

# RX62N、RX621 グループ

## メモリプロテクションユニットの設定例

R01AN1078JJ0100  
Rev.1.00  
2014.07.01

### 要旨

本アプリケーションノートでは、RX62N、RX621 グループのメモリプロテクションユニット（以下、MPU）の設定例として、メモリ保護領域の設定方法と領域サーチ機能、アクセス例外の発生条件およびアクセス例外発生時の例外処理について説明します。

### 動作確認デバイス

RX62N、RX621 グループ

本アプリケーションノートは、動作確認デバイス RX62N、RX621 グループを基に作成しています。

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 .....	2
2. 動作確認条件 .....	3
3. 周辺機能説明 .....	4
4. ハードウェア説明 .....	7
5. ソフトウェア説明 .....	9
6. サンプルコード .....	52
7. 参考ドキュメント .....	52

## 1. 仕様

本アプリケーションノートは MPU で設定した「読み出しのみ許可した領域」、「書き込みのみ許可した領域」、「実行のみ許可した領域」に対して、それぞれに読み出し、書き込み、または実行を行います。

各領域に対して許可されていないアクセスをすると、メモリプロテクションエラーが発生します。

また、領域サーチ機能を使用すると、アクセスの許可/禁止の状態を検出することができます。

本アプリケーションノートは SW を 3 つ、LED を 4 つ使用しています。

- SW1 を押下げる毎に各領域からアクセスする領域を選択します。
- SW2 を押下げる毎に領域サーチ機能の有効/無効を選択します。
- SW3 を押下げると SW1、SW2 で設定したアクセス方法でアクセスを実行します。

LED はプログラムの動作状態を示します。

- LED0 ~ LED1 はアクセスする領域を示します。
- LED2 は領域サーチ機能の有効/無効を示します。
- LED3 は実行結果を示します。

なお、本アプリケーションノートでは読み出し、書き込み、実行のアクセスを条件付コンパイルで切り替えています。詳細は「5. ソフトウェア説明」を参照してください。

表 1.1 に使用する周辺機能と用途を示します。図 1.1 に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
MPU	メモリ保護
コンペアマッチタイマチャンネル 0 (以下、CMT0)	システムタイマ (SW のチャタリング除去、LED 点滅タイマ)

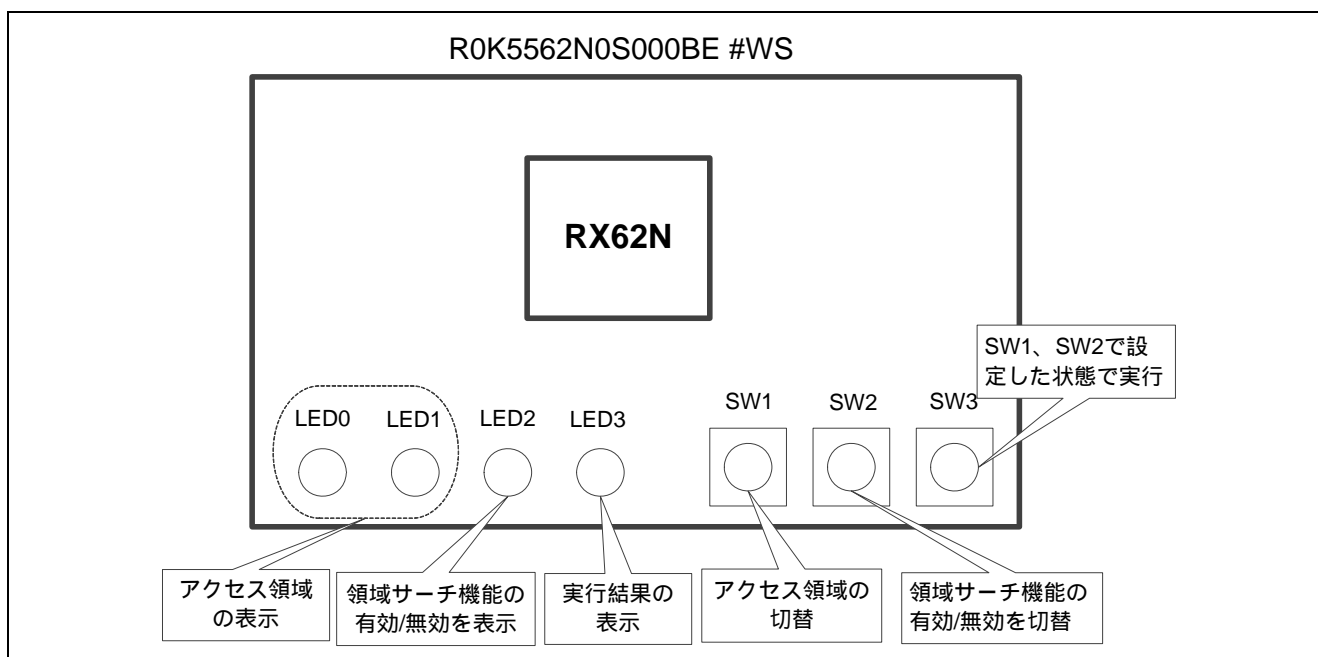


図 1.1 動作概要

## 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	RX62N グループ R5F562N8BDBG
動作周波数	<ul style="list-style-type: none"><li>メインクロック : 12.0MHz</li><li>ICLK : 96MHz</li><li>PCLK : 48MHz</li><li>BCLK : 24MHz</li></ul>
動作電圧	5.0V ( CPU 動作電圧は 3.3V )
統合開発環境	High-performance Embedded Workshop Version 4.09.00.007
Toolchain	RX Standard Toolchain(V1.2.0.0)
	RX Family C/C++ Compiler Driver V.1.02.00.000
	RX Family C/C++ Compiler V.1.02.00.000
	RX Family Assembler V.1.02.00.000
	Optimizing Linkage Editor V.10.02.00.000
	RX Family C/C++ Standard Library Generator V1.02.00.000
iodefine.h のバージョン	V2.0
エンディアン	リトルエンディアン
動作モード	シングルチップモード
サンプルコードのバージョン	Version 1.00
使用ボード	Renesas Starter Kit+ for RX62N ( R0K5562N0S000BE #WS )

### 3. 周辺機能説明

MPU について補足します。基本的な内容はユーザーズマニュアルハードウェア編に記載しています。

RX62N、RX621 グループには MPU が内蔵されており、全アドレス空間 (0000 0000h - FFFF FFFFh) を対象に CPU によるアクセスのアドレスチェックをします。最大 8 つの領域およびバックグラウンド領域に対して、領域毎のアクセス制御情報を設定することができます。各領域のアクセス制御情報は、読み出し許可、書き込み許可、実行許可をサポートしています。

MPU で設定した各領域のアクセス制御情報は CPU のプロセッサモードがユーザモードの時に有効です。スーパーバイザモードのときは、メモリ保護を行いません。アクセス制御情報が有効時に、領域に対する CPU のアクセス制御情報違反 (メモリプロテクションエラー) を検出すると、アクセス例外処理が開始されます。また、領域サーチ機能を使用し、アクセスする領域のアクセス制御情報を調べることで、CPU はアクセス制御情報違反を発生させずに許可されたアクセスのみに限定することができます。

- 【注】
1. MPU は CPU のメモリアクセスのみを監視しているため、DMAC、DTC によるメモリアクセスについては対応していません。
  2. 割り込み処理はスーパーバイザモードで実行されるため、MPU のアクセス制御情報は無効になります。
  3. MPU のアクセス制御情報を有効にする場合は、ユーザモードにする必要があります。
  4. バックグラウンド領域はメモリ保護領域 0~7 で設定された領域を除く全アドレス空間を指します。
  5. メモリプロテクションユニット関連レジスタへは、スーパーバイザモードでのみアクセスすることができます。

なお、ユーザモードからスーパーバイザモードの遷移は、「5.7.17 ユーザモードからスーパーバイザモードに変更」を参照ください。

スーパーバイザモードからユーザモードの遷移は、「5.7.16 スーパーバイザモードからユーザモードに変更」を参照ください。

### 3.1 MPU によるメモリ保護

本アプリケーションノートでは MPU を使用することにより、メモリの領域別にアクセス保護を設定します。図 3.1 にセクション配置とメモリ保護領域設定、表 3.1 にメモリ領域別のアクセス制御情報を示します。

なお、内蔵 RAM 領域に配置している監視領域 A、監視領域 B の実行アクセスを実現するため、PSAMPLE1 プログラムを PRead\_Area に転送し、PSAMPLE2 プログラムを PWrite\_Area に転送しています。

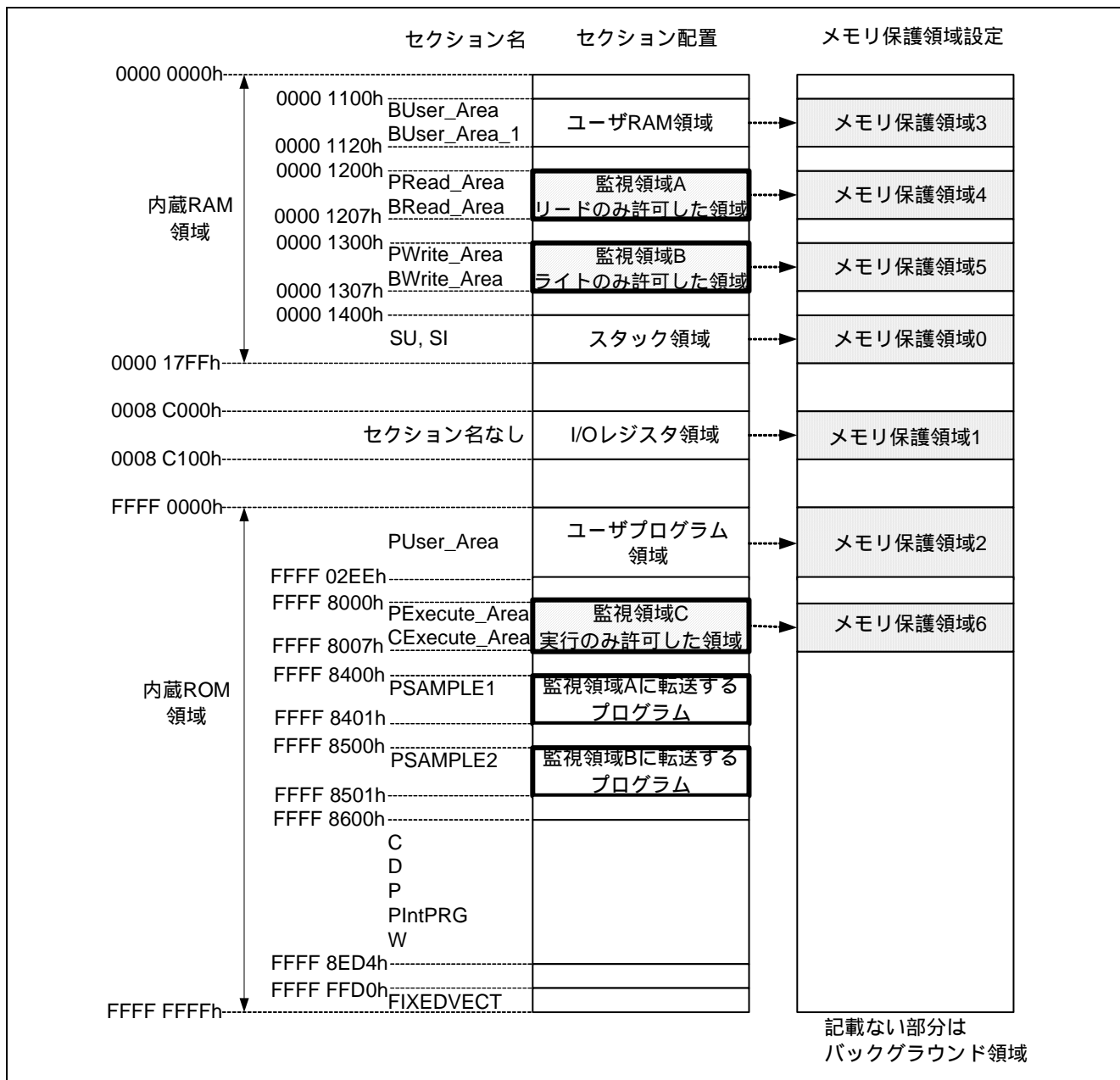


図3.1 セクション配置とメモリ保護領域設定

表3.1 メモリ領域別のアクセス制御情報

領域名 および説明	対応レジスタ	設定アドレス	アクセス制御情報		
			読み出し	書き込み	実行
本サンプルコードで 設定したスタック領域	RSPAGE0 (領域 0 開始)	SU 開始アドレス			
	REPAGE0 (領域 0 終了)	SI 終了アドレス			×
本サンプルコードで 設定した I/O レジスタ領域	RSPAGE1 (領域 1 開始)	0008 C000h			
	REPAGE1 (領域 1 終了)	0008 C100h			×
本サンプルコードで 設定したプログラム領域	RSPAGE2 (領域 2 開始)	PUser_Area 開始アドレス			
	REPAGE2 (領域 2 終了)	PUser_Area 終了アドレス	×	×	
本サンプルコードで 設定した RAM 使用領域	RSPAGE3 (領域 3 開始)	BUser_Area 開始アドレス			
	REPAGE3 (領域 3 終了)	BUser_Area_1 終了アドレス			×
監視領域 A 読み出しのみ 許可した領域	RSPAGE4 (領域 4 開始)	PRead_Area 開始アドレス			
	REPAGE4 (領域 4 終了)	BRead_Area 終了アドレス		×	×
監視領域 B 書き込みのみ 許可した領域	RSPAGE5 (領域 5 開始)	PWrite_Area 開始アドレス			
	REPAGE5 (領域 5 終了)	BWrite_Area 終了アドレス	×		×
監視領域 C 実行のみ許可した領域	RSPAGE6 (領域 6 開始)	PExecute_Area 開始アドレス			
	REPAGE6 (領域 6 終了)	CExecute_Area 終了アドレス	×	×	
未使用	RSPAGE7 (領域 7 開始)	-	-	-	-
	REPAGE7 (領域 7 終了)	-	-	-	-
バックグラウンド領域 *	MPBAC	上記領域を 除く全て	×	×	×

【注】 は許可、×は禁止を表します。

\* MPU のメモリプロテクション機能が有効でないスーパーバイザモードでアクセスする領域はアクセス情報が無効です

## 4. ハードウェア説明

### 4.1 ハードウェア構成例

本アプリケーションノートでは、SW1～SW3を使用してアクセス方法、アクセス実行を操作します。LEDを使用してプログラムの動作状態を表示しています。

図 4.1に本アプリケーションノートの接続例を示します。

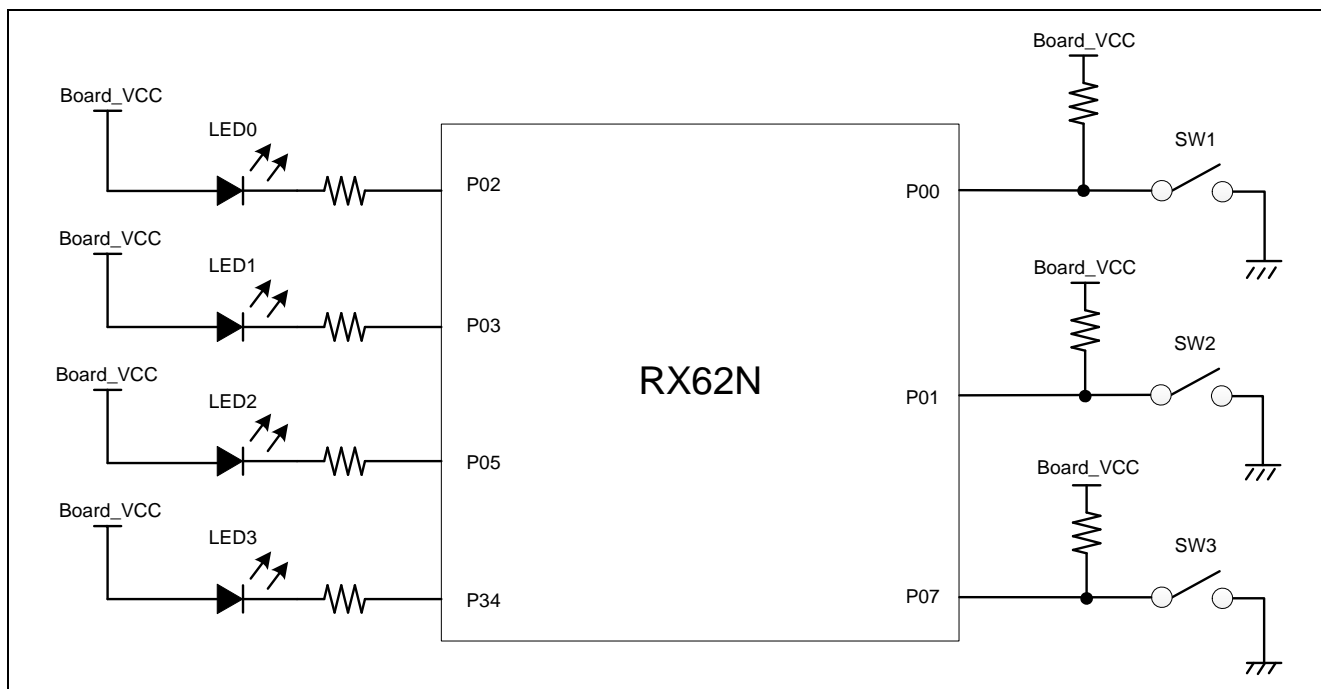


図4.1 接続例

## 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
P02	出力	LED0 制御
P03	出力	LED1 制御
P05	出力	LED2 制御
P34	出力	LED3 制御
P00	入力	SW1
P01	入力	SW2
P07	入力	SW3



## 5. ソフトウェア説明

本サンプルコードでは、指定の領域に対して CPU によるアクセスを読み出しに固定しています。CPU によるアクセスを変更する場合は、main.h ヘッドファイルに記述されている PRG\_TYPE の定義を変更する必要があります。図 5.1 に main.h ヘッドファイルの PRG\_TYPE 定義箇所を示します。図 5.1 に示す PRG\_TYPE 定義を変更し、サンプルコードをリコンパイルすることで、CPU によるアクセスを「読み出し」、「書き込み」、または「実行」の 3 つの動作から選択することができます。表 5.1 に PRG\_TYPE の定義とサンプルコード動作の対応を示します。

< main.h ファイルから抜粋 >

```
#define READ_ACCESS    (0x00000008)
#define WRITE_ACCESS   (0x00000004)
#define EXECUTE_ACCESS (0x00000002)

#define PRG_TYPE      READ_ACCESS
```

図5.1 サンプルコード切り替え箇所（読み出しアクセス選択時の例）

表5.1 PRG\_TYPE の定義とサンプルコード動作の対応

定義	サンプルコード動作
READ_ACCESS	指定の領域へ読み出しアクセスに固定
WRITE_ACCESS	指定の領域へ書き込みアクセスに固定
EXECUTE_ACCESS	指定の領域へ実行アクセスに固定

以下にソフトウェア説明を示します。条件付きコンパイルにより、指定領域に対して CPU によるアクセスを読み出し、書き込み、実行動作に切り替えた場合の差分は、「5.7.9 指定されたアクセス領域にアクセスする関数」のみとなります。「5.7.9 指定されたアクセス領域にアクセスする関数」の箇所は条件付きコンパイルの各条件でフローチャートを記載しています。

