

# SH7786 グループ

## SH7786 INTC CPU 間割り込み使用例

R01AN0808JJ0100  
Rev1.00  
2011.10.01

### 要旨

この資料は、SH7786 の割り込みコントローラの機能にある CPU 間割り込みの使用例を掲載しています。

### 動作確認デバイス

SH7786

### 目次

1.	はじめに .....	2
1.1	仕様 .....	2
1.2	使用機能 .....	2
1.3	適用条件 .....	2
1.4	関連アプリケーションノート .....	3
2.	INTC CPU間割り込み使用例 .....	4
2.1	応用例の説明 .....	4
2.1.1	使用機能の動作概要 .....	4
2.1.2	CPU間割り込み発生要因 .....	5
2.1.3	参考プログラムの説明 .....	6
2.1.4	参考プログラムのレジスタ設定 .....	7
2.1.5	参考プログラムのシーケンス .....	10
2.1.6	参考プログラムの処理手順 .....	12
2.1.7	参考プログラム例 .....	24
2.1.8	プログラム作成の注意点 .....	30
3.	参考ドキュメント .....	31
	ホームページとサポート窓口 .....	31

## 1. はじめに

### 1.1 仕様

本アプリケーションノートでは、割り込みコントローラ(INTC)の CPU 間割り込みを、タイマを使用して発生させる毎に各 CPU で基板上の LED の点灯/消灯を行う使用方法を例にして掲載しています。

### 1.2 使用機能

割り込みコントローラ(INTC CPU 間割り込み)  
 タイマユニット(TMU チャンネル 0, チャンネル 1)  
 汎用入出力ポート(GPIO PortG)

### 1.3 適用条件

評価ボード	アルファプロジェクト製 AP-AH4AD-0A(注 1) 外付けメモリ エリア 0: NOR 型 Flash メモリ 16M バイト Spansion 製 S29GL128P90TFIR20 エリア 2~5: DDR3-SDRAM 256M バイト Micron 製 MT41J64M16LA-187E (2 個)
マイコン	SH7786
動作周波数	内部クロック 533MHz SuperHyway クロック 267MHz 周辺クロック 44MHz DDR3 クロック 533MHz 外部バスクロック 89MHz
エリア 0 バス幅	16bit(MD4 端子=Low レベル, MD5 端子=High レベル, MD6 端子=Low レベル)
クロック動作モード	クロックモード 3 (MD0 端子=High レベル, MD1 端子=High レベル, MD2 端子=Low レベル, MD3 端子=Low レベル)
エンディアン	リトルエンディアン(MD8 端子=High レベル)
アドレスモード	29 ビットアドレスモード(MD10 端子=Low レベル)
ツールチェイン	Super-H RISC engine Standard Toolchain Ver9.3.2.0
コンパイルオプション	High-performance Embedded Workshop で include 指定以外はデフォルト設定 -cpu=sh4a -endian=little -include="\$ (PROJDIR)¥inc¥drv", "\$ (PROJDIR)¥inc" -object="\$ (CONFIGDIR)¥\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo
アセンブラオプション	cpu=sh4a -endian=little -round=zero -denormalize=off -include="\$ (PROJDIR)¥inc" -include="\$ (PROJDIR)¥inc¥drv" -debug -object="\$ (CONFIGDIR)¥\$(FILELEAF).obj" -literal=pool,branch,jump,return -nolist -nologo -chgincpath -errorpath

(注 1) AP-AH4AD-0A の使用方法等の詳細は、「AP-AH4AD-0A Hardware Manual」を参照してください。

表 1.3 に本参考プログラムのセクション配置を示します。

表 1.3 セクション配置

セクション名	セクション用途	領域	配置アドレス(仮想アドレス)	
INTHandler	例外/割り込みハンドラ	ROM	0x00000800	P0 領域 (キャッシング可能, MMU アドレス変換不可)
VECTTBL	リセットベクタテーブル 割り込みベクタテーブル	ROM		
INTTBL	割り込みマスクテーブル	ROM		
PIntPRG	割り込み関数	ROM		
PRResetPRG	リセットプログラム	ROM	0x00002000	
C\$BSEC	未初期化データ領域用アドレス構造	ROM	0x00004000	
C\$DSEC	初期化データ領域用アドレス構造	ROM		
P	CPU0 で使用するプログラム領域	ROM		
D	CPU0 で使用する初期化データ	ROM		
Pcpu1	CPU1 で使用するプログラム領域	ROM	0x00008000	
Dcpu1	CPU 1 で使用する初期化データ	ROM		
RSTHandler_cpu0	CPU0 のリセットハンドラ	ROM	0xA0000000	P2 領域 (キャッシング不可, MMU アドレス変換不可)
RSTHandler_cpu1	CPU1 のリセットハンドラ	ROM	0xA0000100	
B	CPU0 で使用する未初期化データ領域	RAM	0xE4000000	P4 領域 (内部リソースにマッピングされる領域)
R	CPU0 で使用する初期化データ領域	RAM		
Bcpu1	CPU1 で使用する未初期化データ領域	RAM		
Rcpu1	CPU1 で使用する初期化データ領域	RAM		
S	CPU0 で使用するスタック領域	RAM	0xE500E000	
Scpu1	CPU1 で使用するスタック領域	RAM	0XE500E400	

※セクション用途で CPU の指定がないセクションは、共通で使用しています。

#### 1.4 関連アプリケーションノート

本資料の参考プログラムは、「SH7786 グループ アプリケーションノート SH7786 初期設定例 (R01AN0519JJ0101)」の設定条件で動作確認しています。

そちらも合わせてご参照ください。

## 2. INTC CPU間割り込み使用例

### 2.1 応用例の説明

#### 2.1.1 使用機能の動作概要

割り込みコントローラ (INTC) は、割り込み要因の優先順位を判定し、CPU0, CPU1 (SH-4A) への割り込み要求を制御します。INTCには、各割り込みの優先順位を設定するためのレジスタがあり、ユーザがこのレジスタに設定した優先順位に従って割り込み要求を処理します。

以下に、INTCの特長とブロック図を示します。

- 外部割り込みおよび内蔵周辺モジュール割り込みを固定分配モードまたは自動分配モードに設定可能
- 外部割り込みの割り込み優先順位を 15 レベル設定可能
- NMI ノイズキャンセル
- SR.BL ビットが 1 にセットされたときの NMI 要求のマスクが可能
- SR.IMASK ビットを受け付けた割り込みレベルに自動更新可能
- 内蔵周辺モジュール割り込みの優先順位は 30 レベル設定可能
- ユーザモード割り込み禁止機能

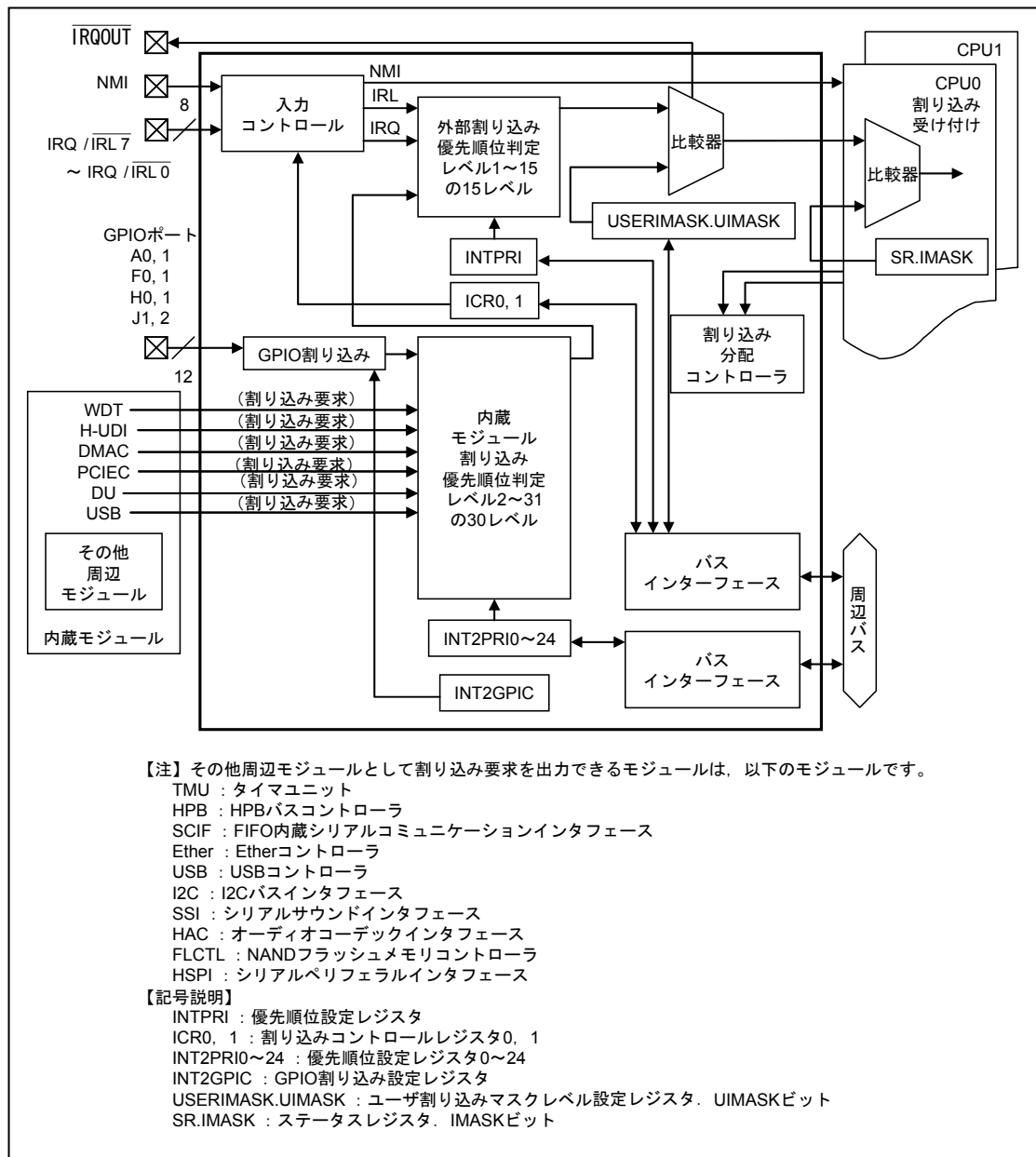


図 2.1.1 INTC ブロック図

## 2.1.2 CPU間割り込み発生要因

CPU間割り込みについては、以下の動作となります。

- CPU間割り込み制御レジスタ(CnINTICI : n=0, 1)への設定によって発生します。
- 1CPUにつき8要因(#0~#7の8フィールド)の入力とし、1フィールドにつき4ビットの論理和で割り込みを発生します。
- (1)要求元CPUごとにフィールドを分けて使用方法と、(2)CPU間で通知する要因ごとにフィールドを分ける方法があります(参考プログラムでは、(1)の方法を使用しています)
- 各CPUへ発生させた割り込みは、フィールド毎に割り込みハンドラのINITCI<sub>n</sub>で処理を行います。本参考プログラムでは、INITCI0にCPU1からCPU0へのCPU間割り込み処理を、INTCI1にCPU0からCPU1へのCPU間割り込み処理を割り当てています。
- 同一フィールドに対して複数のCPUからの書込みがある場合には、CPU毎にフィールド内のビット位置を決めておくことで対応します。
- 割り込み要求を受け付けたCPUはCPU間割り込みクリアレジスタ(CnINTICICLR)でCnINTICIレジスタの対応するビットを0にクリアして割り込み要求をクリアします。

