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

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

1.1　　シリアルフラッシュメモリからのブート

RZ/A1Hは、ブートモード 3 の場合、SPI マルチ I/O バス空間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。

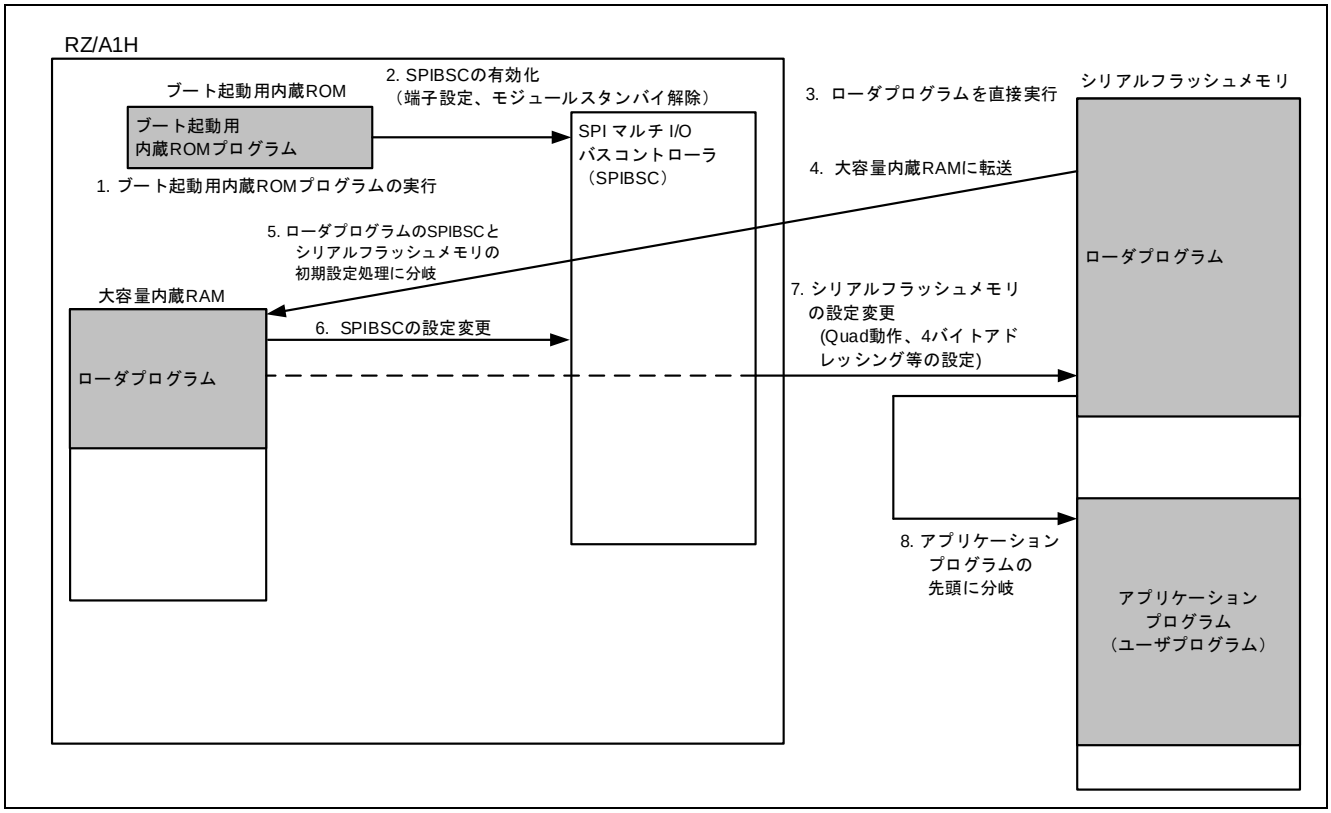


図1.1　シリアルフラッシュブートの動作イメージ

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に格納されたローダプログラムを直接実行します。
4　ローダプログラムのセクション初期化処理により、ローダプログラムをシリアルフラッシュメモリから大容量内蔵 RAM に転送します。
5　大容量内蔵 RAM に転送したローダプログラムの SPIBSC とシリアルフラッシュメモリの初期設定処理に分岐します
6　ローダプログラムにより、SPIBSC の設定を変更します。
7　ローダプログラムにより、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 バス空間の先頭番地（H'1800_0000）にローダプログラムを配置し、ローダプログラムにより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、ローダプログラムにて、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。

