RZ/A1Hグループ
シリアルフラッシュメモリからのブート例

R01AN1960JJ0101
Rev.1.01
2015.05.29

要旨

本アプリケーションノートは、RZ/A1Hの SPI マルチ I/O バスコントローラ（以下、SPIBSC とします）を使用して、ブートモード 3（シリアルフラッシュブート）によってシリアルフラッシュメモリからブートを行う例について説明します。

対象デバイス

RZ/A1H

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

RZ/A1H は、ブートモード 3 の場合、SPI マルチ I/O バス空間（チャネル 0）に配置されたシリアルフラッシュメモリからブートします（以下、シリアルフラッシュブートとします）。図 1.1にシリアルフラッシュブートの動作イメージを示します。

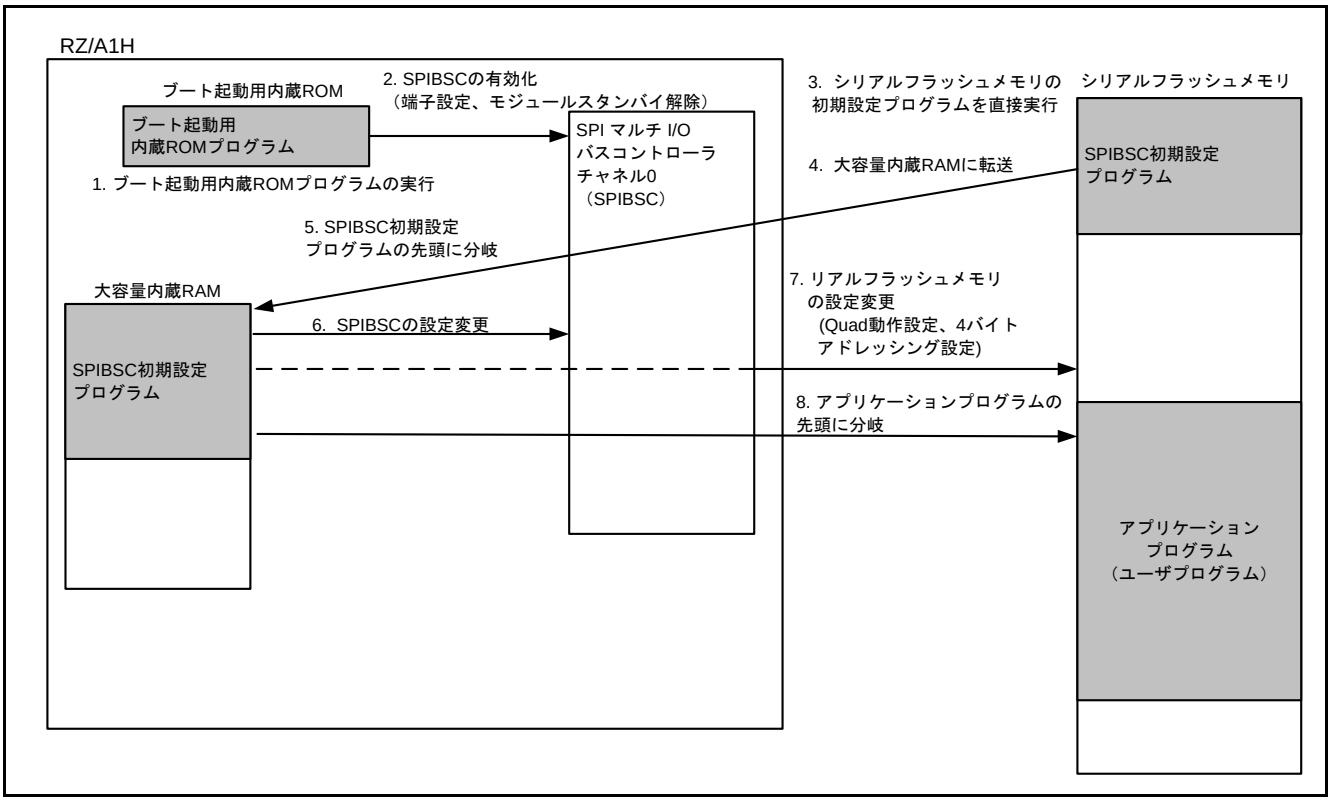

図1.1　シリアルフラッシュブートの動作イメージ

シリアルフラッシュブートの動作イメージについて説明します。

1　RZ/A1H は、シリアルフラッシュブートで起動した場合、パワーオンリセット解除後にブート起動用内蔵 ROM プログラムを実行します。
2　ブート起動用内蔵 ROM プログラムは、SPIBSC を外部アドレス空間リードモードに設定し、SPI マルチ I/O バス空間（チャネル 0）に配置されたプログラムを直接実行できる状態にします。
3　シリアルフラッシュメモリの初期設定プログラムを実行します。
4　シリアルフラッシュメモリから SPIBSC 初期設定プログラムを大容量内蔵 RAM に転送します。
5　大容量内蔵 RAM に転送した SPIBSC 初期設定プログラムの先頭アドレスに分岐します。
6　SPIBSC 初期設定プログラムにより、SPIBSC の設定を変更します。
7　SPIBSC 初期設定プログラムにより、シリアルフラッシュメモリの設定を変更します。
8　アプリケーションプログラムの先頭アドレスに分岐します。

ブート起動用内蔵 ROM プログラムは、一般的なシリアルフラッシュメモリに共通でアクセスできる設定を行っているため、お客様が使用するシリアルフラッシュメモリに最適な設定を行う必要があります。このため、本アプリケーションノートでは、ブート起動用内蔵 ROM プログラムより分岐する SPI マルチ I/O バス空間（チャネル 0）の先頭番地（H'1800_0000）に、お客様が使用するシリアルフラッシュメモリに最適な設定を行う SPIBSC 初期設定プログラムを配置し、SPIBSC 初期設定プログラムより最適な設定を行った後、お客様が作成するアプリケーションプログラム（ユーザプログラム）に分岐する方法を説明します。

本アプリケーションノートでは、SPIBSC 初期設定プログラムにて、お客様が使用するシリアルフラッシュメモリに応じ、最適に設定する方法および、アプリケーションプログラム（ユーザプログラム）の作成方法について説明します。

表 1.1に使用する周辺機能と用途を示します。

表1.1　使用する周辺機能と用途

周辺機能	用途
SPI マルチ I/O バスコントローラ（SPIBSC）	外部アドレス空間リードモードに設定し、CPU が SPI マルチ I/O バス空間（チャネル 0）に接続されたシリアルフラッシュメモリから、直接リードするための信号を生成します。
汎用入出力ポート	SPIBSC にて使用する端子の割り当てに使用します。

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	RZ/A1H
動作周波数	CPU クロック（Iφ）：400MHz 画像処理クロック（Gφ）：266.67MHz 内部バスクロック（Bφ）：133.33MHz 周辺クロック（P1φ）：66.67MHz 周辺クロック（P0φ）：33.33MHz
動作電圧	電源電圧（I/O）：3.3V 電源電圧（内部）：1.18V
統合開発環境	ARM®統合開発環境 ARM Development Studio 5（DS-5™）Version 5.16
C コンパイラ	ARM C/C++ Compiler/Linker/Assembler Ver.5.03 [Build 102]
	コンパイラオプション -O3 -Ospace --cpu=Cortex-A9 --littleend --arm --apcs=/interwork --no_unaligned_access --fpu=vfpv3_fp16 -g --asm
動作モード	ブートモード 3 （シリアルフラッシュブート）
使用ボード	GENMAI ボード • R7S72100 CPU ボード　RTK772100BC00000BR
使用デバイス （ボード上で使用する機能）	シリアルフラッシュメモリ（SPI マルチ I/O バス空間（チャネル 0）に接続） • メーカー：Spansion 社製 • 型名：S25FL512S

【注】 クロックモード 3（EXTAL 端子からの 13.33MHz のクロック入力）で使用時の動作周波数です。

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。

- RZ/A1Hグループ レジスタ定義ヘッダ・ファイル iodefine.h（R01AN1860JJ）
- RZ/A1Hグループ 初期設定例（R01AN1864JJ）
- RZ/A1Hグループ ARM® Development Studio 5（DS-5™）のセミホスティング機能を使用したシリアルフラッシュメモリへのダウンロード例（R01AN1958JJ）

4.　ハードウェア説明

4.1　　ハードウェア構成例

図 4.1にブートモード 3 にてシリアルフラッシュメモリからブートする場合の接続例を示します。

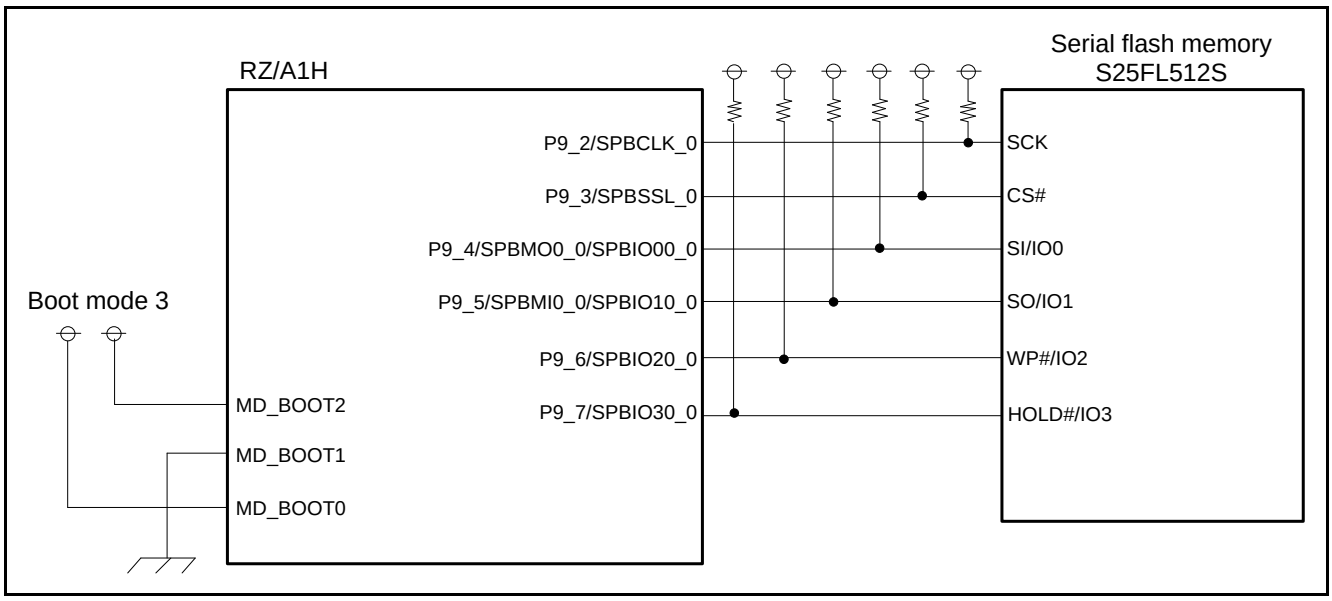

図4.1　シリアルフラッシュメモリからブートする場合の接続例

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
SPBCLK_0	出力	クロック出力
SPBSSL_0	出力	スレーブセレクト
SPBMO0_0/SPBIO00_0	入出力	マスタ送出データ／データ 0
SPBMI0_0/SPBIO10_0	入出力	マスタ入力データ／データ 1
SPBIO20_0	入出力	データ 2
SPBIO30_0	入出力	データ 3
MD_BOOT2	入力	ブートモードの選択 MD_BOOT2："H"、MD_BOOT1："L"、MD_BOOT0："H" （ブートモード 3 に設定）
MD_BOOT1	入力	
MD_BOOT0	入力	

5.　ソフトウェア説明

5.1　　動作概要

ここでは、本アプリケーションノートのサンプルコードの動作概要について説明します。

5.1.1　　シリアルフラッシュブートに関する用語

表 5.1に本アプリケーションノートで説明するシリアルフラッシュブート動作に関する用語を示します。

表5.1　シリアルフラッシュブート動作に関する用語

用語	説明
ブート起動用内蔵 ROM プログラム	ブート起動用内蔵 ROM プログラムは、ブートモード 3（シリアルフラッシュブート）で起動した場合に、SPI マルチ I/O バス空間（チャネル 0）に接続されたシリアルフラッシュメモリに格納されているプログラムを直接実行するための設定を行うプログラムです。 RZ/A1Hはブート起動用内蔵 ROM プログラムの実行完了後、SPI マルチ I/O バス空間（チャネル 0）の先頭アドレスである H'1800_0000 番地に分岐します。なお、ブート起動用内蔵 ROM プログラムでは、一般的なシリアルフラッシュメモリに共通でアクセスできる設定を行っています。 CPU 内のブート起動用内蔵 ROM に格納されているプログラムのため、お客様が作成する必要はありません。
SPIBSC 初期設定プログラム	SPIBSC 初期設定プログラムは、ブート起動用内蔵 ROM プログラムの処理完了後に実行するプログラムです。 SPIBSC 初期設定プログラムは、お客様が使用するシリアルフラッシュメモリに応じて、SPIBSC およびシリアルフラッシュメモリ内のレジスタ設定処理を行い、アプリケーションプログラムの先頭アドレスへ分岐する処理を行います。 SPIBSC 初期設定プログラムは、本アプリケーションノートを参考に、使用するシリアルフラッシュメモリの仕様に応じてお客様が作成してください。なお、サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）を使用する場合に最適な設定を行っています。
アプリケーションプログラム （ユーザプログラム）	アプリケーションプログラムは、お客様がシステムに応じて作成するプログラムです。

5.1.2　サンプルコード全体の動作概要

サンプルコードはブート起動用内蔵 ROM プログラムから実行される SPIBSC 初期設定プログラムとアプリケーションプログラムで構成されています。

1　SPIBSC 初期設定プログラム
SPIBSC 初期設定プログラムは、使用するシリアルフラッシュメモリ（Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S））に最適な設定を行います。SPIBSC 初期設定プログラムはブート起動用内蔵 ROM プログラムより分岐する SPI マルチ I/O バス空間の先頭番地（H'1800_0000）に配置し、ブート起動用内蔵 ROM プログラムから実行できるようにしています。SPIBSC 初期設定プログラム実行後、アプリケーションプログラムの先頭番地に分岐します。

2　アプリケーションプログラム（ユーザプログラム）
アプリケーションプログラムは、SPIBSC 初期設定プログラムにてシリアルフラッシュメモリに最適な設定後に実行するアプリケーションプログラムです。サンプルコードでは、アプリケーションプログラムを H'1808_0000 番地に配置しています。

図 5.1に本アプリケーションノートのサンプルコードの動作概要を示します。

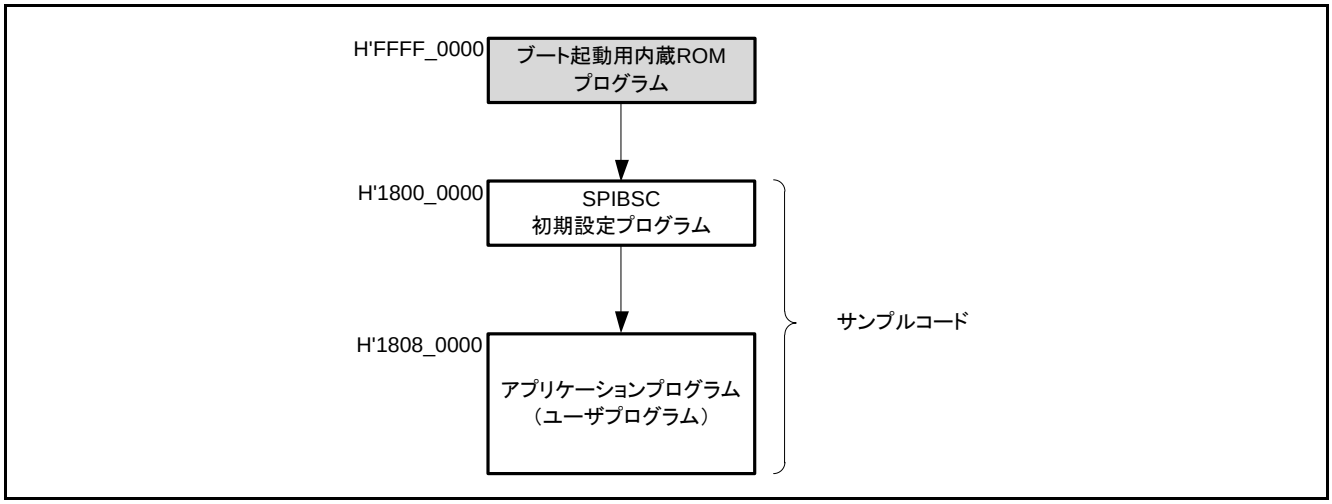

図5.1　サンプルコードの動作概要

5.1.3　SPIBSC 初期設定プログラムの動作概要

SPIBSC 初期設定プログラムは、ブート起動用内蔵 ROM プログラムから実行されるプログラムです。SPIBSC 初期設定プログラムは、ブート起動用内蔵 ROM プログラムより分岐する SPI マルチ I/O バス空間の先頭番地（H'1800_0000）に配置してください。