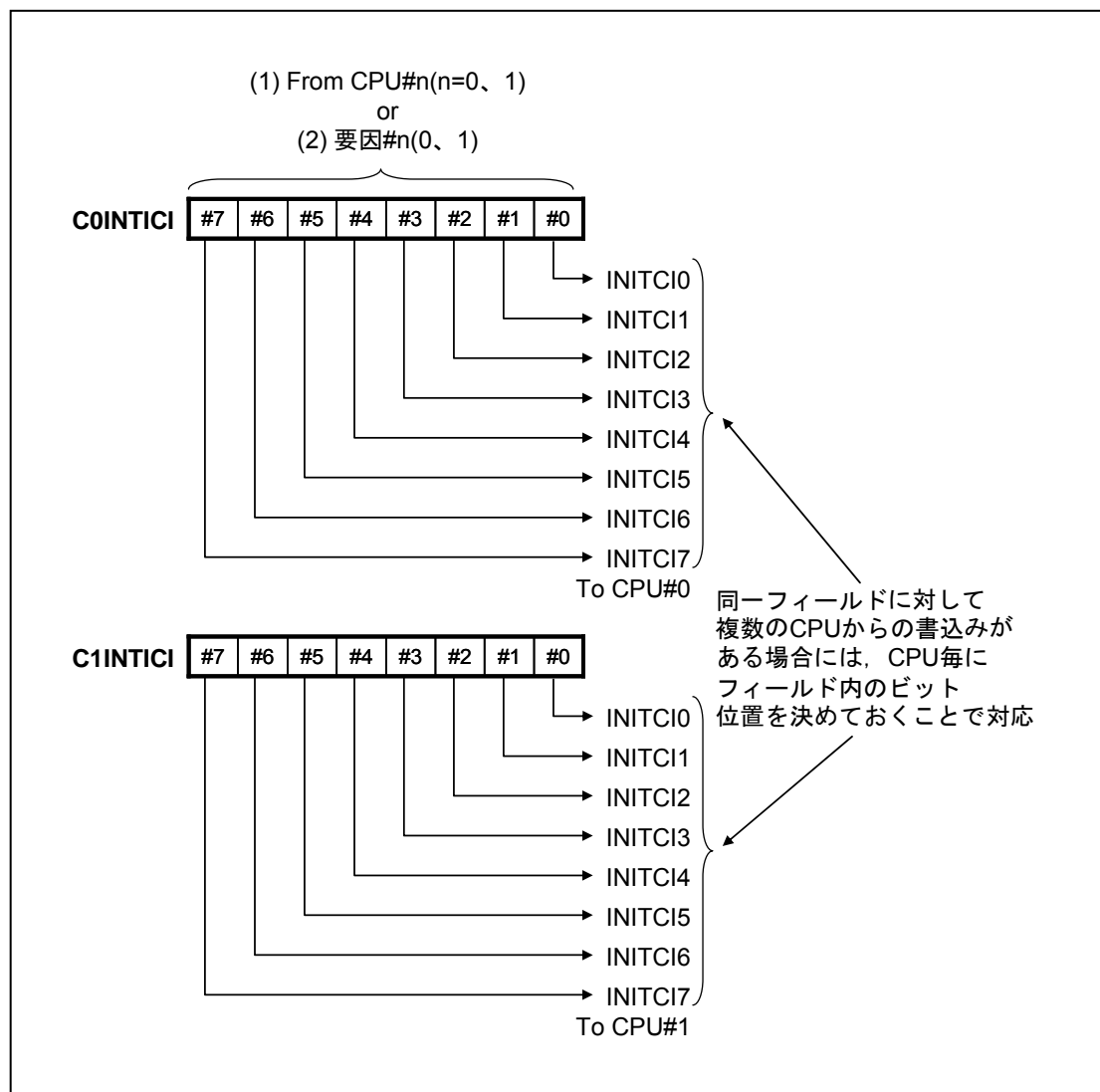


図 2.1.2 CPU 間割り込み

## 2.1.3 参考プログラムの説明

本参考プログラムでは、タイマのアンダーフロー割り込みをトリガにして各 CPU 間割り込みを発生させます。また LED を消灯/点灯させることによって、各 CPU 間割り込みが発生していることを確認できます。

表 2.1.3 に参考プログラムの仕様を示します。

項目	仕様
タイマ	- チャンネル 0(TMU0) - チャンネル 1(TMU1)
タイマ割り込みの発生条件	- TMU0 : 1 秒間隔でアンダーフローを発生 - TMU1 : 0.5 秒間隔でアンダーフローを発生
CPU 間割り込みの発生条件	- CPU0 → CPU1 間割り込み : TMU0 の割り込み - CPU1 → CPU0 間割り込み : TMU1 の割り込み
CPU 間割り込みの割り当て	- CPU0 → CPU1 間割り込み : ICI1 の bit1 - CPU1 → CPU0 間割り込み : ICI0 の bit1 ※ビットの割り当てについては、1 ビット 1 要因として割り当てています。
割り込み優先順位	- TMU0 : 3 - TMU1 : 3 - CPU0 → CPU1 間割り込み(ICIPRI0) : 3 - CPU1 → CPU0 間割り込み(ICIPRI1) : 3
LED 制御	- LD1 : CPU1 で点灯/消灯 - LD2 : CPU0 で点灯/消灯

※一連の制御については、「[2.1.5 参考プログラムのシーケンス](#)」をご参照ください。

## 2.1.4 参考プログラムのレジスタ設定

以下に本参考プログラムのレジスタ設定値を以下に示します。

表 2.1.4.1 INTC CPU 間割り込みレジスタ設定

レジスタ名称(呼称)	アドレス	R/W	サイズ	動作仕様
CPU 間割り込みレジスタ(CPU0) (C0INTICI)	H'FE41 0070	R/W	32	・ CPU0 に対して割り込みを発生 TMU1 割り込み発生後, IC10(bit[3:0])に "B'0001"を設定(CPU1→CPU0 へ割り込み)
CPU 間割り込みレジスタ(CPU1) (C1INTICI)	H'FE41 0074	R/W	32	・ CPU1 に対して割り込みを発生 TMU0 割り込み発生後, IC11(bit[7:4])に "B'0001"を設定(CPU0→CPU1 へ割り込み)
CPU 間割り込みクリアレジスタ(CPU0) (C0INTICICLR)	H'FE41 0080	-/W	32	・ CPU 間割り込み要求のクリア ICICLR0(bit[3:0])="B'0001" : CN0INTICI.IC10 の割り込みをクリア ("0"ライト無効)
CPU 間割り込みクリアレジスタ(CPU1) (C1INTICICLR)	H'FE41 0084	-/W	32	・ CPU 間割り込み要求のクリア ICICLR1(bit[7:4])="B'0001" : CN1INTICI.IC11 の割り込みをクリア ("0"ライト無効)
CPU 間割り込み優先順位設定 レジスタ(CPU0) (C0INTICIPRI)	H'FE41 0090	R/W	32	・ CPU0 に対し優先順位を設定 ICIPRI0(bit[3:0])="B'0011"(優先順位 3)
CPU 間割り込み優先順位設定 レジスタ(CPU1) (C1INTICIPRI)	H'FE41 0094	R/W	32	・ CPU1 に対し優先順位を設定 ICIPRI1(bit[7:4])="B'0011"(優先順位 3)
CPU 間割り込み優先順位クリア レジスタ(CPU0) (C0INTICIPRICLR)	H'FE41 00A0	R/W	32	・ 設定した優先順位をクリア C0INTICIPRI.ICIPRI0 の割り込みをクリア ("0"ライト無効)
CPU 間割り込み優先順位クリア レジスタ(CPU1) (C1INTICIPRICLR)	H'FE41 00A4	R/W	32	・ 設定した優先順位をクリア C1INTICIPRI.ICIPRI1 の割り込みをクリア ("0"ライト無効)

表 2.1.4.2 INTC 内蔵周辺モジュール割り込みレジスタ設定

レジスタ名称(呼称)	アドレス	R/W	サイズ	動作仕様
周辺割り込み優先順位設定レジスタ 0 (INT2PRI0)	H'FE41 0800	R/W	32	・ Event Code 毎に優先順位を設定 TMU-ch0_ch2(bit[28:24])="B'0011" (優先順位 3)
周辺割り込み優先順位設定レジスタ 1 (INT2PRI1)	H'FE41 0804	R/W	32	・ Event Code 毎に優先順位を設定 TMU-ch0_ch2(bit[20:16])="B'0011" (優先順位 3)
割り込み要因レジスタ (マスク状態の影響なし)(CPU0) (C0INT2A0_1)	H'FE41 0A04	R	32	・ モジュールの種別毎に割り込み状態を表示 (マスクの影響なし) TMU-ch0_ch2(bit31)="B'0" : 割り込みなし TMU-ch0_ch2(bit31)="B'1" : 割り込みあり
割り込み要因レジスタ (マスク状態の影響あり)(CPU0) (C0INT2A1_1)	H'FE41 0A14	R	32	・ モジュールの種別毎に割り込み状態を表示 (マスクの影響あり) TMU-ch0_ch2(bit31)="B'0" : 割り込みなし TMU-ch0_ch2(bit31)="B'1" : 割り込みあり
割り込み要因レジスタ (マスク状態の影響なし)(CPU1) (C1INT2A0_1)	H'FE41 0B04	R	32	・ モジュールの種別毎に割り込み状態を表示 (マスクの影響なし) TMU-ch0_ch2(bit30)="B'0" : 割り込みなし TMU-ch0_ch2(bit30)="B'1" : 割り込みあり
割り込み要因レジスタ (マスク状態の影響あり)(CPU1) (C1INT2A1_1)	H'FE41 0B14	R	32	・ モジュールの種別毎に割り込み状態を表示 (マスクの影響あり) TMU-ch0_ch2(bit30)="B'0" : 割り込みなし TMU-ch0_ch2(bit30)="B'1" : 割り込みあり
周辺割り込みマスクレジスタ 1(CPU0) (C0INT2MSK1)	H'FE41 0A24	R/W	32	・ モジュールの種別毎にマスクを設定 TMU-ch0_ch2(bit31)="B'0"ライト : ch0 の マスクを解除
周辺割り込みマスクレジスタ 1(CPU1) (C1INT2MSK1)	H'FE41 0B24	R/W	32	・ モジュールの種別毎にマスクを設定 TMU-ch0_ch2(bit31)="B'0"ライト : ch0 の マスクを解除
周辺割り込みマスククリアレジスタ (CPU0) (C0INT2MSKCLR1)	H'FE41 0A34	R/W	32	・ モジュールの種別毎にマスク解除を設定 TMU-ch0_ch2(bit31)="B'1" : TMU-ch0 の マスクを解除
周辺割り込みマスククリアレジスタ (CPU1) (C1INT2MSKCLR1)	H'FE41 0B34	R/W	32	・ モジュールの種別毎にマスク解除を設定 TMU-ch0_ch2(bit30)="B'1" : TMU-ch1 の マスクを解除
周辺割り込み詳細要因表示レジスタ 01 (INT2B01)	H'FE41 0C04	R	32	・ モジュールの種別を更に個別に表示 bit0 : TMU-ch0_ch2(ch0)の割り込みを表示 bit1 : TMU-ch1_ch2(ch1)の割り込みを表示



表 2.1.4.3 TMU0/1 のレジスタ設定

レジスタ名称(呼称)	アドレス	R/W	サイズ	動作仕様
タイマスタートレジスタ 0 (TSTR0) ※チャンネル 0, 1, 2 共通	H'FFD8 0004	R/W	8	<ul style="list-style-type: none"> <li>・ タイマ 0 のカウント開始/停止 STR0(bit0)="B'0": TCNT0 のカウント動作を停止(プログラム開始後にカウントを開始)</li> <li>・ タイマ 1 のカウント開始/停止 STR1(bit1)="B'0": TCNT1 のカウント動作を停止(プログラム開始後にカウントを開始)</li> </ul>
タイマコンスタントレジスタ 0 (TCOR0)	H'FFD8 0008	R/W	32	<ul style="list-style-type: none"> <li>・ アンダーフロー発生後, TCNT にカウント値を再設定 設定値: H'A7D7</li> </ul>
タイマコンスタントレジスタ 1 (TCOR1)	H'FFD8 0014			
タイマカウンタ 0 (TCNT0/1)	H'FFD8 000C	R/W	32	<ul style="list-style-type: none"> <li>・ アンダーフローを発生させるカウントを設定(本プログラムでは 1s 後に発生するように設定) 設定値: H'A7D7</li> </ul>
タイマカウンタ 1 (TCNT1)	H'FFD8 0018			
タイマコントロールレジスタ 0 (TCR1)	H'FFD8 0010	R/W	16	<ul style="list-style-type: none"> <li>・ アンダーフローフラグ UNF(bit8)="B'0": アンダーフローなし UNF(bit8)="B'1": アンダーフローあり</li> <li>・ アンダーフロー制御 UNIE(bit5)="B'0": アンダーフローによる割り込み(TUNI)を許可</li> <li>・ クロックエッジの選択 CKEG0/1(bit3/4)="B'00": 立ち上がりエッジでカウント</li> <li>・ タイムプリスケール TPSC(bit2~0)="B'100": 1024/Pck でカウント</li> </ul>
タイマコントロールレジスタ 1 (TCR1)	H'FFD8 001C			

表 2.1.4.4 低消費電力モードの設定値

レジスタ名称(呼称)	アドレス	R/W	サイズ	動作仕様
CPU1 スタンバイコントロールレジスタ (C1STBCR)	H'FE40 1004	R/W	32	<ul style="list-style-type: none"> <li>・ CPU1 の起動/停止 MSTP(bit0)="B'0": CPU1 の起動</li> </ul>

※パワーオン時, CPU0 は起動していますが, CPU1 は停止しています。

表 2.1.4.5 GPIO のレジスタ設定(LED1/2 の制御)