1.2　使用する周辺機能

本サンプルコードでは、SPIBSC の設定とともに、クロックパルス発振器、割り込みコントローラ、バスステートコントローラ、汎用入出力ポート、メモリ管理ユニット、1 次キャッシュ（L1 キャッシュ）、および 2 次キャッシュ（L2 キャッシュ）の初期設定を行います。

本アプリケーションノートでは、クロックパルス発振器を CPG、割り込みコントローラを INTC、バスステートコントローラを BSC、OS タイマを OSTM、FIFO 内蔵シリアルコミュニケーションインタフェースを SCIF、汎用入出力ポートを PORT、低消費電力モードを STB、メモリ管理ユニットを MMU とします。

表1.1に使用する周辺機能と用途を、図1.2にサンプルコード実行時の動作環境を示します。

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。
クロックパルス発振器（CPG）	RZ/A1Hの動作周波数の生成
割り込みコントローラ（INTC）	OSTM チャネル 0 の割り込み制御に使用
バスステートコントローラ（BSC）	CS3 空間で SDRAM を使用するための信号の生成（注）
OS タイマ（OSTM）	OSTM チャネル 0 のタイマにより、LED 点灯および消灯の周期を生成
FIFO 内蔵シリアルコミュニケーション インタフェース（SCIF）	SCIF チャネル 0 を用いて、ホスト PC との通信用として使用
汎用入出力ポート（PORT）	SPIBSC、CS3、SCIF チャネル 0 の兼用端子の切り替えに使用、LED の点灯および消灯のための端子制御に使用
低消費電力モード（STB）	RZ/A1Hの周辺 IO のモジュールスタンバイを解除するために使用、保持用内蔵 RAM をライト許可するために使用
メモリ管理ユニット（MMU）、 L1 キャッシュ、L2 キャッシュ	RZ/A1Hの外部アドレス空間において、L1 キャッシュの有効領域の指定やメモリタイプの指定などの変換テーブルを生成。L1 キャッシュおよび L2 キャッシュを有効に設定

【注】 RZ/A1H ボード（Renesas Starter Kit+ for RZ/A1H）には、CS3 空間に SDRAM（Samsung 社製 K4S561632D）が実装されていますが、SDRAM を使用するための BSC および兼用端子の設定は、サンプルコードの初期状態ではソースコード上の設定部分を無効にしています。SDRAM を使用する場合は、ソースコードを変更してください。