5.1 動作概要

サンプルコードは、表 3.1に示すメモリ領域別のアクセス、図 3.1に示すセクション配置とメモリ保護領域設定をしています。図 5.2にシーケンス図を示します。プログラムを動作させるとアクセス方法設定中となります。初期設定はアクセス領域を監視領域 A、領域サーチ機能有効です。

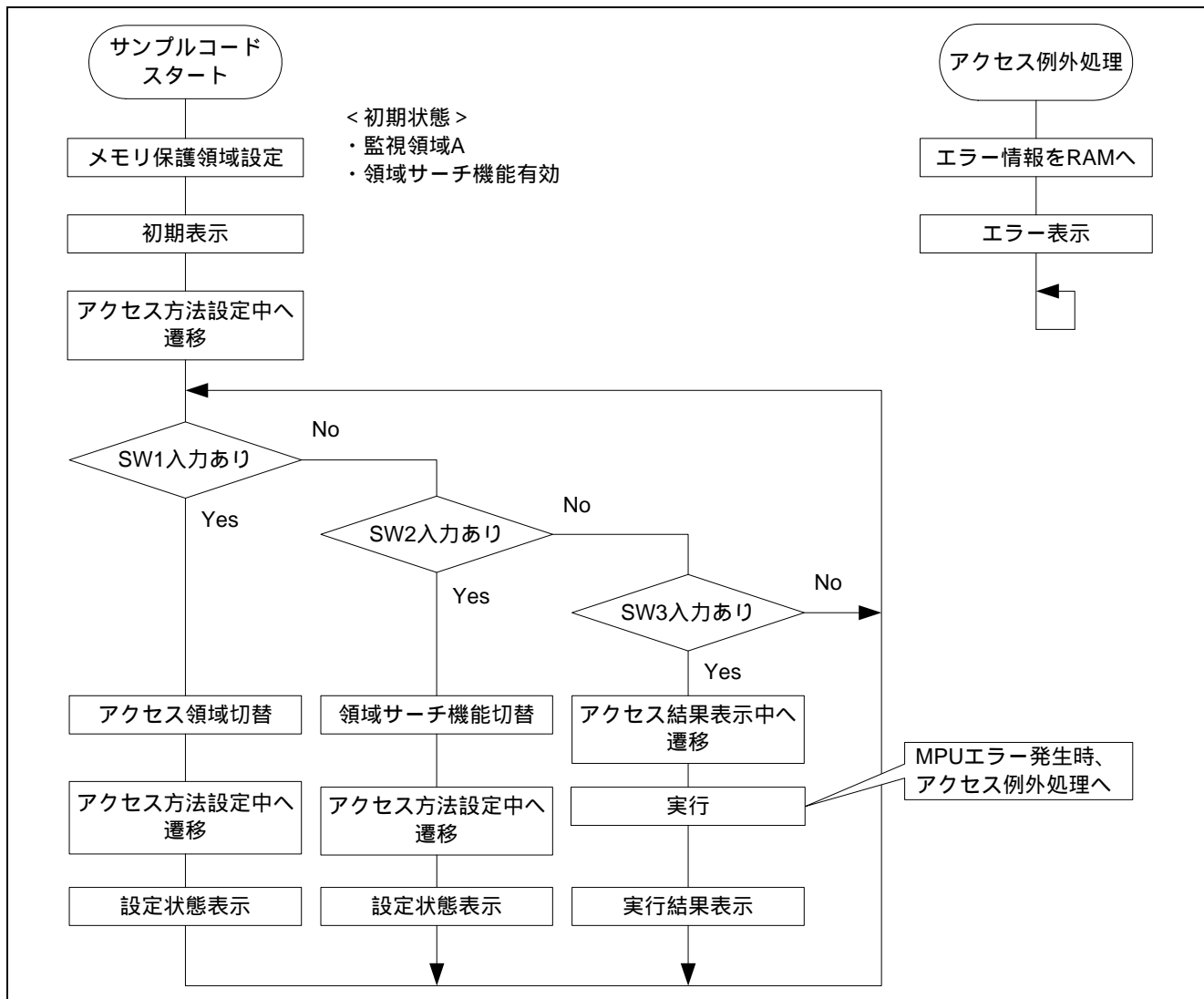


図5.2 シーケンス図

## 5.1.1 アクセス方法設定中

## (1) SW1 押下げ動作

SW1 を押下げる毎にアクセスする領域を監視領域 A (メモリ保護領域 4) 監視領域 B (メモリ保護領域 5) 監視領域 C (メモリ保護領域 6) 監視領域 A (メモリ保護領域 4) と切り替えることができます。図 5.3 に SW1 押下げ動作を示します。

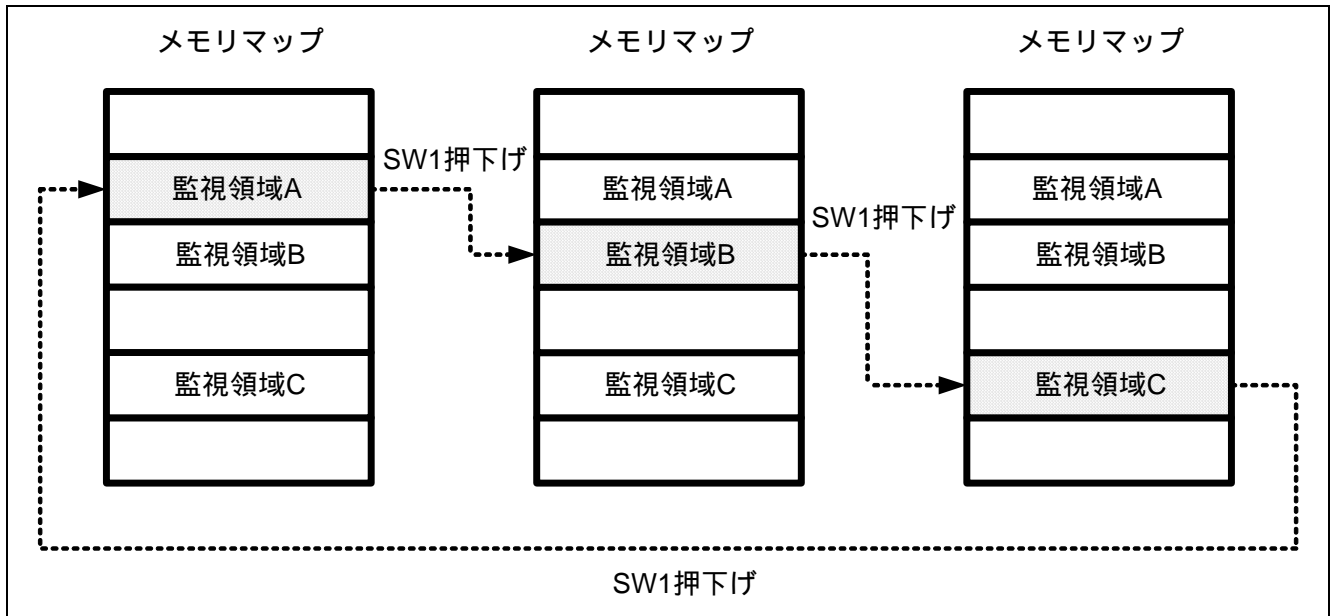


図5.3 SW1 押下げ動作

## (2) SW2 押下げ動作

SW2 を押下げる毎に領域サーチ機能の有効 / 無効を切り替えることができます。領域サーチ機能が有効の場合は、予めアクセスするアドレスに対して、領域サーチ機能を使用します。アクセス制御情報を調べることで、アクセス制御違反を回避し、許可されているアクセスのみに限定します。図 5.4 に SW2 押下げ動作を示します。

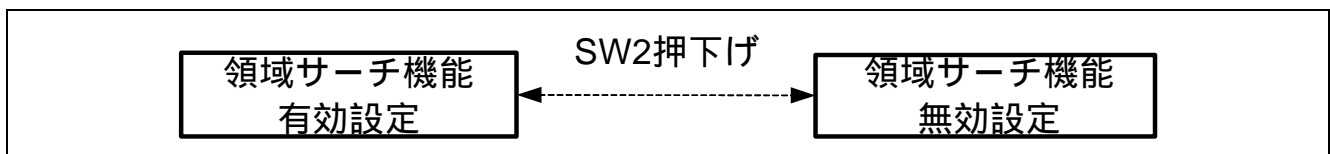


図5.4 SW2 押下げ動作

(3) プログラムのステータス表示

図 5.5にアクセス方法設定中におけるプログラムのステータス表示を示します。

アクセス領域		領域サーチ機能		アクセス結果		アクセス領域	領域サーチ機能	アクセス結果	
LED0	LED1	LED2	LED3	LED0	LED1				
●	●	●	●	監視領域A	有効	-	点灯 ○		
●	●	●	●		無効	-			
●	●	●	●	監視領域B	有効	-	消灯 ●		
●	●	●	●		無効	-			
●	●	●	●	監視領域C	有効	-	点滅 ◐		
●	●	●	●		無効	-			

図5.5 アクセス方法設定中におけるプログラムのステータス表示

## 5.1.2 アクセス結果表示中

## (1) SW3 押下げ動作

アクセス方法設定中に SW3 を押下げるとアクセス結果表示中となり、設定されたアクセス方法でプログラムが実行され、アクセス結果が表示されます。図 5.6 に SW3 押下げ動作を示します。

メモリアクセスマニピュレーションエラーが発生していない場合、アクセス結果表示中に SW1 または SW2 を押下げると、SW1 を押下げればアクセスする領域を切り替え、SW2 を押下げれば領域サーチ機能の有効 / 無効を切り替え後、再度アクセス方法設定中となります。

また、メモリアクセスマニピュレーションエラーが発生していない場合、アクセス結果表示中に SW3 を押下げることにより、再度同様のアクセス方法でプログラムが実行され、アクセス結果が表示されます。

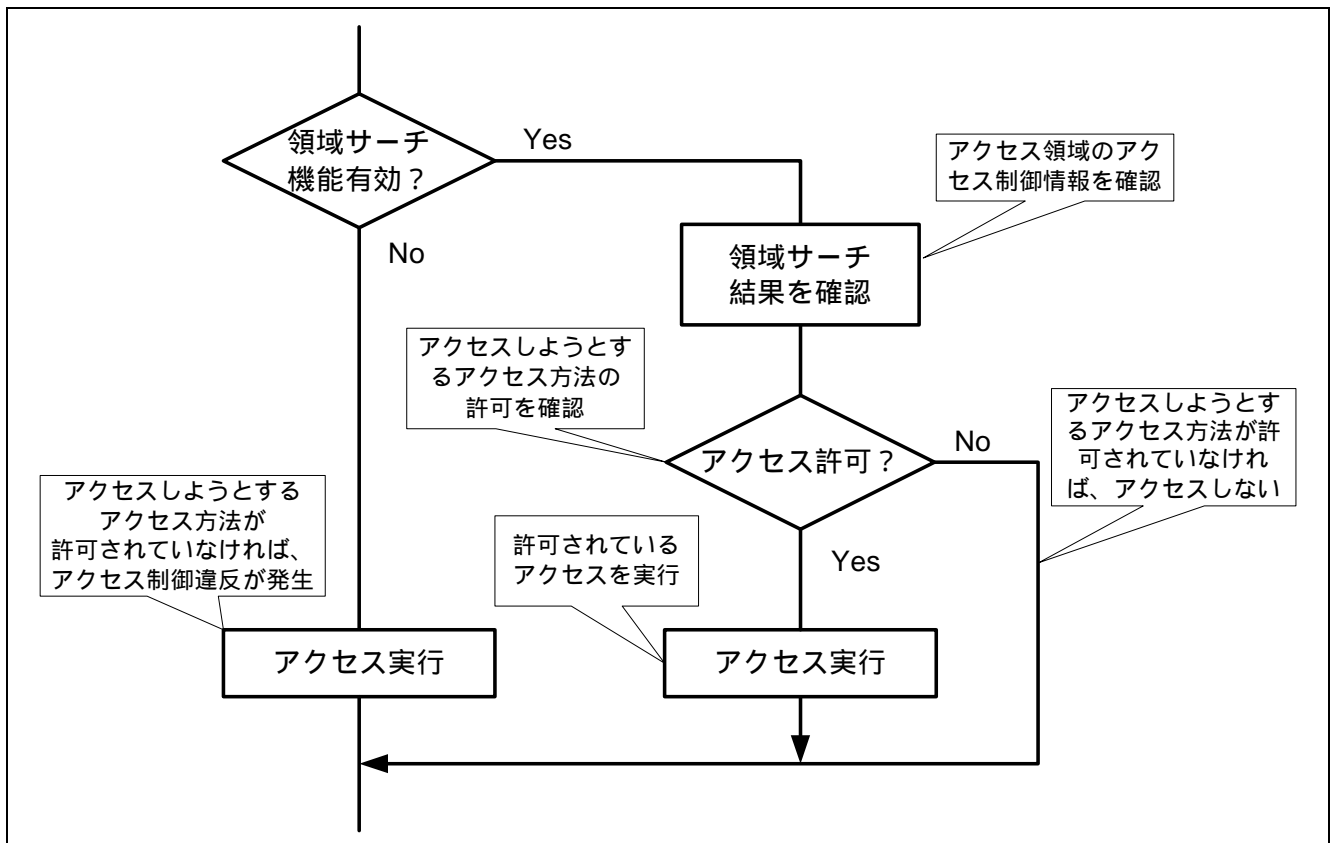


図5.6 SW3 押下げ動作

(2) プログラムのステータス表示

図 5.7にアクセス結果表示中におけるプログラムのステータス表示を示します。

アクセス領域		領域サーチ機能	アクセス結果	アクセス領域	領域サーチ機能	アクセス結果	
LED0	LED1	LED2	LED3				
○	○	○	●	監視領域A	有効	アクセス禁止を検出し、回避	点灯 ○ 消灯 ● 点滅 ●
○	○	○	●			アクセス許可で実行	
○	○	●	○		無効	エラー発生	
○	○	●	●			エラーなし	
○	●	○	●	監視領域B	有効	アクセス禁止を検出し、回避	
○	●	○	●			アクセス許可で実行	
○	●	●	○		無効	エラー発生	
○	●	●	●			エラーなし	
●	○	○	●	監視領域C	有効	アクセス禁止を検出し、回避	
●	○	○	●			アクセス許可で実行	
●	○	●	○		無効	エラー発生	
●	○	●	●			エラーなし	

図5.7 アクセス結果表示中におけるプログラムのステータス表示

## 5.2 ファイル構成

表 5.2 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のうち、内容を変更していないファイルは除きます (typedefine.h の変更による内容変更も除きます)。

表5.2 ファイル構成

ファイル名	概要	備考
main.h	定義ファイル	
resetprg.c	リセット後の初期設定処理	
main.c	メイン処理	
initialize.c	各種初期設定	
vect.h	割り込み関数宣言	Excep_Memory_ProtectionError 関数宣言を追加
vecttbl.c	固定ベクタ割り込みベクタテーブル	アクセス例外処理に Excep_Memory_ProtectionError 関数を追加
inprg.c	ベクタ関数の定義 <ul style="list-style-type: none"> <li>無条件トラップ例外処理</li> <li>CMT0 コンペアマッチ例外処理を使用。</li> </ul>	統合開発環境で自動生成されるファイルで各種割り込み処理が記述されています。本アプリケーションノートでは、本ファイルの Excep_Memory_ProtectionError 関数を削除し、Excep_BRK 関数、Excep_CMTU0_CMT0 関数を変更しています。
Excep_Memory_ProtectionError.src	アクセス例外処理	メモリプロテクションエラーが発生した命令のアドレスをリード
MPU_Error.c	メモリプロテクションエラー発生時のエラー情報の格納およびエラー表示	
stacksct.h	スタックサイズの設定	
iodefine_mpu.h	メモリ保護機能関連レジスタ定義ファイル	iodefine.h のバージョン 2.0 に MPU 関連レジスタの記載がないため追加

## 5.3 定数一覧

表 5.3、表 5.4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 1

定数名	設定値	内容
READ_ACCESS	00000008h	読み出しアクセス用サンプルコード
WRITE_ACCESS	00000004h	書き込みアクセス用サンプルコード
EXECUTE_ACCESS	00000002h	実行アクセス用サンプルコード
PRG_TYPE	READ_ACCESS	読み出しアクセス用のサンプルコードを選択 (表 5.1を参照)
SETUP_STATUS	0	サンプルコードがアクセス方法設定中の状態
RESULT_STATUS	1	サンプルコードがアクセス結果表示中の状態
SET	1	1 にセット
CLEAR	0	ゼロクリア
NO_INPUT	0	SW 入力なし
PUSH_DOWN_SW1	01h	SW1 押下げ
PUSH_DOWN_SW2	02h	SW2 押下げ
PUSH_DOWN_SW3	80h	SW3 押下げ
SW_VALID	1	SW 押しの有効を示す
SW_CHAT_TIMES	3	SW 入力信号を有効とする比較回数
SW_FIX	0	SW 入力の確定判定
ERROR_STATUS_CLEAR	01h	メモリプロテクションエラーステータスレジスタのクリア
NO_ERROR	0	メモリプロテクションエラー未発生
MPU_ERROR_MASK	00000007h	メモリプロテクションエラーステータス用マスク
READ_ERROR	00000002h	オペランドアクセスメモリ プロテクションエラー(リード)が発生
WRITE_ERROR	00000006h	オペランドアクセスメモリ プロテクションエラー(ライト)が発生
EXECUTE_ERROR	00000001h	命令メモリプロテクションエラーが発生
RWE_MASK	0000000Eh	読み出し、書き込み、実行許可/禁止用マスク
READ_MASK	00000008h	読み出し許可/禁止用マスク
WRITE_MASK	00000004h	書き込み許可/禁止用マスク
EXECUTE_MASK	00000002h	実行許可/禁止用マスク
ALL_ENABLE	0000000Eh	読み出し、書き込み、実行許可
READ_ENABLE	00000008h	読み出し許可
WRITE_ENABLE	00000004h	書き込み許可
EXECUTE_ENABLE	00000002h	実行許可
DISABLE	0	禁止
AREA_SEARCH_ENABLE	0001h	領域サーチ機能有効
AREA_A	0	監視領域 A
AREA_B	AREA_A + 1	監視領域 B
AREA_C	AREA_B + 1	監視領域 C