ブート起動用内蔵 ROM プログラムは、SPIBSC を外部アドレス空間リードモードに設定します。設定により、RZ/A1Hは SPI マルチ I/O バス空間へのリードを SPI 通信に変換し、接続されたシリアルフラッシュメモリに対して直接リードが可能となり、SPI マルチ I/O バス空間に配置されたプログラムを直接実行することが可能な状態となります。SPI 通信変換に使用するシリアルフラッシュメモリへのコマンドの設定は、一般的なシリアルフラッシュメモリに共通でアクセスできる設定にしているため、SPIBSC 初期設定プログラムにて、お客様が使用するシリアルフラッシュメモリに最適な設定を行う必要があります。

ブート起動用内蔵 ROM プログラム実行後の設定については、表 5.2および表 5.3を参照してください。

シリアルフラッシュメモリの最適な設定は、SPIBSC モジュール内のレジスタの設定（以下、SPIBSC 設定とします）、およびシリアルフラッシュメモリ内のレジスタ設定（以下、シリアルフラッシュメモリ設定とします）を行う必要があり、サンプルコードの SPIBSC 初期設定プログラムでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）を使用する場合に最適な設定を行っています。

また、SPIBSC 初期設定プログラムは、以下に示す SPIBSC 初期設定プログラム 1 と SPIBSC 初期設定プログラム 2 で構成されており、それぞれ SPI マルチ I/O バス空間から大容量内蔵 RAM に転送し、大容量内蔵 RAM 上で実行します。

1　SPIBSC 初期設定プログラム 1
SPIBSC 初期設定プログラム 1 では、SPIBSC 設定として、遅延設定（次アクセス遅延、SPBSSL ネゲート遅延、クロック遅延）の期間を短く設定し、転送ビットレートの設定、リードキャッシュの有効設定を行います。処理内容が少ないため、比較的小さいプログラムサイズとなっています。

2　SPIBSC 初期設定プログラム 2
SPIBSC 初期設定プログラム 2 では、SPIBSC 設定として、転送ビットレートの設定、データバス幅の 4 ビット設定、4 バイトアドレス出力設定を行います。また、シリアルフラッシュメモリ設定として、レイテンシコードの設定、Quad 動作設定、4 バイトアドレッシング設定を行います。処理内容が多いため、初期設定プログラム 1 よりも大きいプログラムサイズとなっています。

SPIBSC 初期設定プログラム 1 および SPIBSC 初期設定プログラム 2 は、SPI マルチ I/O バス空間に配置されたプログラムで設定することはできないため、大容量内蔵 RAM 上で実行する必要があります。サンプルコードでは、最初に SPIBSC 初期設定プログラム 1 を大容量内蔵 RAM に転送して実行し、可能な限り使用するシリアルフラッシュメモリに最適な設定にした後に、SPIBSC 初期設定プログラム 2 を大容量内蔵 RAM に転送して、実行しています。これにより、全体の SPIBSC 初期設定プログラムの実行時間を短縮しています。

表 5.2および表 5.3に SPIBSC 初期設定プログラムの設定内容を示します。

SPIBSC 初期設定プログラムにて、表 5.2および表 5.3に示す設定を行った後、アプリケーションプログラムの先頭番地に分岐します。サンプルコードでは、アプリケーションプログラムを H'1808_0000 番地に配置しています。

表5.2　SPIBSC 設定およびシリアルフラッシュメモリ設定（1/2）

	項目	ブート起動用内蔵 ROM プログラム	SPIBSC 初期設定プログラム 1	SPIBSC 初期設定プログラム 2
SPIBSC 設定	遅延設定			
	次アクセス遅延設定：			
	SSLDR.SPNDL[2:0]	B'111 (8SPBCLK)	B'000 (1SPBCLK)	B'000 (1SPBCLK)
	SPBSSL ネゲート遅延設定：			
	SSLDR.SLNDL[2:0]	B'111 (8.5SPBCLK)	B'000 (1.5SPBCLK)	B'000 (1.5SPBCLK)
	クロック遅延設定：			
	SSLDR.SCKDL[2:0]	B'111 (8SPBCLK)	B'000 (1SPBCLK)	B'000 (1SPBCLK)
	シリアルクロック：			
	(Bφ=133.33MHz で動作時)	Bφ/ 8=16.67[MHz]	Bφ/ 4=33.33[MHz]	Bφ/ 2=66.67[MHz]
	SPBCR.SPBR[7:0]	0	2	1
	SPBCR.BRDV[1:0]	3	0	0
	CPOL：CMNCR.CPOL	0	0	0
	CPHAT：CMNCR.CPHAT	0	0	0
	CPHAR：CMNCR.CPHAR	0	0	1
	SPBSSL 出力アイドル値固定：	SPBSSL ネゲート期間の出力値を、前回転送の最終ビットに設定	SPBSSL ネゲート期間の出力値を、前回転送の最終ビットに設定	SPBSSL ネゲート期間の出力値を、Hi-z に設定
	SPBIO30, SPBIO31 の設定	CMNCR.MOIIO3[1:0]=B'10	CMNCR.MOIIO3[1:0]=B'10	CMNCR.MOIIO3[1:0]=B'11
	SPBIO20, SPBIO21 の設定	CMNCR.MOIIO2[1:0]=B'10	CMNCR.MOIIO2[1:0]=B'10	CMNCR.MOIIO2[1:0]=B'11
	SPBIO10, SPBIO11 の設定	CMNCR.MOIIO1[1:0]=B'10	CMNCR.MOIIO1[1:0]=B'10	CMNCR.MOIIO1[1:0]=B'11
	SPBIO00, SPBIO01 の設定	CMNCR.MOIIO0[1:0]=B'10	CMNCR.MOIIO0[1:0]=B'10	CMNCR.MOIIO0[1:0]=B'11
	端子の出力値固定：	1 ビット/2 ビット幅の端子の出力値を、前回転送の最終ビットに設定	1 ビット/2 ビット幅の端子の出力値を、前回転送の最終ビットに設定	1 ビット/2 ビット幅の端子の出力値を、Hi-z に設定
	SPBIO30, SPBIO31 の設定	CMNCR.IO3FV[1:0]=B'10	CMNCR.IO3FV[1:0]=B'10	CMNCR.IO3FV[1:0]=B'11
	SPBIO20, SPBIO21 の設定	CMNCR.IO2FV[1:0]=B'10	CMNCR.IO2FV[1:0]=B'10	CMNCR.IO2FV[1:0]=B'11
	SPBIO00, SPBIO01 の設定	CMNCR.IO0FV[1:0]=B'10	CMNCR.IO0FV[1:0]=B'10	CMNCR.IO0FV[1:0]=B'11
	リードキャッシュ：DRCR.RBE	0 (無効)	1 (有効)	1 (有効)
	リードデータバースト長：	1 データ長	1 データ長	2 データ長
	DRCR.RBURST[3:0]	B'0000	B'0000	B'0001
	データバス幅：	1 [bit]	1 [bit]	4 [bit]
	DRENR.DRDB[1:0]	B'00	B'00	B'10
	リードコマンド:	Read (03H)	Read (03H)	Quad I/O Read (ECH)
	DRCMR.CMD[7:0]	H'03	H'03	H'EC
	コマンド：	出力する	出力する	出力する
	DRENR.CDE	1	1	1
	オプショナルコマンド：	出力しない	出力しない	出力しない
	DRENR.OCDE	0	0	0

表5.3　SPIBSC 設定およびシリアルフラッシュメモリ設定（2/2）

	項目	ブート起動用内蔵 ROM プログラム	SPIBSC 初期設定 プログラム 1	SPIBSC 初期設定 プログラム 2
SPIBSC 設定	アドレス指定： DRENR.ADE[3:0]	Address[23:0]を出力 B'0111	Address[23:0]を出力 B'0111	Address[31:0]を出力 B'1111
	アドレスビット幅： DRENR.ADB[1:0]	1 [bit] B'00	1 [bit] B'00	4 [bit] B'10
	オプションデータ： DRENR.OPDE[3:0]	出力しない B'0000	出力しない B'0000	OPD3 を出力 B'1000
	オプションデータビット幅： DRENR.OPDB[1:0]	1 [bit] B'00	1 [bit] B'00	4 [bit] B'10
	ダミーサイクルイネーブル： DRENR.DME	挿入しない 0	挿入しない 0	挿入する 1
	ダミーサイクルビット幅： DRDMCR.DMDB[1:0]	－	－	1 [bit] B'00
	ダミーサイクル数： DRDMCR.DMCYC[2:0]	－	－	4 サイクル B'011
	拡張アドレス： DREAR.EAC[2:0] DREAR.EAV[7:0]	外部アドレス[24:0]が有効 32MB の空間に直接アクセス可能 B'000 H'00	外部アドレス[24:0]が有効 32MB の空間に直接アクセス可能 B'000 H'00	外部アドレス [25:0]が有効 64MB の空間に直接アクセス可能 B'001 H'00
	AC 入力特性調整ビット： CKDLY.CKDLY[3:0]	B'0100	B'0100	B'0100
	AC 出力特性調整ビット： SPODLY.SPODLY[15:0]	H'0000	H'0000	H'0000
シリアル フラッ シュメモ リ設定	Configuration Register 1	変更なし（注）	変更なし（注）	LC[1:0] = B'00 QUAD = 1 (Quad 動作)
その他	CPU の例外処理ベクタの アドレス	ハイベクタ （H'FFFF_0000～）	ローベクタ （H'0000_0000～）	ローベクタ （H'0000_0000～）

【注】 RZ/A1Hのシリアルフラッシュブート（ブートモード 3）では、シリアルフラッシュメモリにリードコマンド（オペコード：03H、アドレスビット：24 ビット、ダミーサイクル：出力しない）を発行するように SPIBSC のレジスタを設定します。このため、シリアルフラッシュメモリのレジスタ設定値が、シリアルフラッシュブート実行時に上記のリードコマンドを正常に受信できない設定となっている場合は、正常にブートできない可能性があります。

5.1.4　アプリケーションプログラム（ユーザプログラム）作成時の注意事項

アプリケーションプログラムは、SPIBSC 初期設定プログラムから分岐するアドレスに配置してください。なお、アプリケーションプログラムは使用するシリアルフラッシュメモリにおいて、SPIBSC 初期設定プログラムとは異なるセクタに配置してください。

サンプルコードでは、アプリケーションプログラムは H'1808_0000 番地に配置しています。

図 5.2にサンプルコードのプログラム配置を示します。

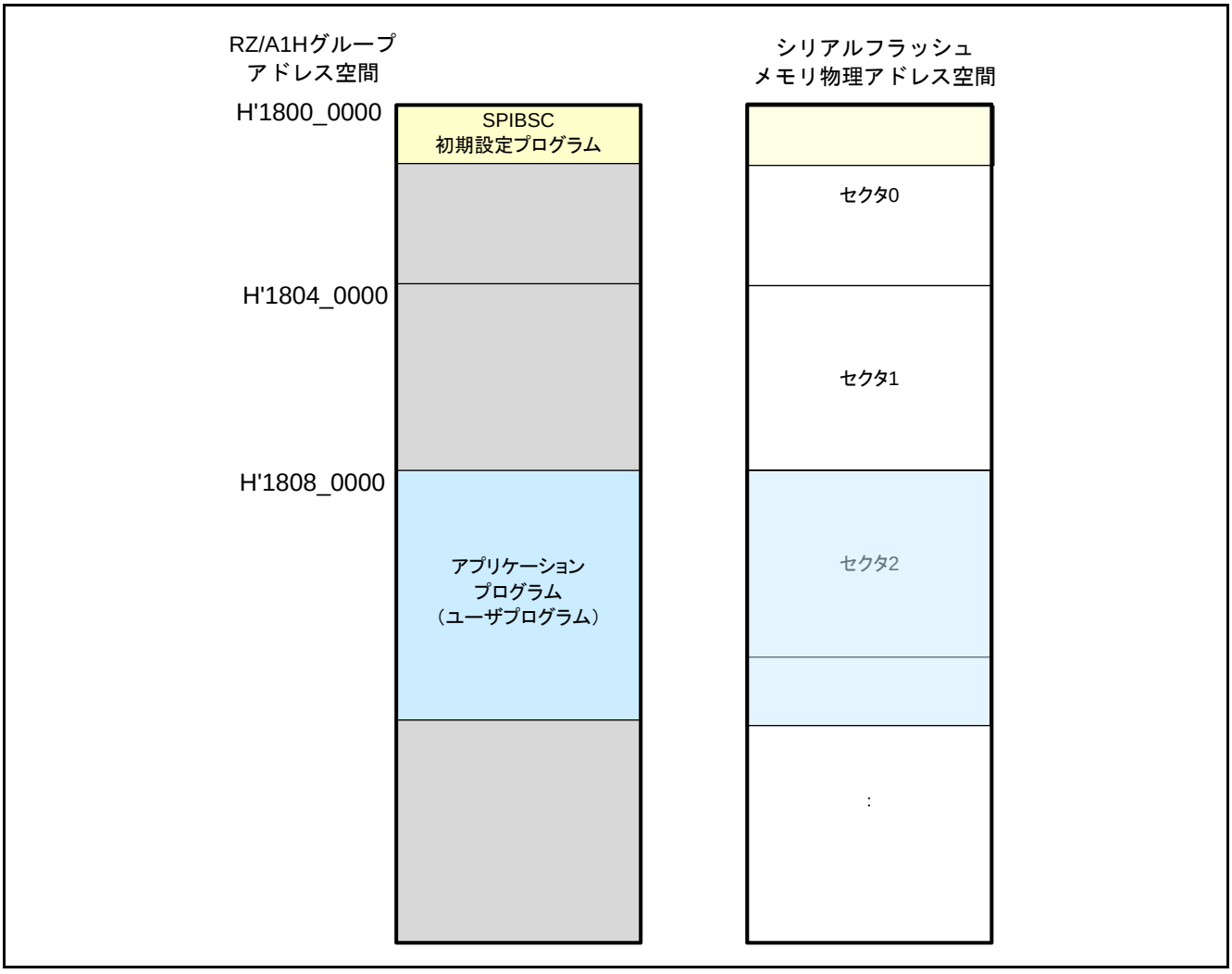

図5.2　サンプルコードのプログラム配置

5.2　サンプルコード実行時の周辺機能の設定およびメモリ配置

5.2.1　周辺機能の設定

表 5.4にサンプルコード実行時の周辺機能の設定内容を示します。

表5.4　周辺機能の設定内容

モジュール	設定内容
CPG	CPU クロック（Iφ）：400MHz 内部バスクロック（Bφ）：133.33MHz 周辺クロック（P1φ）：66.67MHz 周辺クロック（P0φ）：33.33MHz
SPI マルチ I/O バスコントローラ（SPIBSC）	外部アドレス空間リードモードに設定し、CPU が SPI マルチ I/O バス空間（チャネル 0）に接続されたシリアルフラッシュメモリから、直接リードするための信号を生成するための設定 設定内容は表 5.2および表 5.3を参照
PORT	PORT9 のマルチプレクス端子機能を設定 P9_2：SPBCLK_0 P9_3：SPBSSL_0 P9_4：SPBMO0_0/SPBIO00_0 P9_5：SPBMI0_0/SPBIO10_0 P9_6：SPBIO20_0 P9_7：SPBIO30_0
STB	保持用内蔵 RAM へのライト許可および周辺機能へのクロック供給 STBCR2～STBCR12 でクロックの供給および停止制御が可能なすべての周辺機能のクロックを供給

