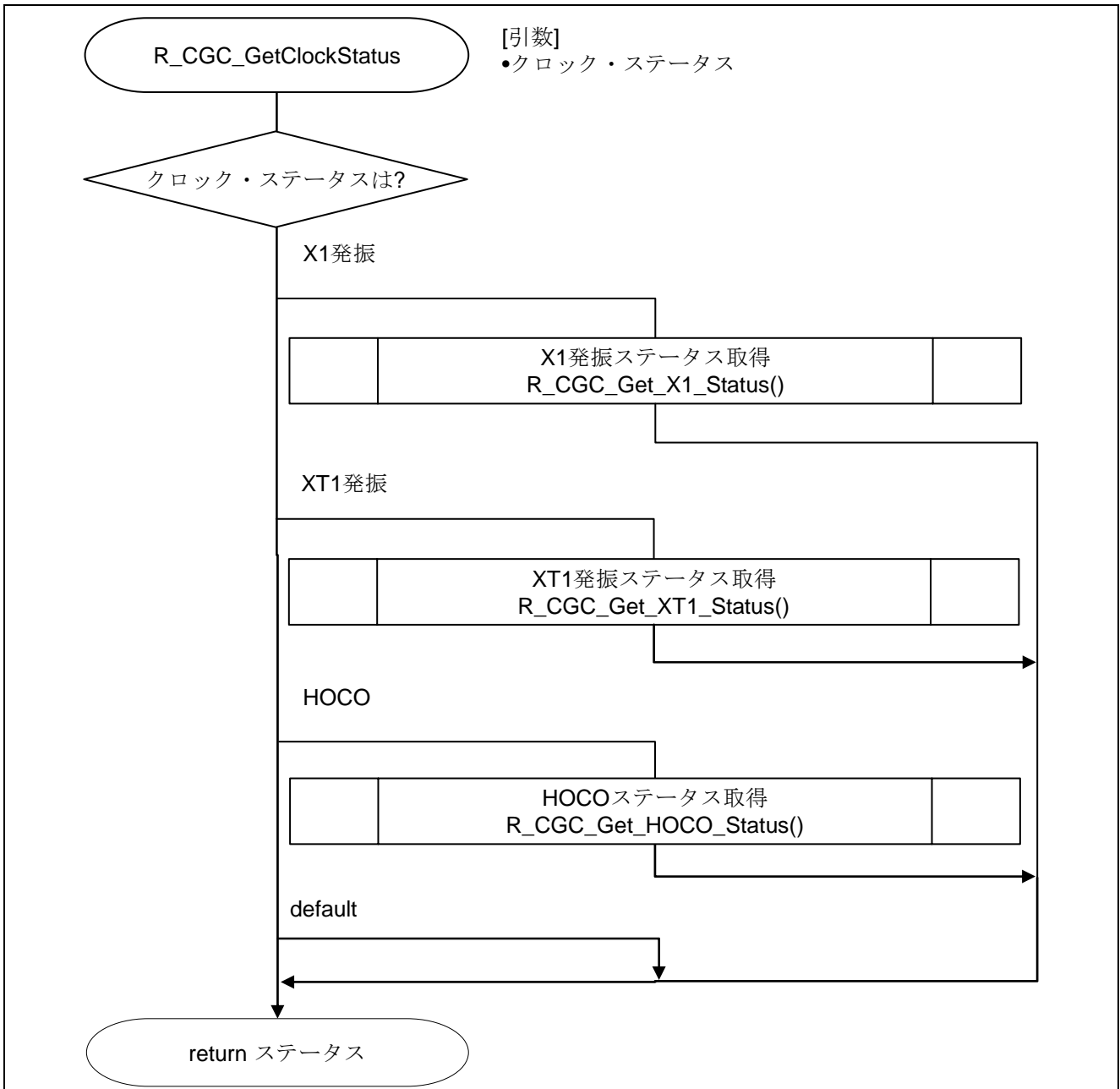


図 5.16 クロック・ステータス取得

5.8.16 X1 発振クロック・ステータス取得

図 5.17にX1 発振クロック・ステータス取得のフローチャートを示します。

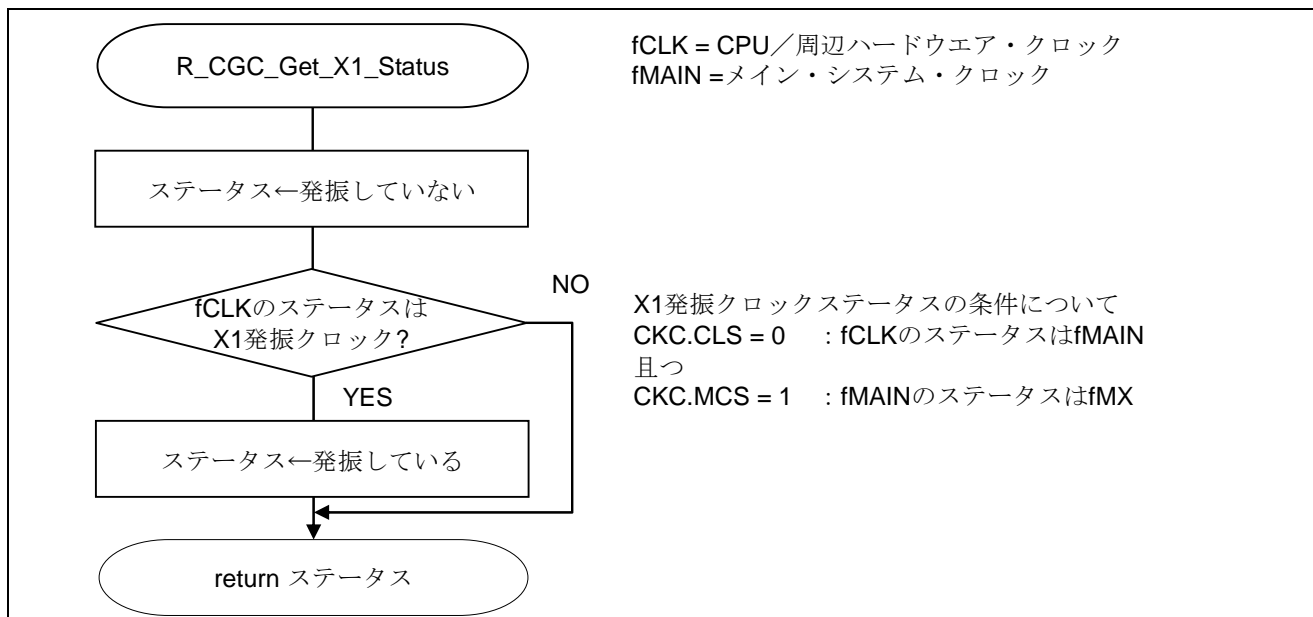


図 5.17 X1 発振クロック・ステータス取得

5.8.17 XT1 発振クロック・ステータス取得

図 5.18にXT1 発振クロック・ステータス取得のフローチャートを示します。

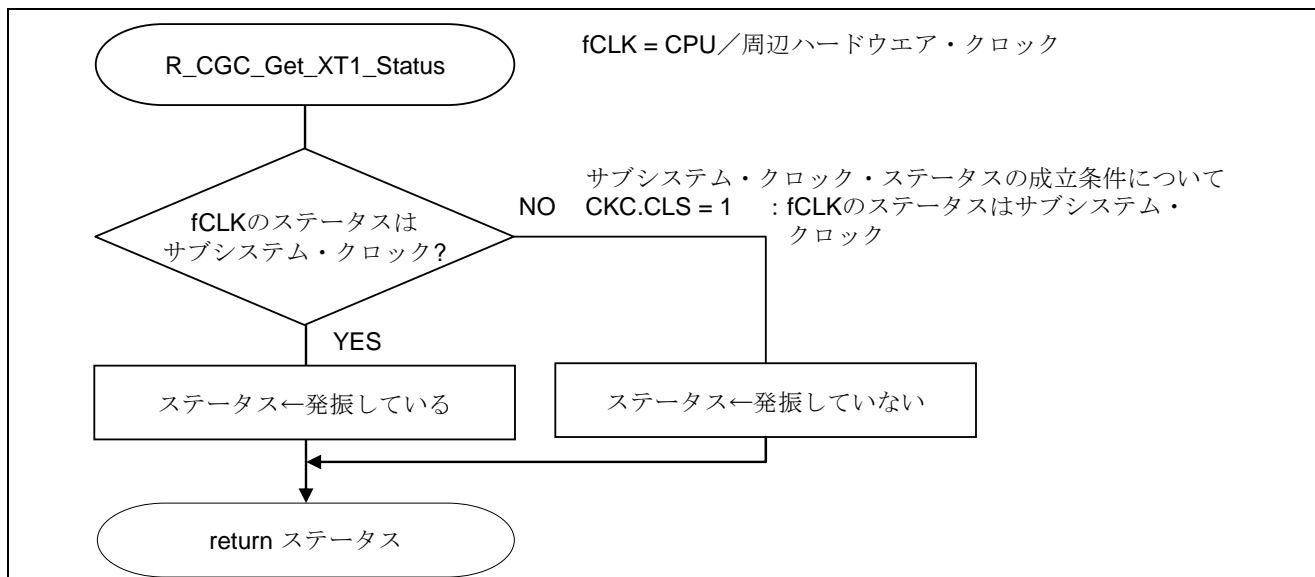


図 5.18 XT1 発振クロック・ステータス取得

5.8.18 HOCO クロック・ステータス取得

図 5.19にHOCO クロック・ステータス取得のフローチャートを示します。

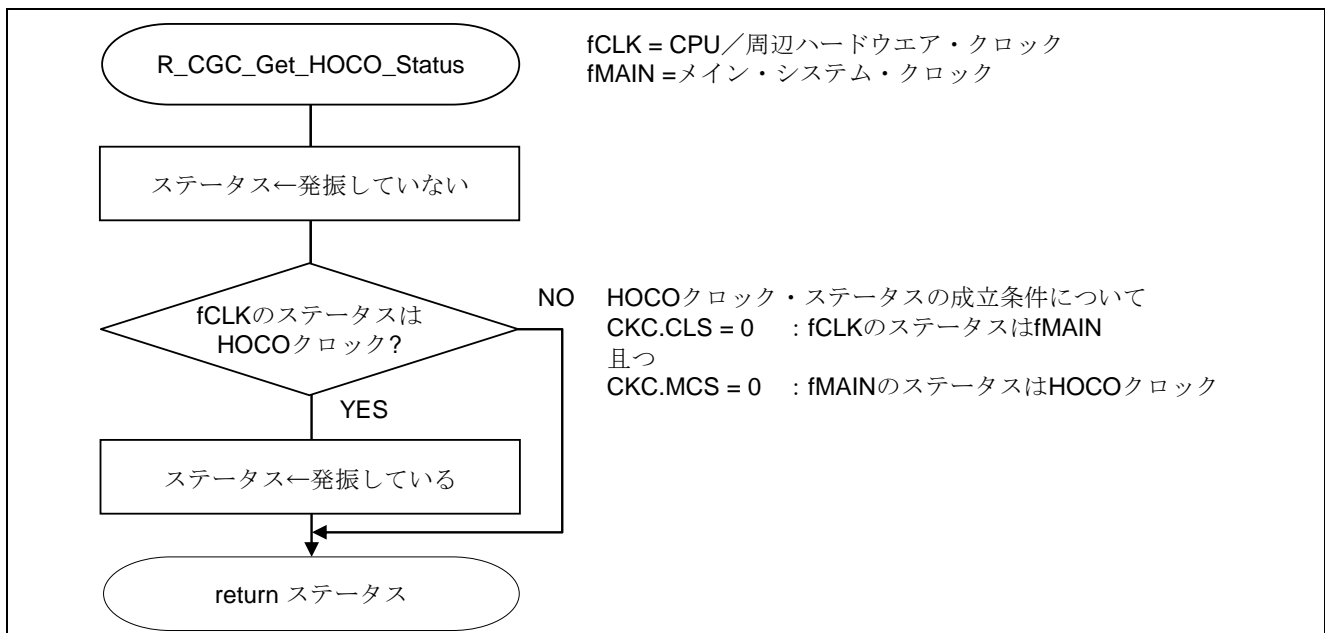