表5.4 サンプルコードで使用する定数 2

定数名	設定値	内容
LED0_MASK	04h	LED0 制御用マスク
LED1_MASK	08h	LED1 制御用マスク
LED2_MASK	20h	LED2 制御用マスク
LED3_MASK	10h	LED3 制御用マスク
SET_BLINK_CNT	20	LED 点滅周期設定
BLINK_TIMING	0	LED 点滅タイミング判定
MPU_REG_MASK	FFFFFFF0h	MPU レジスタ書き込み用マスク
SCOPE_ICLK	00000000h	I クロック倍率を ×8 に指定
SCOPE_PCLK	00000100h	P クロック倍率を ×4 に指定
SCOPE_BCLK	00020000h	B クロック倍率を ×2 に指定
LED3_ACTIVE	10h	LED3 該当ビットを出力に設定
SET_PORT0_DDR	2Ch	LED0,1,2 用にポート出力設定、 SW1,2,3 用にポートを入力設定
TURN_OFF_LED_0_1_2	2Ch	LED0,1,2 消灯
TURN_OFF_LED_3	10h	LED3 消灯
ENABLE_INPUT_BUF_SW1	01h	SW1 用入力制御バッファを有効
ENABLE_INPUT_BUF_SW2	02h	SW2 用入力制御バッファを有効
ENABLE_INPUT_BUF_SW3	80h	SW3 用入力制御バッファを有効
ACCESS_CONTROL_INF0	0Dh	メモリ保護領域 0 アクセス制御情報
ACCESS_CONTROL_INF1	0Dh	メモリ保護領域 1 アクセス制御情報
ACCESS_CONTROL_INF2	03h	メモリ保護領域 2 アクセス制御情報
ACCESS_CONTROL_INF3	0Dh	メモリ保護領域 3 アクセス制御情報
ACCESS_CONTROL_INF4	09h	メモリ保護領域 4 アクセス制御情報
ACCESS_CONTROL_INF5	05h	メモリ保護領域 5 アクセス制御情報
ACCESS_CONTROL_INF6	03h	メモリ保護領域 6 アクセス制御情報
IO_DEFINE_STARTADDRESS	0008C000h	ユーザモード時に使用する I/O レジスタの 開始アドレス
IO_DEFINE_ENDADDRESS	0008C100h	ユーザモード時に使用する I/O レジスタの 終了アドレス
BACKGROUND_ACCESS_CONTROL	0000h	バックグラウンド領域アクセス制御情報
MPU_ENABLE	0001h	MPU 有効
E_OK	0	MPU レジスタ正常書き込み
WRITE_FAULT	1	MPU レジスタ書き込み失敗
SYSTEM_TIME	3A97h	システムタイマの周期設定 ( 10ms )

## 5.4 変数一覧

表 5.5にグローバル変数を、表 5.6にstatic 型変数を、表 5.7にconst 型変数を示します。

表5.5 グローバル変数

型	変数名	内容	使用関数
uint32_t	g_rd_isp	ユーザモードからスーパーバイザモードに変更する際のPSW退避データを格納	Change_PSW_PM_to_SuperVisorMode、 Excep_BRK
uint32_t	g_mpu_err_status	メモリプロテクションエラーステータスを格納 (NO_ERROR: エラーなし、 READ_ERROR: リードエラー、 WRITE_ERROR: ライトエラー、 EXECUTE_ERROR: 実行エラー)	chip_init、 output_resultLED、 MPU_Error
uint32_t	g_mpu_err_instruction_adrs	メモリプロテクションエラーが発生した命令のアドレスを格納	chip_init、 Excep_Memory_Protection Error
uint32_t	g_mpu_err_operand_adrs	メモリプロテクションエラーが発生したオペランドアクセスアドレスを格納	chip_init、MPU_Error
uint32_t	g_mpu_err_area	メモリプロテクションエラーが発生した領域および領域のアクセス制御情報を格納	chip_init、MPU_Error
uint32_t	g_mpu_hit_area	領域サーチ機能を使用したアドレスでサーチヒットした領域および領域のアクセス制御情報を格納	chip_init、 Memory_Protection、 output_resultLED、 Area_Access
uint8_t	g_access_area	アクセスする領域データを格納 (AREA_A: 監視領域 A、 AREA_B: 監視領域 B、 AREA_C: 監視領域 C)	chip_init、 Memory_Protection、 output_setupLED、 Area_Access_Select
uint8_t	g_area_search_flg	領域サーチ機能の有効/無効を格納 (0: 無効、1: 有効)	chip_init、 Memory_Protection、 output_setupLED、 output_resultLED、 Area_Access
uint8_t	g_system_tmr_flg	システムタイマ周期フラグを格納 (0: 周期経過待ち、1: 周期経過)	chip_init、main、 Excep_CMTU0_CMT0

表5.6 static 型変数

型	変数名	内容	使用関数
static uint8_t	g_sw_status	確定した SW 入力データを格納	main、get_sw、Memory_Protection
static uint8_t	g_last_sw	前回の SW 入力データを格納	main、get_sw
static uint8_t	g_sw_chat_cnt	SW のチャタリング除去用カウンタ	main、get_sw
static uint8_t	g_sw_update	SW 入力データの確定更新を格納 (0:更新なし、1:更新あり)	main、get_sw、Memory_Protection、control_LED
static uint8_t	g_led_cycle_cnt	LED 点滅用カウンタ	main、control_LED
static uint8_t	g_prg_status	プログラムのステータスを格納	main、Memory_Protection、control_LED
uint32_t	g_Area_A_operand	監視領域 A のオペランド	areaAccess_Select
uint32_t	g_Area_B_operand	監視領域 B のオペランド	areaAccess_Select

表5.7 const 型変数

型	変数名	内容	使用関数
uint32_t	g_Area_C_operand	監視領域 C のオペランド	areaAccess_Select

## 5.5 関数一覧

表 5.8に関数を示します。

表5.8 関数

関数名	説明
PowerON_Reset_PC	スタートアッププログラム
Main	メイン処理
get_sw	SW 入力検出
Memory_Protection	アクセス領域とアクセス方法の設定およびアクセス制御
control_LED	SW 入力の有効による LED 表示の更新および LED 点滅更新
output_setupLED	アクセス方法設定の LED 制御
output_resultLED	アクセス結果表示の LED 制御
areaAccess_Select	指定されたアクセス領域の判定
Area_Access	指定されたアクセス領域にアクセスする関数
areaSearch	領域サーチ機能で指定のアドレスに対応する領域と領域のアクセス制御情報を取得
operand_Access_Read	指定されたオペランドを読み出しする関数
operand_Access_Write	指定されたオペランドに書き込みする関数
Area_A_prg	監視領域 A を実行する関数
Area_B_prg	監視領域 B を実行する関数
Area_C_prg	監視領域 C を実行する関数
Change_PSW_PM_to_UserMode	スーパーバイザモードからユーザモードに変更
Change_PSW_PM_to_SuperVisorMode	ユーザモードからスーパーバイザモードに変更
init	初期設定関数の呼び出し
chip_init	MCU 初期設定、動作クロックの設定、使用 RAM の初期化
mpu_init	MPU の初期化
verify_mpu_reg	MPU 関連レジスタのベリファイ
cmt0_init	CMT0 の設定
Excep_BRK	無条件トラップの例外関数
Excep_Memory_ProtectionError	メモリプロテクションエラー発生時のアクセス例外関数
MPU_Error	メモリプロテクションエラー情報の格納およびエラー表示
Excep_CMTU0_CMT0	CMT0 コンペアマッチ例外関数

## 5.6 関数仕様

サンプルコードの関数仕様を示します。

PowerON_Reset_PC	
概要	スタートアッププログラム
ヘッダ	なし
宣言	void PowerON_Reset_PC(void)
説明	PowerON_Reset_PC 関数はリセット後に呼び出されます。組み込み関数や標準ライブラリ関数を使用して CPU レジスタの設定を行います。その後、init 関数、main 関数を呼び出します。
引数	なし
リターン値	なし
main	
概要	メイン処理
ヘッダ	main.h
宣言	void main(void)
説明	Change_PSW_PM_to_UserMode 関数、get_sw 関数、Memory_Protection 関数、control_LED 関数を呼び出します。
引数	なし
リターン値	なし
get_sw	
概要	SW 入力検出
ヘッダ	main.h
宣言	static uint8_t get_sw(uint8_t *p_last_sw, uint8_t *p_sw_chat_cnt, uint8_t *p_sw_status)
説明	各 SW の入力検出をします。
引数	uint8_t *p_last_sw : 前回の SW 入力データ uint8_t *p_sw_chat_cnt : SW 入力信号の一致回数カウンタ uint8_t *p_sw_status : 確定 SW 入力データ
リターン値	SW 入力の有効 / 無効を返す CLEAR : SW 入力が無効 SW_VALID : SW 入力がある
Memory_Protection	
概要	アクセス領域とアクセス方法の設定およびアクセス制御
ヘッダ	main.h
宣言	static void Memory_Protection( uint8_t sw_status, uint8_t sw_update, uint8_t *p_prg_status )
説明	引数によって示される SW 入力状態に応じて、アクセス領域の切り替え、領域サーチ機能の有効 / 無効切り替え、指定された領域へアクセスする関数を呼び出します。
引数	uint8_t sw_status : 確定 SW 入力データ uint8_t sw_update : SW 入力の有効 / 無効 uint8_t *p_prg_status : プログラムのステータス
リターン値	なし

control_LED	
概要	SW 入力の有効による LED 表示の更新および LED 点滅更新
ヘッダ	main.h
宣言	static void control_LED(uint8_t sw_update, uint8_t prg_status, uint8_t *p_led_cycle_cnt)
説明	引数によって示されるプログラムのステータスを LED に表示します。また、LED の点滅制御をします。
引数	uint8_t sw_update : SW 入力の有効 / 無効 uint8_t prg_status : プログラムのステータス uint8_t *p_led_cycle_cnt : LED 点滅制御カウンタ
リターン値	なし
output_setupLED	
概要	アクセス方法設定の LED 制御
ヘッダ	main.h
宣言	void output_setupLED(void)
説明	アクセス方法設定の LED 制御をします。
引数	なし
リターン値	なし
output_resultLED	
概要	アクセス結果表示の LED 制御
ヘッダ	main.h
宣言	void output_resultLED (uint8_t prg_status, uint8_t *p_led_cycle_cnt)
説明	アクセス結果表示の LED 制御をします。
引数	uint8_t prg_status : プログラムのステータス uint8_t *p_led_cycle_cnt : LED 点滅制御カウンタ
リターン値	なし
areaAccess_Select	
概要	指定されたアクセス領域の判定
ヘッダ	main.h
宣言	static void areaAccess_Select (void)
説明	アクセス領域を判定し、Area_Access 関数を呼び出します。
引数	なし
リターン値	なし
areaAccess	
概要	指定されたアクセス領域にアクセスする関数
ヘッダ	main.h
宣言	static void Area_Access(uint32_t *p_access_operand, void (*prg)())
説明	引数に応じたアクセス領域にアクセスする関数を呼び出します。
引数	uint32_t *p_access_operand : アクセスするオペランドアドレス void (*prg)() : 実行アクセスする先頭アドレス
リターン値	なし

---

areaSearch	
概要	領域サーチ機能で指定のアドレスに対応する領域と領域のアクセス制御情報を取得
ヘッダ	なし
宣言	static uint32_t areaSearch(uint32_t *p_access_address)
説明	Change_PSW_PM_to_SuperVisorMode 関数を呼び出します。引数のアドレスに対応する領域と領域のアクセス制御情報を取得します。Change_PSW_PM_to_UserMode を呼び出します。
引数	uint32_t *p_access_address : 領域サーチ機能の対象アドレス
リターン値	引数のアドレスに対応する領域と領域のアクセス制御情報

---

operand_Access_Read	
概要	指定されたオペランドを読み出しする関数
ヘッダ	なし
宣言	static void operand_Access_Read(uint32_t *p_access_operand)
説明	引数のアドレスに読み出しアクセスします。
引数	uint32_t *p_access_operand : 読み出しアクセスするアドレス
リターン値	なし

---

operand_Access_Write	
概要	指定されたオペランドに書き込みする関数
ヘッダ	なし
宣言	static void operand_Access_Write(uint32_t *p_access_operand)
説明	引数のアドレスに書き込みアクセスします。
引数	uint32_t *p_access_operand : 書き込みアクセスするアドレス
リターン値	なし

---

---

Area_A_prg	
概要	監視領域 A を実行する関数
ヘッダ	なし
宣言	static void Area_A_prg(void)
説明	監視領域 A を実行します。
引数	なし
リターン値	なし

---

Area_B_prg	
概要	監視領域 B を実行する関数
ヘッダ	なし
宣言	static void Area_B_prg(void)
説明	監視領域 B を実行します。
引数	なし
リターン値	なし

---

Area_C_prg	
概要	監視領域 C を実行する関数
ヘッダ	なし
宣言	static void Area_C_prg(void)
説明	監視領域 C を実行します。
引数	なし
リターン値	なし

---

Change_PSW_PM_to_UserMode	
概要	スーパーバイザモードからユーザモードに変更
ヘッダ	なし
宣言	void Change_PSW_PM_to_UserMode(void)
説明	プロセッサモードをスーパーバイザモードからユーザモードに切り替えます。
引数	なし
リターン値	なし

---

Change_PSW_PM_to_SuperVisorMode	
概要	ユーザモードからスーパーバイザモードに変更
ヘッダ	なし
宣言	static void Change_PSW_PM_to_SuperVisorMode(void)
説明	無条件トラップ例外処理を発生させます。無条件トラップ例外処理実行終了後、プロセッサモードがスーパーバイザモードで復帰します。
引数	なし
リターン値	なし

---



init	
概要	初期設定関数の呼び出し
ヘッダ	なし
宣言	void init(void)
説明	周辺機能の初期化やメモリ保護領域の設定を行う chip_init 関数、mpu_init 関数、cmt0_init 関数、output_setupLED 関数を呼び出します。
引数	なし
リターン値	なし
chip_init	
概要	MCU 初期設定、動作クロックの設定、使用 RAM の初期化
ヘッダ	main.h
宣言	static void chip_init(void)
説明	クロックの設定および LED、SW を使うための汎用ポート、使用 RAM の初期設定等を行います。
引数	なし
リターン値	なし
mpu_init	
概要	MPU の初期化
ヘッダ	iodefine_mpu.h
宣言	static void mpu_init(void)
説明	メモリ保護領域を設定し、MPU の機能を有効にします。
引数	なし
リターン値	なし
verify_mpu_reg	
概要	MPU 関連レジスタのベリファイ
ヘッダ	iodefine_mpu.h
宣言	static uint8_t verify_mpu_reg(volatile __evenaccess uint32_t *target, uint32_t expect_val)
説明	verify_mpu_reg 関数は MPU レジスタに書き込まれた値が正しいことを確認します。プログラムの動作に直接影響はしません。
引数	volatile __evenaccess uint32_t *p_target : ベリファイ対象のレジスタアドレス uint32_t expect_val : ベリファイするレジスタの期待値
リターン値	MPU レジスタの書き込み結果 E_OK : 正常書き込み WRITE_FAULT : 書き込み失敗

cmt0_init	
概要	CMT0 の設定
ヘッダ	なし
宣言	static void cmt0_init(void)
説明	システムタイマ周期を設定。
引数	なし
リターン値	なし
Excep_BRK	
概要	無条件トラップの例外関数
ヘッダ	vect.h
宣言	void Excep_BRK(void)
説明	無条件トラップの例外関数。例外処理実行終了後、スーパーバイザモードで復帰
引数	なし
リターン値	なし
Excep_Memory_ProtectionError	
概要	メモリプロテクションエラー発生時のアクセス例外関数
ヘッダ	vect.h
宣言	void Excep_Memory_ProtectionError(void)
説明	アクセス例外処理でメモリプロテクションエラーが発生した命令のアドレスをリードし、MPU_Error 関数を呼び出します。
引数	なし
リターン値	なし
MPU_Error	
概要	メモリプロテクションエラー情報の格納およびエラー表示
ヘッダ	vect.h
宣言	void MPU_Error (void)
説明	メモリプロテクションエラー情報の格納およびエラー表示後、本関数内で無限ループします。
引数	なし
リターン値	なし
Excep_CMTU0_CMT0	
概要	CMT0 コンペアマッチ例外関数
ヘッダ	vect.h
宣言	void Excep_CMTU0_CMT0 (void)
説明	CMT0 コンペアマッチ例外関数。システムタイマ経過フラグをセットします。
引数	なし
リターン値	なし