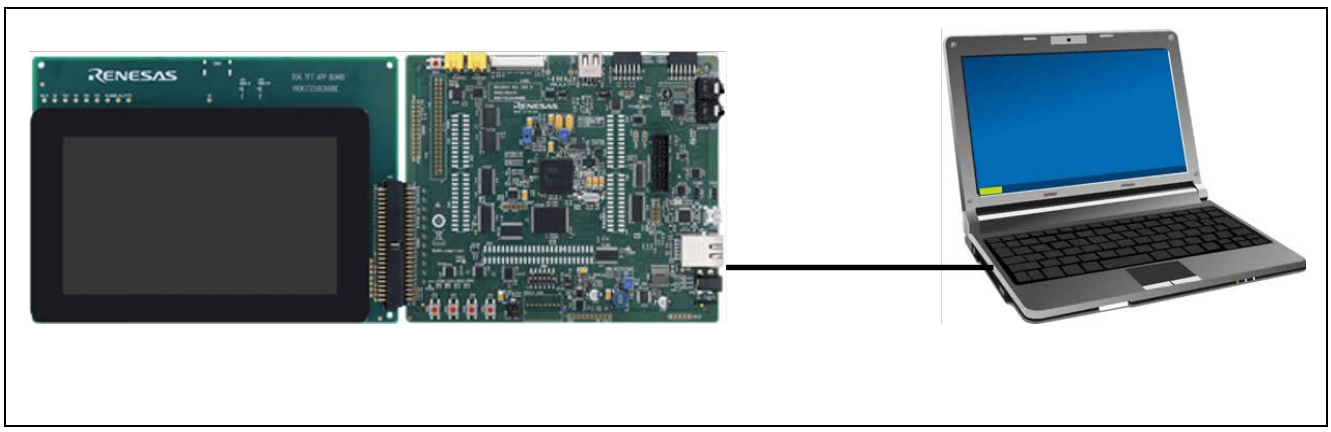


図1.2　動作環境

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 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	e2 studio v7.8.0
C コンパイラ	GNU ARM Embedded Toolchain 6-2017-q2-update
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	RZ/A1H ボード YR0K77210C000BE（以下、Renesas Starter Kit+ for RZ/A1Hとします）
ターミナルソフトの通信設定	• 通信速度：115200bps • データ長：8 ビット • パリティ：なし • ストップビット長：1 ビット • フロー制御：なし
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 バス空間に接続） - メーカ：CYPRESS 社 - 型名：S25FL512S • RL78/G1C（USB 通信とシリアル通信を変換し、ホスト PC との通信に使用） • LED0

【注】 クロックモード 0（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。

- RZ/A1H グループ レジスタ定義ヘッダ・ファイル iodefine.h（R01AN1860JJ）

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。

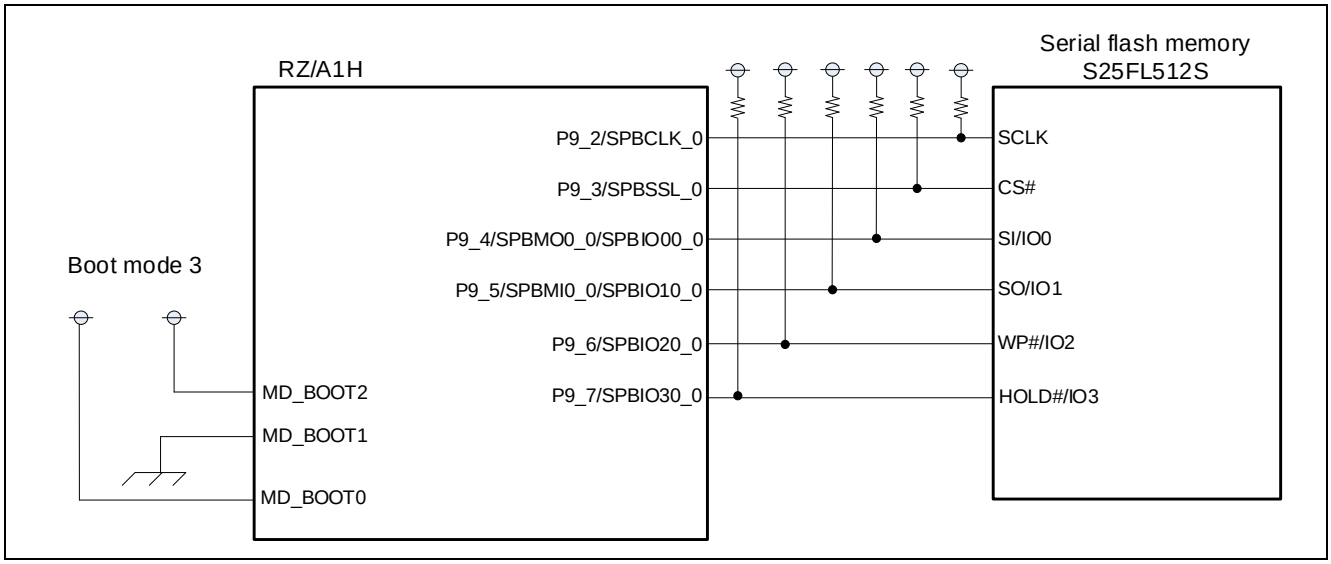


図4.1　シリアルフラッシュメモリからブートする場合の接続例

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_BOOT0 SW6-1	入力	ブートモードの選択 MD_BOOT0：1、MD_BOOT1：0、MD_BOOT2：1 （ブートモード 3 に設定）
MD_BOOT1 SW6-2	入力	
MD_BOOT2 SW6-3	入力	
P7_1	出力	LED0 の点灯および消灯
TxD2(P3_0)	出力	シリアル送信データ信号

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 バス空間の先頭アドレスである H'1808_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。 RZ/A1H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。
ローダプログラム	ローダプログラムは､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。 ローダプログラムは、お客様が使用するシリアルフラッシュメモリに合わせて､SPIBSC およびシリアルフラッシュメモリ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。 ローダプログラムは、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合わせてお客様が作成してください。なお、サンプルコードでは、CYPRESS社製シリアルフラッシュメモリ（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合わせ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ローダ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　ローダプログラム