レジスタ名称(呼称)	アドレス	R/W	サイズ	動作仕様
ポート G コントロールレジスタ (PGCR)	H'FFCC000C	R/W	16	<ul style="list-style-type: none"> <li>・ PG5 モード PG5MD0/1(bit10/11)="BH'01": 出力設定</li> <li>・ PG6 モード PG6MD0/1="B'01": 出力設定</li> </ul>
ポート G データレジスタ (PGDR)	H'FFCC002C	R/W	8	<ul style="list-style-type: none"> <li>・ LED1 制御 PG5DT(bit5)="B'0": 消灯 PG5DT(bit5)="B'1": 点灯</li> <li>・ LED2 制御 PG6DT(bit6)="B'0": 消灯 PG6DT(bit6)="B'1": 点灯</li> </ul>

※使用していないレジスタや設定していないビットは初期値のままです。

## 2.1.5 参考プログラムのシーケンス

参考プログラムのシーケンスを以下に示します。

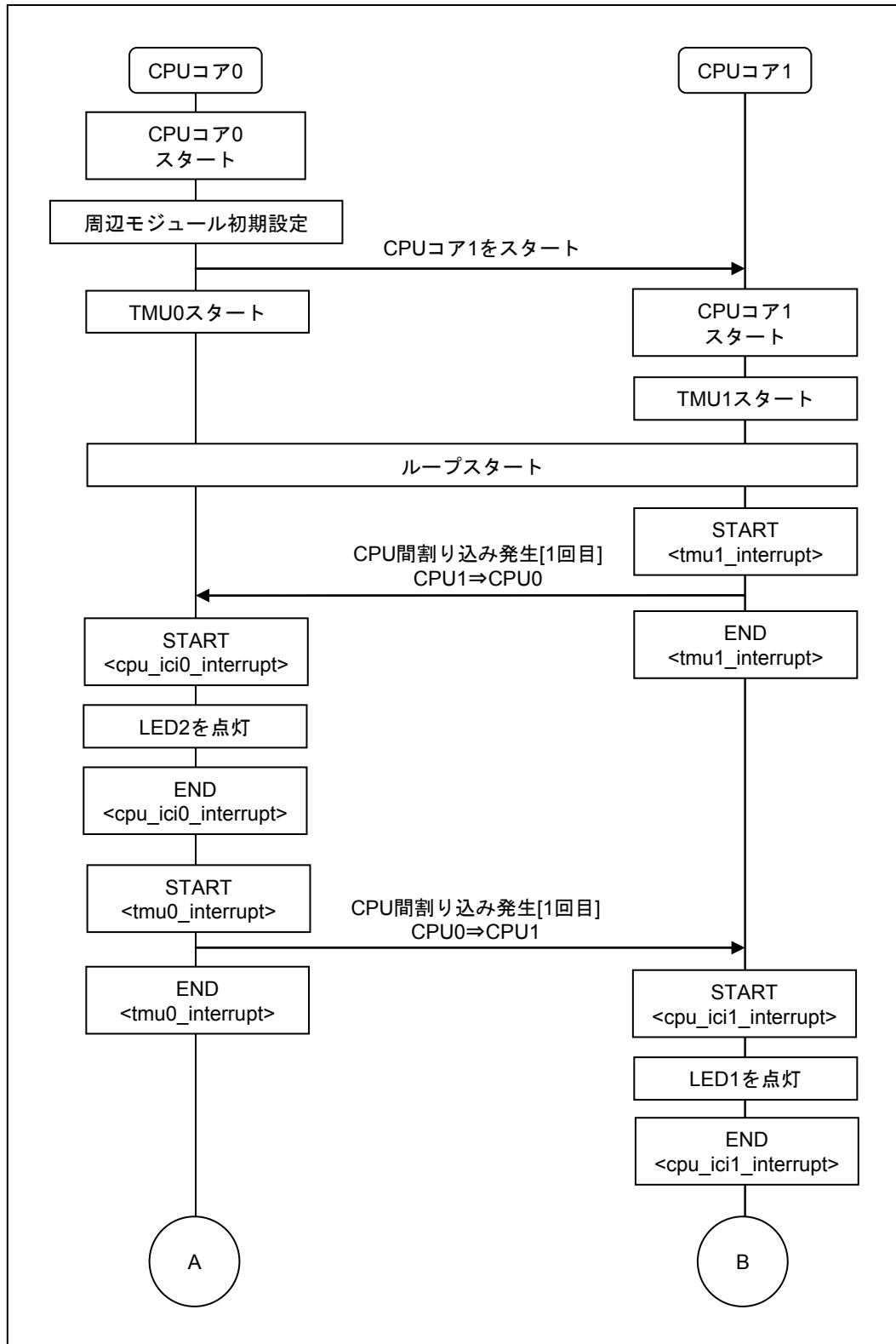


図 2.1.5.1 参考プログラムのシーケンス 1