図 5.19 HOCO クロック・ステータス取得

5.8.19 クロック停止

図 5.20にクロック停止のフローチャートを示します。

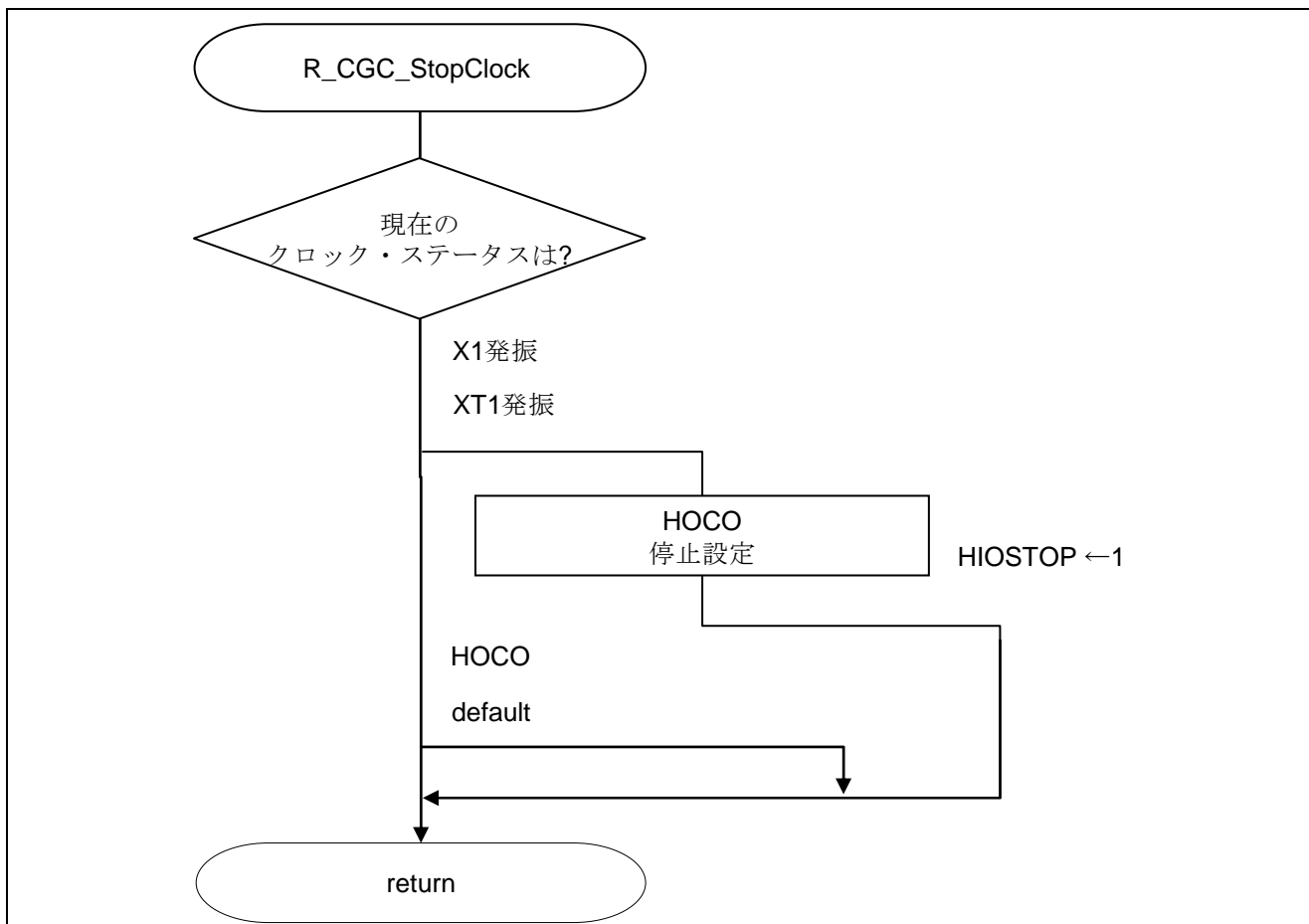


図 5.20 クロック停止

5.8.20 TAU0 チャンネル0 パラメータ取得

図 5.21にTAU0 チャンネル0 パラメータ取得のフローチャートを示します。

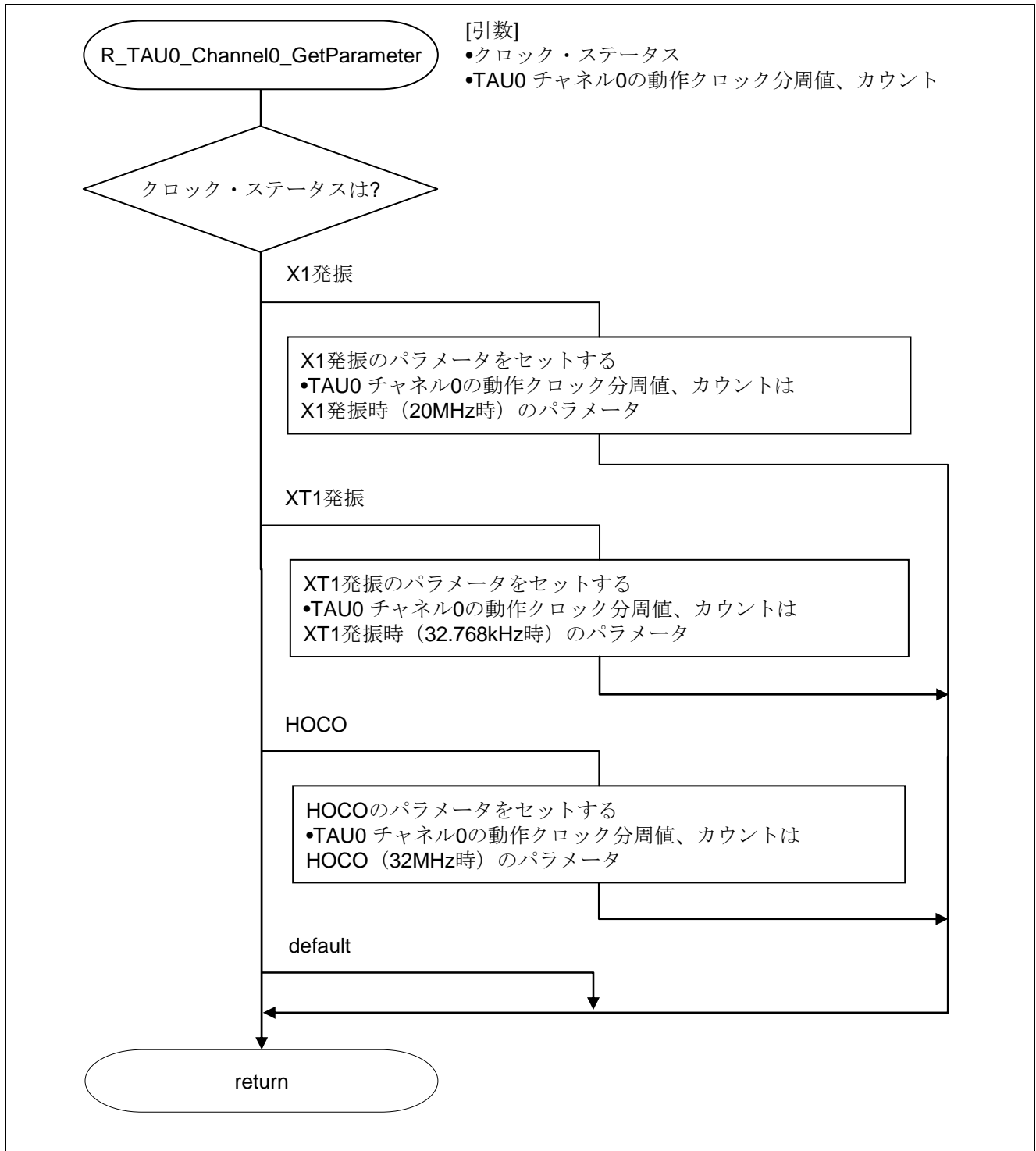


図 5.21 TAU0 チャンネル0 パラメータ取得

5.8.21 TAU0 チャンネル0 リスタート

図 5.22にTAU0 チャンネル0 リスタートのフローチャートを示します。

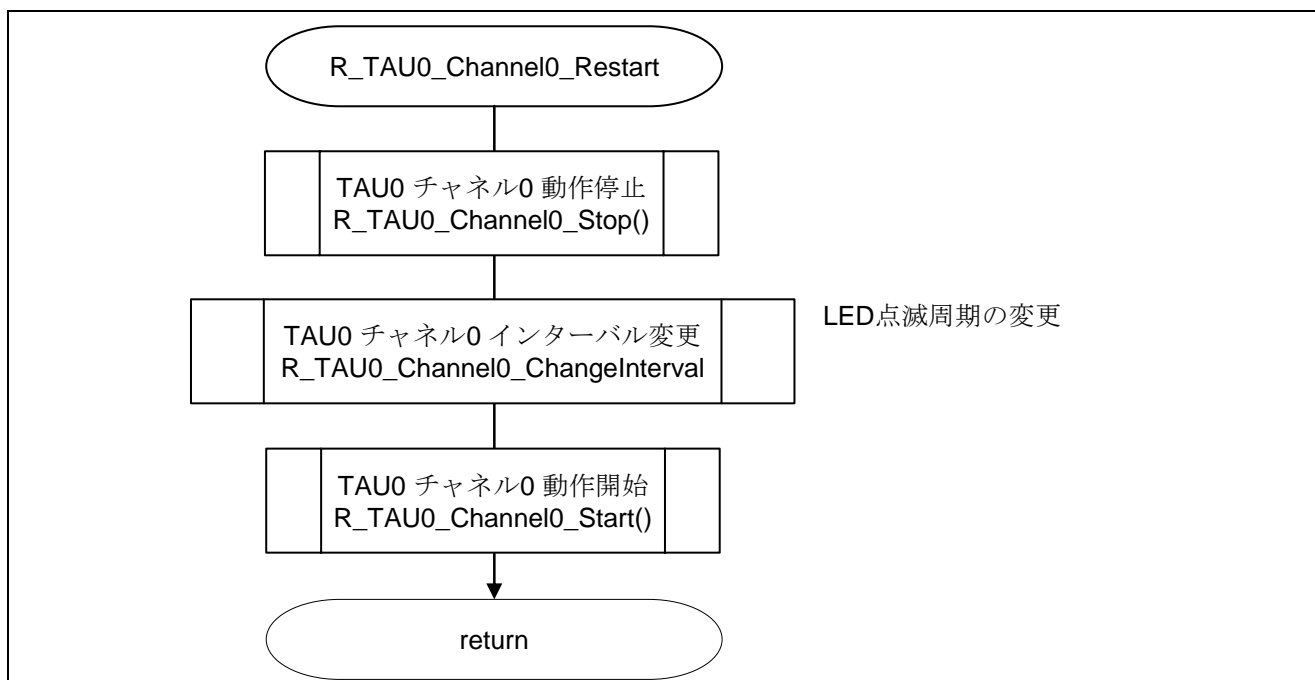


図 5.22 TAU0 チャンネル0 リスタート