5.2.2　メモリマップ

図 5.3にRZ/A1Hグループのアドレス空間とサンプルコードが動作するGENMAI ボードのメモリマップを示します。

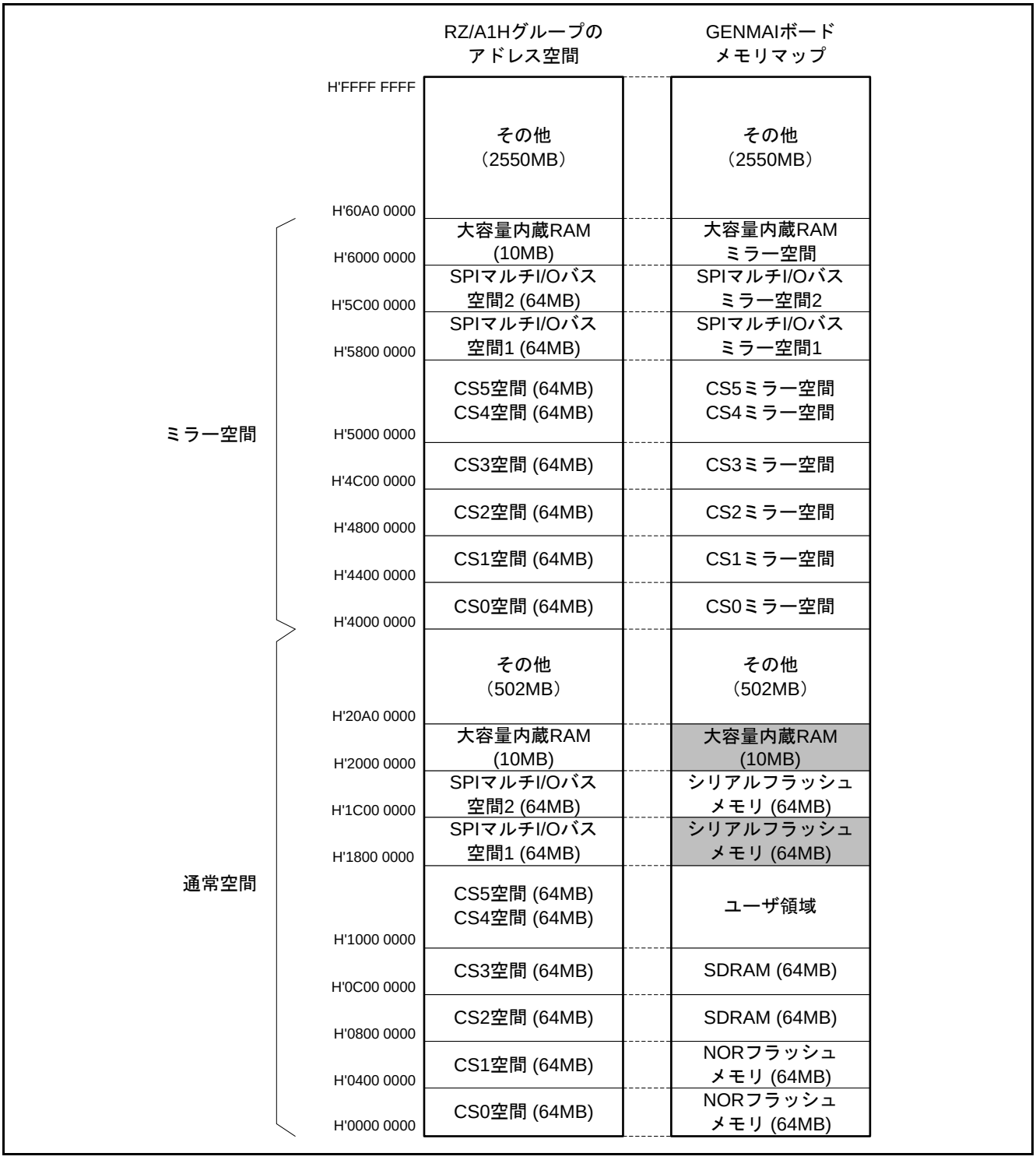

図5.3　RZ/A1Hグループのアドレス空間とGENMAI ボードメモリマップ

5.2.3　サンプルコードのセクション配置

表 5.5にSPIBSC 初期設定プログラムで使用するセクションを、表 5.6および表 5.7にアプリケーションプログラムで使用するセクションを示します。

表5.5　SPIBSC 初期設定プログラムで使用するセクション

領域の名前	内容	タイプ	ロード領域	実行領域
VECTOR_TABLE	例外処理ベクタテーブル	Code	S-FLASH	S-FLASH
CODE_SPIBSC_INIT1	SPIBSC 初期設定プログラム 1 用プログラムコード領域	Code	S-FLASH	LRAM
CODE_IO_REGRW	IO レジスタのリード/ライト関数のプログラムコード領域	Code	S-FLASH	LRAM
CODE_SPIBSC_INIT2	SPIBSC 初期設定プログラム 2 用プログラムコード領域	Code	S-FLASH	LRAM
DATA_SPIBSC_INIT2	SPIBSC 初期設定プログラム 2 用初期値ありデータ領域	RW Data	S-FLASH	LRAM
BSS_SPIBSC_INIT2	SPIBSC 初期設定プログラム 2 用初期値なし領域	ZI Data	ー	LRAM
RESET_HANDLER	リセットハンドラ処理のプログラムコード領域	Code	S-FLASH	S-FLASH
CODE	デフォルトのプログラムコード領域 C ソースでセクション名を定義しない Code タイプのセクションは、すべてこの領域に配置されます	Code	S-FLASH	S-FLASH
SVC_STACK	スタック領域	ZI Data	ー	LRAM

【注】 表中のロード領域および実行領域において、S-FLASH はシリアルフラッシュメモリの領域を、LRAM は大容量内蔵 RAM の領域を表します。

表5.6　アプリケーションプログラムで使用するセクション（1/2）

領域の名前	内容	タイプ	ロード領域	実行領域
VECTOR_TABLE	例外処理ベクタテーブル	Code	S-FLASH	S-FLASH
RESET_HANDLER	リセットハンドラ処理のプログラムコード領域 この領域は以下のセクションから構成されています ・INITCA9CACHE（L1 キャッシュ設定） ・INIT_TTB（MMU 設定） ・RESET_HANDLER（リセットハンドラ）	Code	S-FLASH	S-FLASH
CODE_BASIC_SETUP	保持用内蔵 RAM のライト許可のためのプログラムコード領域	Code	S-FLASH	S-FLASH
InRoot	この領域は C 標準ライブラリなどのルート領域に配置するセクションから構成されています	Code および RO Data	S-FLASH	S-FLASH
CODE_FPU_INIT	NEON および VFP 初期設定のプログラムコード領域 この領域は以下のセクションから構成されています ・CODE_FPU_INIT ・FPU_INIT	Code	S-FLASH	S-FLASH
CODE_RESET	ハードウェア初期設定のプログラムコード領域 この領域は以下のセクションから構成されています ・CODE_RESET（スタートアップ処理） ・INIT_VBAR（ベクタベース設定）	Code	S-FLASH	S-FLASH
CODE	デフォルトのプログラムコード領域 C ソースでセクション名を定義しない Code タイプのセクションは、すべてこの領域に配置されます	Code	S-FLASH	S-FLASH
CONST	デフォルトの定数データ領域 C ソースでセクション名を定義しない RO Data タイプのセクションは、すべてこの領域に配置されます	RO Data	S-FLASH	S-FLASH

表5.7　アプリケーションプログラムで使用するセクション（2/2）

領域の名前	内容	タイプ	ロード領域	実行領域
VECTOR_MIRROR_TABLE	例外処理ベクタテーブル （大容量内蔵 RAM に転送して実行するためのセクション）	Code	S-FLASH	LRAM
CODE_HANDLER_JMPTBL	IRQ 割り込みハンドラのユーザ定義関数のプログラムコード領域	Code	S-FLASH	LRAM
CODE_HANDLER	IRQ 割り込みハンドラのプログラムコード領域 この領域は以下のセクションから構成されています ・CODE_HANDLER ・IRQ_FIQ_HANDLER	Code	S-FLASH	LRAM
CODE_IO_REGRW	IO レジスタのリード/ライト関数のプログラムコード領域	Code	S-FLASH	LRAM
CODE_CACHE_OPERATION	L1 および L2 キャッシュ設定処理のプログラムコード領域（注 3）	Code	S-FLASH	LRAM
DATA_HANDLER_JMPTBL	IRQ 割り込みハンドラのユーザ定義関数の登録テーブルデータ領域	RW Data	S-FLASH	LRAM
ARM_LIB_STACK	アプリケーションスタック領域	ZI Data	―	LRAM
IRQ_STACK	IRQ モードのスタック領域	ZI Data	―	LRAM
FIQ_STACK	FIQ モードのスタック領域	ZI Data	―	LRAM
SVC_STACK	スーパバイザ（SVC）モードのスタック領域	ZI Data	―	LRAM
ABT_STACK	アボート（ABT）モードのスタック領域	ZI Data	―	LRAM
TTB	MMU 変換テーブル領域	ZI Data	―	LRAM
ARM_LIB_HEAP	アプリケーションヒープ領域	ZI Data	―	LRAM
DATA	デフォルトの初期値ありデータ領域 C ソースでセクション名を定義しない RW Data タイプのセクションは、すべてこの領域に配置されます	RW Data	S-FLASH	LRAM
BSS	デフォルトの初期値なしデータ領域 C ソースでセクション名を定義しない ZI Data タイプのセクションは、すべてこの領域に配置されます	ZI Data	―	LRAM

【注】 1. 表中のロード領域および実行領域において、S-FLASH はシリアルフラッシュメモリの領域を、LRAM は大容量内蔵 RAM の領域を表します。
2. セクションの名前は基本的に領域と同じ名前にしていますが、RESET_HANDLER、InRoot、CODE_FPU_INIT、CODE_RESET、CODE、CONST、CODE_HANDLER、DATA、BSS の各領域は複数のセクションから構成されています。領域とセクションについては、ARM コンパイラツールチェーンのマニュアルを参照してください。
3. このセクションは、キャッシュ無効領域に配置する必要があります。

5.3　定数一覧

表 5.8および表 5.9にサンプルコードの SPIBSC 初期設定プログラムで使用する定数を示します。

表5.8　サンプルコードで使用する定数（1/2）

定数名	設定値	内容
SPIBSC_1BIT	0	リードコマンド発行時のビット幅を 1 ビットに設定
SPIBSC_4BIT	2	リードコマンド発行時のビット幅を 4 ビットに設定
SPIBSC_CMNCR_BSZ_SINGLE	0	チャネルに接続しているシリアルフラッシュの個数を 1 個に設定
SPIBSC_CMNCR_BSZ_DUAL	1	チャネルに接続しているシリアルフラッシュの個数を 2 個に設定
SPIBSC_OUTPUT_DISABLE	0	リードコマンド発行時のコマンドを出力しない設定
SPIBSC_OUTPUT_ENABLE	1	リードコマンド発行時のコマンドを出力する設定
SPIBSC_OUTPUT_ADDR_24	0x07	24 ビットのアドレスを出力
SPIBSC_OUTPUT_ADDR_32	0x0f	32 ビットのアドレスを出力
SPIBSC_OUTPUT_OPD_3	0x08	リードコマンド発行時のオプションナルデータイネーブル OPD3 を出力
SPIBSC_OUTPUT_OPD_32	0x0c	リードコマンド発行時のオプションナルデータイネーブル OPD3,OPD2 を出力
SPIBSC_OUTPUT_OPD_321	0x0e	リードコマンド発行時のオプションナルデータイネーブル OPD3,OPD2,OPD1 を出力
SPIBSC_OUTPUT_OPD_3210	0x0f	リードコマンド発行時のオプションナルデータイネーブル OPD3,OPD2,OPD1,OPD0 を出力
SPIBSC_OUTPUT_SPID_8	0x08	SPI 動作モード時に転送データイネーブルを 8（または 16）ビット転送に設定
SPIBSC_OUTPUT_SPID_16	0x0c	SPI 動作モード時に転送データイネーブルを 16（または 32）ビット転送に設定
SPIBSC_OUTPUT_SPID_32	0x0f	SPI 動作モード時に転送データイネーブルを 32（または 64）ビット転送に設定
SPIBSC_SPISSL_NEGATE	0	SPI 動作モード時に転送終了後の SPBSSL 信号状態をネゲートに設定
SPIBSC_SPISSL_KEEP	1	SPI 動作モード時に転送終了後から次アクセス開始まで SPBSSL 信号レベルを保持する設定
SPIBSC_SPIDATA_DISABLE	0	SPI 動作モード時にデータリードしない設定
SPIBSC_SPIDATA_ENABLE	1	SPI 動作モード時にデータリードする設定
SPIBSC_DUMMY_CYC_DISABLE	0	ダミーサイクルを挿入しない設定
SPIBSC_DUMMY_CYC_ENABLE	1	ダミーサイクルを挿入する設定
SPIBSC_DUMMY_1CYC	0	SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換 する時にシリアルフラッシュメモリに出力するダミーサイクル数を 1 に設定
SPIBSC_DUMMY_2CYC	1	SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換 する時にシリアルフラッシュメモリに出力するダミーサイクル数を 2 に設定

表5.9　サンプルコードで使用する定数（2/2）

定数名	設定値	内容
SPIBSC_DUMMY_3CYC	2	SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換 する時にシリアルフラッシュメモリに出力するダミーサイクル数を 3 に設定
SPIBSC_DUMMY_4CYC	3	SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換 する時にシリアルフラッシュメモリに出力するダミーサイクル数を 4 に設定
SPIBSC_DUMMY_5CYC	4	SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換 する時にシリアルフラッシュメモリに出力するダミーサイクル数を 5 に設定
SPIBSC_DUMMY_6CYC	5	SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換 する時にシリアルフラッシュメモリに出力するダミーサイクル数を 6 に設定
SPIBSC_DUMMY_7CYC	6	SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換 する時にシリアルフラッシュメモリに出力するダミーサイクル数を 7 に設定
SPIBSC_DUMMY_8CYC	7	SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換 する時にシリアルフラッシュメモリに出力するダミーサイクル数を 8 に設定
SPIBSC_SDR_TRANS	0	SPI 動作モード時の転送モードを SDR 転送に設定
SPIBSC_DDR_TRANS	1	SPI 動作モード時の転送モードを DDR 転送に設定
SF_REQ_PROTECT	0	シリアルフラッシュメモリ内のレジスタ設定情報を「プロテクト」に設定
SF_REQ_UNPROTECT	1	シリアルフラッシュメモリ内のレジスタ設定情報を「プロテクト解除」に設定
SF_REQ_SERIALMODE	2	シリアルフラッシュメモリ内のレジスタ設定情報を「Serial モード」に設定
SF_REQ_QUADMODE	3	シリアルフラッシュメモリ内のレジスタ設定情報を「Quad モード」に設定