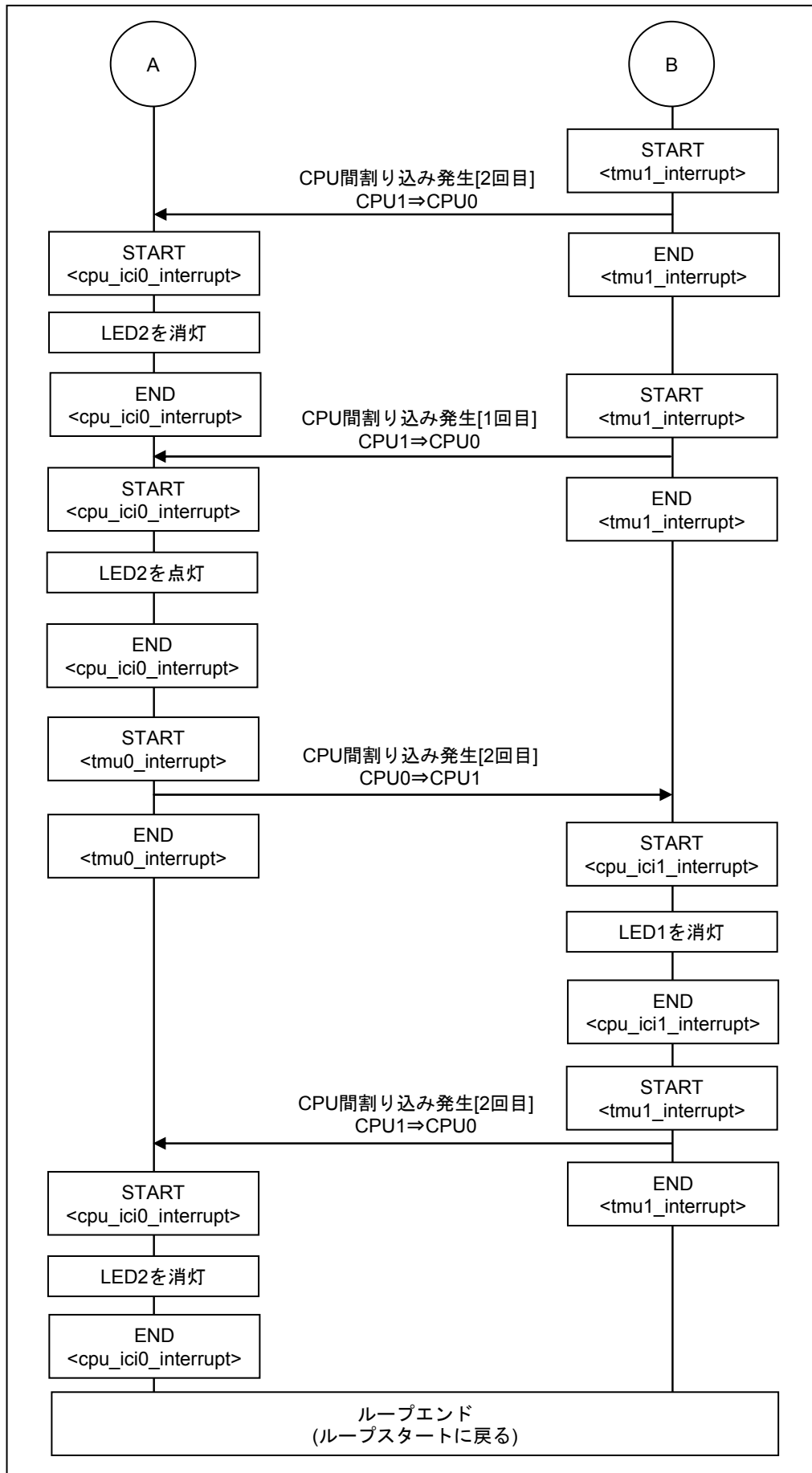


図 2.1.5.2 参考プログラムのシーケンス 2

## 2.1.6 参考プログラムの処理手順

以下に参考プログラムの処理フローを示します。

## 2.1.6.1 CPUコア0 メインフロー

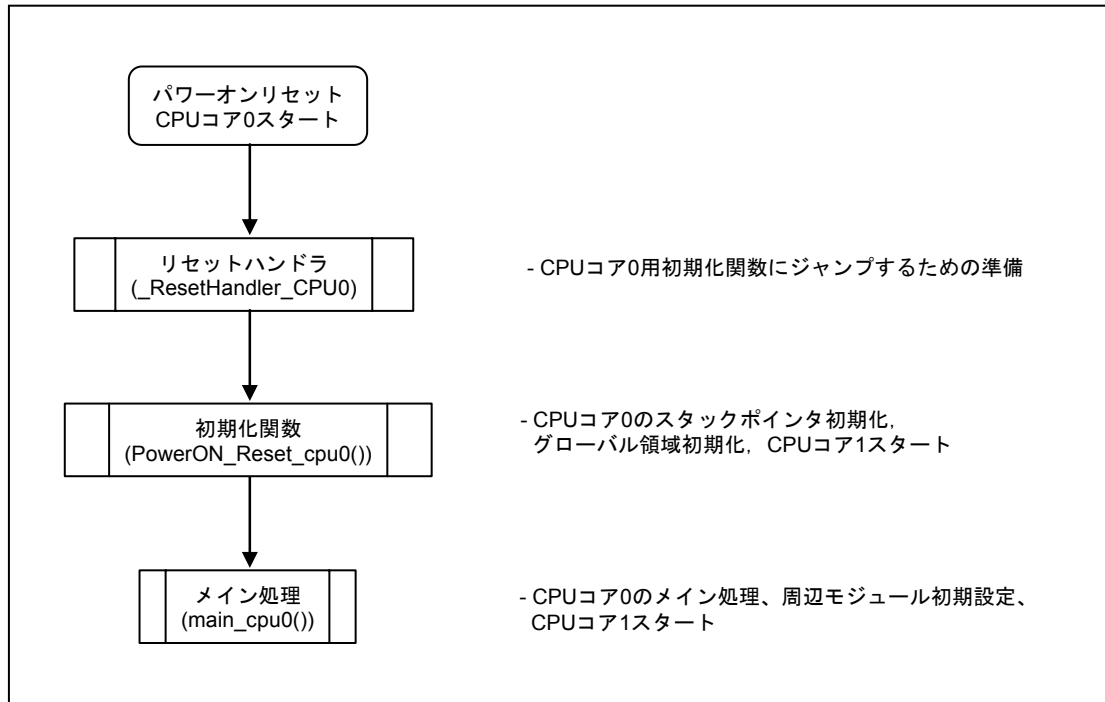


図 2.1.6.1 CPU コア 0 メインフロー

## 2.1.6.2 CPUコア0 リセットハンドラ(ResetHandler\_CPU0)

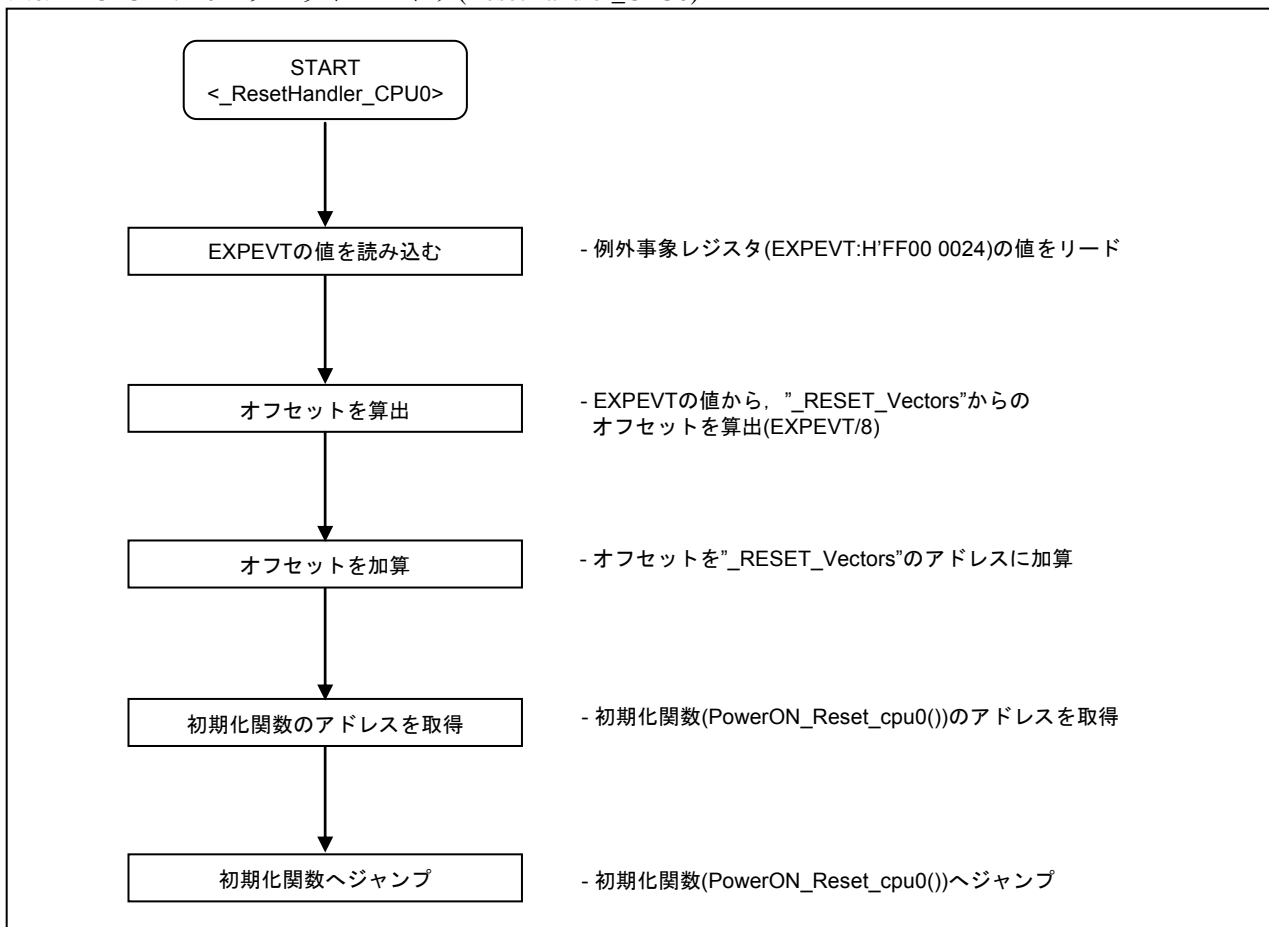


図 2.1.6.2 リセットハンドラフロー

## 2.1.6.3 CPUコア 0 初期化(PowerON\_Reset\_CPU0)

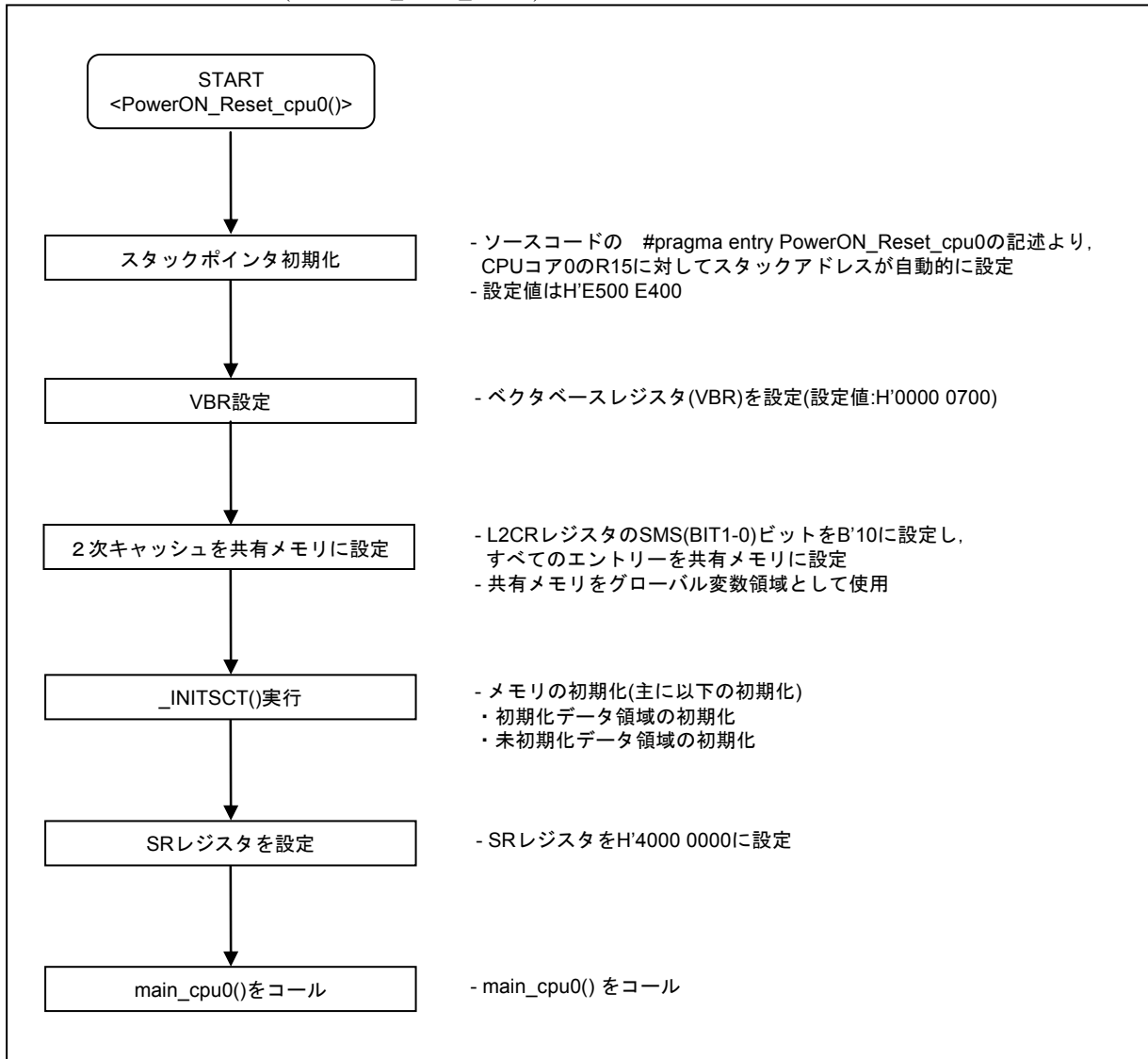


図 2.1.6.3 CPU コア 0 初期化フロー

## 2.1.6.4 CPUコア 0 メイン処理 1(main\_cpu0)

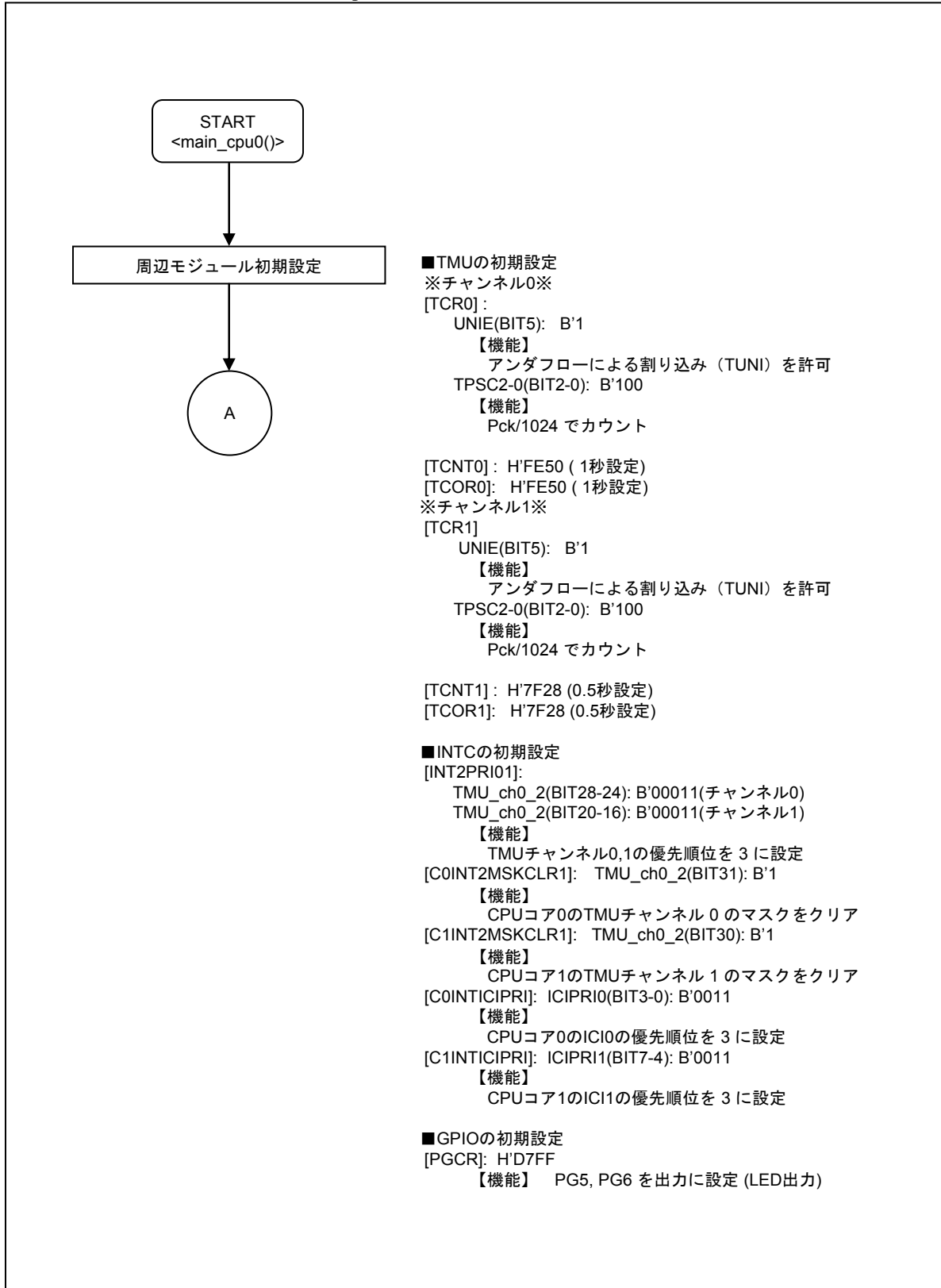


図 2.1.6.4 CPU コア 0 メイン処理 1 フロー