## 5.7 フローチャート

## 5.7.1 スタートアッププログラム

図 5.8にスタートアッププログラムのフローチャートを示します。

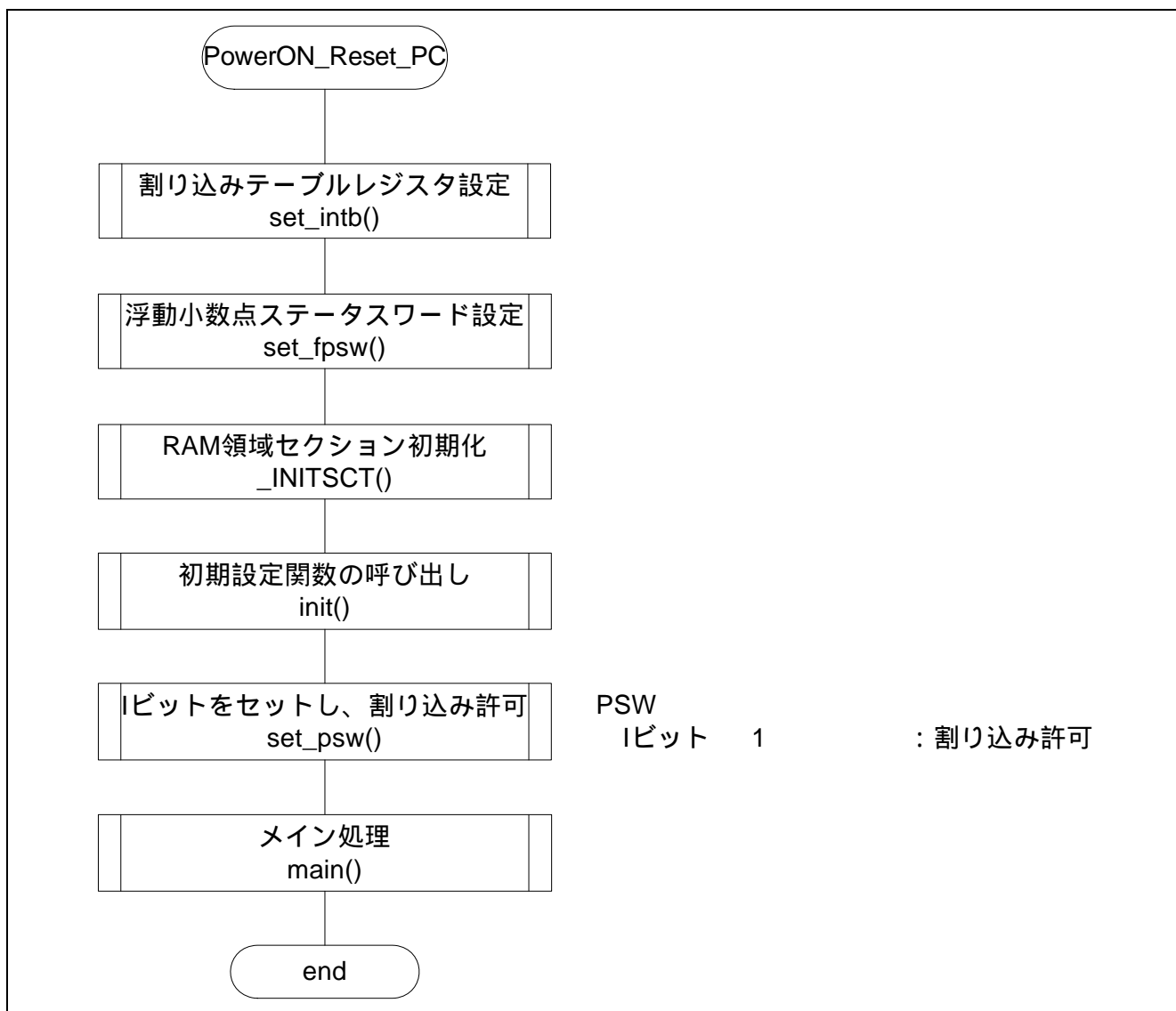


図5.8 スタートアッププログラム

## 5.7.2 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。

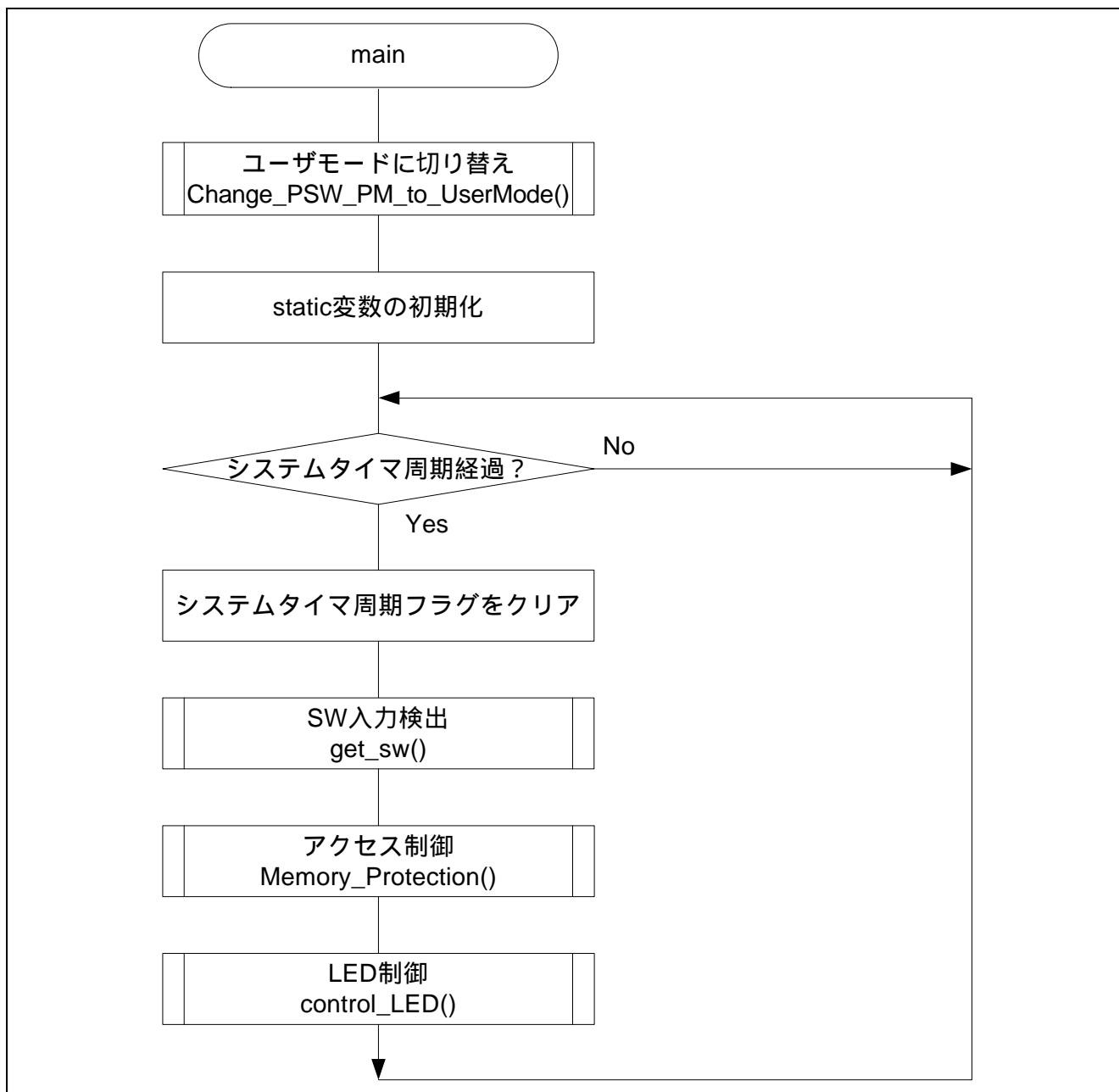


図5.9 メイン処理

## 5.7.3 SW 入力検出

図 5.10にSW 入力検出のフローチャートを示します。

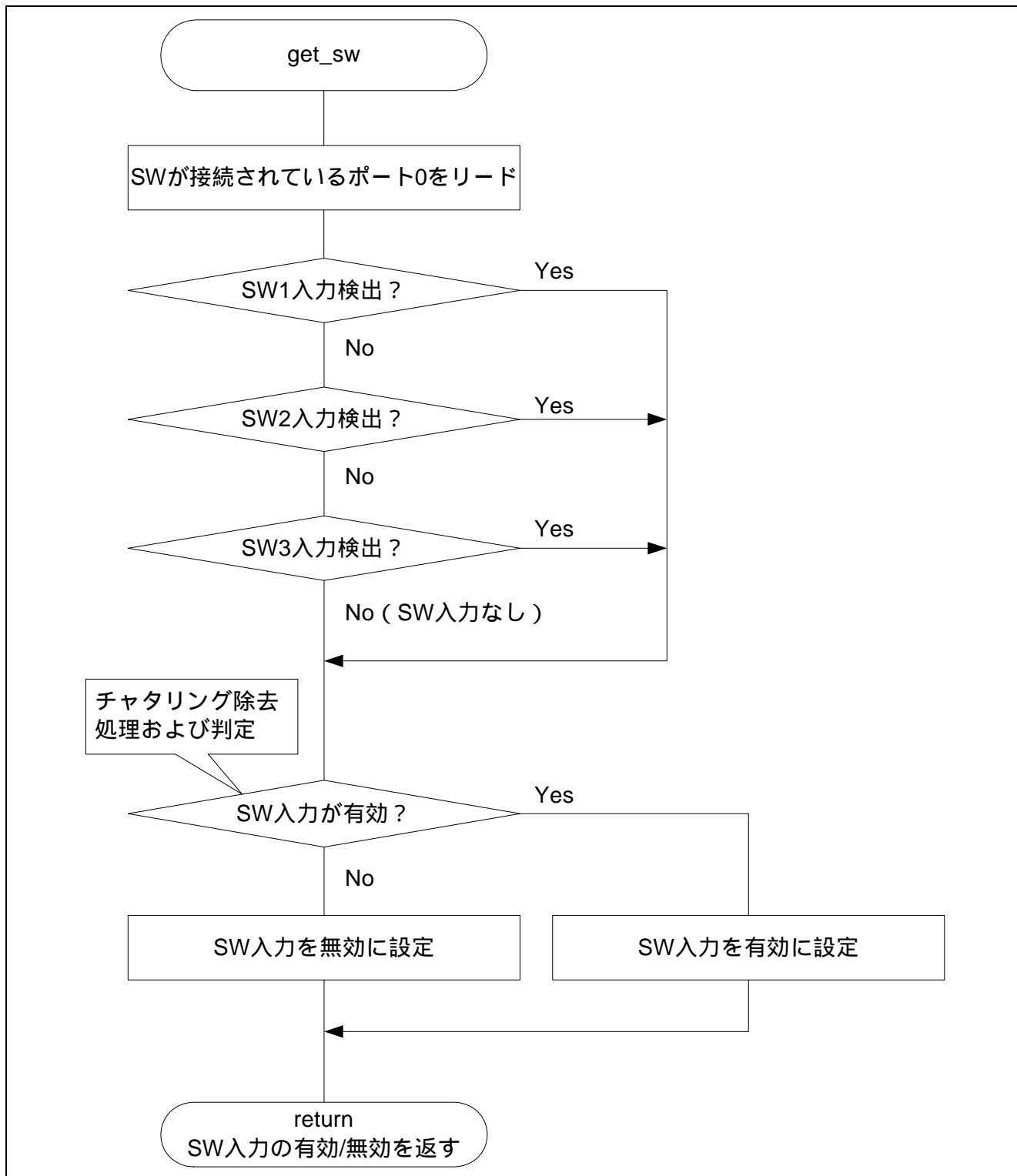


図5.10 SW 入力検出

5.7.4 アクセス領域とアクセス方法の設定およびアクセス制御

図 5.11にアクセス領域とアクセス方法の設定およびアクセス制御のフローチャートを示します。

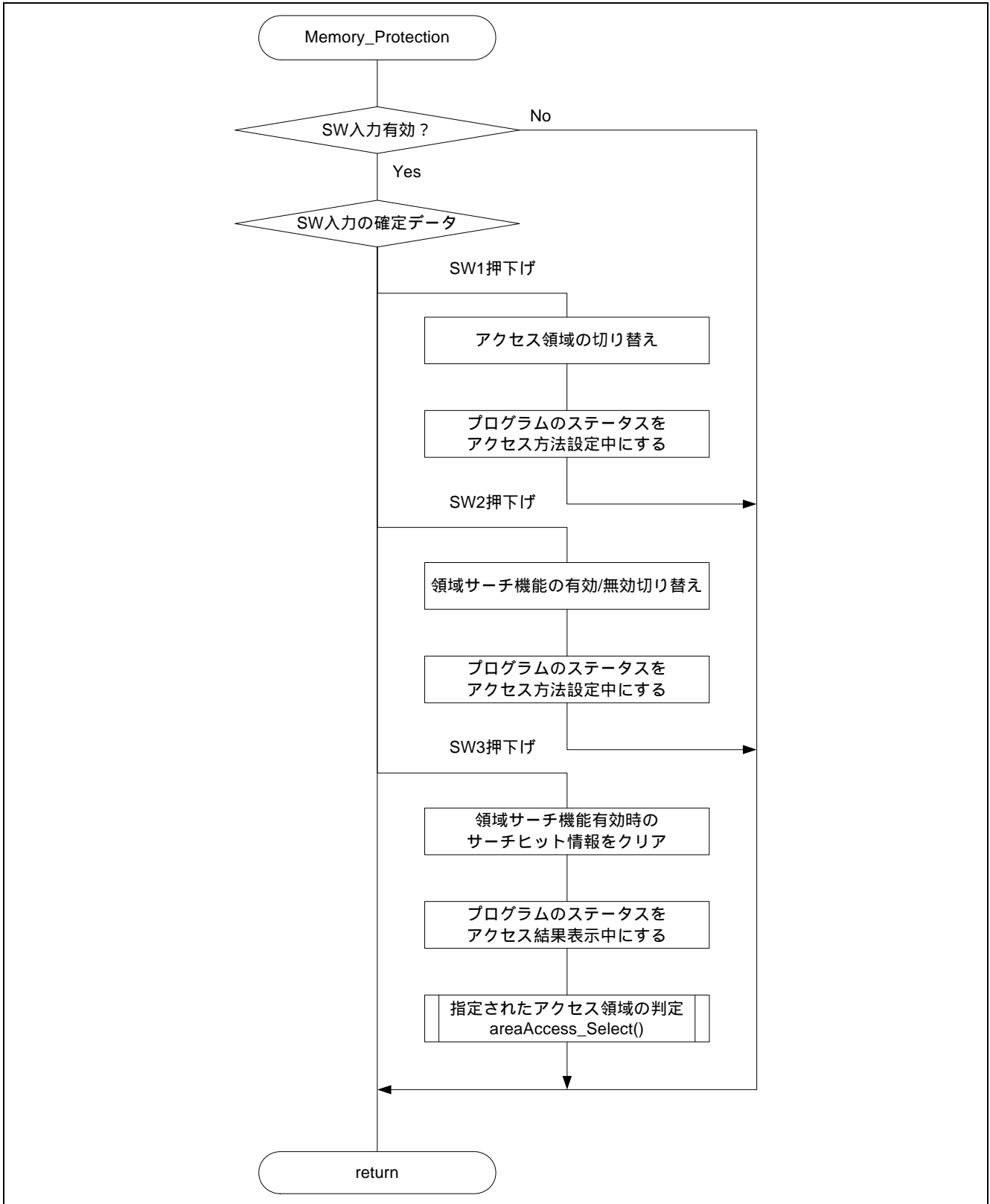


図5.11 アクセス領域とアクセス方法の設定およびアクセス制御

## 5.7.5 SW 入力の有効による LED 表示の更新および LED 点滅更新

図 5.12にSW 入力の有効による LED 表示の更新および LED 点滅更新のフローチャートを示します。

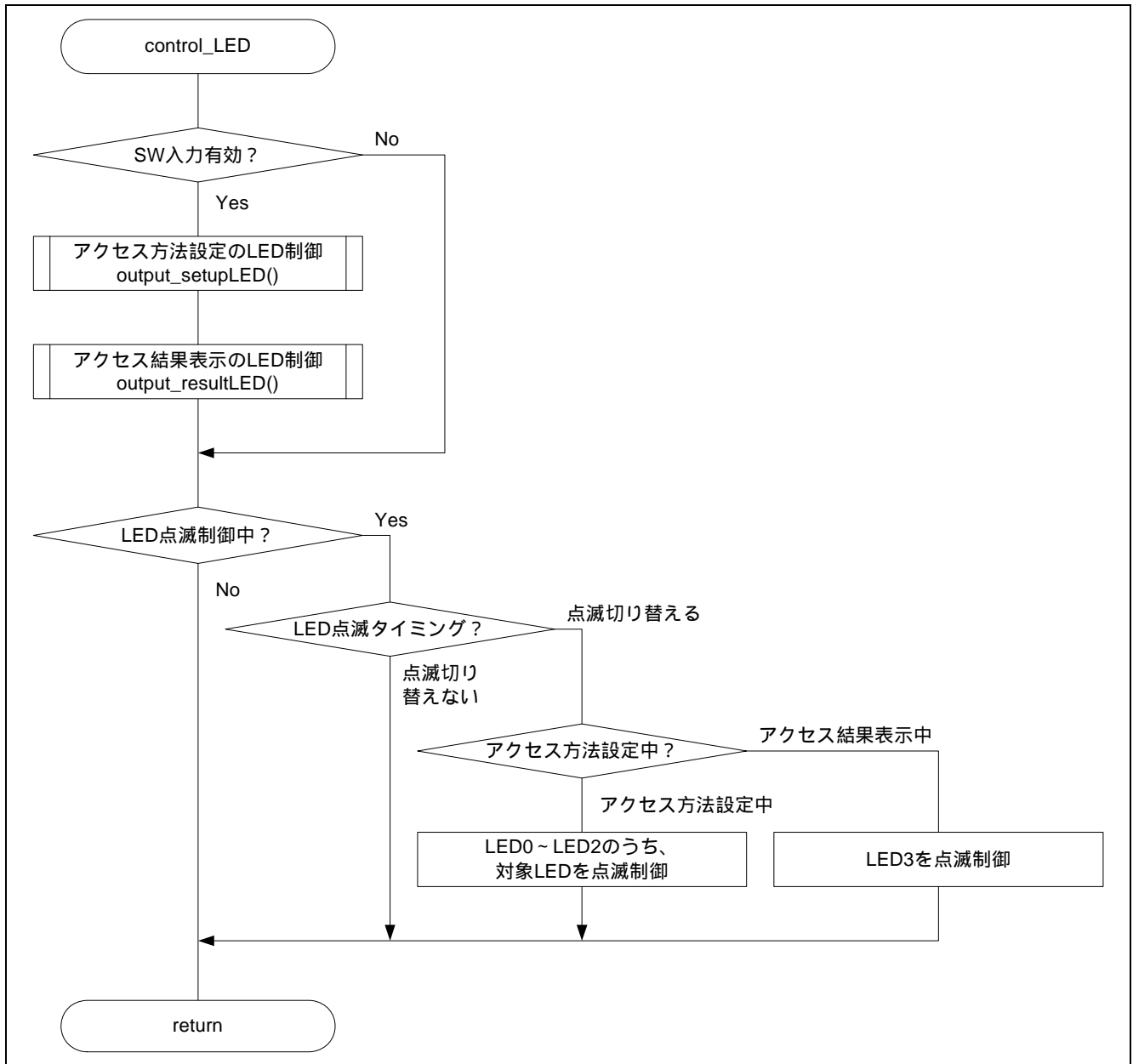


図5.12 SW 入力の有効による LED 表示の更新および LED 点滅更新

5.7.6 アクセス方法設定のLED制御

図 5.13にアクセス方法設定のLED制御のフローチャートを示します。

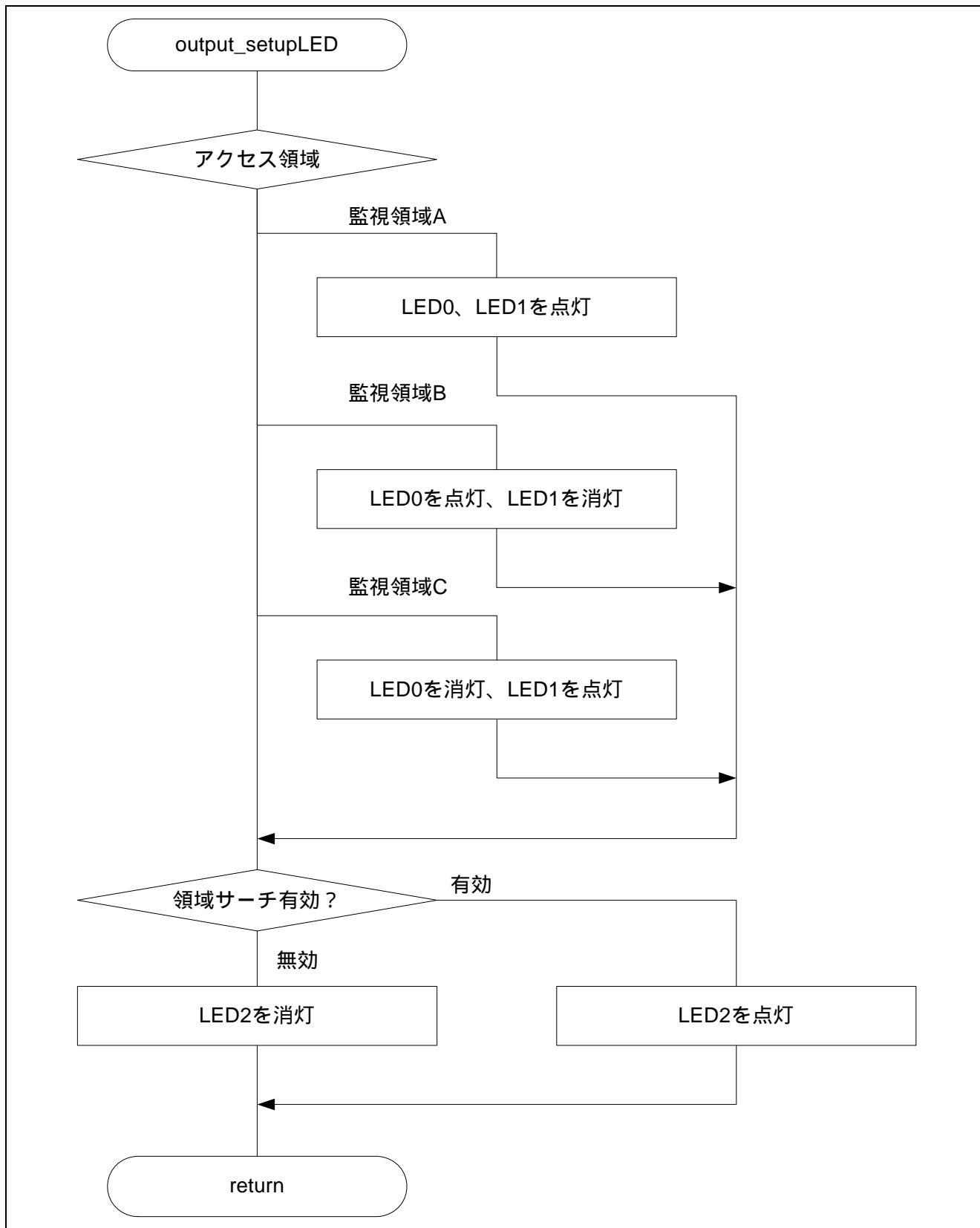


図5.13 アクセス方法設定のLED制御



5.7.7 アクセス結果表示のLED制御

図 5.14にアクセス結果表示のLED制御のフローチャートを示します。