ローダプログラムは、使用するシリアルフラッシュメモリ（CYPRESS 社製シリアルフラッシュメモリ（S25FL512S））に最適な設定を行います。ローダ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'1808_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。ローダプログラム実行後、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、ローダ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。

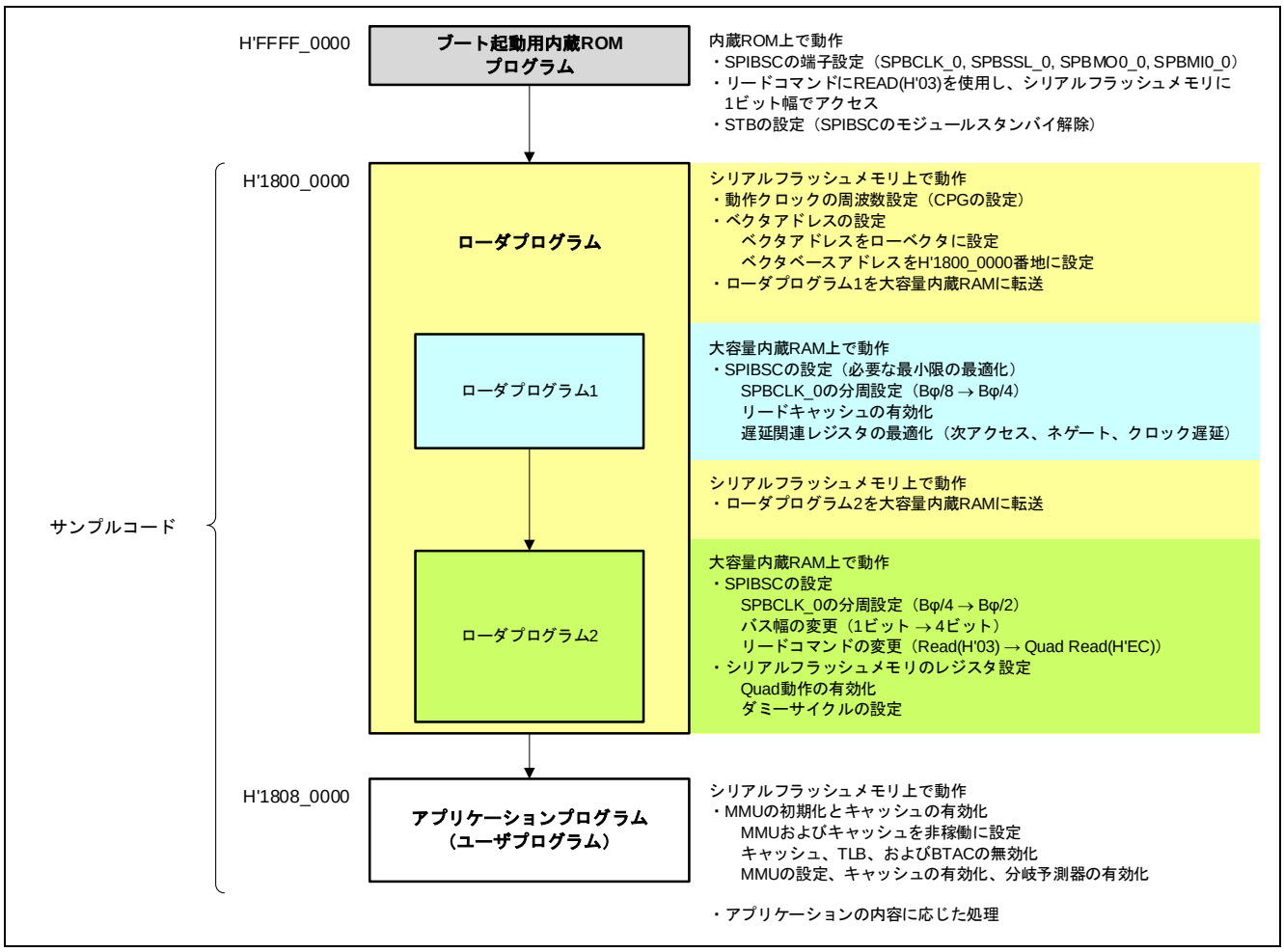


図5.1　サンプルコードの動作概要

5.1.3　ローダプログラムの動作概要

ローダプログラムは、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。ローダプログラムは、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、ローダプログラムにて、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～表5.4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、サンプルコードのローダプログラムでは、CYPRESS社製シリアルフラッシュメモリ（S25FL512S）を使用する場合に最適な設定を行っています。