5.4　構造体/共用体一覧

表 5.10～表 5.17にサンプルコードの SPIBSC 初期設定プログラムで使用する構造体を示します。

表5.10　SPIBSC 外部アドレスリード設定構造体（st_spibsc_cfg_t）（1/3）

メンバ名	内容
uint8_t　udef_cmd	リードコマンド • SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するリードコマンドを設定します。 • 本メンバに設定した値をデータリードコマンド設定レジスタ（DRCMR）の CMD[7:0]に設定します。
uint8_t　udef_cmd_width	リードコマンドビット幅 • リードコマンド発行時のビット幅を設定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値をデータリードイネーブル設定レジスタ（DRENR）の CDB[1:0]に設定します。
uint8_t　udef_opd3 uint8_t　udef_opd2 uint8_t　udef_opd1 uint8_t　udef_opd0	オプショナルデータ • SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するオプショナルデータを設定します。 • 本メンバに設定した値をデータリードオプション設定レジスタ（DROPR）の OPD3[7:0]、OPD2[7:0]、OPD1[7:0]、OPD0[7:0]に設定します。
uint8_t　udef_opd_enable	オプションナルデータイネーブル • オプショナルデータを発行するかどうかを選択します。 • 設定可能な値： SPIBSC_OUTPUT_DISABLE：出力しない SPIBSC_OUTPUT_OPD_3：OPD3 を出力 SPIBSC_OUTPUT_OPD_32：OPD3,OPD2 を出力 SPIBSC_OUTPUT_OPD_321：OPD3,OPD2,OPD1 を出力 SPIBSC_OUTPUT_OPD_3210：OPD3,OPD2,OPD1,OPD0 を出力 • 本メンバに設定した値をデータリードイネーブル設定レジスタ（DRENR）の OPDE[3:0]に設定します。
uint8_t　udef_opd_width	オプショナルデータビット幅 • オプショナルデータ発行時のビット幅を設定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値をデータリードイネーブル設定レジスタ（DRENR）の OPDB[1:0]に設定します。

表5.11　SPIBSC 外部アドレスリード設定構造体（st_spibsc_cfg_t）（2/3）

メンバ名	内容
uint8_t　udef_dmycyc_num	ダミーサイクル数 • SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するダミーサイクル数を設定します。 • 設定可能な値： SPIBSC_DUMMY_1CYC：1 サイクル SPIBSC_DUMMY_2CYC：2 サイクル SPIBSC_DUMMY_3CYC：3 サイクル SPIBSC_DUMMY_4CYC：4 サイクル SPIBSC_DUMMY_5CYC：5 サイクル SPIBSC_DUMMY_6CYC：6 サイクル SPIBSC_DUMMY_7CYC：7 サイクル SPIBSC_DUMMY_8CYC：8 サイクル • 本メンバに設定した値をデータリードダミーサイクル設定レジスタ（DRDMCR）の DMCYC[2:0]に設定します。
uint8_t　udef_dmycyc_enable	ダミーサイクルイネーブル • ダミーサイクルを挿入するかを選択します。 • 設定可能な値： SPIBSC_DUMMY_CYC_DISABLE：挿入しない SPIBSC_DUMMY_CYC_ENABLE：挿入する • 本メンバに設定した値をデータリードイネーブル設定レジスタ（DRENR）の DME に設定します。
uint8_t　udef_dmycyc_width	ダミーサイクルビット幅 • ダミーサイクル発行時のビット幅を設定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値をデータリードダミーサイクル設定レジスタ（DRDMCR）の DMDB[1:0]に設定します。
uint8_t　udef_data_width	データリードビット幅 • SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅を設定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値をデータリードイネーブル設定レジスタ（DRENR）の DRDB[1:0]に設定します。

表5.12　SPIBSC 外部アドレスリード設定構造体（st_spibsc_cfg_t）（3/3）

メンバ名	内容
uint8_t　udef_spbr	ビットレート • SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するシリアルクロック（SPBCLK）のビットレートを設定します。 • 設定可能な値： ビットレート分周設定（udef_brdv）と合わせて設定を行ってください。 • 本メンバに設定した値をビットレート設定レジスタ（SPBCR）の SPBR[7:0]に設定します。
uint8_t　udef_brdv	ビットレート分周設定 • SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するシリアルクロック（SPBCLK）のビットレートを設定します。 • 設定可能な値： ビットレート（udef_spbr）と合わせて設定を行ってください。 • 本メンバに設定した値をビットレート設定レジスタ（SPBCR）の BRDV[1:0]に設定します。
uint8_t　udef_addr_width	アドレスビット幅 • SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスのビット幅を設定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値をデータリードイネーブル設定レジスタ（DRENR）の ADB[1:0]に設定します。
uint8_t　udef_addr_mode	アドレスイネーブル • SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 • 設定可能な値： SPIBSC_OUTPUT_ADDR_24：24 ビットのアドレスを出力 SPIBSC_OUTPUT_ADDR_32：32 ビットのアドレスを出力 • 本メンバに設定した値をデータリードイネーブル設定レジスタ（DRENR）の ADE[3:0]に設定します。

表5.13　SPIBSC SPI モード設定構造体（st_spibsc_spimd_reg_t）（1/5）

メンバ名	内容
uint32_t　cdb	• コマンドビット幅 • SPI 動作モード時のコマンドビット幅を指定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の CDB[1:0]に設定します。
uint32_t　ocdb	オプショナルコマンドビット幅 • SPI 動作モード時のオプショナルコマンドビット幅を指定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の OCDB[1:0]に設定します。
uint32_t　adb	アドレスビット幅 • SPI 動作モード時のアドレスビット幅を指定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の ADB[1:0]に設定します。
uint32_t　opdb	オプショナルデータビット幅 • SPI 動作モード時のオプショナルデータビット幅を指定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の OPDB[1:0]に設定します。
uint32_t　spidb	転送データビット幅 • SPI 動作モード時の転送データビット幅を指定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の SPIDB[1:0]に設定します。
uint32_t　cde	SPI 動作モード時にコマンドを出力するかを設定します。 • 設定可能な値： SPIBSC_OUTPUT_DISABLE：出力しない SPIBSC_OUTPUT_ENABLE：出力する • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の CDE に設定します。

表5.14　SPIBSC SPI モード設定構造体（st_spibsc_spimd_reg_t）（2/5）

メンバ名	内容
uint32_t　ocde	オプショナルコマンドイネーブル • SPI 動作モード時にオプショナルコマンドを出力するかを設定します。 • 設定可能な値： SPIBSC_OUTPUT_DISABLE：出力しない SPIBSC_OUTPUT_ENABLE：出力する • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の OCDE に設定します。
uint32_t　ade	アドレスイネーブル • SPI 動作モード時にアドレスを出力するかを設定します。 • 設定可能な値： • SPIBSC_OUTPUT_DISABLE：出力しない SPIBSC_OUTPUT_ADDR_24：ADR[23:0]を出力 SPIBSC_OUTPUT_ADDR_32：ADR[31:0]を出力 • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の ADE[3:0]に設定します。
uint32_t　opde	オプショナルデータイネーブル • SPI 動作モード時にオプショナルデータを出力するかを設定します。 • 設定可能な値： SPIBSC_OUTPUT_DISABLE：出力しない SPIBSC_OUTPUT_OPD_3：OPD3 を出力 SPIBSC_OUTPUT_OPD_32：OPD3,OPD2 を出力 SPIBSC_OUTPUT_OPD_321：OPD3,OPD2,OPD1 を出力 SPIBSC_OUTPUT_OPD_3210：OPD3,OPD2,OPD1,OPD0 を出力 • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の OPDE[3:0]に設定します。
uint32_t　spide	転送データイネーブル • SPI 動作モード時にデータ転送を行うかを設定します。 • 設定可能な値： SPIBSC_OUTPUT_DISABLE：出力しない SPIBSC_OUTPUT_SPID_8：8（または 16）ビット転送 SPIBSC_OUTPUT_SPID_16：16（または 32）ビット転送 SPIBSC_OUTPUT_SPID_32：32（または 64）ビット転送 • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の SPIDE[3:0]に設定します。
uint32_t　sslkp	SPBSSL 信号レベル保持 • SPI 動作モード時に転送終了後の SPBSSL 信号状態を設定します。 • 設定可能な値： SPIBSC_SPISSL_NEGATE：転送終了時にネゲート SPIBSC_SPISSL_KEEP：転送終了後から次アクセス開始まで SPBSSL 信号レベルを保持 • 本メンバに設定した値を SPI モードコントロールレジスタ（SMCR）の SSLKP に設定します。

表5.15　SPIBSC SPI モード設定構造体（st_spibsc_spimd_reg_t）（3/5）

メンバ名	内容
uint32_t　spire	データリードイネーブル • SPI 動作モード時にデータリードするかを設定します。 • 設定可能な値： SPIBSC_SPIDATA_DISABLE：データリードしない SPIBSC_SPIDATA_ENABLE：データリードする • 本メンバに設定した値を SPI モードコントロールレジスタ（SMCR）の SPIRE に設定します。
uint32_t　spiwe	データライトイネーブル • SPI 動作モード時にデータライトするかを設定します。 • 設定可能な値： SPIBSC_SPIDATA_DISABLE：データライトしない SPIBSC_SPIDATA_ENABLE：データライトする • 本メンバに設定した値を SPI モードコントロールレジスタ（SMCR）の SPIWE に設定します。
uint32_t　dme	ダミーサイクルイネーブル • SPI 動作モード時にダミーサイクル挿入するかどうかを設定します。 • 設定可能な値： SPIBSC_DUMMY_CYC_DISABLE：挿入しない SPIBSC_DUMMY_CYC_ENABLE：挿入する • 本メンバに設定した値を SPI モードイネーブル設定レジスタ（SMENR）の DME に設定します。
uint32_t　adder	アドレス DDR イネーブル • SPI 動作モード時に出力するアドレスの SDR/DDR 転送を選択します。 • 設定可能な値： SPIBSC_SDR_TRANS：SDR 転送 • 本メンバに設定した値を SPI モード DDR イネーブルレジスタ（SMDRENR）の ADDRE に設定します。
uint32_t　opdre	オプションデータ DDR イネーブル • SPI 動作モード時に出力するオプションデータの SDR/DDR 転送を選択します。 • 設定可能な値： • SPIBSC_SDR_TRANS：SDR 転送 • 本メンバに設定した値を SPI モード DDR イネーブルレジスタ（SMDRENR）の OPDRE に設定します。
uint32_t　spidre	転送データ DDR イネーブル • SPI 動作モード時に転送するデータの SDR/DDR 転送を選択します。 • 設定可能な値： SPIBSC_SDR_TRANS：SDR 転送 • 本メンバに設定した値を SPI モード DDR イネーブルレジスタ（SMDRENR）の SPIDRE に設定します。

表5.16　SPIBSC SPI モード設定構造体（st_spibsc_spimd_reg_t）（4/5）

メンバ名	内容
uint8_t　dmdb	ダミーサイクルビット幅 • SPI 動作モード時のダミーサイクルのビット幅を設定します。 • 設定可能な値： SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅 • 本メンバに設定した値を SPI モードダミーサイクル設定レジスタ（SMDMCR）の DMDB[1:0]に設定します。
uint8_t　dmcyc	ダミーサイクル数 • SPI 動作モード時のダミーサイクル数を設定します。 • 設定可能な値： SPIBSC_DUMMY_1CYC：1 サイクル SPIBSC_DUMMY_2CYC：2 サイクル SPIBSC_DUMMY_3CYC：3 サイクル SPIBSC_DUMMY_4CYC：4 サイクル SPIBSC_DUMMY_5CYC：5 サイクル SPIBSC_DUMMY_6CYC：6 サイクル SPIBSC_DUMMY_7CYC：7 サイクル SPIBSC_DUMMY_8CYC：8 サイクル • 本メンバに設定した値を SPI モードダミーサイクル設定レジスタ（SMDMCR）の DMCYC[2:0]に設定します。
uint8_t　cmd	コマンド • SPI 動作モード時に出力するコマンドを設定します。 • 本メンバに設定した値を SPI モードコマンド設定レジスタ（SMCMR）の CMD[7:0]に設定します。
uint8_t　ocmd	オプショナルコマンド • SPI 動作モード時に出力するオプショナルコマンドを設定します。 • 本メンバに設定した値を SPI モードコマンド設定レジスタ（SMCMR）の OCMD[7:0]に設定します。
uint32_t　addr	アドレス • SPI 動作モード時に出力するアドレスを設定します。 • 本メンバに設定した値を SPI モードアドレス設定レジスタ（SMADR）の ADR[31:0]に設定します。
uint8_t　opd[4]	オプションデータ • SPI 動作モード時に出力するオプションデータを設定します。 • 本メンバに設定した値を SPI モードオプション設定レジスタ（SMOPR）の OPDn[7:0]に以下のように設定します。 OPD3[7:0]←opd[0] OPD2[7:0]←opd[1] OPD1[7:0]←opd[2] OPD0[7:0]←opd[3]

表5.17　SPIBSC SPI モード設定構造体（st_spibsc_spimd_reg_t）（5/5）

メンバ名	内容
uint32_t　smrdr[2]	リードデータ格納バッファ • SPI 動作モード時にリードしたデータ（SPI モードリードデータレジスタ n（SMRDRn））を以下のように格納します。 SMRDR0→smrdr[0] SMRDR1→smrdr[1]
uint32_t　smwdr[2]	ライトデータ格納バッファ • SPI 動作モード時にライトするデータ（SPI モードライトデータレジスタ n（SMWDRn））を以下のように格納します。 SMWDR0←swrdr[0] SMWDR1←swrdr[1]

5.5　変数一覧

表 5.18にグローバル変数を示します。

表5.18　グローバル変数

型	変数名	内容
st_spibsc_cfg_t	g_spibsc_cfg	SPIBSC 外部アドレスリード設定内容格納変数 • SPIBSC 外部アドレスリード設定内容を格納します。
st_spibsc_spimd_reg_t	g_spibsc_spimd_reg	SPIBSC SPI モード動作設定内容格納変数 • SPIBSC SPI モードで使用する場合に、SPIBSC 設定内容を格納します。 サンプルコードでは、API 関数およびユーザ定義関数内でシリアルフラッシュ制御関数を実行する際の引数として共用で使用しています。

5.6　関数一覧

サンプルコードは、周辺機能を使用するためのインタフェース関数（API 関数）、ユーザシステムの用途に合わせてユーザで準備が必要なユーザ定義関数（API 関数からコールされる関数）、サンプルコードを動作させるために必要なサンプル関数から構成されています。

サンプルコードの SPIBSC 初期設定プログラムの関数について、表 5.19にサンプル関数一覧を、表 5.20にAPI 関数一覧を、表 5.21にユーザ定義関数一覧を示します。アプリケーションプログラムの関数は、「RZ/A1Hグループ 初期設定例」の関数と基本的に同じ仕様ですので、「RZ/A1Hグループ 初期設定例」のアプリケーションノートを参照してください。

表5.19　サンプル関数一覧

関数名	概要
reset_handler	リセットハンドラ
init_spibsc_init1_section	SPIBSC 初期設定プログラム 1 展開関数 SPIBSC 初期設定プログラム 1 を大容量内蔵 RAM で実行するために、ROM 領域（シリアルフラッシュメモリ）から大容量内蔵 RAM に展開します。
spibsc_init1	SPIBSC 初期設定プログラム 1 実行関数 使用するシリアルフラッシュメモリに最適な設定（STEP1）を行います。
init_spibsc_init2_section	SPIBSC 初期設定プログラム 2 展開関数 SPIBSC 初期設定プログラム 2 を大容量内蔵 RAM で実行するために、ROM 領域（シリアルフラッシュメモリ）から大容量内蔵 RAM にコピーします。
spibsc_init2	SPIBSC 初期設定プログラム 2 実行関数 使用するシリアルフラッシュメモリに最適な設定（STEP2）を行います。