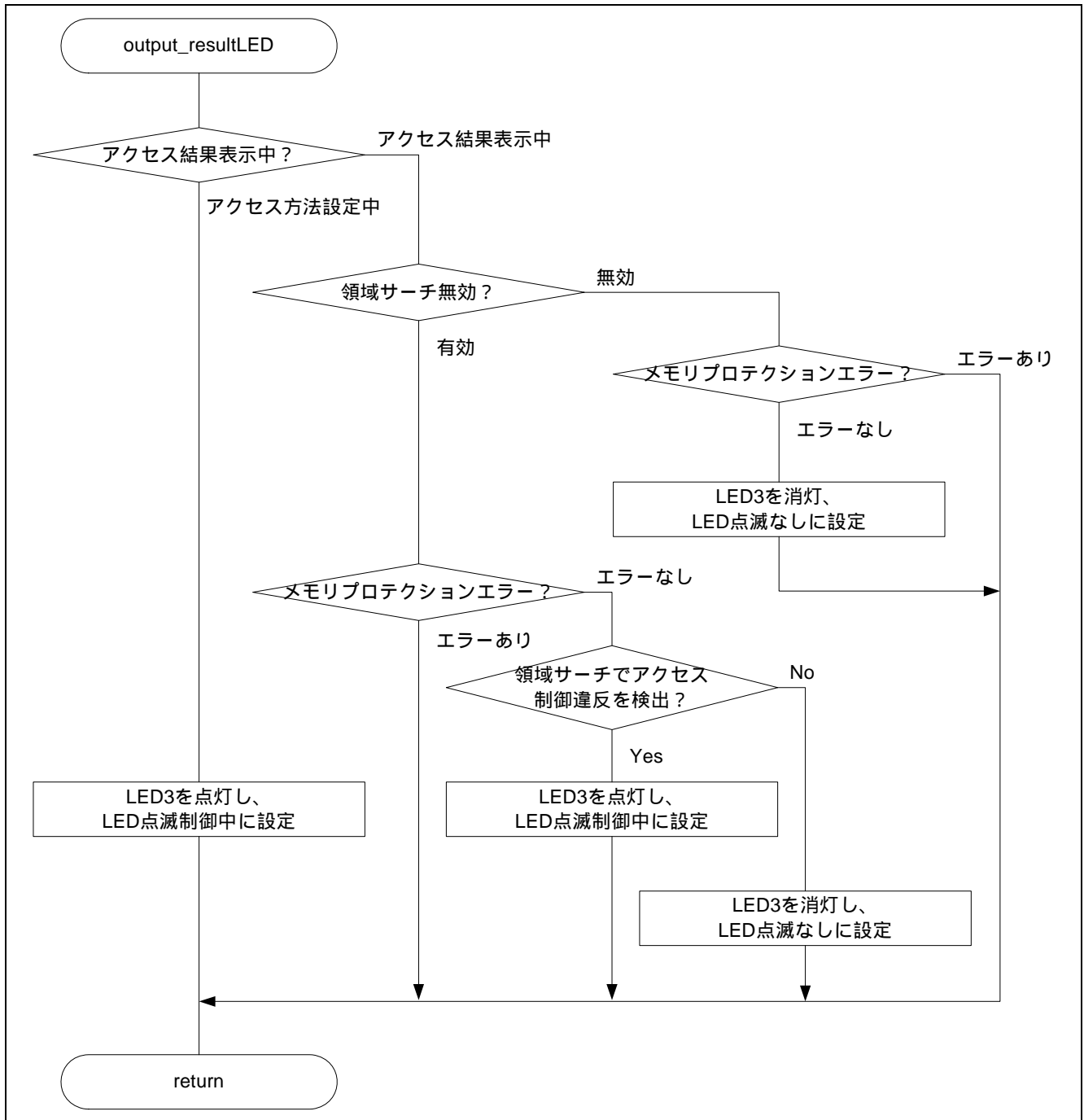


図5.14 アクセス結果表示のLED制御

5.7.8 指定されたアクセス領域の判定

図 5.15に指定されたアクセス領域の判定のフローチャートを示します。

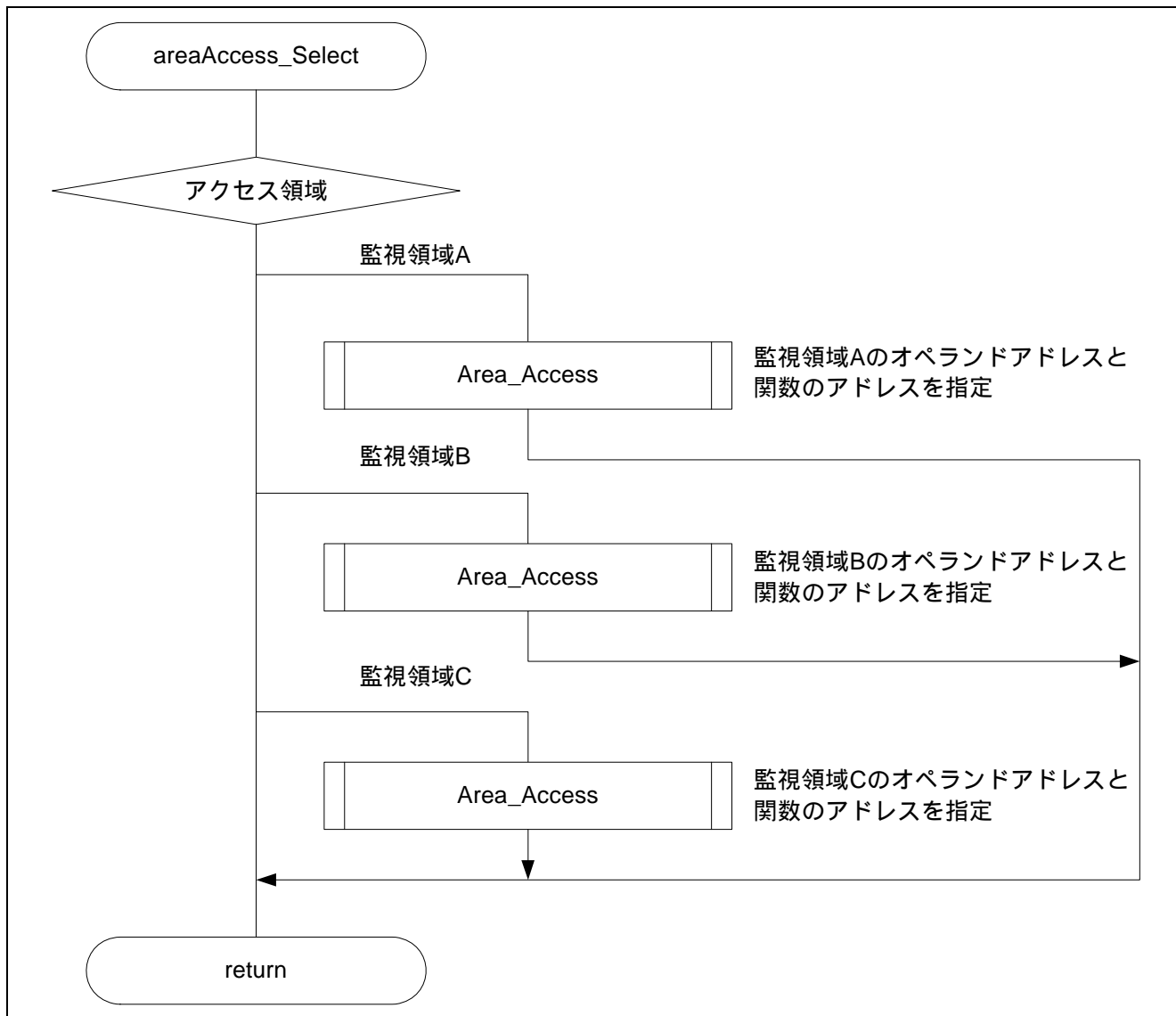


図5.15 指定されたアクセス領域の判定

## 5.7.9 指定されたアクセス領域にアクセスする関数

表 5.1のPRG\_TYPEの定義とサンプルコード動作の対応に示す条件付きコンパイルで動作が異なる関数です。条件付きコンパイルの各条件で説明します。

## (1) 指定領域へ読み出しアクセス

指定の領域へ読み出しアクセスに固定を選択した場合に動作する、図 5.16に指定されたアクセス領域にアクセスする関数 (READ\_ACCESS) のフローチャートを示します。

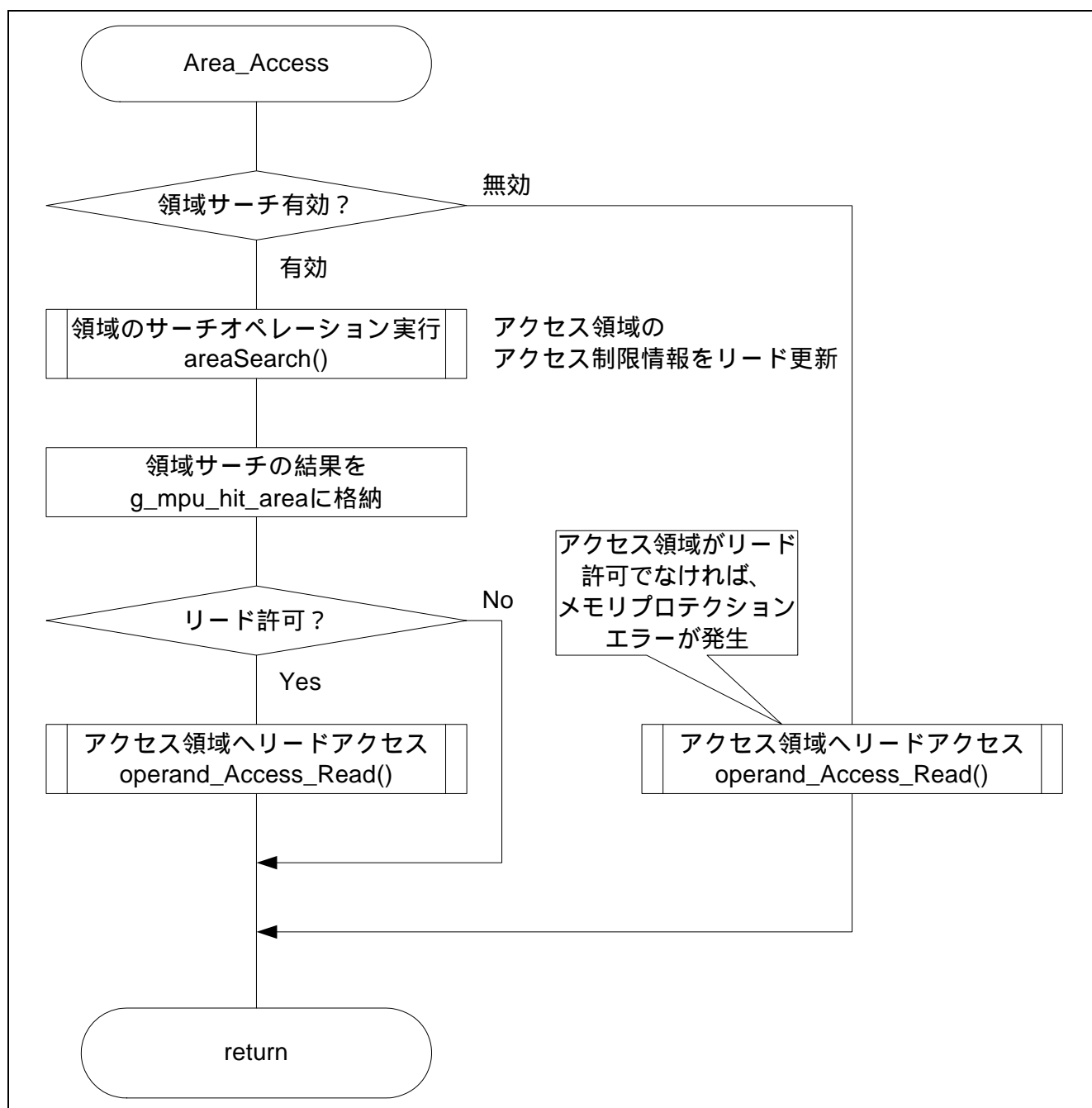


図5.16 指定されたアクセス領域にアクセスする関数 (READ\_ACCESS)

## (2) 指定領域へ書き込みアクセス

指定の領域へ書き込みアクセスに固定を選択した場合に動作する、図 5.17に指定されたアクセス領域にアクセスする関数 ( WRITE\_ACCESS ) のフローチャートを示します。

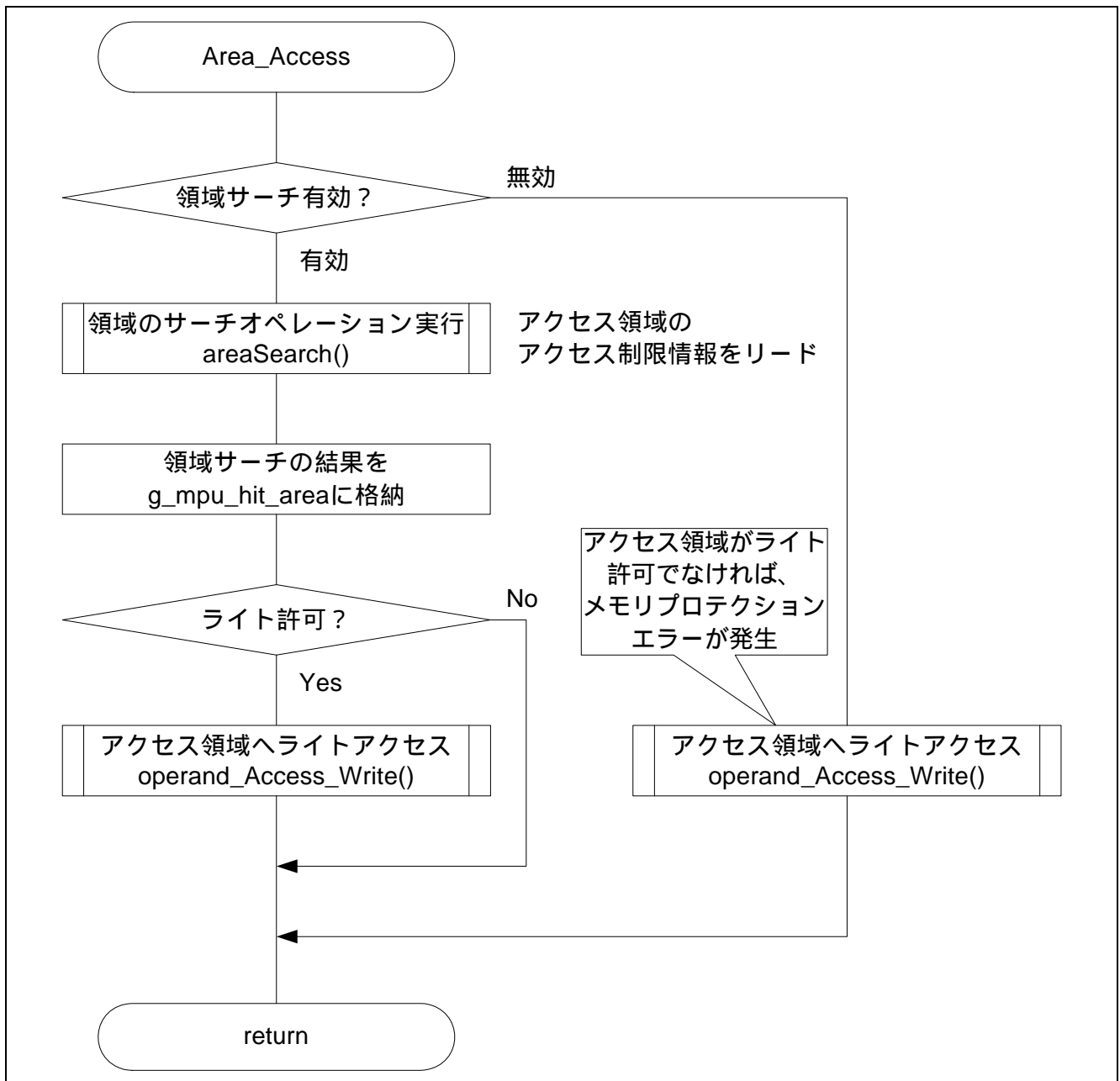


図5.17 指定されたアクセス領域にアクセスする関数 ( WRITE\_ACCESS )

## (3) 指定領域へ実行アクセス

指定の領域へ実行アクセスに固定を選択した場合に動作する、図 5.18に指定されたアクセス領域にアクセスする関数 ( EXECUTE\_ACCESS ) のフローチャートを示します。

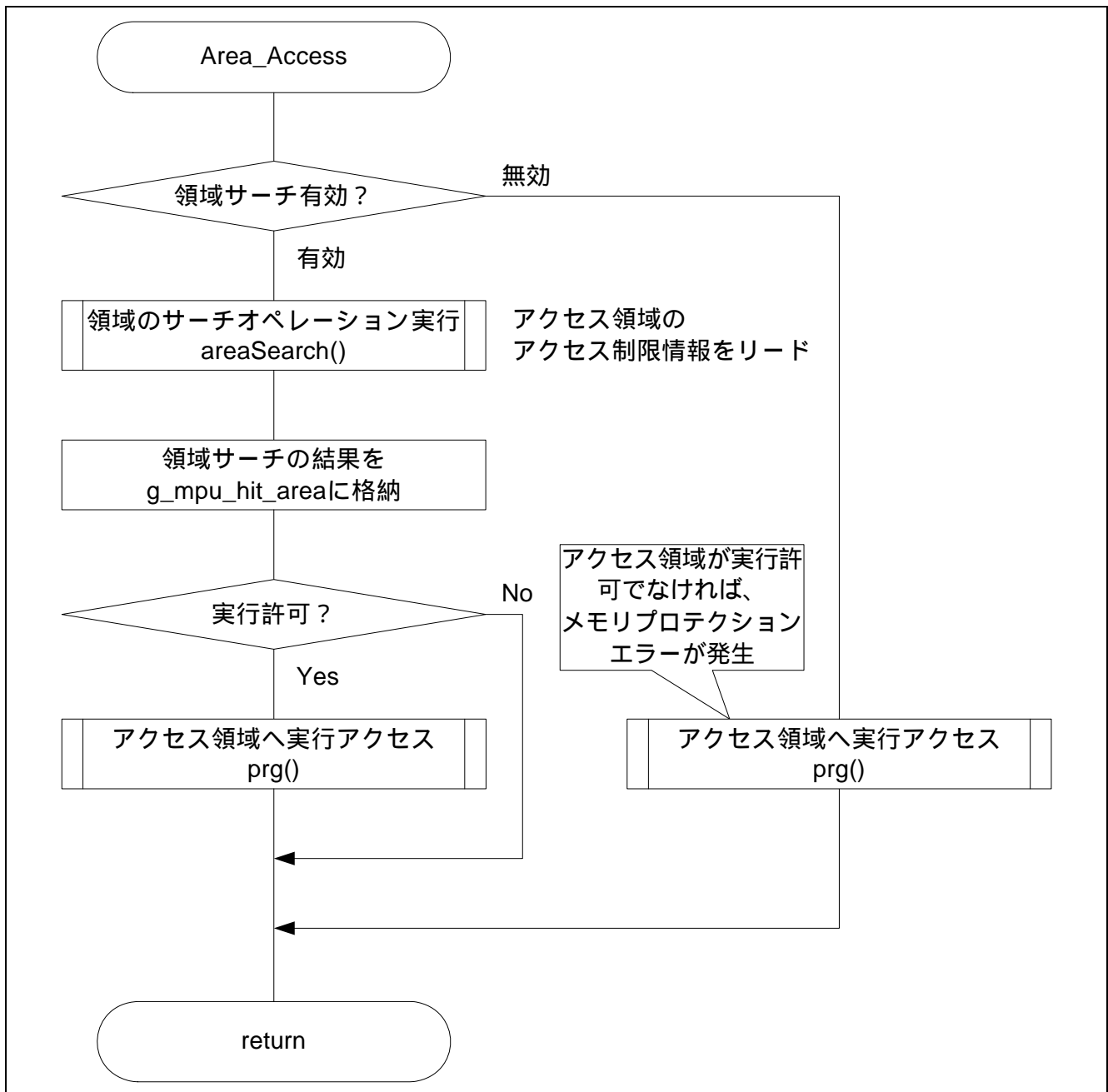


図5.18 指定されたアクセス領域にアクセスする関数 ( EXECUTE\_ACCESS )