また、ローダプログラムは、以下に示すローダプログラム 1 とローダプログラム 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　ローダプログラム 1
ローダプログラム 1 では、遅延（次アクセス遅延、SPBSSL ネゲート遅延、クロック遅延）の期間を短くし、転送ビットレートを設定しリードキャッシュを有効にするために、SPIBSC のレジスタを設定します。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　ローダプログラム 2
ローダプログラム 2 では、データバス幅を 4 ビットにし、使用するリードコマンドに合わせて転送ビットレートをさらに最適化し、4 バイトアドレスを出力するために、SPIBSC のレジスタを設定します。また、シリアルフラッシュメモリのダミーサイクル数、Quad 動作の有効化、4 バイトアドレッシングへの変更を行うために、シリアルフラッシュメモリ（S25FL512S）のレジスタを設定します。処理内容が多いため、ローダプログラム 1 よりも大きいプログラムサイズとなっています。

ローダプログラム 1 およびローダプログラム 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、最初にローダプログラム 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、ローダプログラム 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、全体のローダプログラムの実行時間を短縮しています。

表5.2～表5.4に、ブート起動用内蔵 ROM プログラムおよびローダプログラムの設定内容を示します。

ローダプログラムにて、表5.2～表5.4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　ブート起動用内蔵 ROM プログラムおよびローダプログラムの設定内容（1/3）

	項目	ブート起動用内蔵 ROM プログラム実行後	SPIBSC 初期設定 プログラム 1 実行後	SPIBSC 初期設定 プログラム 2 実行後
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'01	CMNCR.IO3FV[1:0]=B'01	CMNCR.IO3FV[1:0]=B'11
	SPBIO20, SPBIO21 の設定	CMNCR.IO2FV[1:0]=B'00	CMNCR.IO2FV[1:0]=B'00	CMNCR.IO2FV[1:0]=B'11
	SPBIO00, SPBIO01 の設定	CMNCR.IO0FV[1:0]=B'00	CMNCR.IO0FV[1:0]=B'00	CMNCR.IO0FV[1:0]=B'11
	シリアルフラッシュ接続数： CMNCR.BSZ[1:0]	1 個 B'00	1 個 B'00	1 個 B'00
	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： DRCR.RBURST[3:0]	1 データ長（8 バイト） B'0000	1 データ長（8 バイト） B'0000	4 データ長（32 バイト） B'0011
	データバス幅： DRENR.DRDB[1:0]	1 [bit] B'00	1 [bit] B'00	4 [bit] B'10
	リードコマンド： DRCMR.CMD[7:0]	Read H'03	Read H'03	QuadIO Read (4B address) H'EC
	コマンドイネーブル： DRENR.CDE	出力する 1	出力する 1	出力する 1
	オプショナルコマンドイネーブル： DRENR.OCDE	出力しない 0	出力しない 0	出力しない 0

表5.3　ブート起動用内蔵 ROM プログラムおよびローダプログラムの設定内容（2/3）

	項目	ブート起動用内蔵 ROM プログラム実行後	SPIBSC 初期設定 プログラム 1 実行後	SPIBSC 初期設定 プログラム 2 実行後
SPIBSC 設定	アドレスイネーブル： 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
	オプションデータイネーブル： DRENR.OPDE[3:0]	出力しない B'0000	出力しない B'0000	OPD3 を出力（注） B'1000
	オプションデータビット幅： DRENR.OPDB[1:0]	－	－	4 [bit] B'10
	オプションデータ： DROPR.OPD3[7:0] DROPR.OPD2[7:0] DROPR.OPD1[7:0] DROPR.OPD0[7:0]	 － － － －	 － － － －	 H'00 － － －
	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
	転送フォーマット： DRDRENR.ADDRE DRDRENR.OPDRE DRDRENR.DRDRE	アドレス、オプションデータ、データは SDR 転送 0 0 0	アドレス、オプションデータ、データは SDR 転送 0 0 0	アドレス、オプションデータ、データは SDR 転送 0 0 0
	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

【注】 S25FL512Sは、アドレスサイクルに続く MODE のサイクル期間に、H'Ax(don't care "x")が入力されると、High Performance Read Mode に遷移します。RZ/A1Hの外部アドレス空間リードモードは High Performance Read Mode のデータ転送に対応していませんので、サンプルコードでは QuadIO Read コマンド発行時に、OPD3 から H'00 を出力するように設定し、S25FL512Sが Hith Performance Read Mode に遷移しないようにしています。