## 2.1.6.5 CPUコア0 メイン処理 2(main\_cpu0)

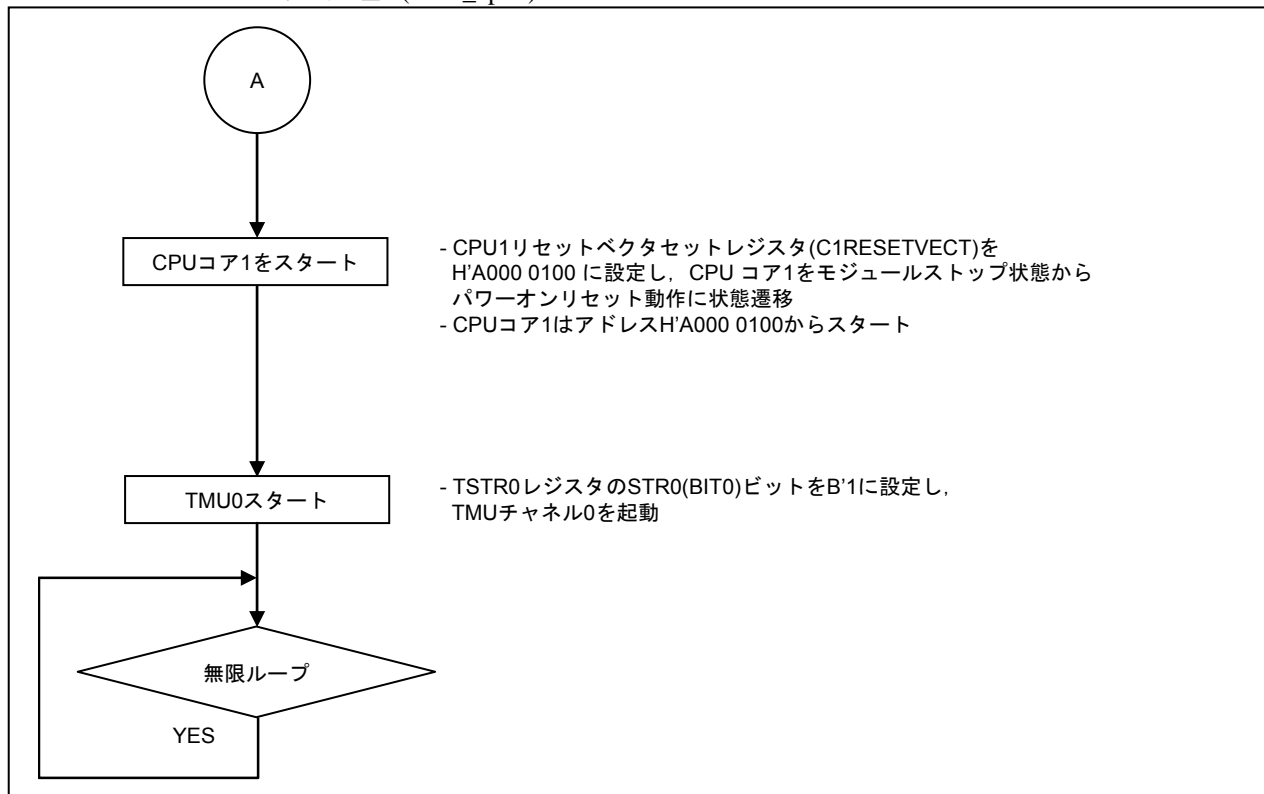


図 2.1.6.5 CPU コア0 メイン処理 2 フロー

## 2.1.6.6 CPUコア 0 TMU0 割り込み処理 1(tmu0\_interrupt)

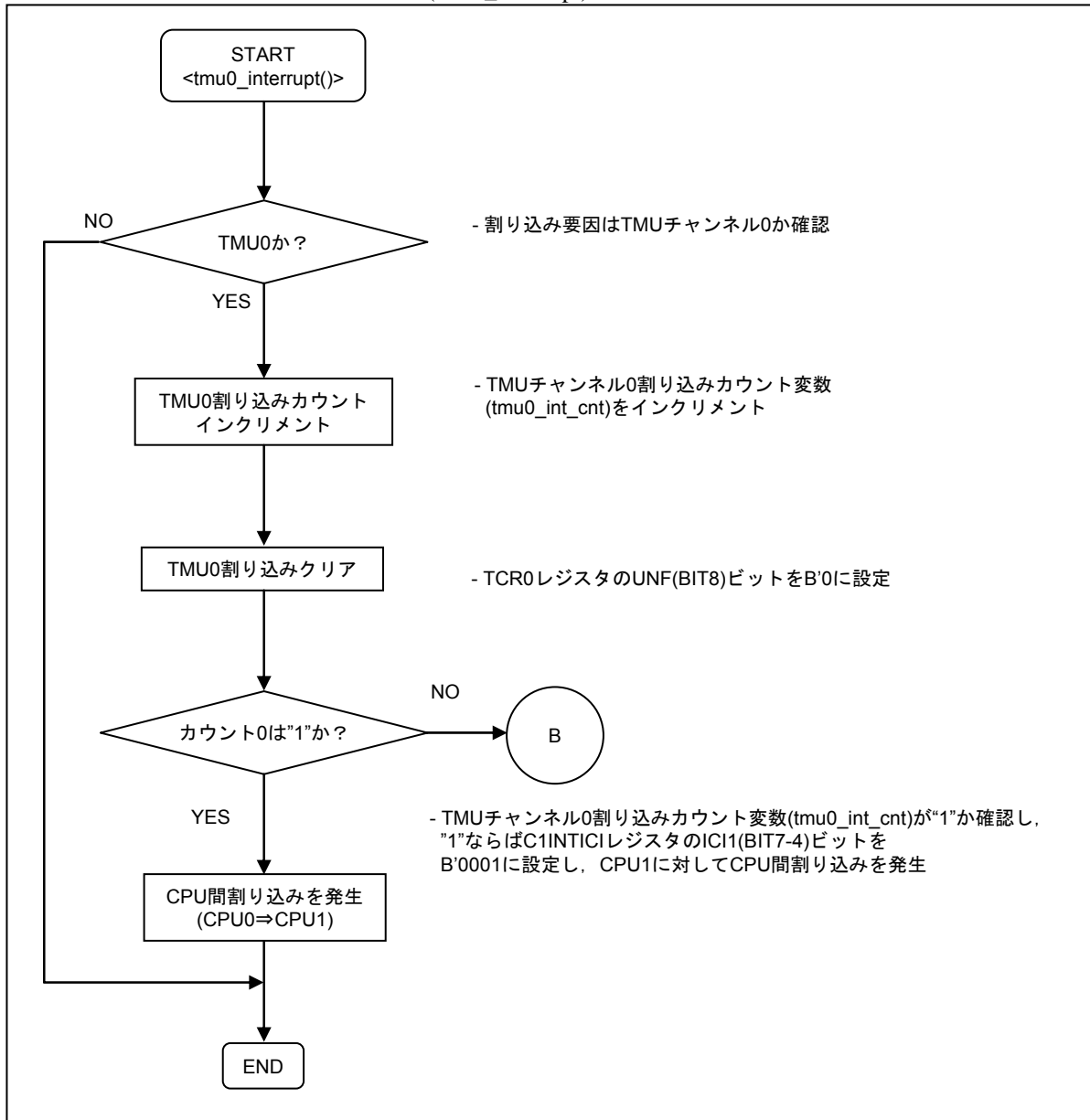


図 2.1.6.6 CPU コア 0 メイン処理 3 TMU0 割り込み 1 フロー



## 2.1.6.7 CPUコア 0 TMU0 割り込み処理 2(tmu0\_interrupt)

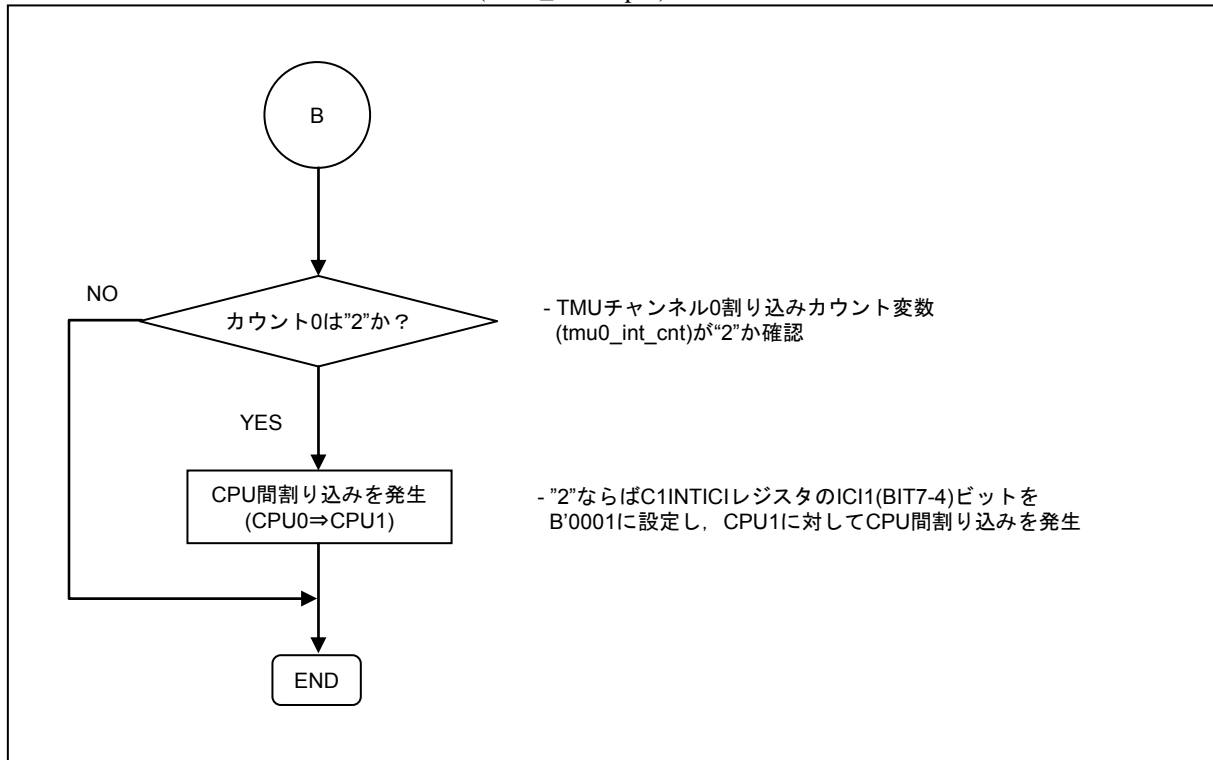


図 2.1.6.7 CPU コア 0 メイン処理 TMU 割り込み 2 フロー

## 2.1.6.8 CPUコア 0 CPU間割り込み処理 1(cpu\_ici0\_interrupt)

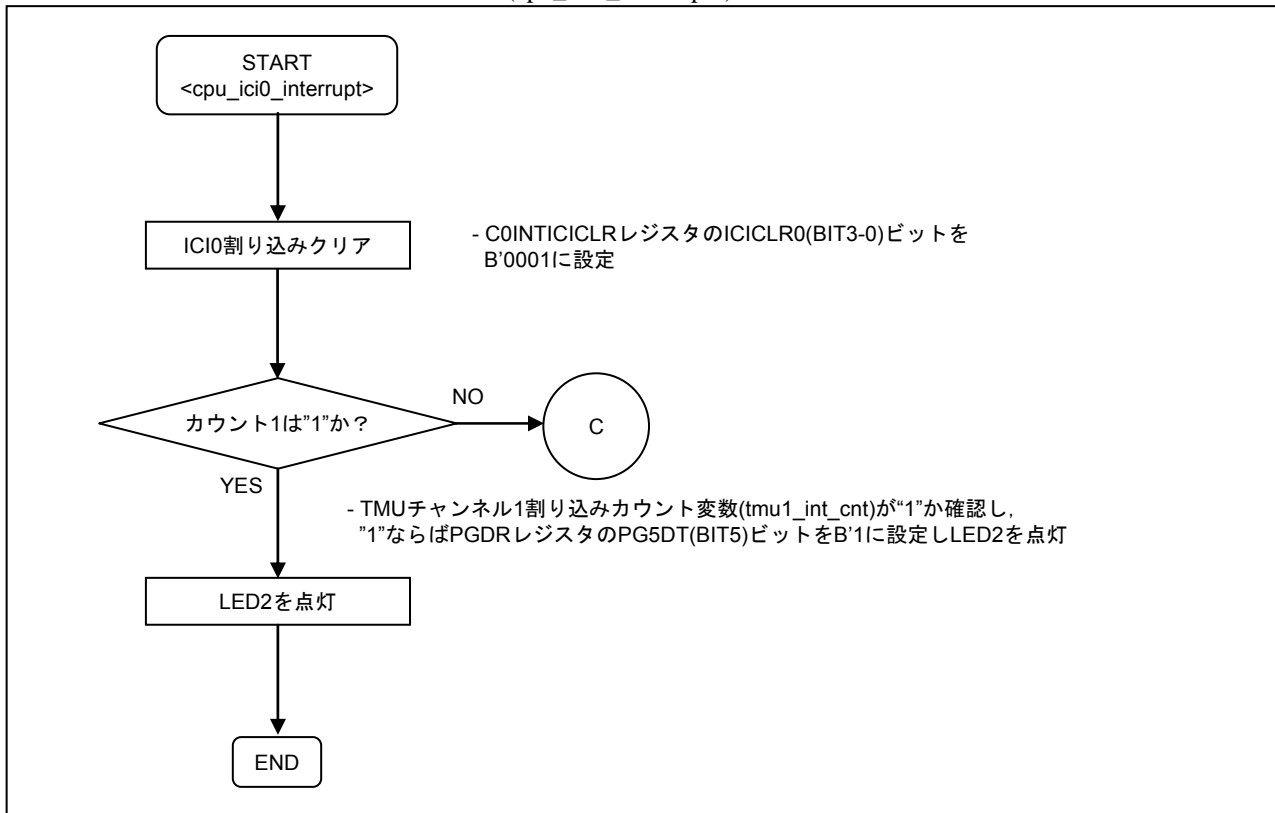


図 2.1.6.8 CPU コア 0 メイン処理 CPU 間割り込み 1 フロー