## 5.7.10 領域サーチ機能で指定のアドレスに対応する領域と領域のアクセス制御情報を取得

図 5.19に領域サーチ機能で指定のアドレスに対応する領域と領域のアクセス制御情報を取得のフローチャートを示します。

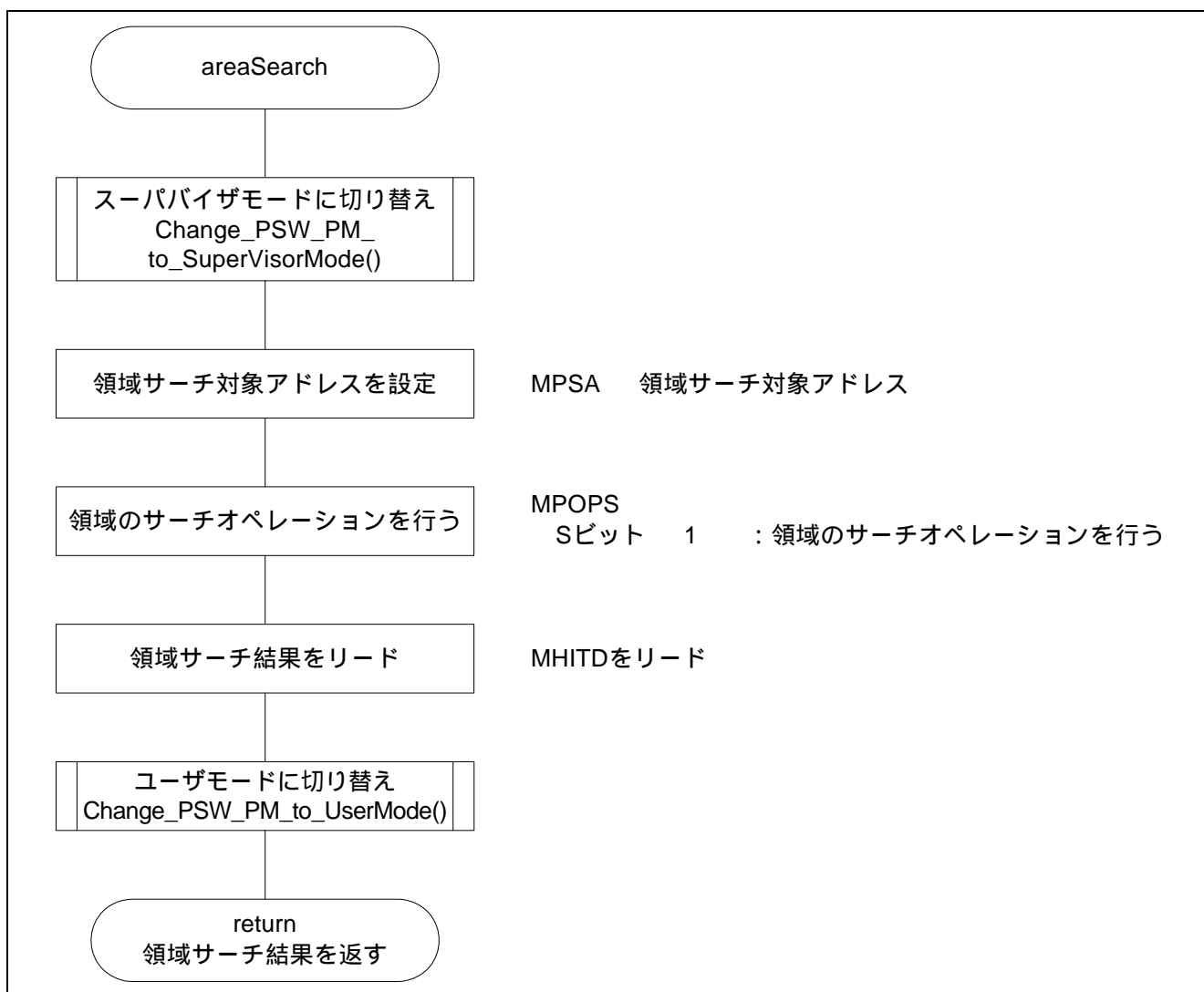


図5.19 領域サーチ機能で指定のアドレスに対応する領域と領域のアクセス制御情報を取得

## 5.7.11 指定されたオペランドを読み出しする関数

図 5.20に指定されたオペランドを読み出しする関数のフローチャートを示します。

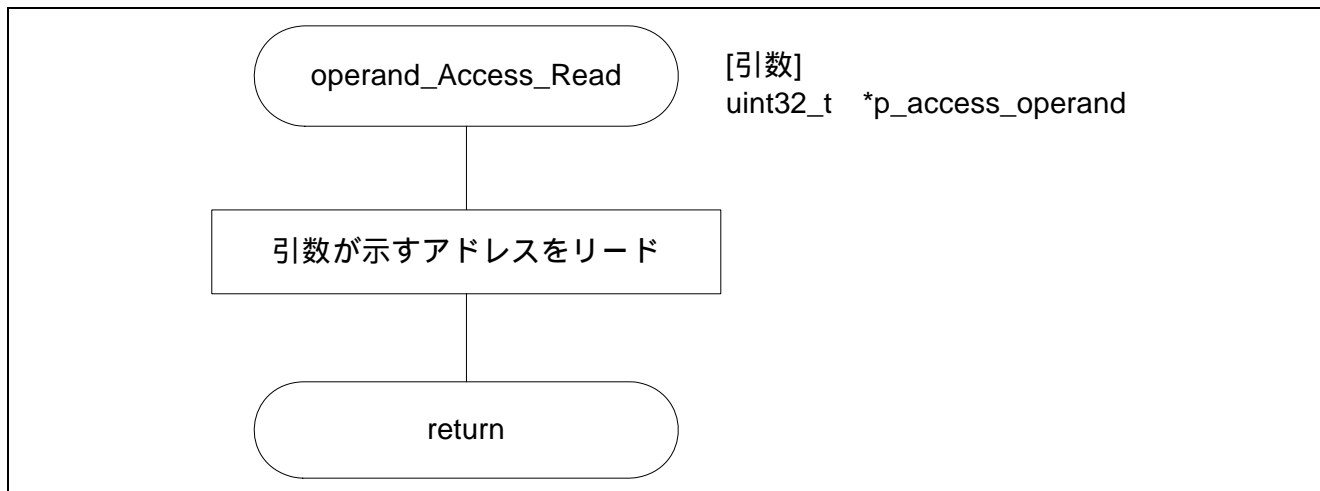


図5.20 指定されたオペランドを読み出しする関数

## 5.7.12 指定されたオペランドを書き込みする関数

図 5.21に指定されたオペランドを書き込みする関数のフローチャートを示します。

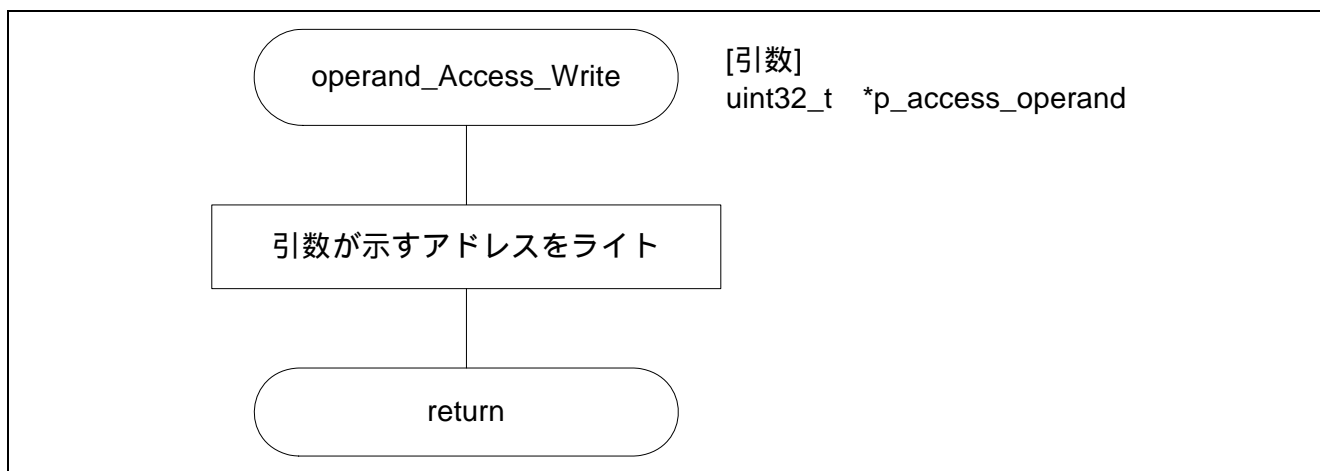


図5.21 指定されたオペランドを書き込みする関数

## 5.7.13 監視領域 A を実行する関数

図 5.22に監視領域 A を実行する関数のフローチャートを示します。

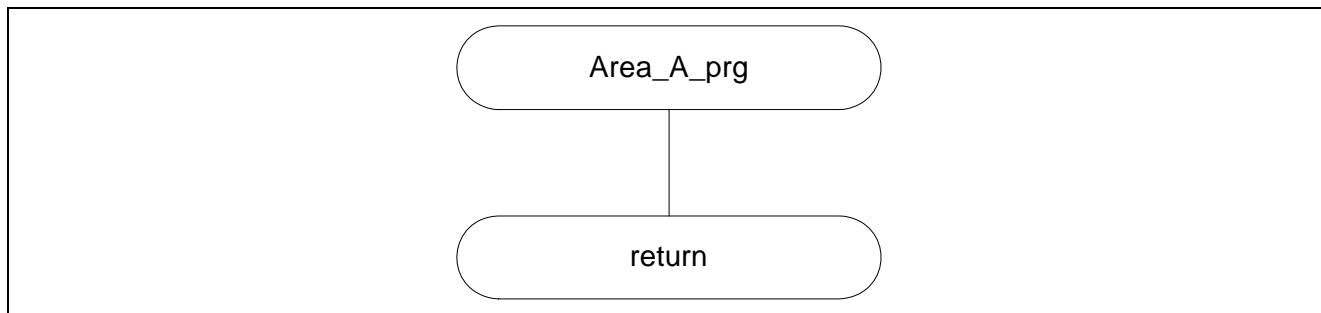


図5.22 監視領域 A を実行する関数

## 5.7.14 監視領域 B を実行する関数

図 5.23に監視領域 B を実行する関数のフローチャートを示します。

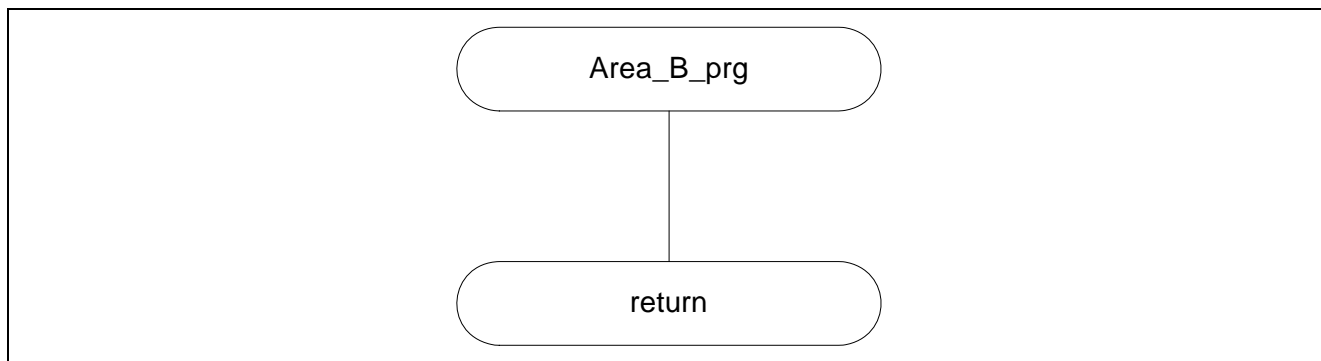


図5.23 監視領域 B を実行する関数

## 5.7.15 監視領域 C を実行する関数

図 5.24に監視領域 C を実行する関数のフローチャートを示します。

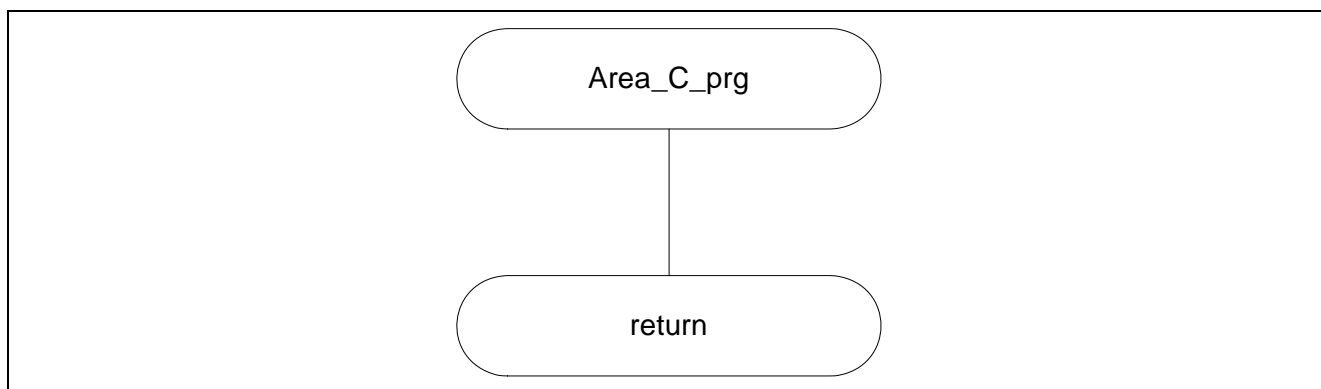


図5.24 監視領域 C を実行する関数



## 5.7.16 スーパーバイザモードからユーザモードに変更

図 5.25 にスーパーバイザモードからユーザモードに変更のフローチャートを示します。

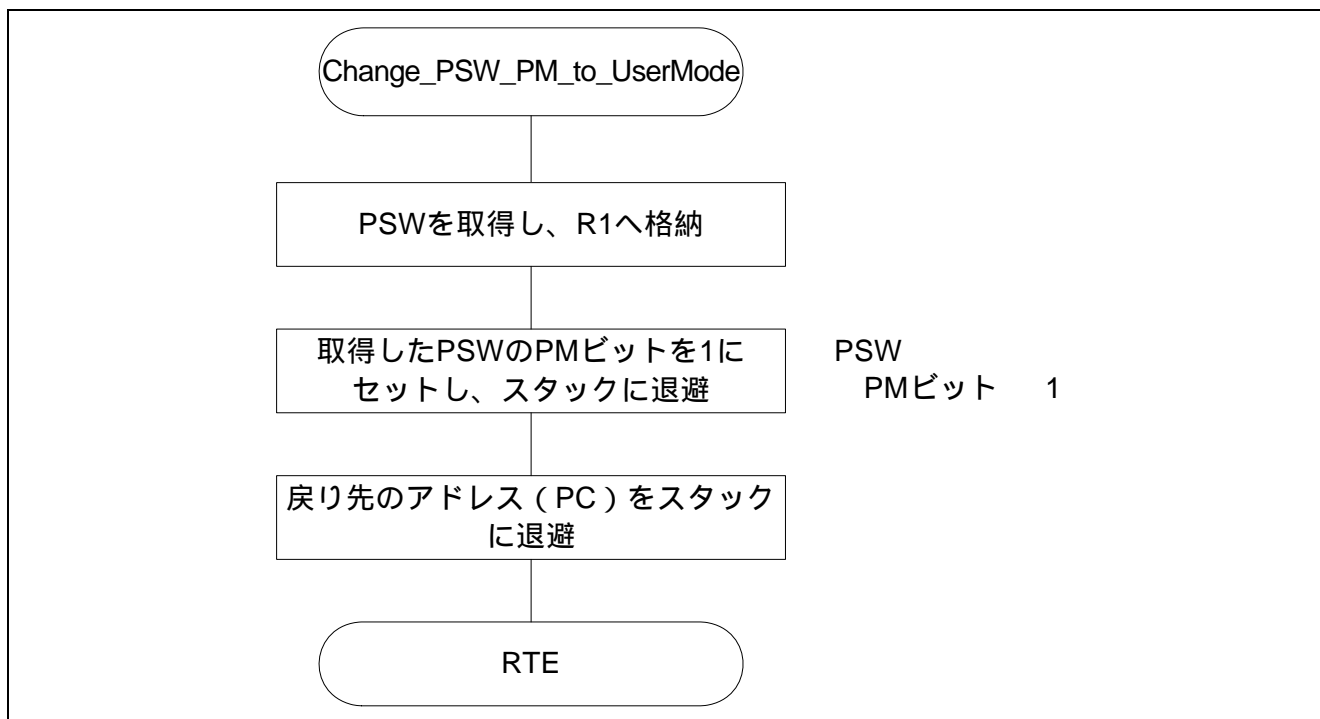


図5.25 スーパーバイザモードからユーザモードに変更

## 5.7.17 ユーザモードからスーパーバイザモードに変更

図 5.26 にユーザモードからスーパーバイザモードに変更のフローチャートを示します。

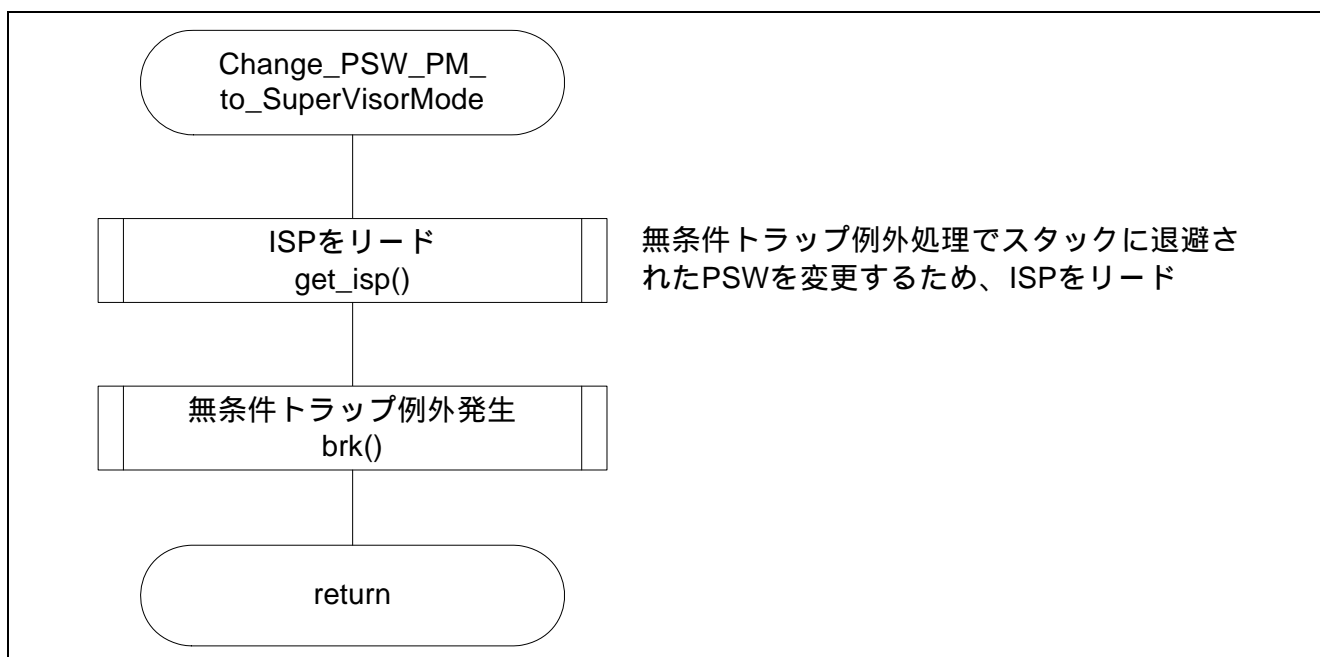


図5.26 ユーザモードからスーパーバイザモードに変更