表5.20　API 関数一覧

関数名	概要
R_SFLASH_Exmode_Setting	SPIBSC 初期設定関数 SPIBSC にてシリアルフラッシュメモリを制御するために必要な初期設定および SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行います。また初期設定に合わせて、シリアルフラッシュメモリ内のレジスタ設定を行います。初期設定後、外部アドレスリードモードに設定します。
R_SFLASH_WaitTend	SPIBSC データ転送終了待ち関数 SPIBSC より、データ転送が終了するのを待ちます。
R_SFLASH_Exmode	SPIBSC 外部アドレスリードモード設定関数 SPIBSC を外部アドレスリードモードに設定します。
R_SFLASH_Set_Config	SPIBSC 外部アドレスリードモード設定関数 SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行います。
R_SFLASH_SpibscStop	SPIBSC 停止関数 SPIBSC を停止します。
R_SFLASH_Spimode	SPIBSC SPI モード設定関数 SPIBSC を SPI モードに設定します。
R_SFLASH_Exmode_Init	SPIBSC 外部アドレスリードモード初期設定関数 SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行います。初期設定後、外部アドレスリードモードに設定します。
R_SFLASH_Spimode_Init	SPIBSC SPI モード初期設定関数 SPIBSC を SPI モードで使用するために必要な初期設定を行います。初期設定後、SPI モードに設定します。
R_SFLASH_EraseSector	シリアルフラッシュ消去関数 SPIBSC の SPI モードを使用して、シリアルフラッシュの消去を行います。 本関数はサンプルコードでは使用しません。
R_SFLASH_ByteProgram	シリアルフラッシュ書き込み関数 SPIBSC の SPI モードを使用して、シリアルフラッシュにデータを書き込みます。 本関数はサンプルコードでは使用しません。
R_SFLASH_ByteRead	シリアルフラッシュ読み出し関数 SPIBSC の SPI モードを使用して、シリアルフラッシュからデータを読み出します。 本関数はサンプルコードでは使用しません。
R_SFLASH_Spibsc_Transfer	シリアルフラッシュ制御関数 引数に従い、シリアルフラッシュメモリにコマンドを発行します。
R_SFLASH_Ctrl_Protect	シリアルフラッシュメモリプロテクト解除関数 引数に従い、シリアルフラッシュメモリ内のレジスタを設定し、プロテクトを解除します。

表5.21　ユーザ定義関数一覧

関数名	概要
Userdef_SPIBSC_Set_Config	SPIBSC 外部アドレスリード設定関数 使用するシリアルフラッシュメモリに応じて、SPIBSC 外部アドレスリードモードの設定する内容を決定します。サンプルコードは、本関数で設定された内容を基に SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行います。 サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）を使用する場合の SPIBSC 初期設定となっています。
Userdef_SFLASH_Set_Mode	シリアルフラッシュメモリ内レジスタ設定関数 使用するシリアルフラッシュメモリ応じて、SPIBSC を外部アドレスリードモードで使用する場合に必要なシリアルフラッシュメモリ内レジスタの設定を行います。 サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）へのレジスタ設定を行っています。
Userdef_SFLASH_Write_Enable	シリアルフラッシュメモリライト許可関数 使用するシリアルフラッシュメモリ応じて、シリアルフラッシュメモリ内レジスタを設定し、ライト許可に設定します。 サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）へのレジスタ設定を行っています。
Userdef_SFLASH_Busy_Wait	シリアルフラッシュメモリレディー待ち関数 使用するフラッシュメモリに応じて、シリアルフラッシュメモリ内のレジスタを読み出し、シリアルフラッシュメモリがレディー状態に遷移するのを待ちます。 サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）のレディー待ちを行っています。
Userdef_SFLASH_Ctrl_Protect	シリアルフラッシュメモリプロテクト解除関数 使用するフラッシュメモリに応じて、シリアルフラッシュメモリ内のレジスタを設定し、プロテクトを解除します。 サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）のプロテクトを解除しています。

5.7　関数仕様

サンプルコードの SPIBSC 初期設定プログラムの関数仕様を示します。

reset_handler	
概　要	SPIBSC 初期設定プログラムのリセットハンドラ
宣　言	reset_handler
説　明	SPIBSC 初期設定プログラムのエントリ関数です。
引　数	なし
リターン値	なし

init_spibsc_init1_section	
概　要	SPIBSC 初期設定プログラム 1 展開関数
宣　言	void init_spibsc_init1_section (void);
説　明	SPIBSC 初期設定プログラム 1 を大容量内蔵 RAM で実行するために、ROM 領域（シリアルフラッシュメモリ）から大容量内蔵 RAM に展開します。
引　数	なし
リターン値	なし

spibsc_init1	
概　要	SPIBSC 初期設定プログラム 1
宣　言	void spibsc_init1 (void);
説　明	使用するシリアルフラッシュメモリに最適な設定（STEP1）を行います。 • SSL の遅延サイクル数を設定 • SPBCLK 動作周波数を変更：Bφ/8→Bφ/4 • SPIBSC のリードキャッシュの有効化
引　数	なし
リターン値	なし

init_spibsc_init2_section	
概　要	SPIBSC 初期設定プログラム 2 のセクション初期化関数
宣　言	void init_spibsc_init2_section (void);
説　明	SPIBSC 初期設定プログラム 2 を大容量内蔵 RAM に転送します。
引　数	なし
リターン値	なし

spibsc_init2	
概　要	SPIBSC 初期設定プログラム 2
宣　言	void spibsc_init2 (void);
説　明	使用するシリアルフラッシュメモリに最適な設定（STEP2）を行います。 ● SPBCLK 動作周波数を変更：Bφ/4→Bφ/2 ● リードコマンドの変更：03H→EBH ● シリアルフラッシュメモリ内レジスタの設定 ステータスレジスタ：QUAD ビットに 1 を設定 コンフィグレーションレジスタ：レイテンシコードに 00H を設定
引　数	なし
リターン値	なし

R_SFLASH_Exmode_Setting		
概　要	SPIBSC 初期設定関数	
宣　言	int32_t R_R_SFLASH_Exmode_Setting (uint32_t ch_no, uint32_t dual, st_spibsc_cfg_t *spibsccfg);	
説　明	SPIBSC にてシリアルフラッシュメモリを制御するために必要な初期設定および SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行います。また初期設定に合わせて、シリアルフラッシュメモリ内のレジスタ設定を行います。初期設定後、外部アドレスリードモードに設定します。 本関数内で SPIBSC 外部アドレスモード初期設定関数（R_SFLASH_Exmode_Init）を実行します。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	uint32_t dual	チャネルに接続しているシリアルフラッシュの個数 1 個の場合：SPIBSC_CMNCR_BSZ_SINGLE (0) 2 個の場合：SPIBSC_CMNCR_BSZ_DUAL (1)
	st_spibsc_cfg_t *spibsccfg	SPIBSC 外部アドレスリード設定 設定内容は表 5.10～表 5.12を参照してください。
リターン値	0：正常終了 -1: エラー	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_Set_Config		
概　要	SPIBSC 外部アドレスリード設定関数	
宣　言	void R_SFLASH_Set_Config(uint32_t ch_no, st_spibsc_cfg_t *spibsccfg);	
説　明	使用するシリアルフラッシュメモリに応じて、SPIBSC を外部アドレスリードモードで使用するための設定内容を決定します。 本関数内で、ユーザ定義関数（SPIBSC 外部アドレスリード設定関数 Userdef_SPIBSC_Set_Config）を実行します。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	st_spibsc_cfg_t *spibsccfg	SPIBSC 外部アドレスリード設定 設定内容は表 5.10～表 5.12を参照してください。
リターン値	0：正常終了	
	-1: エラー	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_SpibscStop		
概　要	SPIBSC 停止関数	
宣　言	int32_t R_SFLASH_SpibscStop(uint32_t ch_no);	
説　明	SPIBSC を停止します。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
リターン値	なし	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_WaitTend		
概　要	SPIBSC データ転送終了待ち関数	
宣　言	void R_SFLASH_WaitTend(uint32_t ch_no);	
説　明	SPIBSC より、データ転送が終了するのを待ちます。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
リターン値	なし	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_Exmode		
概　要	SPIBSC 外部アドレスモード設定関数	
宣　言	int32_t R_SFLASH_Exmode(uint32_t ch_no);	
説　明	SPIBSC を外部アドレスリードモードに設定します。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
リターン値	0：設定成功	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_Spimode		
概　要	SPIBSC SPI モード設定関数	
宣　言	int32_t R_SFLASH_Spimode(uint32_t ch_no);	
説　明	SPIBSC を SPI モードに設定します。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
リターン値	0：設定成功	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_Exmode_Init		
概　要	SPIBSC 外部アドレスモード初期設定関数	
宣　言	int32_t R_SFLASH_Exmode_Init(uint32_t ch_no, uint32_t dual, st_spibsc_cfg_t *spibsccfg)	
説　明	SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行います。初期設定後、外部アドレスリードモードに設定します。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	uint32_t dual	チャネルに接続しているシリアルフラッシュの個数 1 個の場合：SPIBSC_CMNCR_BSZ_SINGLE (0) 2 個の場合：SPIBSC_CMNCR_BSZ_DUAL (1)
	st_spibsc_cfg_t *spibsccfg	SPIBSC 外部アドレスリード設定 設定内容は表 5.10～表 5.12を参照してください。
リターン値	0：正常終了	
	-1: エラー	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_Spimode_Init		
概　要	SPIBSC SPI モード初期設定関数	
宣　言	int32_t R_SFLASH_Spimode_Init(uint32_t ch_no, uint32_t dual, uint8_t data_width, uint8_t spbr, uint8_t brdv, uint8_t addr_mode);	
説　明	SPIBSC を SPI モードで使用するために必要な初期設定を行います。初期設定後、SPI モードに設定します。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	int32_t bsz	チャネルに接続しているシリアルフラッシュの個数 1 個の場合：1 2 個の場合：2
	uint8_t data_width	データリードビット幅 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅
	uint8_t spbr	ビットレート ビットレート分周設定（brdv）と合わせてシリアルクロック（SPBCLK）のビットレートを設定します。
	uint8_t brdv	ビットレート分周設定 ビットレート分周設定（spbr）と合わせてシリアルクロック（SPBCLK）のビットレートを設定します。
	uint8_t addr_mode	アドレスモード設定 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 SPIBSC_OUTPUT_ADDR_24：24 ビットアドレス出力 SPIBSC_OUTPUT_ADDR_32：32 ビットアドレス出力
リターン値	0：設定成功	
	-1：設定失敗	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_EraseSector

概　要	シリアルフラッシュ消去関数	
宣　言	int32_t R_SFLASH_EraseSector(uint32_t addr, uint32_t ch_no, uint32_t dual, uint8_t data_width, uint8_t addr_mode);	
説　明	SPIBSC の SPI モードを使用して、シリアルフラッシュの消去を行います。	
引　数	uint32_t addr	消去するシリアルフラッシュメモリのアドレス
	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	uint32_t dual	チャネルに接続しているシリアルフラッシュの個数 SPIBSC_CMNCR_BSZ_SINGLE：1 個 SPIBSC_CMNCR_BSZ_DUAL　：2 個
	uint8_t data_width	データリードビット幅 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅
	uint8_t addr_mode	アドレスモード設定 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 SPIBSC_OUTPUT_ADDR_24：24 ビットアドレス出力 SPIBSC_OUTPUT_ADDR_32：32 ビットアドレス出力
リターン値	0：設定成功	
	-1：設定失敗	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_ByteProgram		
概　要	シリアルフラッシュ書き込み関数	
宣　言	int32_t R_SFLASH_ByteProgram(uint32_t addr, uint8_t *buf, int32_t size, uint32_t ch_no, uint32_t dual, uint8_t data_width, uint8_t addr_mode);	
説　明	SPIBSC の SPI モードを使用して、シリアルフラッシュにデータを書き込みます。	
引　数	uint32_t addr	書き込むシリアルフラッシュメモリのアドレス
	uint8_t *buf	書き込みデータ格納バッファ
	int32_t size	書き込みサイズ（バイト単位）
	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	uint32_t dual	チャネルに接続しているシリアルフラッシュの個数 SPIBSC_CMNCR_BSZ_SINGLE：1 個 SPIBSC_CMNCR_BSZ_DUAL　：2 個
	uint8_t data_width	データリードビット幅 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅
	uint8_t addr_mode	アドレスモード設定 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 SPIBSC_OUTPUT_ADDR_24：24 ビットアドレス出力 SPIBSC_OUTPUT_ADDR_32：32 ビットアドレス出力
リターン値	0：設定成功	
	-1：設定失敗	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_ByteRead		
概　要	シリアルフラッシュ読み出し関数	
宣　言	int32_t R_SFLASH_ByteRead(uint32_t addr, uint8_t *buf, int32_t size, uint32_t ch_no, uint8_t data_width, uint8_t addr_mode);	
説　明	SPIBSC の SPI モードを使用して、シリアルフラッシュからデータを読み出します。	
引　数	uint32_t addr	読み出すシリアルフラッシュメモリのアドレス
	uint8_t *buf	読み出しデータ格納バッファ
	int32_t size	読み出しサイズ（バイト単位）
	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	uint32_t dual	チャネルに接続しているシリアルフラッシュの個数 SPIBSC_CMNCR_BSZ_SINGLE：1 個 SPIBSC_CMNCR_BSZ_DUAL　：2 個
	uint8_t data_width	データリードビット幅 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅
	uint8_t addr_mode	アドレスモード設定 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 SPIBSC_OUTPUT_ADDR_24：24 ビットアドレス出力 SPIBSC_OUTPUT_ADDR_32：32 ビットアドレス出力
リターン値	0：設定成功	
	-1：設定失敗	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_Spibsc_Transfer		
概　要	シリアルフラッシュ制御関数	
宣　言	int32_t R_SFLASH_Spibsc_Transfer(uint32_t ch_no, st_spibsc_spimd_reg_t * regset);	
説　明	SPIBSC の SPI モードを使用して、シリアルフラッシュからデータを読み出します。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	st_spibsc_spimd_reg_t * regset	SPIBSC SPI モード設定 設定内容は表 5.13～表 5.17を参照してください。
リターン値	0：設定成功	
	-1：設定失敗	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

R_SFLASH_Ctrl_Protect

概　要	シリアルフラッシュメモリプロテクト解除関数	
宣　言	int32_t R_SFLASH_Ctrl_Protect(en_sf_req_t req, uint32_t ch_no, uint32_t dual, uint8_t data_width);	
説　明	シリアルフラッシュメモリ内のレジスタを設定し、プロテクトを解除します。	
引　数	en_sf_req_t req	レジスタ設定情報 SF_REQ_PROTECT：プロテクトに設定 SF_REQ_UNPROTECT：プロテクト解除に設定
	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	uint32_t dual	チャネルに接続しているシリアルフラッシュの個数 SPIBSC_CMNCR_BSZ_SINGLE：1 個 SPIBSC_CMNCR_BSZ_DUAL　：2 個
	uint8_t data_width	データリードビット幅 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅
リターン値	なし	
備考	シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 を指定してください。	

Userdef_SPIBSC_Set_Config

概　要	SPIBSC 外部アドレスリード設定関数	
宣　言	void Userdef_SPIBSC_Set_Config(uint32_t ch_no, st_spibsc_cfg_t *spibsccfg);	
説　明	使用するシリアルフラッシュメモリに応じて、SPIBSC 外部アドレスリードモードの設定する内容を決定します。本関数にて、引数 spibsccfg にて指定された領域に SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行ってください。 引数 spibsccfg に設定する内容については、表 5.10-表 5.12および6.3.1リードコマンド波形の変更を参照してください。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	st_spibsc_cfg_t *spibsccfg	SPIBSC 外部アドレスリード設定 設定内容は表 5.10～表 5.12を参照してください。
リターン値	なし	
備考	• サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）を使用する場合の SPIBSC 初期設定となっています。 • シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 が指定されています。	

Userdef_SFLASH_Set_Mode		
概　要	シリアルフラッシュメモリ内レジスタ設定関数	
宣　言	int32_t Userdef_SFLASH_Set_Mode(uint32_t ch_no, uint32_t dual, en_sf_req_t req, uint8_t data_width, uint8_t addr_mode);	
説　明	本関数内で使用するシリアルフラッシュメモリ応じて、SPIBSC を外部アドレスリードモードで使用する場合に必要なシリアルフラッシュメモリ内レジスタを設定する処理を実装してください。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	uint32_t dual	チャネルに接続しているシリアルフラッシュの個数 SPIBSC_CMNCR_BSZ_SINGLE：1 個 SPIBSC_CMNCR_BSZ_DUAL　：2 個
	en_sf_req_t req	レジスタ設定情報 SF_REQ_SERIALMODE：Serial モードに設定 SF_REQ_QUADMODE：Quad モードに設定
	uint8_t data_width	データリードビット幅 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅
	uint8_t addr_mode	アドレスモード設定 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 SPIBSC_OUTPUT_ADDR_24：24 ビットアドレス出力 SPIBSC_OUTPUT_ADDR_32：32 ビットアドレス出力
リターン値	0：設定成功 -1：設定失敗	
備考	• サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）へのレジスタ設定を行っています。 • シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 が指定されています。	