## 2.1.6.9 CPUコア 0 CPU間割り込み処理 2(cpu\_ici0\_interrupt)

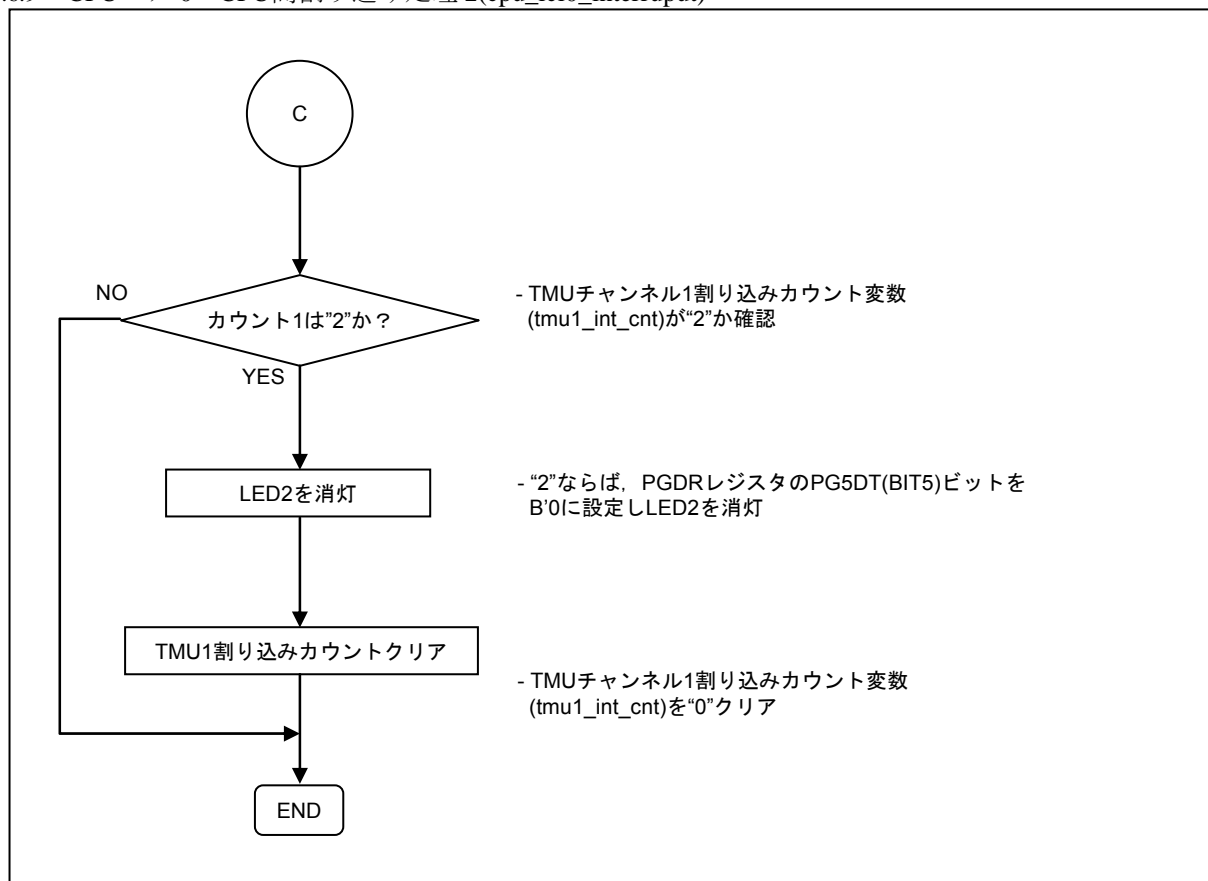


図 2.1.6.9 CPU コア 0 メイン処理 CPU 間割り込み 2 フロー

## 2.1.6.10 CPUコア 1 メインフロー

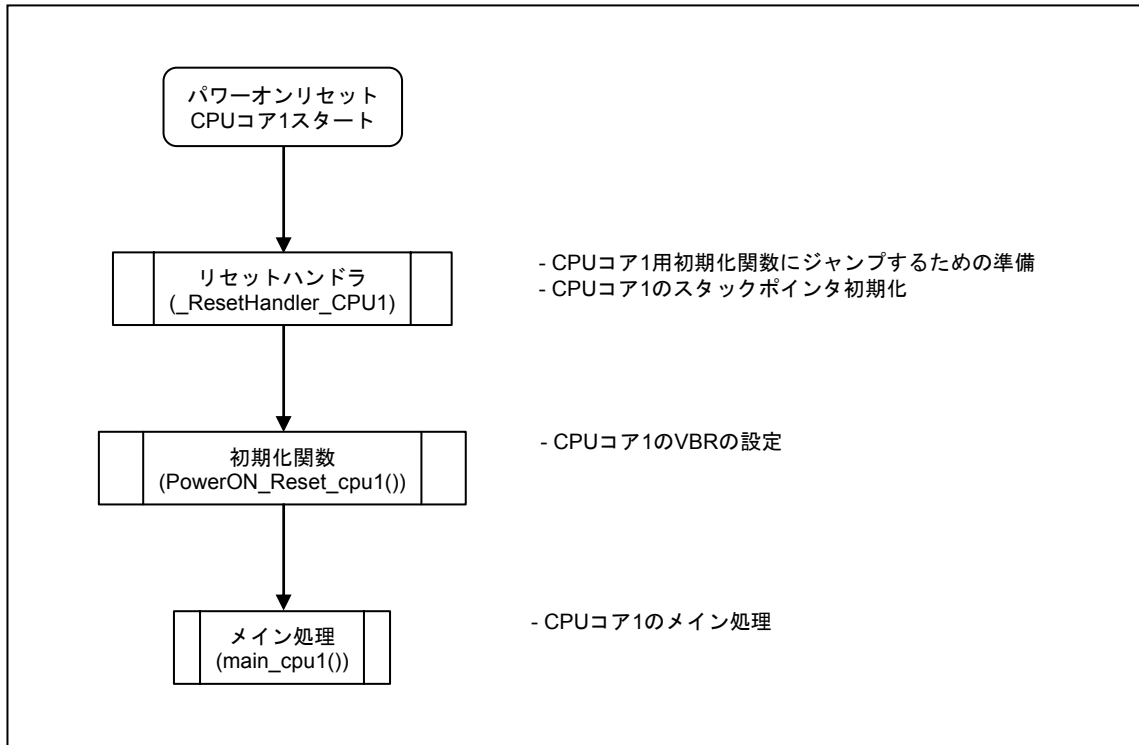


図 2.1.6.10 CPU コア 1 メインフロー

## 2.1.6.11 CPUコア 1 リセットハンドラ(Resethandler\_CPU1)

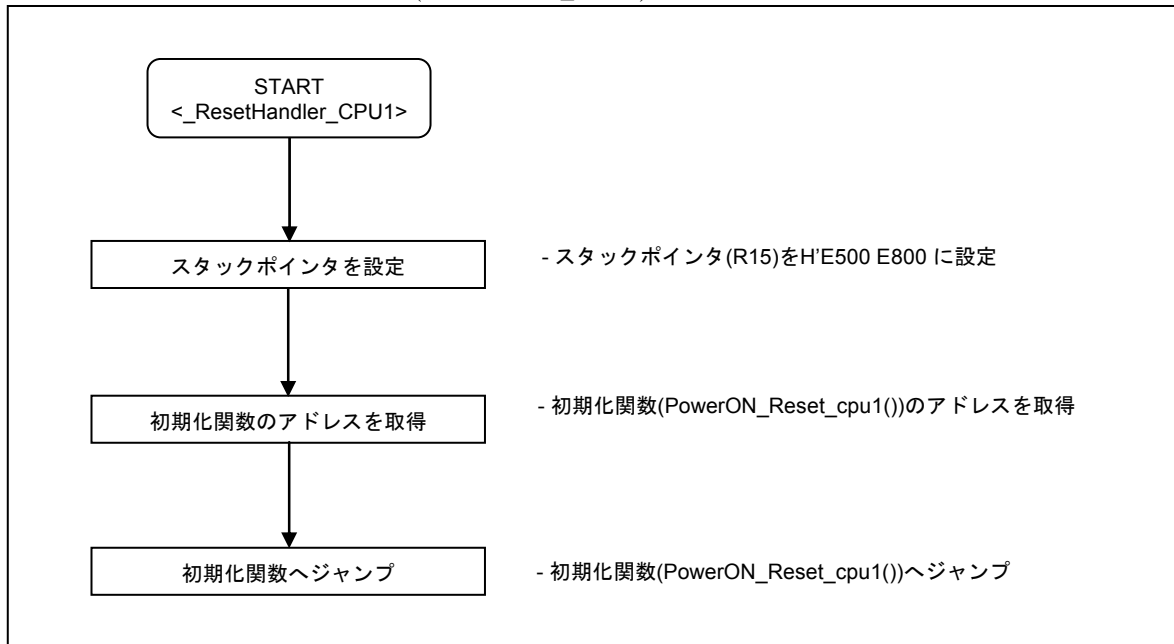


図 2.1.6.11 CPU コア 1 リセットハンドラフロー

## 2.1.6.12 CPUコア 1 初期化(PowerON\_Rset\_CPU1)

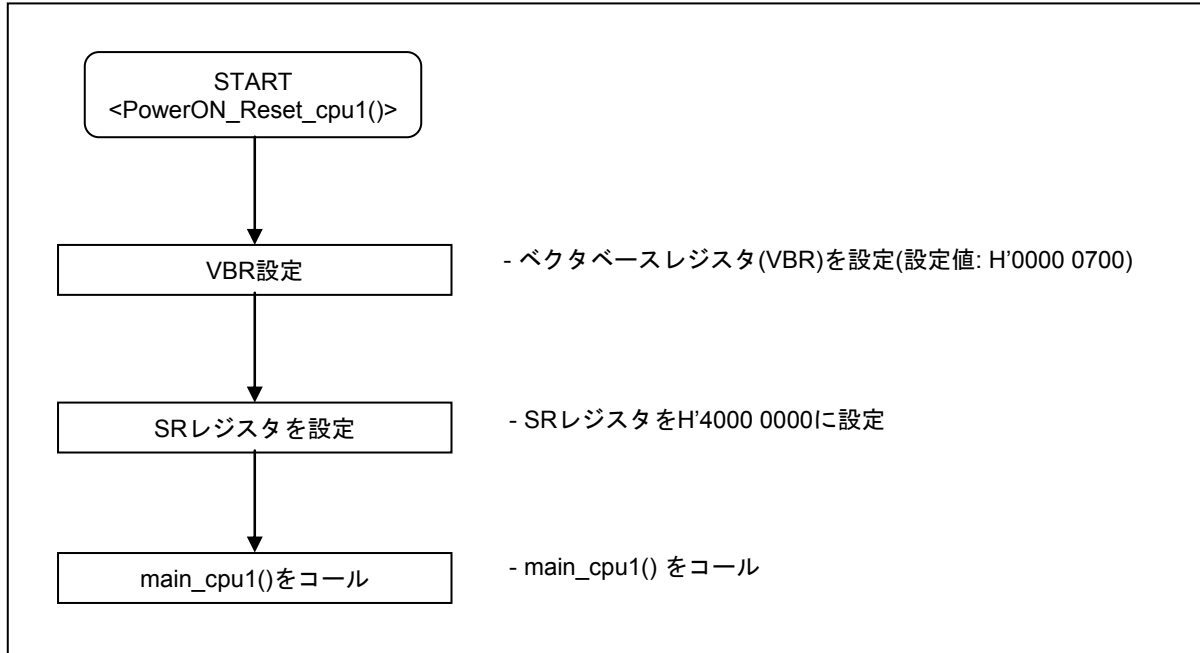


図 2.1.6.12 CPU コア 1 初期化フロー

## 2.1.6.13 CPUコア 1 メイン処理(main\_cpu1)

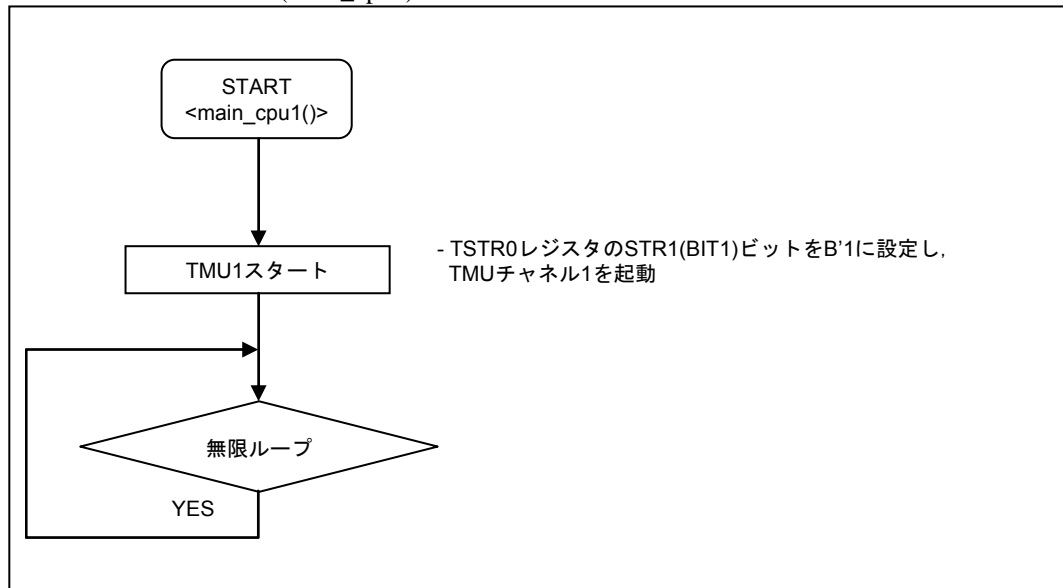


図 2.1.6.13 CPU コア 1 メイン処理(main\_cpu1)