## 5.7.18 初期設定関数の呼び出し

図 5.27に初期設定関数の呼び出しのフローチャートを示します。

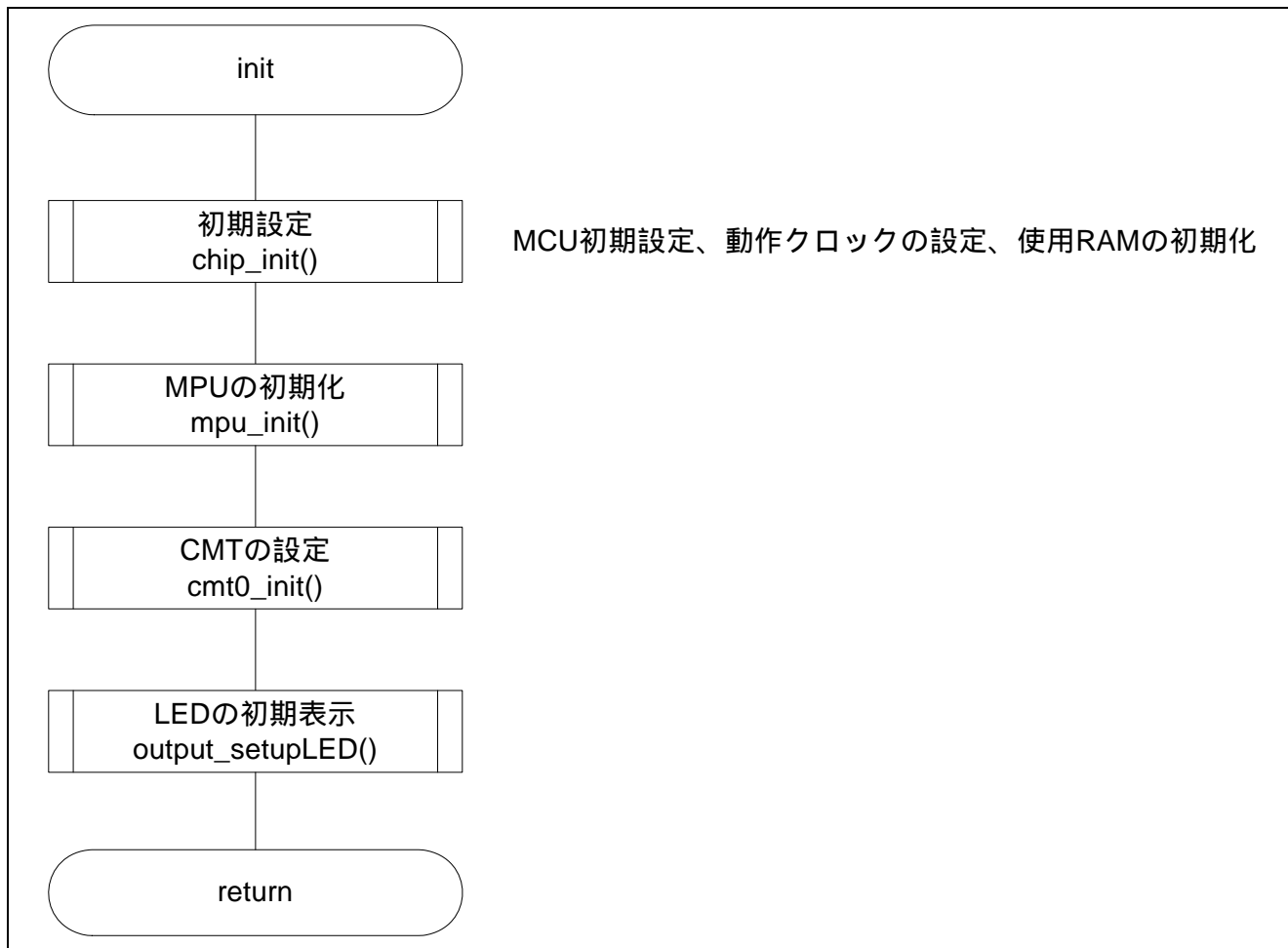


図5.27 初期設定関数の呼び出し

5.7.19 MCU 初期設定、動作クロックの設定、使用 RAM の初期化

図 5.28にMCU 初期設定、動作クロックの設定、使用 RAM の初期化のフローチャートを示します。

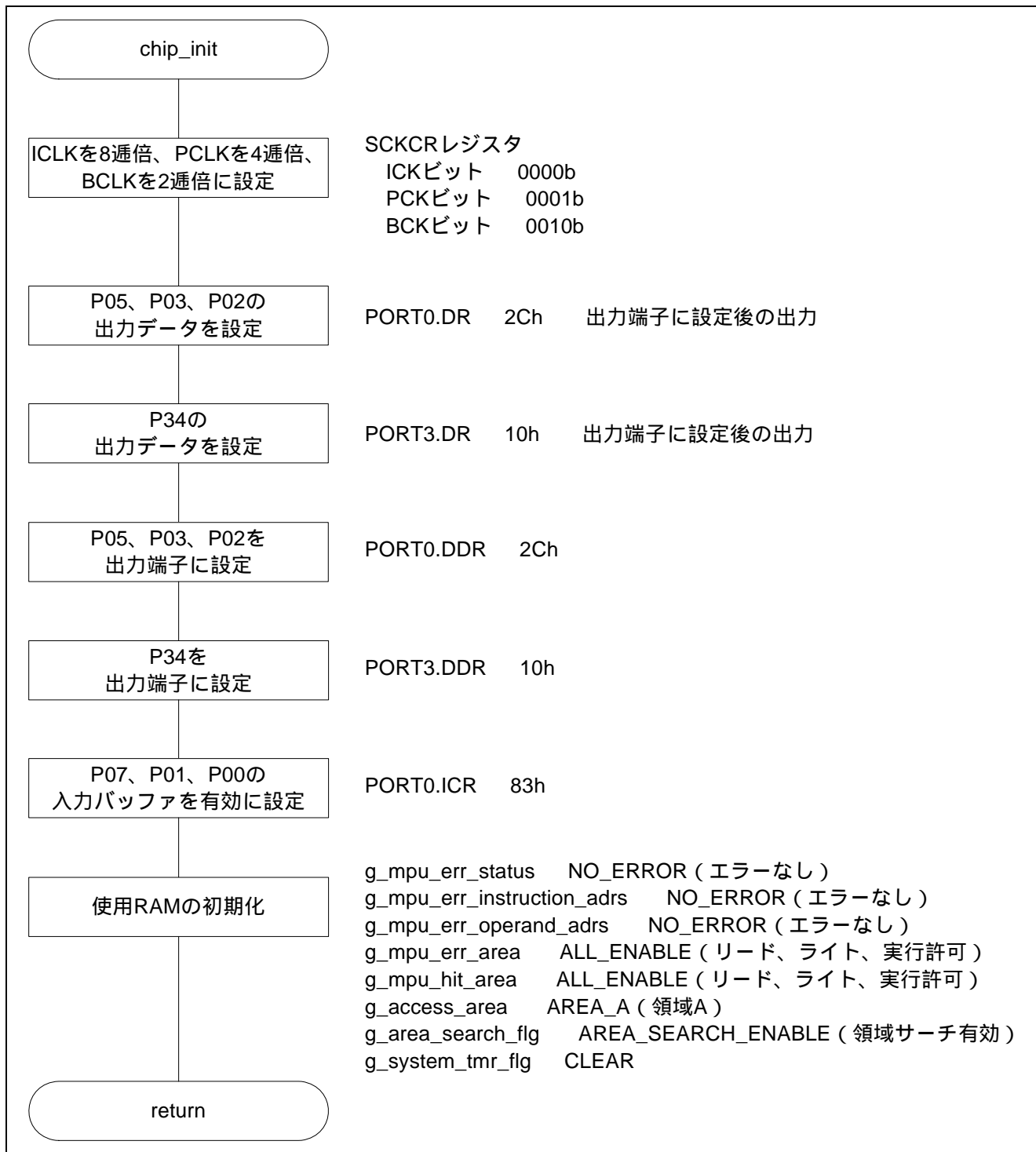


図5.28 MCU 初期設定、動作クロックの設定、使用 RAM の初期化

5.7.20 MPU の初期化

図 5.29、図 5.30にMPU の初期化 1、MPU の初期化 2のフローチャートを示します。

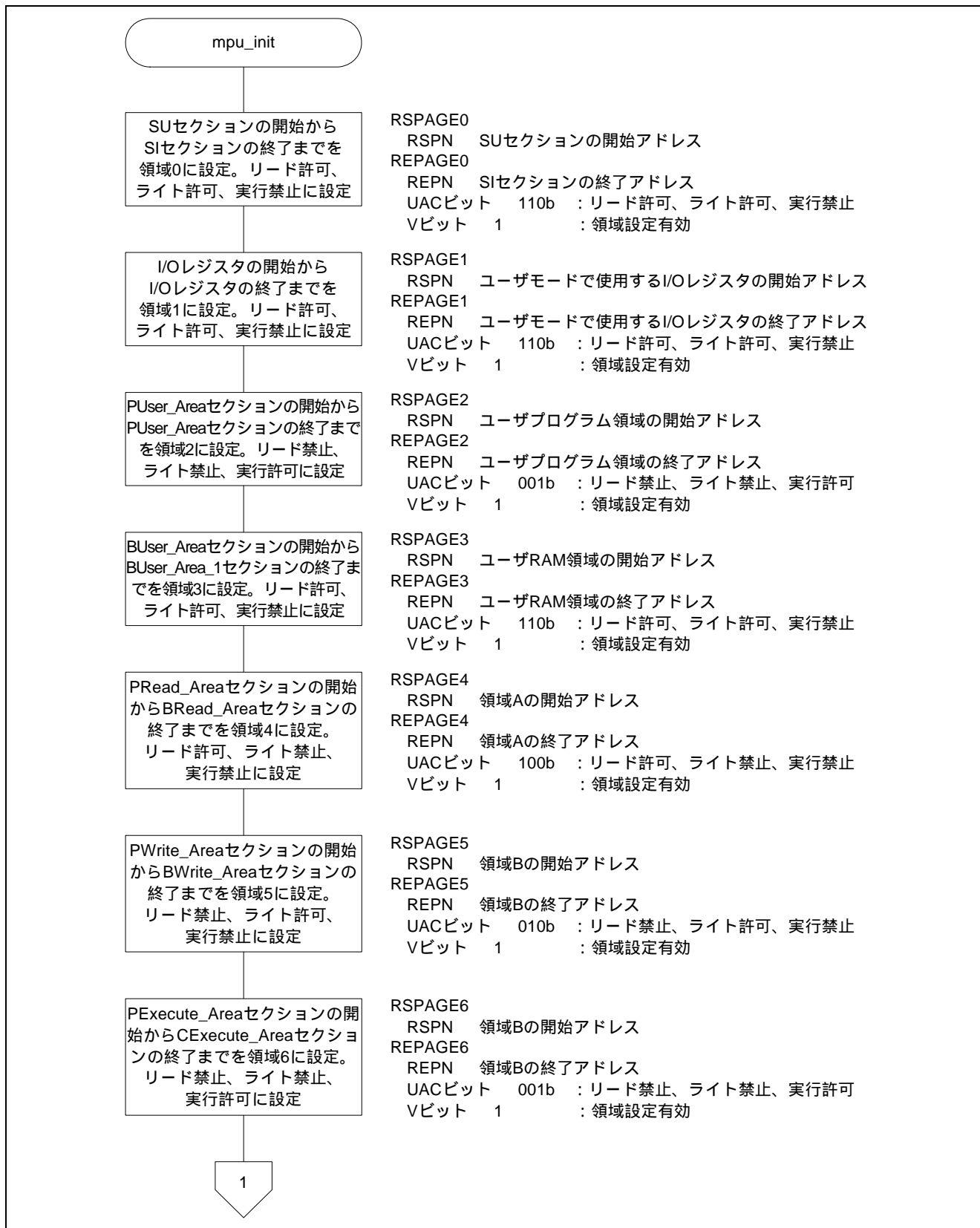


図5.29 MPU の初期化 1

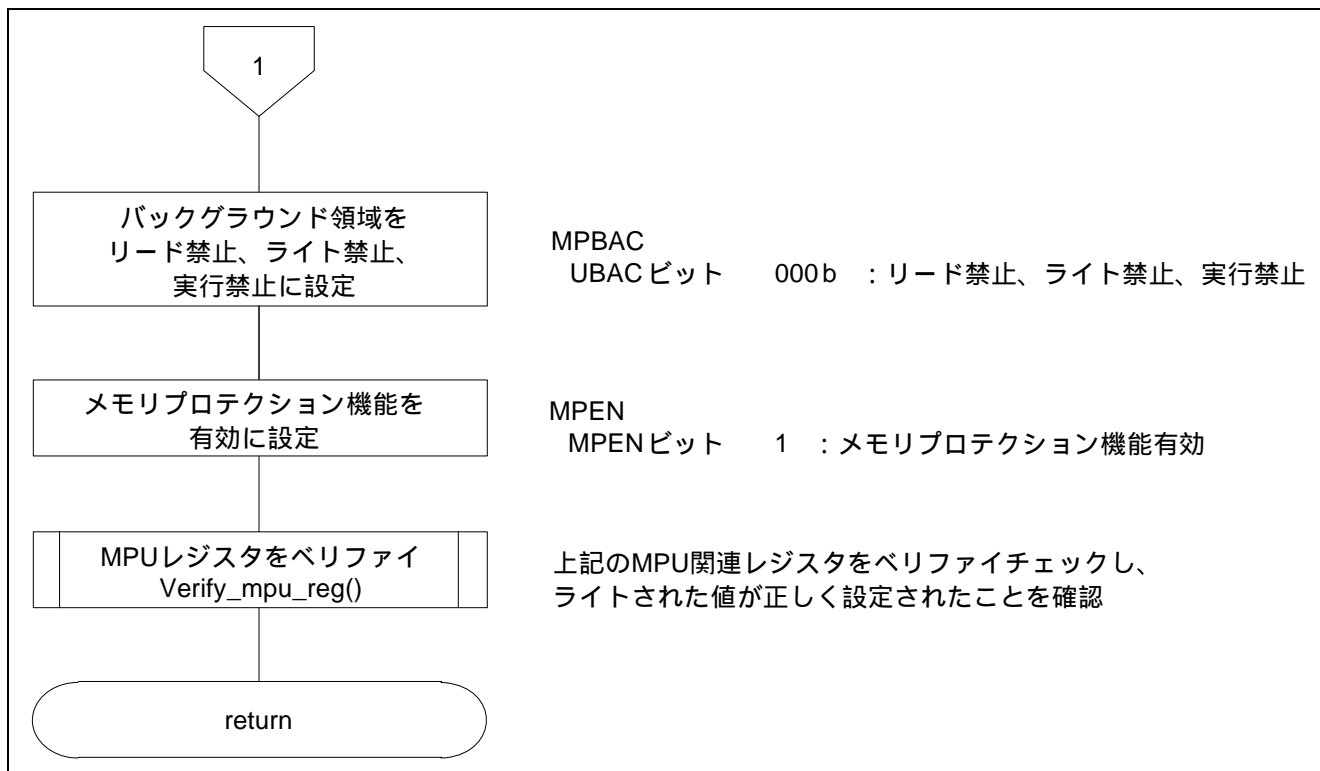


図5.30 MPUの初期化2

## 5.7.21 MPU 関連レジスタのベリファイ

図 5.31にMPU 関連レジスタのベリファイのフローチャートを示します。

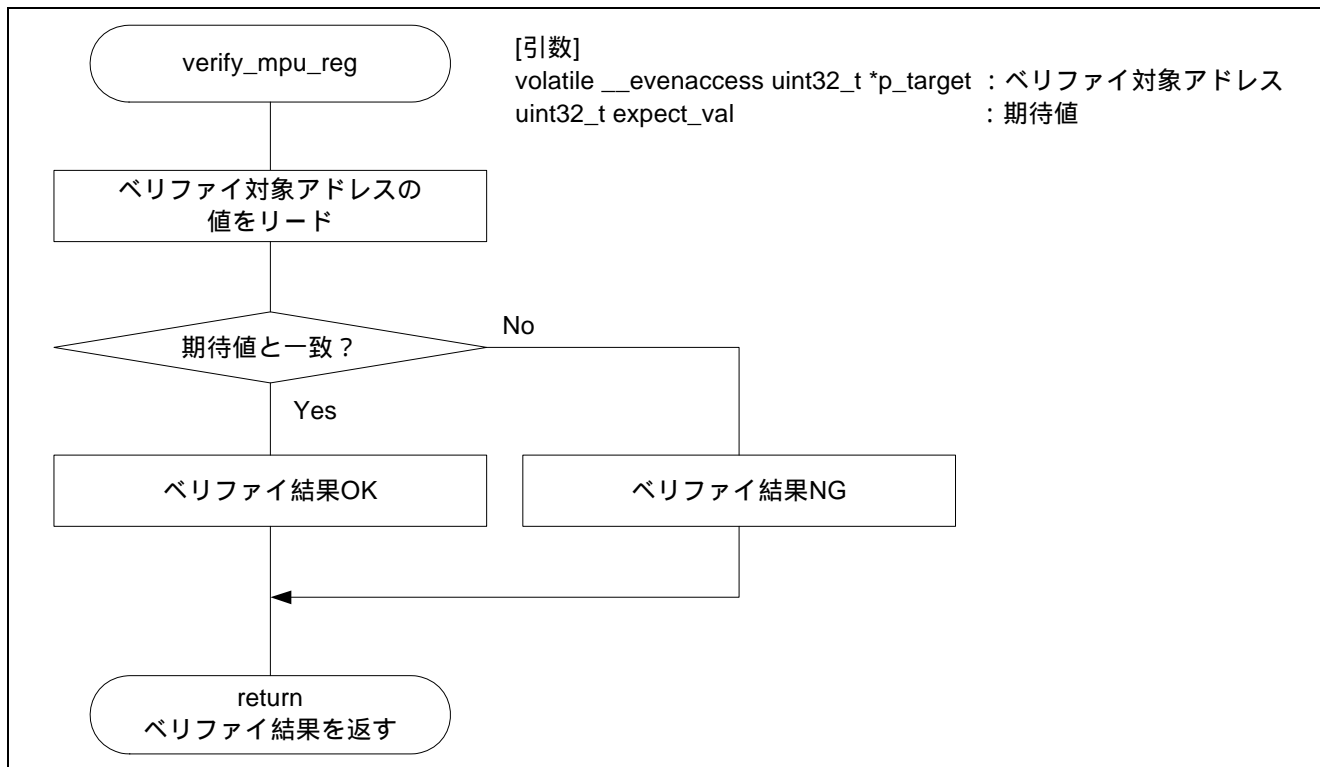


図5.31 MPU 関連レジスタのベリファイ

## 5.7.22 CMT0 の設定

図 5.32にCMT0 の設定のフローチャートを示します。

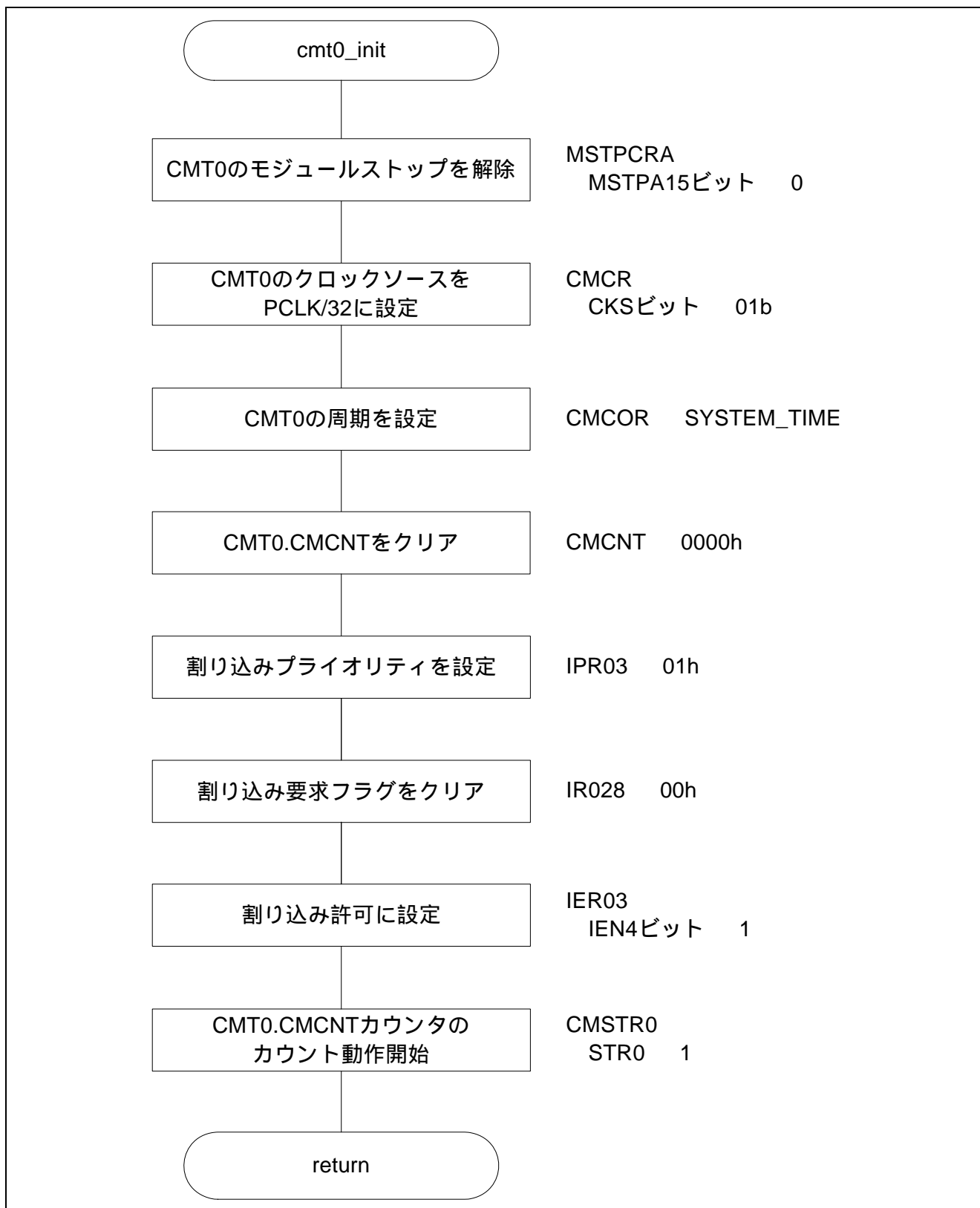


図5.32 CMT0 の設定

## 5.7.23 無条件トラップの例外関数

図 5.33に無条件トラップの例外関数のフローチャートを示します。

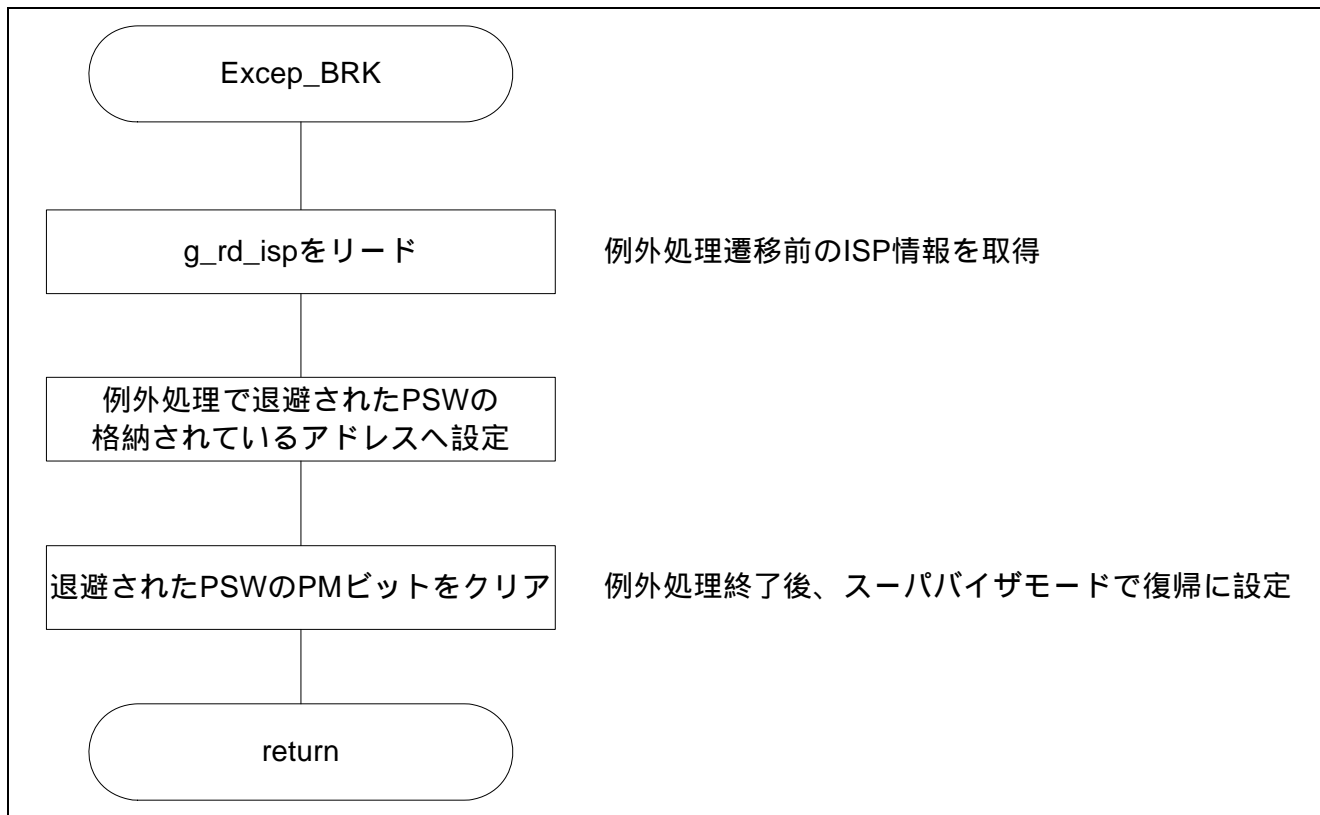


図5.33 無条件トラップの例外関数



## 5.7.24 メモリアプロテクションエラー発生時のアクセス例外関数

図 5.34にメモリアプロテクションエラー発生時のアクセス例外関数のフローチャートを示します。

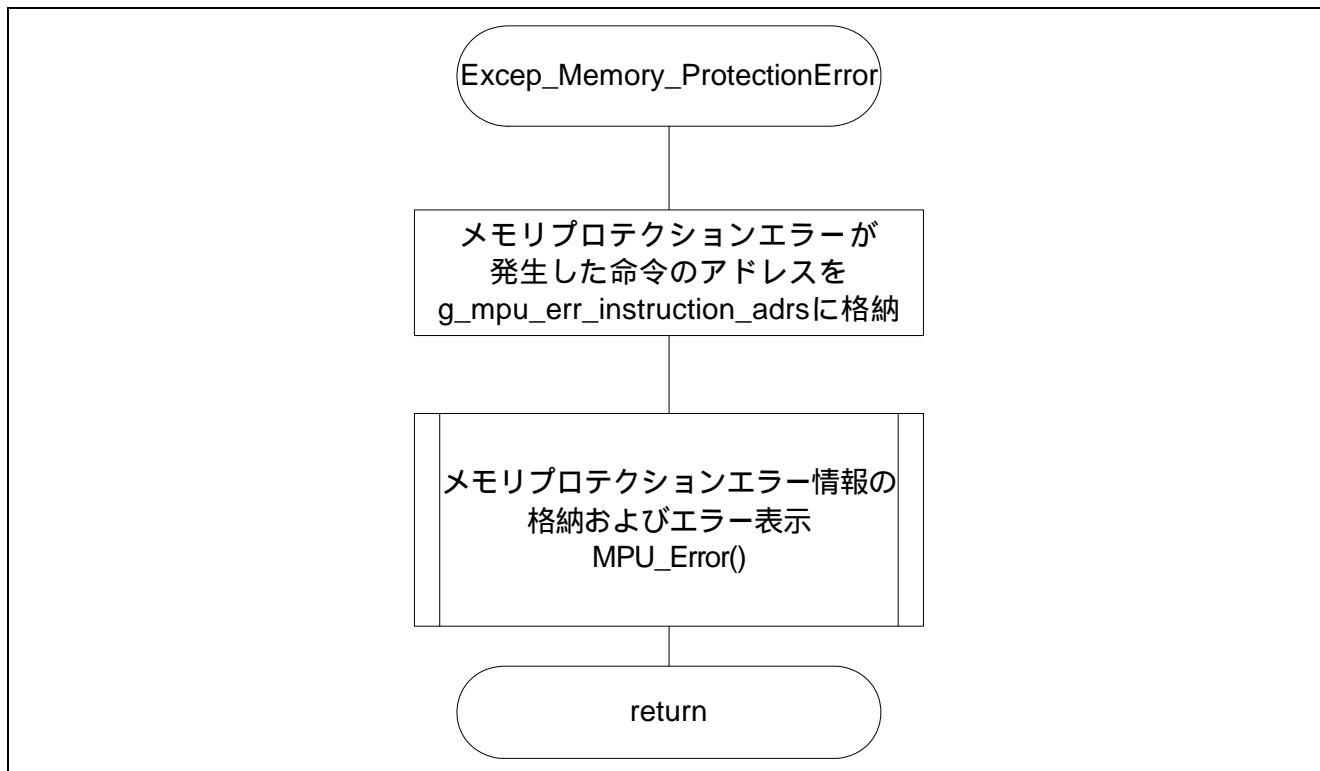


図5.34 メモリアプロテクションエラー発生時のアクセス例外関数

5.7.25 メモリプロテクションエラー情報の格納およびエラー表示

図 5.35にメモリプロテクションエラー情報の格納およびエラー表示のフローチャートを示します。

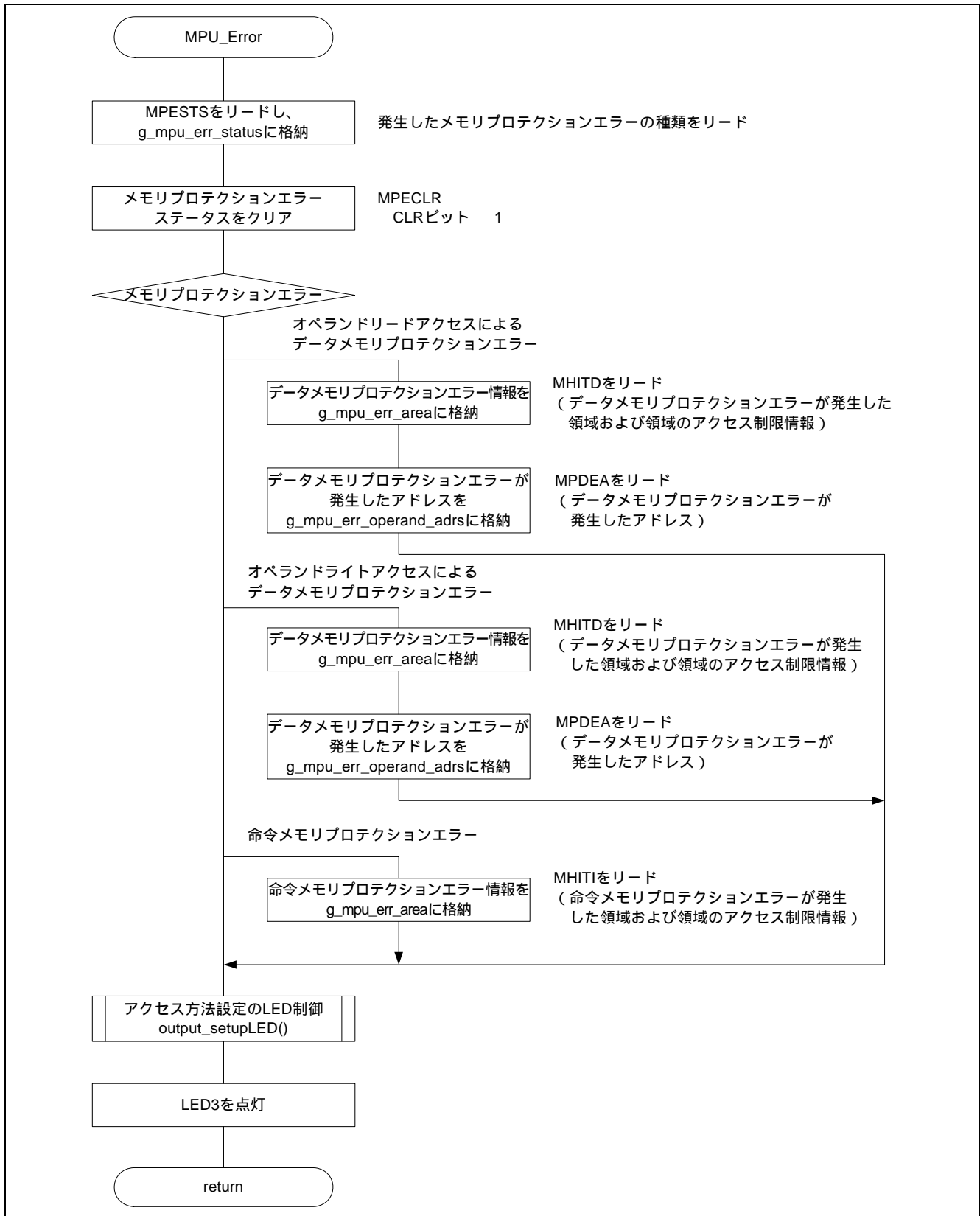


図5.35 メモリプロテクションエラー情報の格納およびエラー表示