5.8.22 TAU0 チャンネル0 のインターバル変更

図 5.23にTAU0 チャンネル0 のインターバル変更のフローチャートを示します。

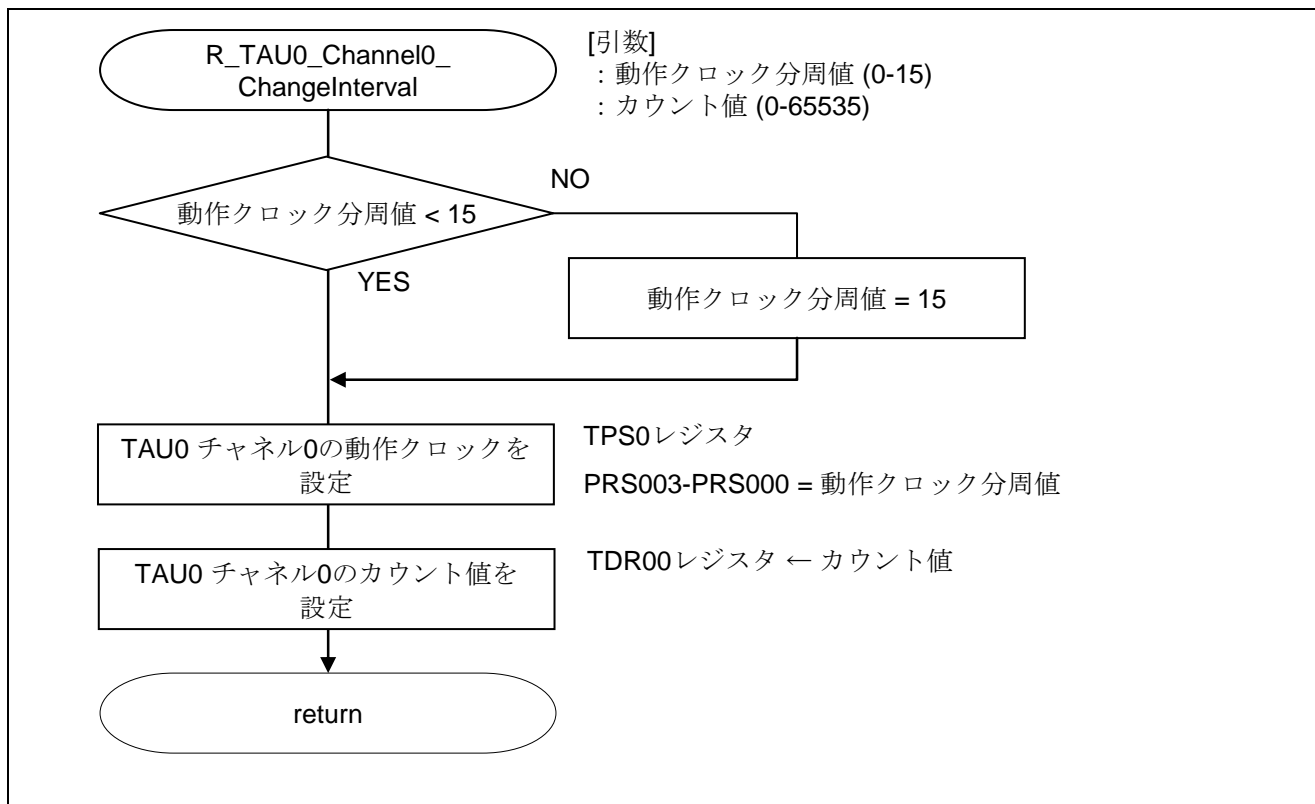


図 5.23 TAU0 チャンネル0 のインターバル変更

5.8.23 TAU0 チャンネル0 の動作停止設定

図 5.24にTAU0 チャンネル0 の動作停止設定のフローチャートを示します。

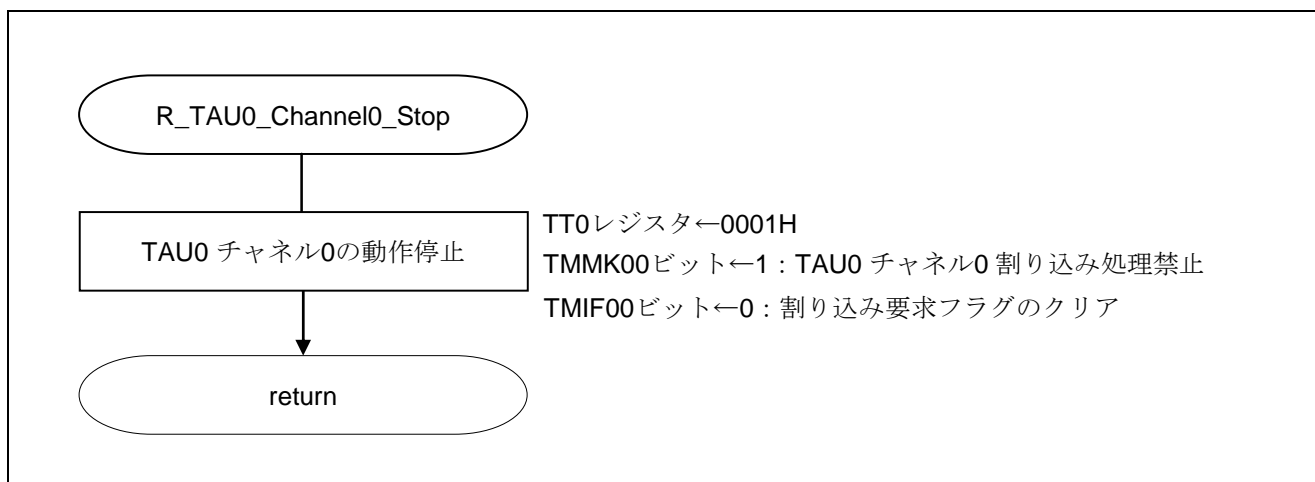


図 5.24 TAU0 チャンネル0 の動作停止設定

5.8.24 TAU0 チャンネル0 インターバル・タイマ割り込み

図 5.25にTAU0 チャンネル0 インターバル・タイマのフローチャートを示します。

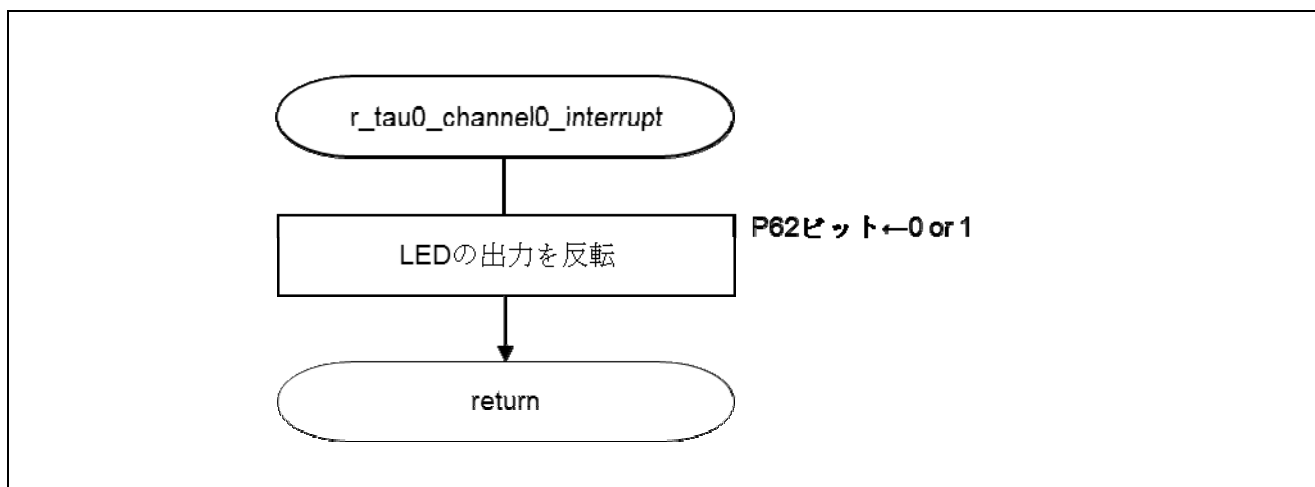


図 5.25 TAU0 チャンネル0 インターバル・タイマ割り込み