Userdef_SFLASH_Write_Enable	
概　要	シリアルフラッシュメモリライト許可関数
宣　言	int32_t Userdef_SFLASH_Write_Enable(uint32_t ch_no);
説　明	本関数内で使用するシリアルフラッシュメモリ応じて、シリアルフラッシュメモリ内レジスタを設定し、ライト許可に設定する処理を実装してください。
引　数	uint32_t ch_no　　SPIBSC のチャネル番号（0 または 1）
リターン値	0：設定成功 -1：設定失敗
備考	• サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）へのレジスタ設定を行っています。 • シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 が指定されています。

Userdef_SFLASH_Busy_Wait	
概　要	シリアルフラッシュメモリレディー待ち関数
宣　言	int32_t Userdef_SFLASH_Busy_Wait(uint32_t ch_no, uint32_t dual, uint8_t data_width);
説　明	本関数内で使用するフラッシュメモリに応じて、シリアルフラッシュメモリ内のレジスタを読み出し、シリアルフラッシュメモリがレディー状態に遷移する処理を実装してください。
引　数	uint32_t ch_no　　SPIBSC のチャネル番号（0 または 1） uint32_t dual　　チャネルに接続しているシリアルフラッシュの個数 SPIBSC_CMNCR_BSZ_SINGLE：1 個 SPIBSC_CMNCR_BSZ_DUAL　：2 個 uint8_t data_width　　データリードビット幅 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅
リターン値	なし
備考	• サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）のレディー待ちを行っています。 • シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 が指定されています。

Userdef_SFLASH_Ctrl_Protect		
概　要	シリアルフラッシュメモリプロテクト解除関数	
宣　言	int32_t Userdef_SFLASH_Ctrl_Protect(en_sf_req_t req, uint32_t ch_no, uint32_t dual, uint8_t data_width);	
説　明	本関数内で使用するフラッシュメモリに応じて、シリアルフラッシュメモリ内のレジスタを設定し、プロテクトを解除する処理を実装してください。	
引　数	en_sf_req_t req	レジスタ設定情報 SF_REQ_PROTECT：プロテクトに設定 SF_REQ_UNPROTECT：プロテクト解除に設定
	uint32_t ch_no	SPIBSC のチャネル番号（0 または 1）
	uint32_t dual	チャネルに接続しているシリアルフラッシュの個数 SPIBSC_CMNCR_BSZ_SINGLE：1 個 SPIBSC_CMNCR_BSZ_DUAL　：2 個
	uint8_t data_width	データリードビット幅 SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT：1 ビット幅 SPIBSC_4BIT：4 ビット幅
リターン値	なし	
備考	• サンプルコードでは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）のプロテクトを解除しています。 • シリアルフラッシュメモリからブートする場合は、第 1 引数の ch_no には 0 が指定されています。	

5.8　フローチャート

5.8.1　SPIBSC 初期設定プログラム（全体）

図 5.4にSPIBSC 初期設定プログラム（全体）のフローチャートを示します。

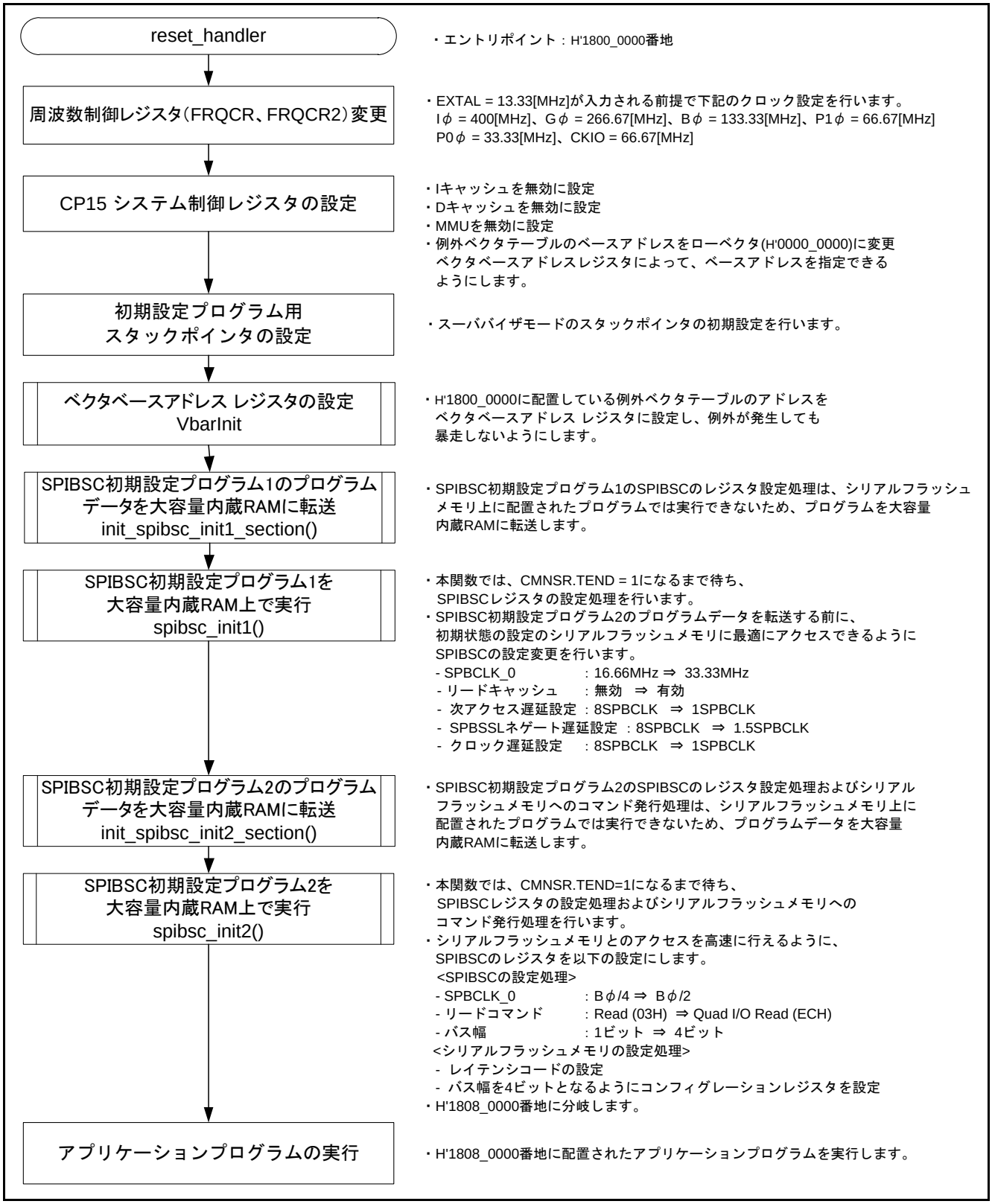

図5.4　SPIBSC 初期設定プログラム（全体）のフローチャート

5.8.2　SPIBSC 初期設定プログラム 1（STEP1）

SPIBSC 初期設定プログラム 1 では、通信時に挿入される各遅延サイクル数を変更し、SPI クロック周波数を Bφ/4 に変更し、リードキャッシュを有効する処理を行います。

SPIBSC 初期設定プログラム 1 の処理は、SPIBSC のレジスタを変更するため、SPI マルチ I/O バス空間（チャネル 0）に配置されたプログラムでは実行できないため、大容量内蔵 RAM に展開し、大容量内蔵 RAM 上で実行します。

図 5.5にSPIBSC 初期設定プログラム 1 のフローチャートを示します。

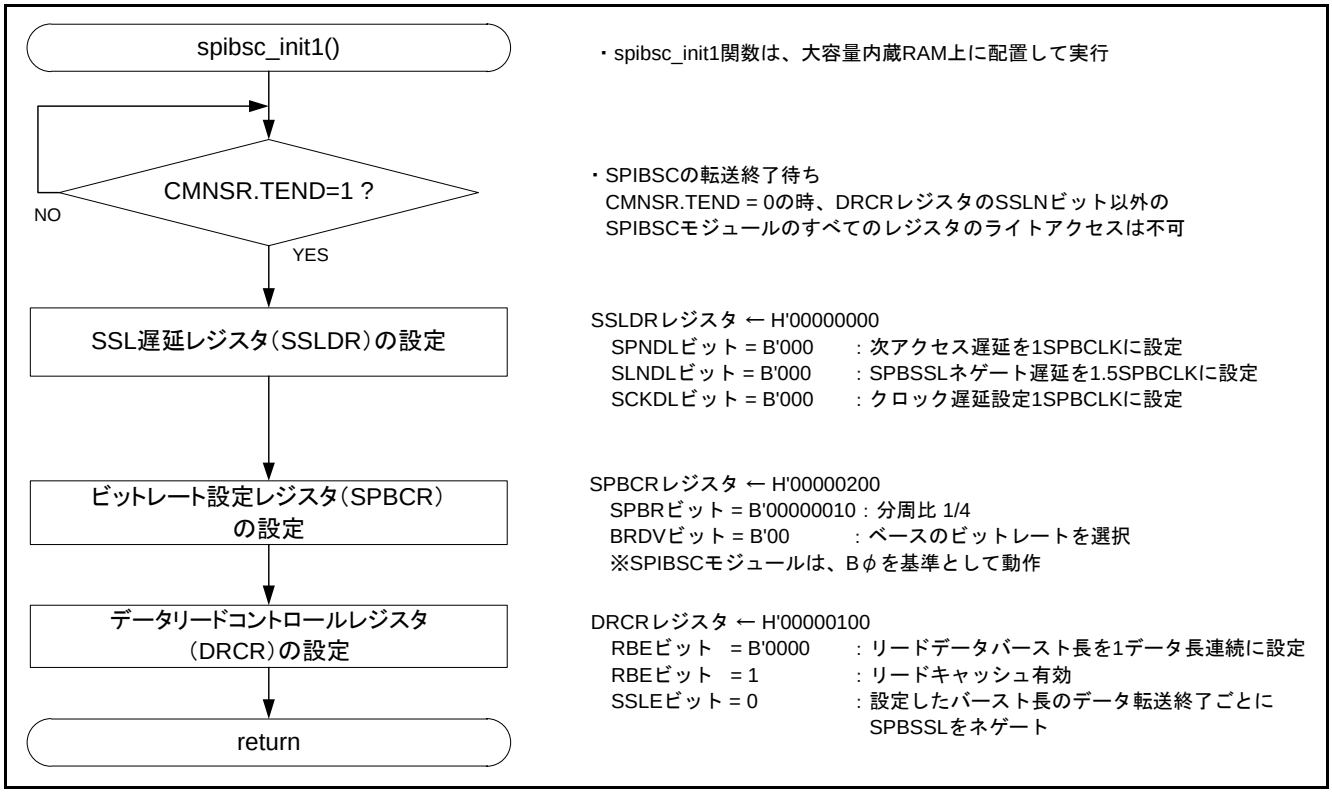

図5.5　SPIBSC 初期設定プログラム 1 のフローチャート

5.8.3　SPIBSC 初期設定プログラム 2（STEP2）

SPIBSC 初期設定プログラム 2 では、シリアルフラッシュメモリが Quad モードで動作できるように、シリアルフラッシュメモリ内のレジスタ（Configuration Register 1 の Quad ビット）の設定処理を行います。シリアルフラッシュメモリ内のレジスタ設定後、SPIBSC を外部アドレス空間リードモードで使用する場合にシリアルフラッシュメモリに発行するリードコマンドを Quad I/O Read（3-byte Address EBH または 4-byte Address ECH）に設定し、SPI クロック周波数を Bφ/2 に変更します。

SPIBSC 初期設定プログラム 2 の処理は、SPIBSC のレジスタを変更するため、SPI マルチ I/O バス空間（チャネル 0）に配置されたプログラムでは実行できませんので、大容量内蔵 RAM に展開し、大容量内蔵 RAM 上で実行します。

図 5.6にSPIBSC 初期設定プログラム 2 のフローチャートを示します。

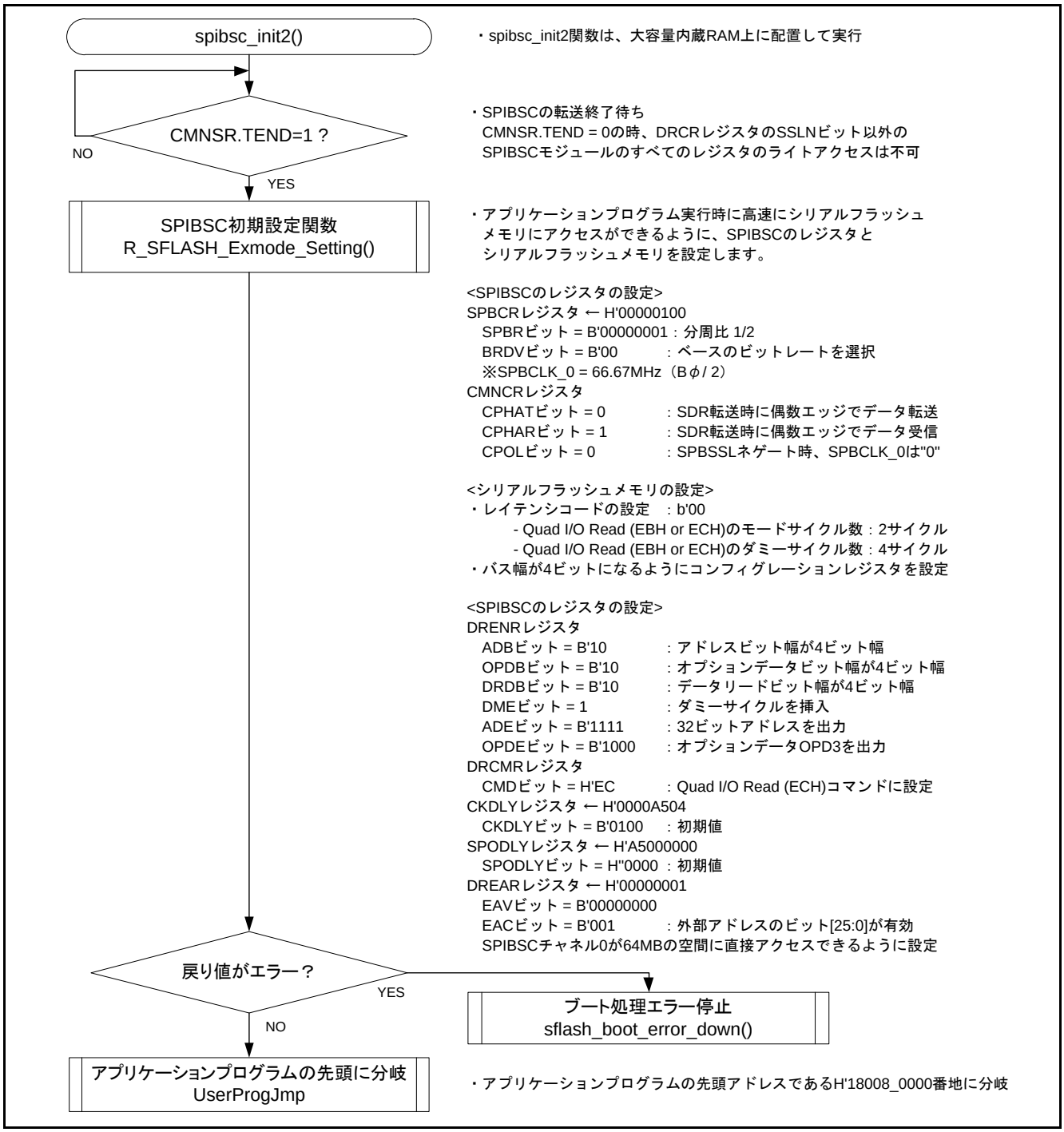

図5.6　SPIBSC 初期設定プログラム 2 のフローチャート

6.　応用例

本章では、サンプルコードの応用例として、お客様が使用するシリアルフラッシュメモリに応じたサンプルコードの変更方法について説明します。

6.1　サンプルコードの条件

サンプルコードは、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）を表 6.1に示す条件で使用する場合に最適な設定を行っています。