## 5.7.26 CMT0 コンペアマッチ例外関数

図 5.36にCMT0 コンペアマッチ例外関数のフローチャートを示します。

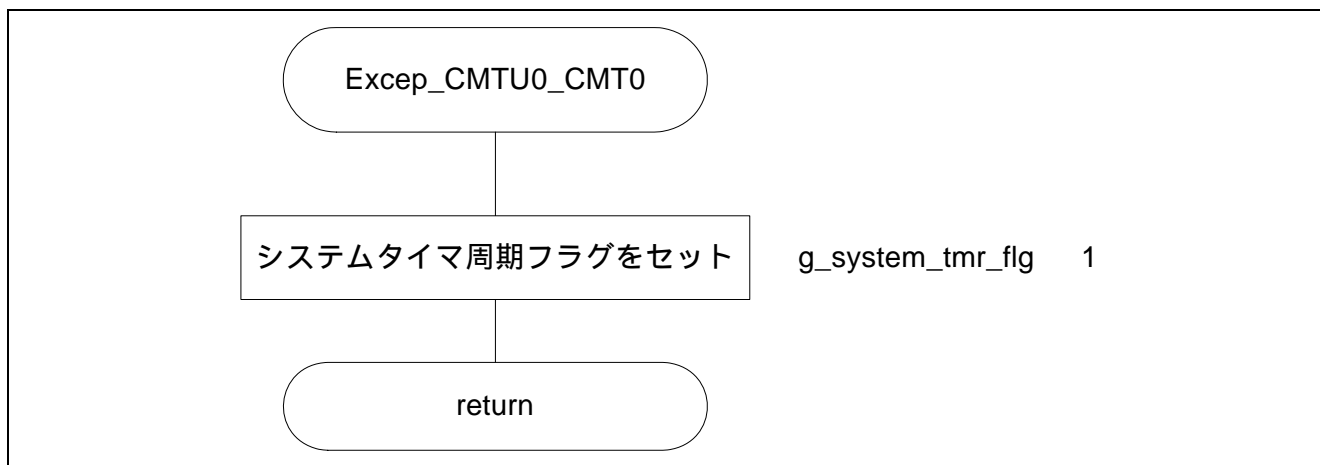


図5.36 CMT0 コンペアマッチ例外関数

## 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

- ハードウェアマニュアル  
RX62N、RX621 ユーザーズマニュアル ハードウェア編 Rev.1.30  
(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。)
- C コンパイラマニュアル  
RX ファミリー C/C++ コンパイラ、  
アセンブラ、最適化リンケージエディタ  
コンパイラパッケージ V.1.01 ユーザーズマニュアル Rev.1.00  
(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。)

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/contact/>

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.07.01		初版発行

## 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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違くと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

## 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のですが、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。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用途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

■ 営業お問合せ窓口

<http://www.renesas.com>

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

ルネサス エレクトロニクス株式会社 〒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。

総合お問合せ窓口 : <http://japan.renesas.com/contact/>