5.8.25 INTP0 外部割り込み

図 5.26にINTP0 外部割り込みのフローチャートを示します。

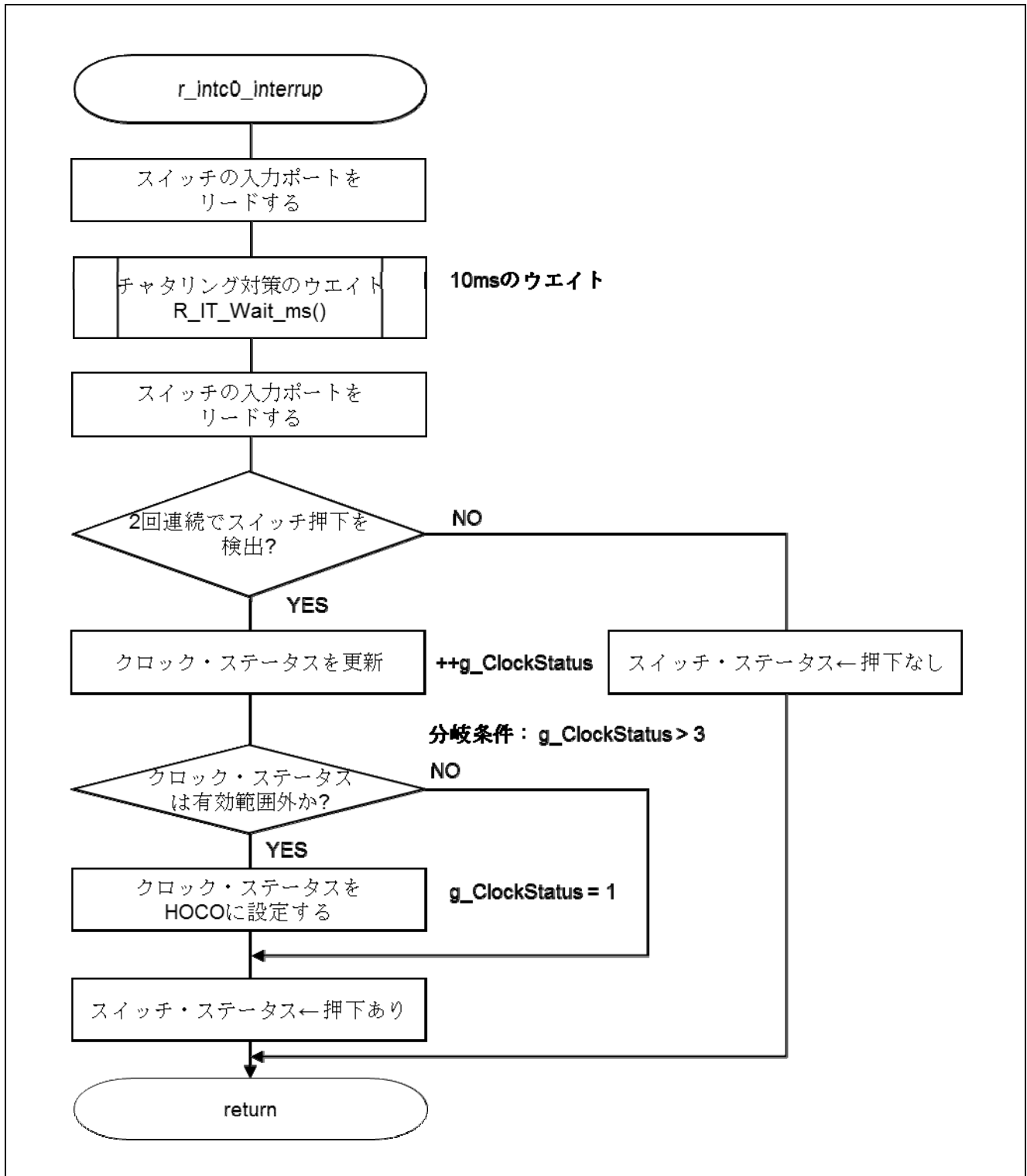


図 5.26 INTP0 外部割り込み

5.8.26 1ms 単位のウェイト

図 5.27に1ms 単位のウェイトのフローチャートを示します。

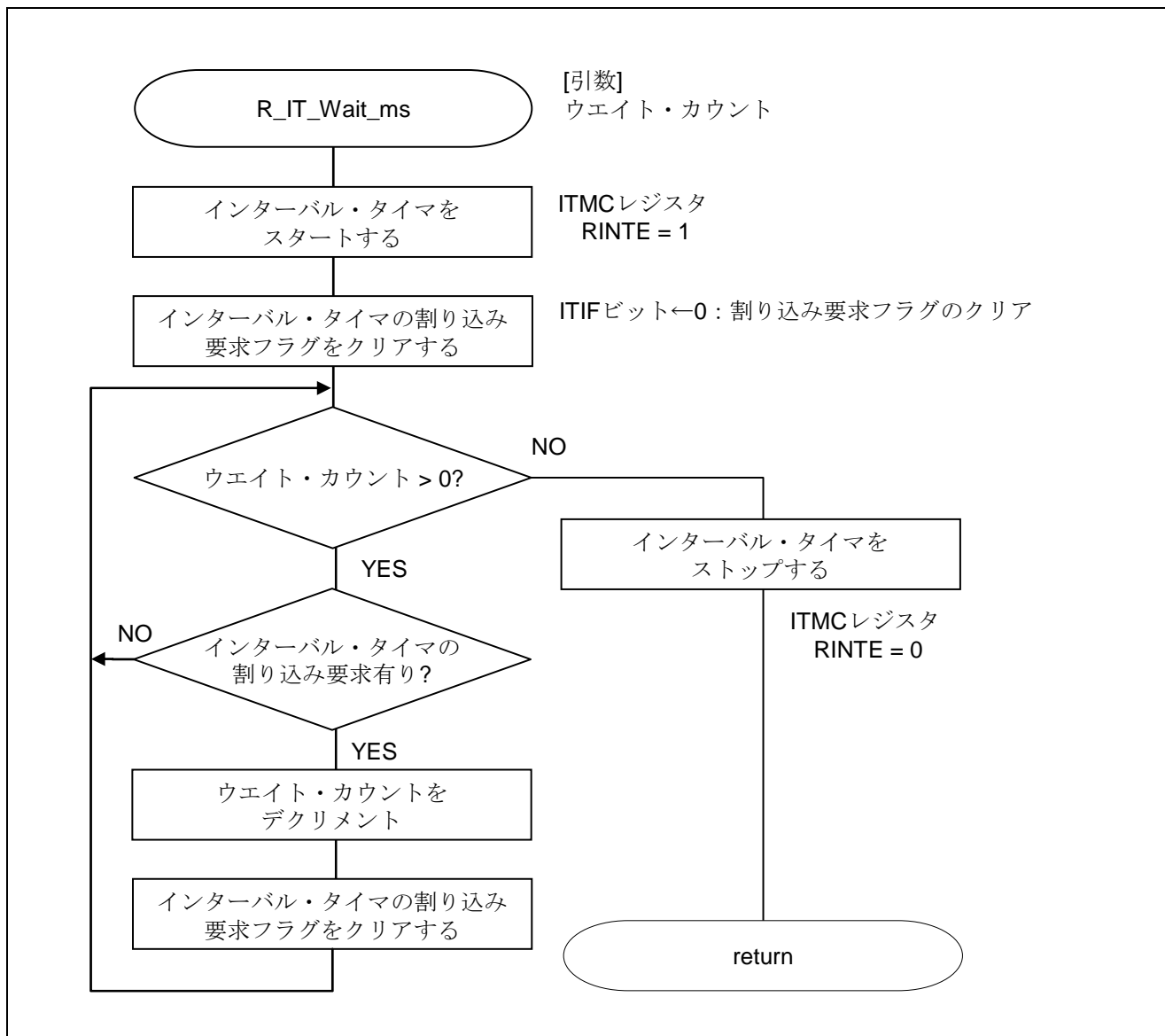


図 5.27 1ms 単位のウェイト

6. 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。

7. 参考ドキュメント

RL78/G13 ユーザーズマニュアル ハードウェア編 Rev.1.00 (R01UH0146J)

RL78 ファミリ ユーザーズマニュアル ソフトウェア編 Rev.1.00 (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/テクニカルニュース

(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。)

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/inquiry>

改訂記録	RL78/G13 クロック発生回路（クロック切り替え）
------	--------------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2015.05.28	—	初版発行

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。

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）がありません。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っていると、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。型名が異なる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、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したものです。誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 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に関し、当社は、何らの責任を負うものではありません。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. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じて、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、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い合わせください。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。お客様がかかる法令を遵守しないこと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。
9. 本資料に記載されている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。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事情報に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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。

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とその総株主の議決権の過半数を直接または間接に保有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。



ルネサス エレクトロニクス株式会社

営業お問い合わせ窓口

<http://www.renesas.com>

営業お問い合わ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 〒100-0004 千代田区大手町2-6-2 (日本ビル)

技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問い合わせ窓口：<http://japan.renesas.com/contact/>