表5.4　ブート起動用内蔵 ROM プログラムおよびローダプログラムの設定内容（3/3）

	項目	ブート起動用内蔵 ROM プログラム実行後	SPIBSC 初期設定 プログラム 1 実行後	SPIBSC 初期設定 プログラム 2 実行後
兼用端子の設定	P9_2	SPBCLK_0	SPBCLK_0	SPBCLK_0
	P9_3	SPBSSL_0	SPBSSL_0	SPBSSL_0
	P9_4	SPBMO0_0 / SPBIO00_0	SPBMO0_0 / SPBIO00_0	SPBMO0_0 / SPBIO00_0
	P9_5	SPBMI0_0 / SPBIO10_0	SPBMI0_0 / SPBIO10_0	SPBMI0_0 / SPBIO10_0
	P9_6	P9_6	P9_6	SPBIO20_0
	P9_7	P9_7	P9_7	SPBIO30_0
シリアルフラッシュメモリ設定	Configuration Register	変更なし（注）	変更なし（注）	Quad 動作 Enable QUAD=1
	Configuration Register	変更なし（注）	変更なし（注）	LC[1:0] = B'00
その他	動作クロックの設定 EXTAL から 13.33MHz 入力時	Iφ=133.33[MHz] Bφ=133.33[MHz] P1φ=66.67[MHz] P0φ=33.33[MHz]	Iφ=400[MHz] Bφ=133.33[MHz] P1φ=66.67[MHz] P0φ=33.33[MHz]	Iφ=400[MHz] Bφ=133.33[MHz] P1φ=66.67[MHz] P0φ=33.33[MHz]
	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：H'03、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）

(1)　アプリケーションプログラム（ユーザプログラム）の動作

リセット解除後に、ブート起動用内蔵 ROM プログラム、ローダプログラムの順にプログラムが実行され、H'1808_0000 番地に配置されているアプリケーションプログラムに分岐します。

アプリケーションプログラムでは、スタックポインタ、MMU の設定を行い、main に分岐します。

main 関数では、SystemInit 関数にて、STB、BSC、INTC、PORT などの周辺機能の初期設定と、L1 キャッシュおよび L2 キャッシュを有効にするための設定を行い、IRQ 割り込みおよび FIQ 割り込みを許可にしています。また、シリアルインタフェースで接続されたホスト PC 上のターミナルに文字列を出力し、OSTM チャネル 0 をインターバルタイマモードに設定して、タイマを起動します。500ms の周期で OSTM チャネル 0 の割り込みを発生させ、Renesas Starter Kit+ for RZ/A1Hの LED を割り込み処理により 500ms ごとに点灯および消灯を繰り返す処理を行います。（サンプルコードでは、MMU の設定およびキャッシュを有効にするための設定はアプリケーションプログラムで行い、ローダプログラムでは行っていません）。

(2)　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、ローダ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、ローダプログラムとは異なるシリアルフラッシュメモリのセクタに配置してください。

Renesas Starter Kit+ for RZ/A1Hに搭載されている CYPRESS 社製シリアルフラッシュメモリ（S25FL512S）のセクタサイズは 4KB 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セクタ 16 の 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。