## 2.1.6.14 CPUコア 1 TMU1 割り込み処理 1(tm\_u1\_interrupt)

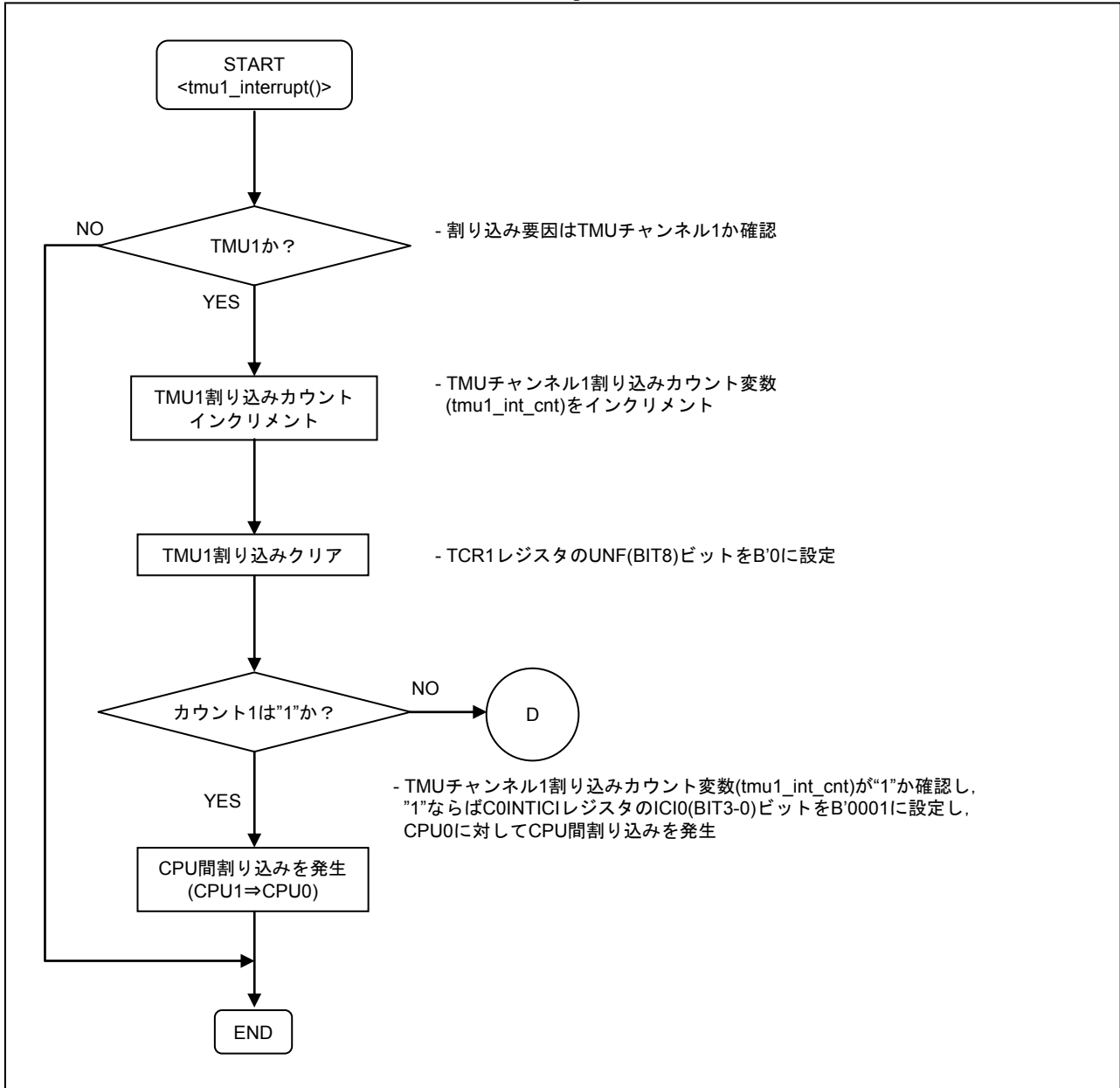


図 2.1.6.14 CPU コア 1 メイン処理フロー

## 2.1.6.15 CPUコア 1 TMU1 割り込み処理 2(tmu1\_interrupt)

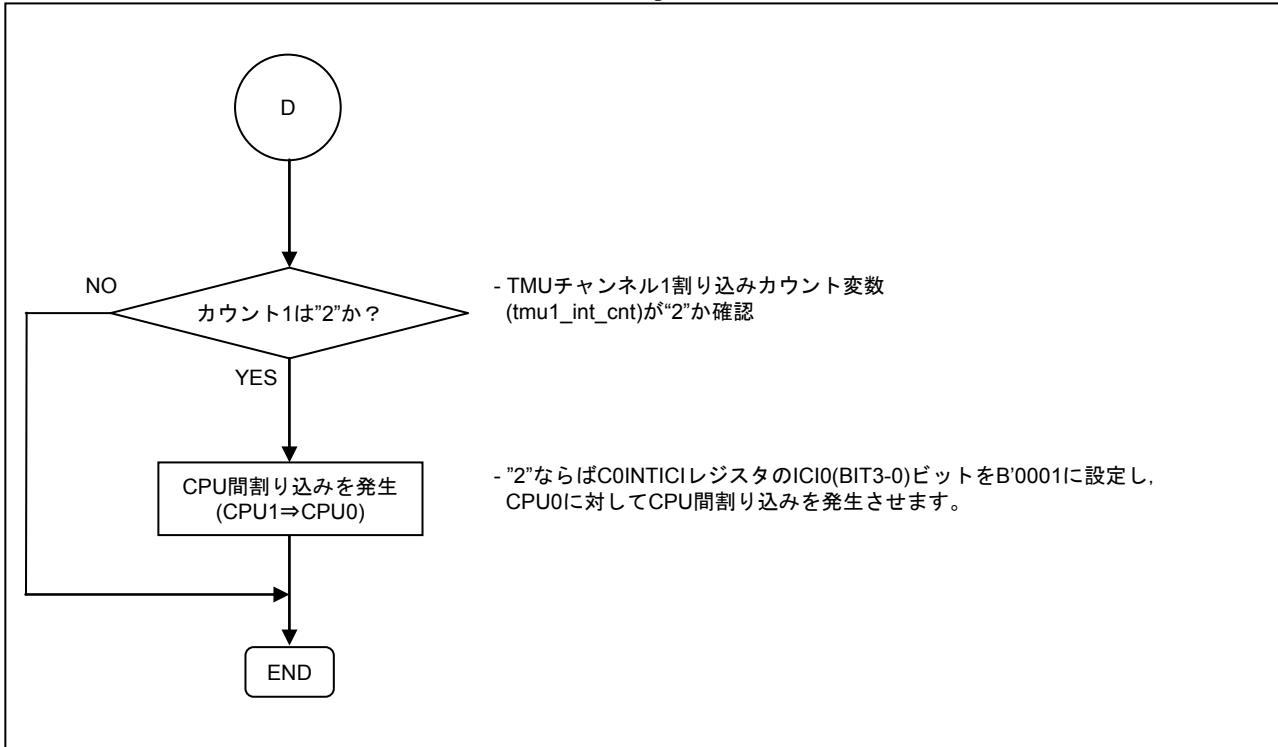


図 2.1.6.15 CPU コア 1 メイン処理 2 TMU1 割り込み 2 フロー

## 2.1.6.16 CPUコア 1 CPU間割り込み処理 1(cpu\_ici1\_interrupt)

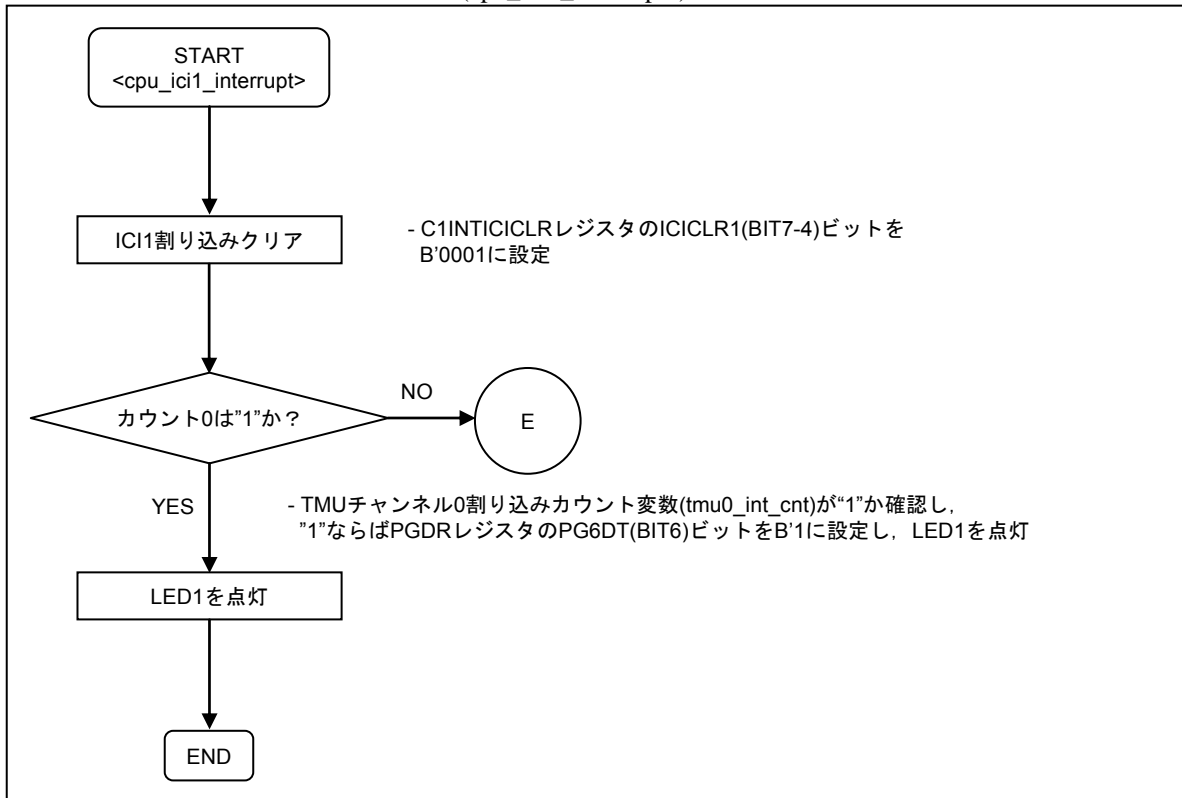


図 2.1.6.16 CPU コア 1 メイン処理 3 CPU 間割り込み 1 フロー

## 2.1.6.17 CPUコア 1 CPU間割り込み処理 2(cpi\_ici1\_interrupt)

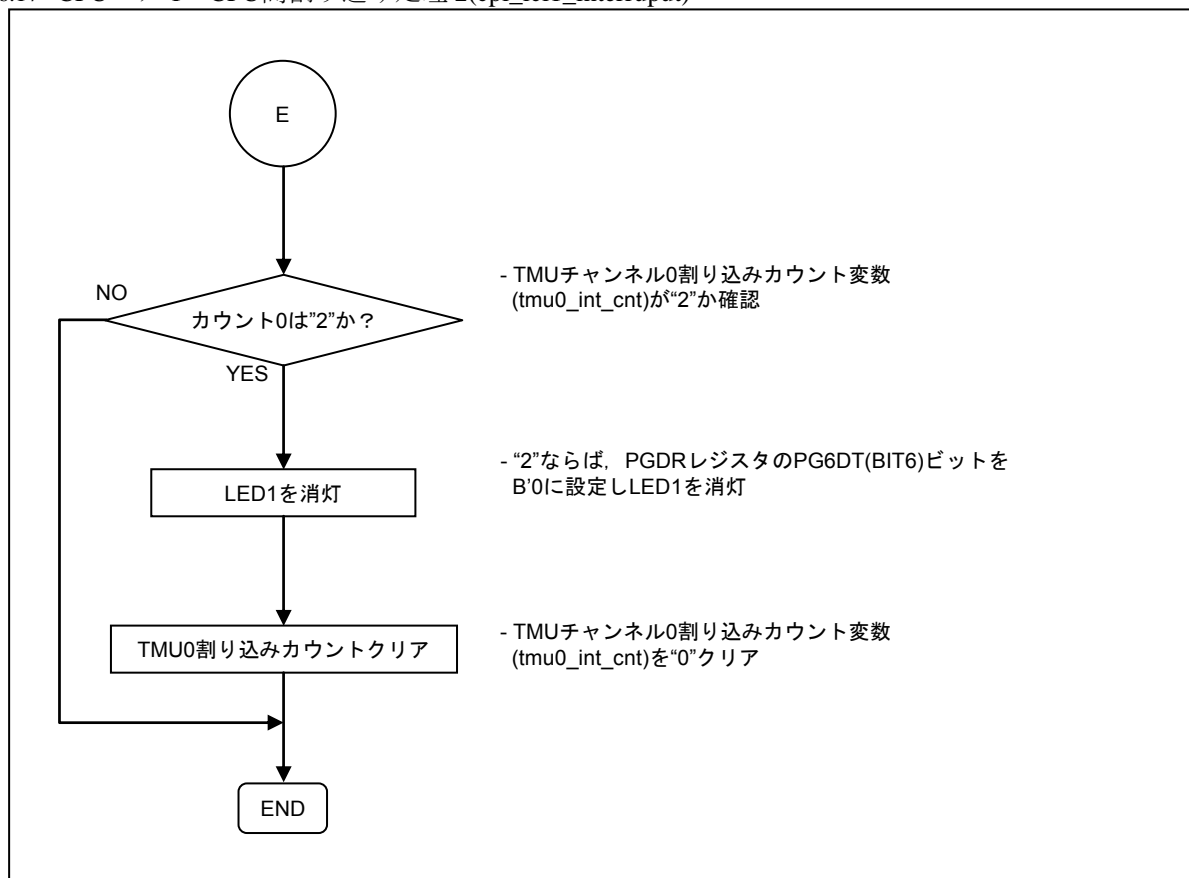


図 2.1.6.17 CPU コア 1 メイン処理 3 CPU 間割り込み 2 フロー

## 2.1.7 参考プログラム例

以下に参考プログラム例を示します。

```

001 /*****
002 ;* DISCLAIMER
003 ;
004 ;* This software is supplied by Renesas Electronics Corporation. and is only
005 ;* intended for use with Renesas products. No other uses are authorized.
006 ;
007 ;* This software is owned by Renesas Electronics Corporation. and is protected under
008 ;* all applicable laws, including copyright laws.
009 ;
010 ;* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
011 ;* REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
012 ;* INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
013 ;* PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
014 ;* DISCLAIMED.
015 ;
016 ;* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
017 ;* ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
018 ;* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
019 ;* FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
020 ;* AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
021 ;
022 ;* Renesas reserves the right, without notice, to make changes to this
023 ;* software and to discontinue the availability of this software.
024 ;* By using this software, you agree to the additional terms and
025 ;* conditions found by accessing the following link:
026 ;* http://www.renesas.com/disclaimer
027 ;*****/
028 /* Copyright (C) 2011. Renesas Electronics Corporation., All Rights Reserved.*/
029 /*****"FILE COMMENT"***** Technical reference data *****/
030 ;* System Name : SH7786 INTC Sample Program
031 ;* File Name : sh7786_intc_sample.c
032 ;* Abstract : Main Program
033 ;* Version : Ver 1.00
034 ;* Device : SH7786
035 ;* Tool-Chain : High-performance Embedded Workshop (Version 4.09.00.007)
036 ;* : C/C++ Compiler Package for SuperH Family (V.9.3.2.0)
037 ;* OS : None
038 ;* H/W Platform : SH-4A Board P/N:AP-SH4AD-0A (Manufacturer:ALPHA PROJECT)
039 ;* Description : Main routines and TMU and ICI interrupt handlers
040 ;* : for CPU CORE0 and CORE1
041 ;* Operation :

```



```
042 ;* Limitation      :
043 ;*                  :
044 ;*****
045 ;* History          : 5.Aug.2011 Ver. 1.00 First Release
046 ;*""FILE COMMENT END""*****/
047 /*****/
048 /*                                                         */
049 /* FILE             :sh7786_intc_sample.c                 */
050 /* DATE              :Wed, Jun 22, 2011                   */
051 /* DESCRIPTION      :Main Program                         */
052 /* CPU TYPE         :Other                                 */
053 /*                                                         */
054 /* This file is generated by Renesas Project Generator (Ver.4.16). */
055 /*                                                         */
056 /*****/
057
058 //#include "typedefine.h"
059 #include "iodefine.h"
060 #include "typedefine.h"
061 #ifdef __cplusplus
062 //#include <ios>                // Remove the comment when you use ios
063 // _SINT ios_base::Init::init_cnt; // Remove the comment when you use ios
064 #endif
065
066 void main_cpu0(void);
067 void main_cpu1(void);
068 void tmu0_interrupt(void);
069 void tmu1_interrupt(void);
070 void cpu_ici0_interrupt(void);
071 void cpu_ici1_interrupt(void);
072 static void init_devices(void);
073
074 #ifdef __cplusplus
075 extern "C" {
076 void abort(void);
077 }
078 #endif
079
080 /*****/
081 /*                                                         */
082 /* Sections for CPU Core 0 */
```

```
083 /* */
084 /*****/
085
086 #pragma section
087
088 static int tmu0_int_cnt = 0;
089 extern int tmu1_int_cnt;
090
091 /* main() for CPU CORE0 */
092 void main_cpu0(void)
093 {
094     tmu0_int_cnt = 0;
095     /* initialize peripherals */
096     init_devices();
097
098     /* Start CPU1 */
099     *(volatile _UINT *)0xFE401008 = 0xA0000100;          /* Set C1RESETVEC reg */
100     *(volatile _UINT *)0xFE401004 = 0x2;                /* Set C1STBCR reg to CPU1 active mode */
101
102     TMU0.TSTR0.BIT.STR0 = 1;                             /* start TMU0 */
103
104     /* Infinite loop */
105     while (1);
106 }
107
108 /* TMU channel0 interrupt handler */
109 void tmu0_interrupt(void)
110 {
111     if (INTC.COINT2A0_1.BIT._TMU00 == 0)                /* If not TMU0 interrupt then just return */