これらの条件を変更する場合のサンプルコードの変更方法について説明します。

表6.1　サンプルコードの条件

条件	設定内容	備考
シリアルフラッシュメモリ	Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）	―
データバス幅	4 ビット	データリード時のビット幅
アドレスバイト数	4 バイト	アドレス指定時に発行するバイト数
SPIBSC チャネル 0 に接続されたシリアルフラッシュメモリ数	1 個	―

6.2　シリアルフラッシュメモリを変更しない場合のサンプルコード変更方法

表 6.2に使用するシリアルフラッシュメモリを変更しない（"S25FL512S"を使用する）場合のサンプルコードの変更方法を示します。

表6.2　シリアルフラッシュを変更しない場合のサンプルコード変更方法

条件	変更内容	変更方法
データバス幅	1 ビット	マクロ定義"SPIBSC_BUS_WITDH"（注 1）に（1）を定義
	4 ビット	マクロ定義"SPIBSC_BUS_WITDH"（注 1）に（4）を定義
アドレスバイト数	3 バイト	マクロ定義"SPIBSC_OUTPUT_ADDR"（注 2）に（SPIBSC_OUTPUT_ADDR_24）を定義
	4 バイト	マクロ定義"SPIBSC_OUTPUT_ADDR"（注 2）に（SPIBSC_OUTPUT_ADDR_32）を定義
SPIBSC チャネル 0 に接続されたシリアルフラッシュメモリ数	1 個	マクロ定義"DEF_SPIBSC_DUAL_MODE"（注 3）に（SPIBSC_CMNCR_BSZ_SINGLE）定義
	2 個	マクロ定義"DEF_SPIBSC_DUAL_MODE"（注 3）に（SPIBSC_CMNCR_BSZ_DUAL）定義

【注】 1. マクロ定義"SPIBSC_BUS_WITDH"は spibsc_ioset_userdef.c ファイル内に定義されています。
2. マクロ定義"SPIBSC_OUTPUT_ADDR"は spibsc_ioset_userdef.c ファイル内に定義されています。
3. マクロ定義"DEF_SPIBSC_DUAL_MODE"は spibsc_init2.c ファイル内に定義されています。

6.3　シリアルフラッシュメモリを変更する場合のサンプルコード変更方法

シリアルフラッシュメモリを変更する場合、使用するシリアルフラッシュメモリの仕様に合わせてサンプルコードを変更する必要があります。

表 6.3にサンプルコードの変更のポイント示します。

表6.3　サンプルコードの変更のポイント

変更のポイント	内容
リードコマンド波形	外部アドレス空間リードモードにより、SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する際に、シリアルフラッシュメモリに出力する信号を、使用するシリアルフラッシュメモリのリードコマンドに合わせて変更します。
シリアルフラッシュメモリのレジスタ設定	使用するシリアルフラッシュメモリ応じて、SPIBSC を外部アドレスリードモードで使用する場合に必要なシリアルフラッシュメモリのレジスタの設定を行います。
シリアルフラッシュメモリライト許可	使用するシリアルフラッシュメモリ応じて、シリアルフラッシュメモリのレジスタを設定し、ライト許可に設定します（注 1）。
シリアルフラッシュメモリレディー待ち	使用するフラッシュメモリに応じて、シリアルフラッシュメモリのレジスタを読み出し、シリアルフラッシュメモリがレディー状態に遷移するのを待ちます。
シリアルフラッシュメモリプロテクト解除	使用するフラッシュメモリに応じて、シリアルフラッシュメモリのレジスタを設定し、プロテクトを解除します（注 2）。

【注】 1. シリアルフラッシュメモリにより、シリアルフラッシュメモリのレジスタを設定するために、ライト許可が必要な場合があります。
2. シリアルフラッシュメモリにより、シリアルフラッシュメモリ内のレジスタを設定するために、プロテクト解除が必要な場合があります。

6.3.1　リードコマンド波形の変更

外部アドレス空間リードモードにより、SPI マルチ I/O バス空間（チャネル 0）へのリードを SPI 通信に変換する際にシリアルフラッシュメモリに出力する信号を使用するシリアルフラッシュメモリのリードコマンドに合わせて変更します。

SPIBSC は、SPIBSC 制御レジスタ設定より、外部アドレス空間リードモードにてシリアルフラッシュメモリに出力する信号を変更することが可能です。

サンプルコードでは、SPIBSC 制御レジスタに設定する値をグローバル変数（SPIBSC 外部アドレスリード設定内容格納変数：g_spibsc_cfg）にて変更することが可能です。g_spibsc_cfg の設定は SPIBSC 外部アドレスリード設定関数（R_SFLASH_Set_Config）内から実行されるユーザ定義関数（SPIBSC 外部アドレスリード設定関数：Userdef_SPIBSC_Set_Config）にて行っています。

図 6.1にSPIBSC 制御レジスタと SPIBSC 外部アドレスリード時にシリアルフラッシュメモリに出力される波形の関係を、にサンプルコードの SPIBSC 制御レジスタ設定値を示します。

本設定例を参考に、使用するシリアルフラッシュメモリのリードコマンドに合わせて g_spibsc_cfg の設定を行ってください。

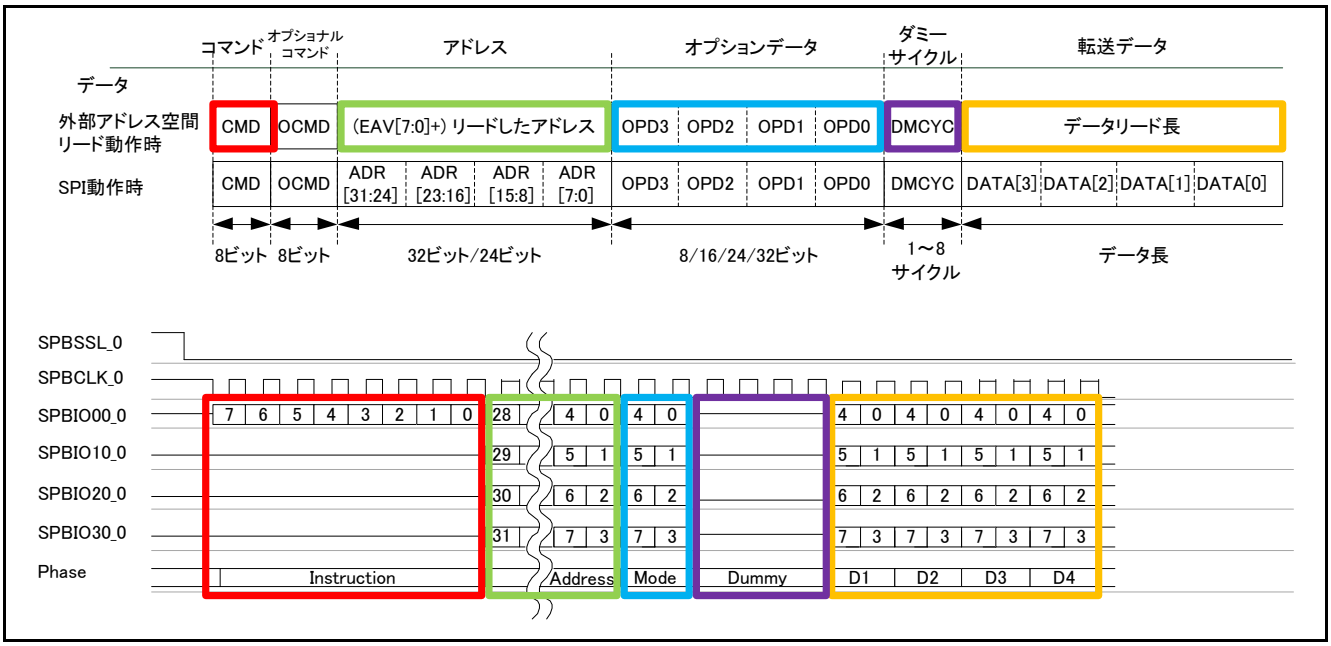

図6.1　SPIBSC 制御レジスタと SPIBSC 外部アドレスリード時にシリアルフラッシュメモリに出力される波形の関係

6.3.2　シリアルフラッシュメモリ内レジスタ設定

6.3.1リードコマンド波形の変更で設定したリードコマンド波形により、シリアルフラッシュメモリからリードする場合、シリアルフラッシュメモリ内のレジスタを設定する必要があります。

サンプルコードでは、ユーザ定義関数（シリアルフラッシュメモリ内レジスタ設定関数：Userdef_SFLASH_Set_Mode）に処理を実装しており、Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）内の Configuration Register 1 内の QUAD ビットを 1（=Quad）に設定しています。

なおシリアルフラッシュメモリ内のレジスタの設定にはシリアルフラッシュ制御関数（R_SFLASH_Spibsc_Transfer）を使用します。また Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）内のレジスタを設定する場合、Write Enable Command（WREN）より、Write Enable Latch (WEL) bit を 1 に設定する必要があるため、Configuration Register 1 設定前に Write Enable Command（WREN）を発行しています。サンプルコードではユーザ定義関数（シリアルフラッシュメモリライト許可関数：Userdef_SFLASH_Write_Enable）にて、Write Enable Command（WREN）発行処理を実装しています。

図 6.2にシリアルフラッシュメモリ内レジスタ設定フロー示します。

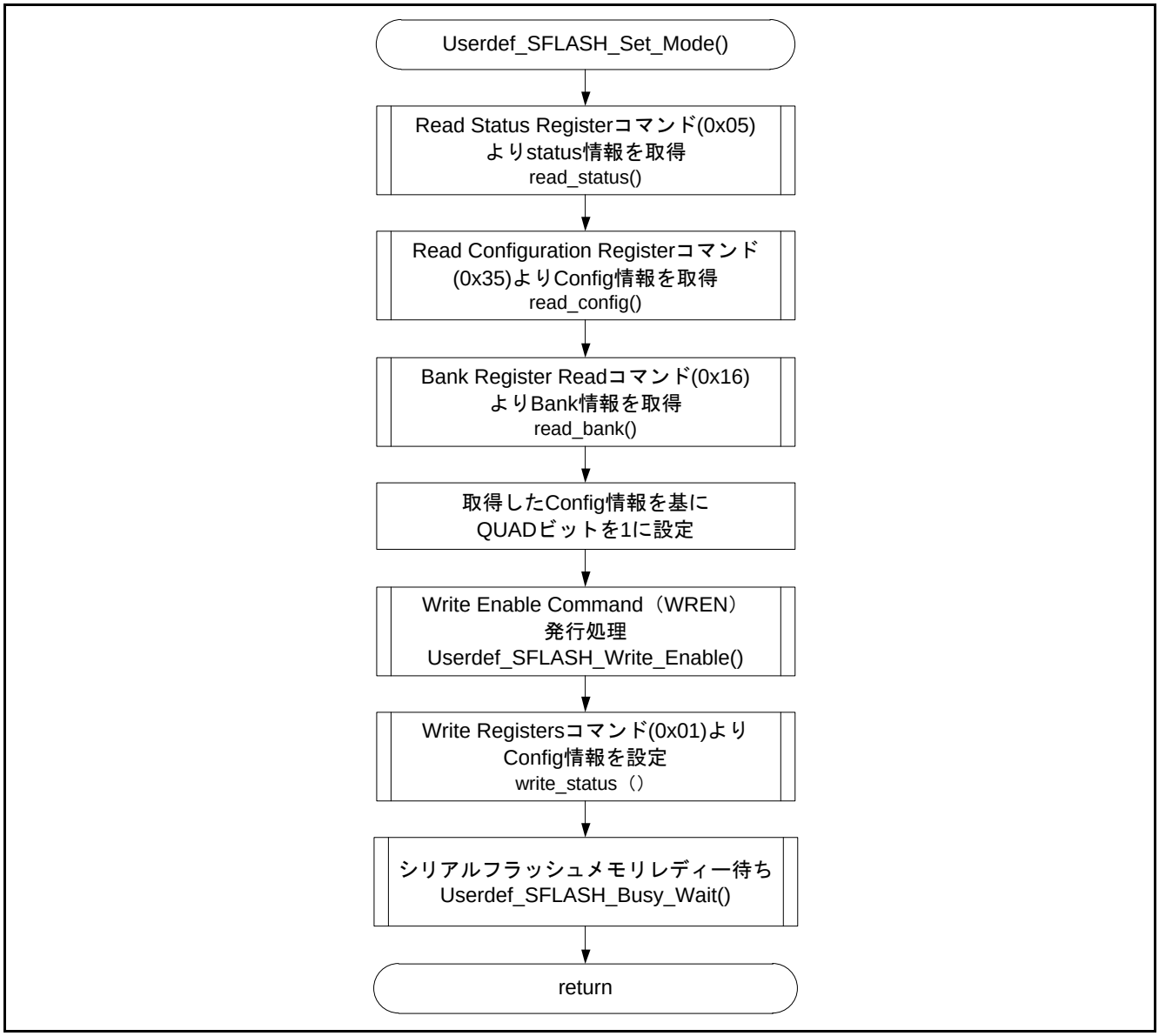

図6.2　シリアルフラッシュメモリ内レジスタ設定フロー

6.3.3　シリアルフラッシュメモリライト許可

6.3.2シリアルフラッシュメモリ内レジスタ設定より、シリアルフラッシュメモリ内レジスタを設定する際に必要なライト許可処理をユーザ定義関数（シリアルフラッシュメモリライト許可関数：Userdef_SFLASH_Write_Enable）に実装してください。

サンプルコードでは、シリアルフラッシュ制御関数（R_SFLASH_Spibsc_Transfer）を使用して、Write Enable Command（WREN）発行処理を実装しています。

図 6.3にシリアルフラッシュメモリライト許可フローを示します。

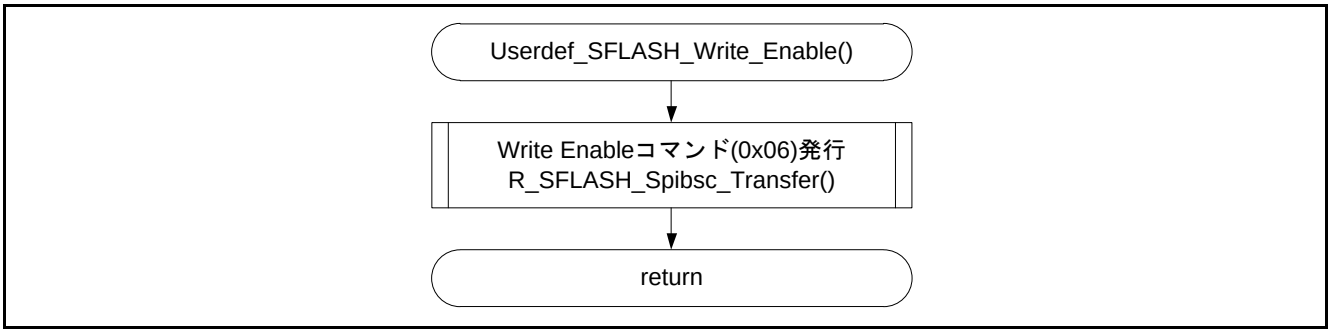

図6.3　シリアルフラッシュメモリライト許可フロー

6.3.4　シリアルフラッシュメモリレディー待ち

シリアルフラッシュメモリに対して、ステータスレジスタへの書き込み(Write Regidter)、書き込みコマンド（Quad Page Program）または消去コマンド（Sector Erase）を発行した場合、シリアルフラッシュメモリはビジー状態に遷移します。ビジー状態からレディー状態への遷移を待つ処理をユーザ定義関数（シリアルフラッシュメモリレディー待ち関数：Userdef_SFLASH_Busy_Wait）に実装してください。