112         return;
113
114     tmu0_int_cnt++;
115     TMU0.TCR0.BIT.UNF = 0;                              /* clear TMU0 interrupt flag */
116
117     if (tmu0_int_cnt == 1)
118         INTC.C1INTIC1.BIT.IC11 = 0x1;                  /* set IC11 bits */
119     else if (tmu0_int_cnt == 2) {
120         INTC.C1INTIC1.BIT.IC11 = 0x1;                  /* set IC11 bits */
121     }
122 }
123
```

```

124 /* INTC ICI0 interrupt handler */
125 void cpu_ici0_interrupt(void)
126 {
127     INTC.C0INTICICLR.BIT.ICICLR0 = 0x1;          /* clear ICICLR0 bits */
128
129     if (tmu1_int_cnt == 1)
130         GPIO.PGDR.BIT.PG5DT = 1;                /* turn on LED2 */
131     else if (tmu1_int_cnt == 2) {
132         GPIO.PGDR.BIT.PG5DT = 0;                /* turn off LED2 */
133         tmu1_int_cnt = 0;
134     }
135 }
136
137 /* initialize peripherals */
138 static void init_devices(void)
139 {
140     /* initialize TMU */
141     TMU0.TCR0.WORD = 0x0024;                      /* set TMU0 interrupt enabled and select
1/1024 prescalar */
142     TMU0.TCNT0 = 0xFE50;                          /* set TMU0 TCNT to 1 second */
143     TMU0.TCOR0 = 0xFE50;                          /* set TMU0 TCOR to 1 second */
144     TMU0.TCR1.WORD = 0x0024;                      /* set TMU1 interrupt enabled and select
1/1024 prescalar */
145     TMU0.TCNT1 = 0x7F28;                          /* set TMU1 TCNT to 0.5 second */
146     TMU0.TCOR1 = 0x7F28;                          /* set TMU1 TCOR to 0.5 second */
147
148     /* initialize INTC */
149     INTC.INT2PRI1.BIT.TMU00 = 0x3;                /* set TMU0 priority to 3 */
150     INTC.C0INT2MSKCLR1.BIT._TMU00 = 1;          /* clear CPU0 TMU0 interrupt mask */
151     INTC.INT2PRI1.BIT.TMU01 = 0x3;                /* set TMU1 priority to 3 */
152     INTC.C1INT2MSKCLR1.BIT._TMU01 = 1;          /* clear CPU1 TMU1 interrupt mask */
153     INTC.C0ICIPRI.BIT.ICIPRI0 = 0x3;            /* set cpu0 ICI0 priority to 3 */
154     INTC.C1ICIPRI.BIT.ICIPRI1 = 0x3;            /* set cpu1 ICI1 priority to 3 */
155
156     /* initialize GPIO */
157     GPIO.PGCR.WORD = 0xD7FF;                      /* set PG5 and PG6 to output mode */
158 }
159
160 /******
161 /*
162 /*     Sections for CPU Core 1
163 /*

```

```
165
166 #pragma section cpu1
167
168 static int tmu1_int_cnt = 0;
169
170 /* main() for CPU CORE1 */
171 void main_cpu1(void)
172 {
173     tmu1_int_cnt = 0;
174
175     TMU0.TSTR0.BIT.STR1 = 1;           /* start TMU1 */
176
177     /* Infinite loop */
178     while(1);
179 }
180
181 /* TMU channel1 interrupt handler */
182 void tmu1_interrupt(void)
183 {
184     if (INTC.C1INT2A0_1.BIT._TMU01 == 0) /* If not TMU1 interrupt then just return */
185         return;
186
187     tmu1_int_cnt++;
188     TMU0.TCR1.BIT.UNF = 0;           /* clear TMU1 interrupt flag */
189
190     if (tmu1_int_cnt == 1)
191         INTC.C0INTIC1.BIT.ICI0 = 0x1; /* set ICI0 bits */
192     else if (tmu1_int_cnt == 2) {
193         INTC.C0INTIC1.BIT.ICI0 = 0x1; /* set ICI0 bits */
194     }
195 }
196
197 /* INTC ICI1 interrupt handler */
198 void cpu_ici1_interrupt(void)
199 {
200     INTC.C1INTICICLR.BIT.ICICLR1 = 0x1; /* clear ICICLR1 bits */
201
202     if (tmu0_int_cnt == 1)
203         GPIO.PGDR.BIT.PG6DT = 1;     /* turn on LED1 */
204     else if (tmu0_int_cnt == 2) {
205         GPIO.PGDR.BIT.PG6DT = 0;     /* turn off LED1 */

```

```
206         tmu0_int_cnt = 0;
207     }
208 }
209
210 #ifdef __cplusplus
211 void abort(void)
212 {
213
214 }
215 #endif
216
217
```

### 2.1.8 プログラム作成の注意点

INTC の CPU 間割り込みを使用したプログラムを作成する際の注意点を以下に示します。

#### 1. CPU0/1 の動作状態について

CPU0/1 の動作状態は、「SH7786 グループ ハードウェアマニュアル 20. 低消費電力モード」に記載されている CPU0/1 スタンバイコントロールレジスタ(C0STBCR, C1STBCR)の MSTP0 ビット, MSTP1 ビットの状態で確認できます。また CPU0 はパワーオンリセット解除後から動作していますが, CPU1 はモジュールストップ状態になっています。CPU1 を起動するには, MSTP ビットを 0 に設定してください。

#### 2. CPU0/1 スタンバイコントロールレジスタ(C0STBCR, C1STBCR) RESET0/1 ビットについて

CPU0/1 の起動/再起動時にソフトウェアリセットを発行できますが, リセットを発行する場合は, CPU0/1 がモジュールストップ状態(MSTP0/1 が"1")の時に発行してください。また, MSTP0/1 が"0"である時に REEST0/1 ビットに"1"を設定しないでください。詳細は、「SH7786 グループ ハードウェアマニュアル 20.5.3 CPU コアモジュールストップ制御について」をご参照ください。

### 3. 参考ドキュメント

#### ソフトウェアマニュアル

SH4-A ソフトウェアマニュアル(RJJ09B0090)

(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)

#### ハードウェアマニュアル

SH7786 グループハードウェアマニュアル(RJJ09B0533)

(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)

#### ルネサスマイコン開発環境マニュアル

SuperH マルチコアマイコン E-10A エミュレータ ユーザーズマニュアル (HS0005KCU04H 用)  
(RJJ10J2095)

SuperH マルチコアマイコン E-10A エミュレータ ユーザーズマニュアル 別冊 SH7786 ご使用時の補  
足説明(RJJ10J2598)

#### 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/>

お問合せ先

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

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	2011.10.1	-	初版発行