Spansion 社製シリアルフラッシュメモリ（型名 : S25FL512S）では、レディー状態への遷移をシリアルフラッシュメモリ内のレジスタ（Status Register 1）にて確認することができます。サンプルコードでは、シリアルフラッシュ制御関数（R_SFLASH_Spibsc_Transfer）を使用して、Read Status Register-1 発行して Status Register-1 を読み出し、レディー状態への遷移を確認する処理を実装しています。

図 6.4にシリアルフラッシュメモリレディー待ちフローを示します。

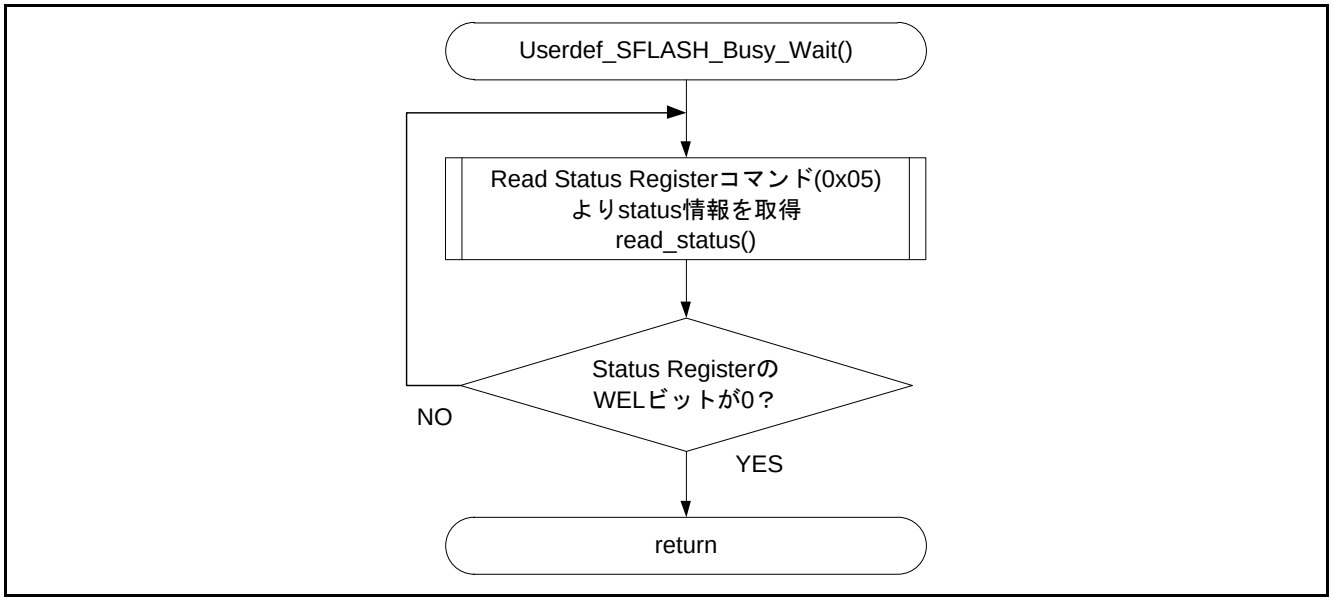

図6.4　シリアルフラッシュメモリレディー待ちフロー

6.3.5　シリアルフラッシュメモリプロテクト解除

シリアルフラッシュメモリ仕様より、シリアルフラッシュメモリ内のデータを変更する場合、シリアルフラッシュメモリ内のレジスタを操作し、プロテクトを解除する必要があります。

Spansion 社製シリアルフラッシュメモリ（型名：S25FL512S）では、シリアルフラッシュメモリがプロテクト状態にある場合、シリアルフラッシュメモリへの書き込みおよび消去を実行することができません。プロテクトを解除するためには、Configuration Register 1 内の FREEZE ビットを 0 に設定した後に、Status Register-1 内の Block Protection（BP2, BP1, BP0）ビットに 0 を設定する必要があります。サンプルコードでは、シリアルフラッシュ制御関数（R_SFLASH_Spibsc_Transfer）を使用して、これらの処理を実装しています。

図 6.5にシリアルフラッシュメモリプロテクト解除フローを示します。

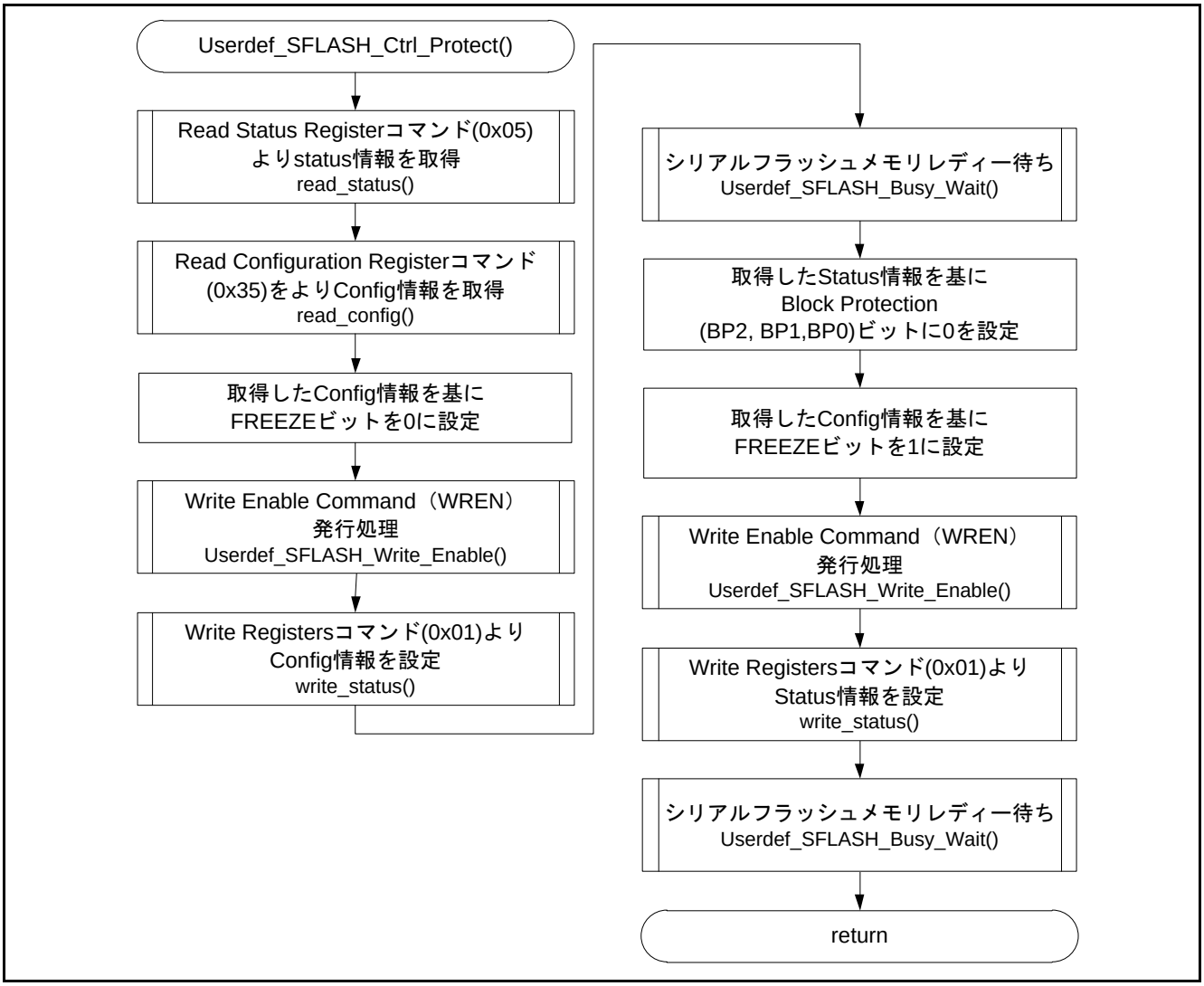

図6.5　シリアルフラッシュメモリプロテクト解除フロー

6.4　シリアルフラッシュメモリ 2 個接続する場合のプログラム配置

シリアルフラッシュメモリを 1 個接続する場合と 2 個接続する場合では、SPI マルチ I/O 空間に配置されるシリアルフラッシュメモリの物理アドレスが異なります。ただし、ブート起動用内蔵 ROM プログラム後の SPIBSC の設定では、シリアルフラッシュメモリが 1 個接続されている設定のため、ブート起動用内蔵 ROM プログラムから実行される SPIBSC 初期設定プログラムは、シリアルフラッシュメモリが 1 個接続されている場合のシリアルフラッシュメモリの物理アドレスに配置する必要があります。

アプリケーションプログラムは、SPIBSC 初期設定プログラムでシリアルフラッシュメモリが 2 個接続されている設定に変更した後に実行しますので、シリアルフラッシュメモリが 2 個接続されている場合のシリアルフラッシュメモリの物理アドレスに配置することができます。

図 6.6にシリアルフラッシュメモリを 2 個接続する場合のプログラム配置の例を示します。

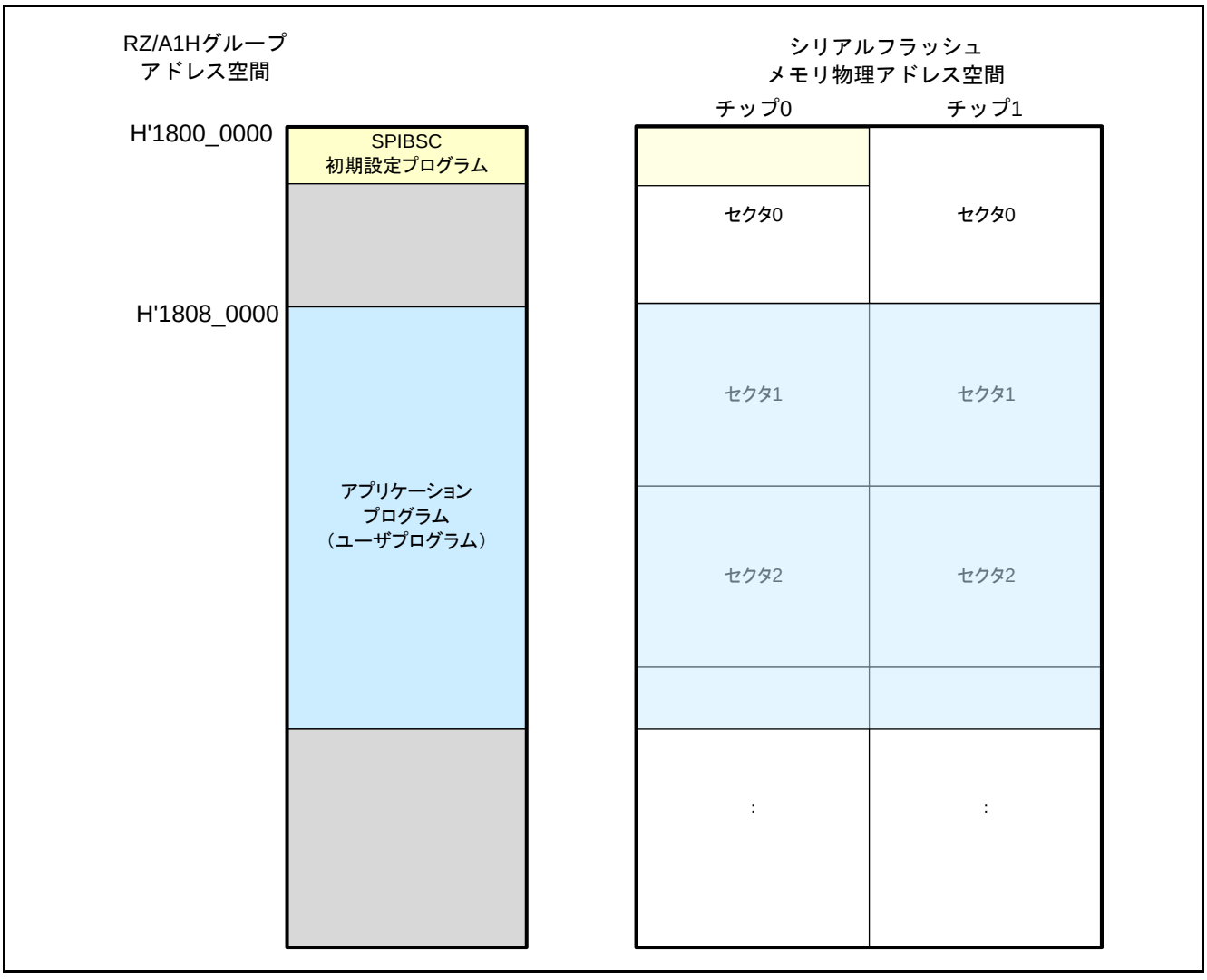

図6.6　シリアルフラッシュメモリを 2 個接続する場合のプログラム配置の例

7. 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。

8. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RZ/A1Hグループ ユーザーズマニュアル ハードウェア編
（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。）

R7S72100 RTK772100BC00000BR（GENMAI）ユーザーズマニュアル
（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。）

ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C
（最新版を ARM ホームページから入手してください。）

ARM Generic Interrupt Controller Architecture Specification Architecture version 1.0
（最新版を ARM ホームページから入手してください。）

ARM Cortex™-A9（Revision: r3p0）Technical Reference Manual
（最新版を ARM ホームページから入手してください。）

ARM CoreLink™ Level 2 Cache Controller L2C-310（Revision: r3p2）Technical Reference Manual
（最新版を ARM 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。）

ユーザーズマニュアル：開発環境

ARM ソフトウェア開発ツール（ARM Compiler toolchain、ARM DS-5 等）に関しては、ARM ホームページから入手してください。
（最新版を ARM 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/

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
Rev.1.00	2014.06.06	ー	初版発行
Rev.1.01	2015.05.29	P10, P11	表 5.2　SPIBSC 設定およびシリアルフラッシュメモリ設定 • サンプルコードで使用する SPIBSC 関連レジスタの設定内容の記載を拡充し、表 5.2と表 5.3に表を分割。
		P16, P17	表 5.6　アプリケーションプログラムで使用するセクション（1/2） 表 5.7　アプリケーションプログラムで使用するセクション（2/2） • 表 5.6および表 5.7にアプリケーションプログラムで使用するセクションの情報を記載。
		P46	図 5.6　SPIBSC 初期設定プログラム 2 のフローチャート • SPIBCS レジスタの各ビットの設定値を追記。 • CKDLY ビット、SPODLY ビットの設定値の誤りを修正。 （変更前） CKDLY ビット = B'1000、SPODLY ビット = H'6363 （変更後） CKDLY ビット = B'0100、SPODLY ビット = H'0000 サンプルコードでも上記の設定値に修正。 • EAC ビットの設定値の誤りを修正。 （変更前）EAC ビット = B'000 （変更後）EAC ビット = B'001 サンプルコードでも上記の設定値に修正。
		ー	Rev.1.00 の「表　サンプルコードの SPIBSC 制御レジスタ設定内容」を表 5.2および表 5.3に内容を統合し、表を削除。

製品ご使用上の注意事項

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

1. 未使用端子の処理
   【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。
   CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。
2. 電源投入時の処置
   【注意】電源投入時は，製品の状態は不定です。
   電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。
   外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。
   同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。
3. リザーブアドレス（予約領域）のアクセス禁止
   【注意】リザーブアドレス（予約領域）のアクセスを禁止します。
   アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
4. クロックについて
   【注意】リセット時は、クロックが安定した後、リセットを解除してください。
   プログラム実行中のクロック切り替え時は､切り替え先クロックが安定した後に切り替えてください。
   リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
5. 製品間の相違について
   【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。
   同じグループのマイコンでも型名が違うと、内部ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、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も、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て、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が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/