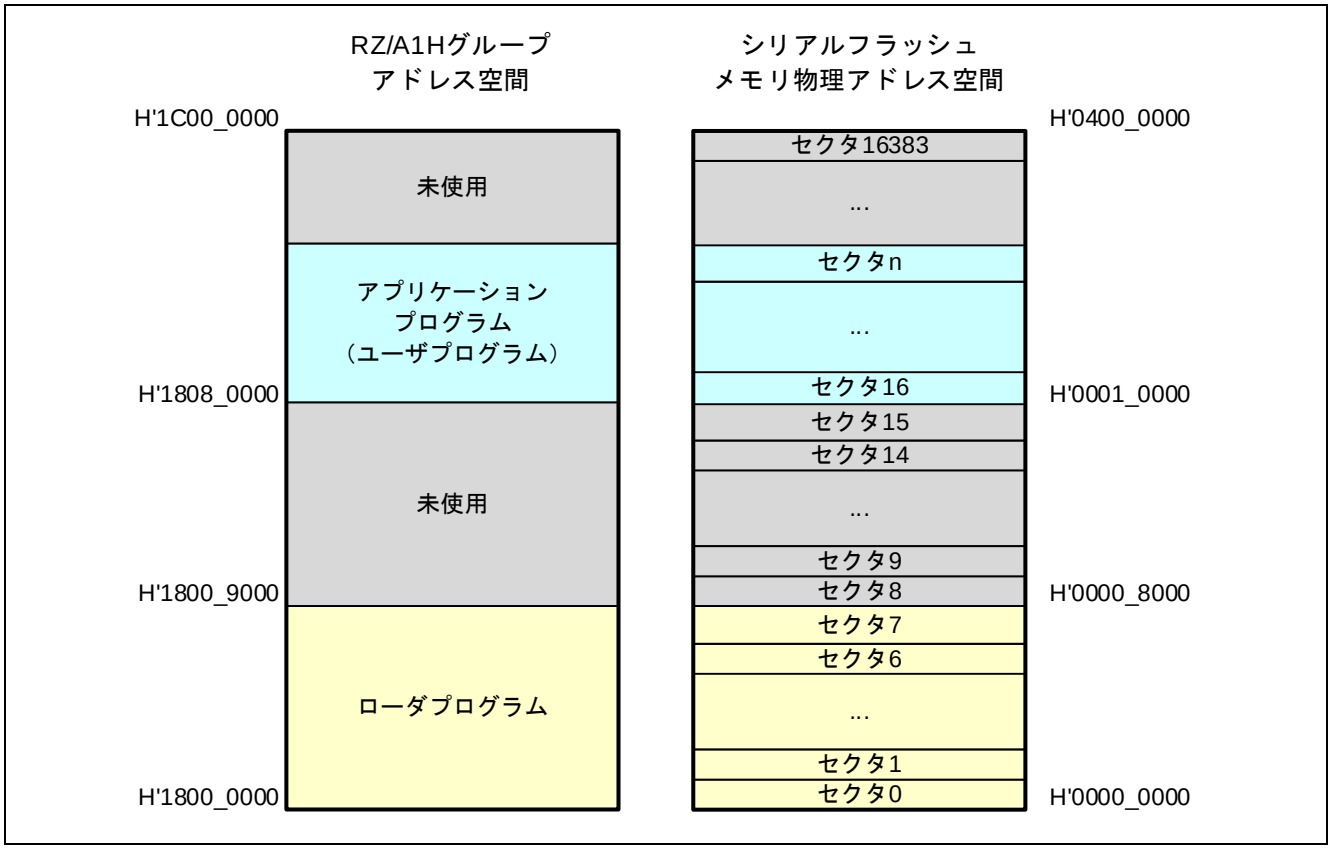


図5.2　サンプルコードのプログラム配置

アプリケーションプログラムの先頭アドレスは、以下の内容を変更することにより、アドレス配置の変更が可能です。

- ローダプログラムのプロジェクト
  アプリケーションプログラムの先頭アドレスへの分岐は、ローダプログラム 2（spibsc_init2.c）にて行っています。「spibsc_config.h」のマクロ定義"DEF_USER_PROGRAM_TOP"により、分岐先を指定してください。
- アプリケーションプログラムのプロジェクト
  スキャッタファイルで、アプリケーションプログラムの"VECTOR_TABLE"のセクションが、"DEF_USER_PROGRAM_TOP"で指定したアドレスと一致するように配置アドレスを変更してください。

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5.5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5.5　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
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～表5.4を参照
PORT	PORT9、PORT7、PORT3 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 P7_1：LED の点灯および消灯 P3_0：TxD2 P3_2：RxD2
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
OSTM	チャネル 0 をインターバルタイマモードに設定 P0φ=33.33MHz の時に 500ms ごとに割り込み要求を発生するように、タイマカウントを設定
INTC	INTC の初期設定および OSTM チャネル 0 割り込み（割り込み ID が 134）ハンドラの登録と実行
SCIF	チャネル 0 を調歩同期式モードに設定 ・データ長：8 ビット ・ストップビット長：1 ビット ・パリティ：なし P1φ=66.67MHz の時に、クロックソースを分周なし、ビットレート値に 17 を設定し、ビットレートが 115200bps となるように設定

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Renesas Starter Kit+ for RZ/A1Hのメモリマップを示します。

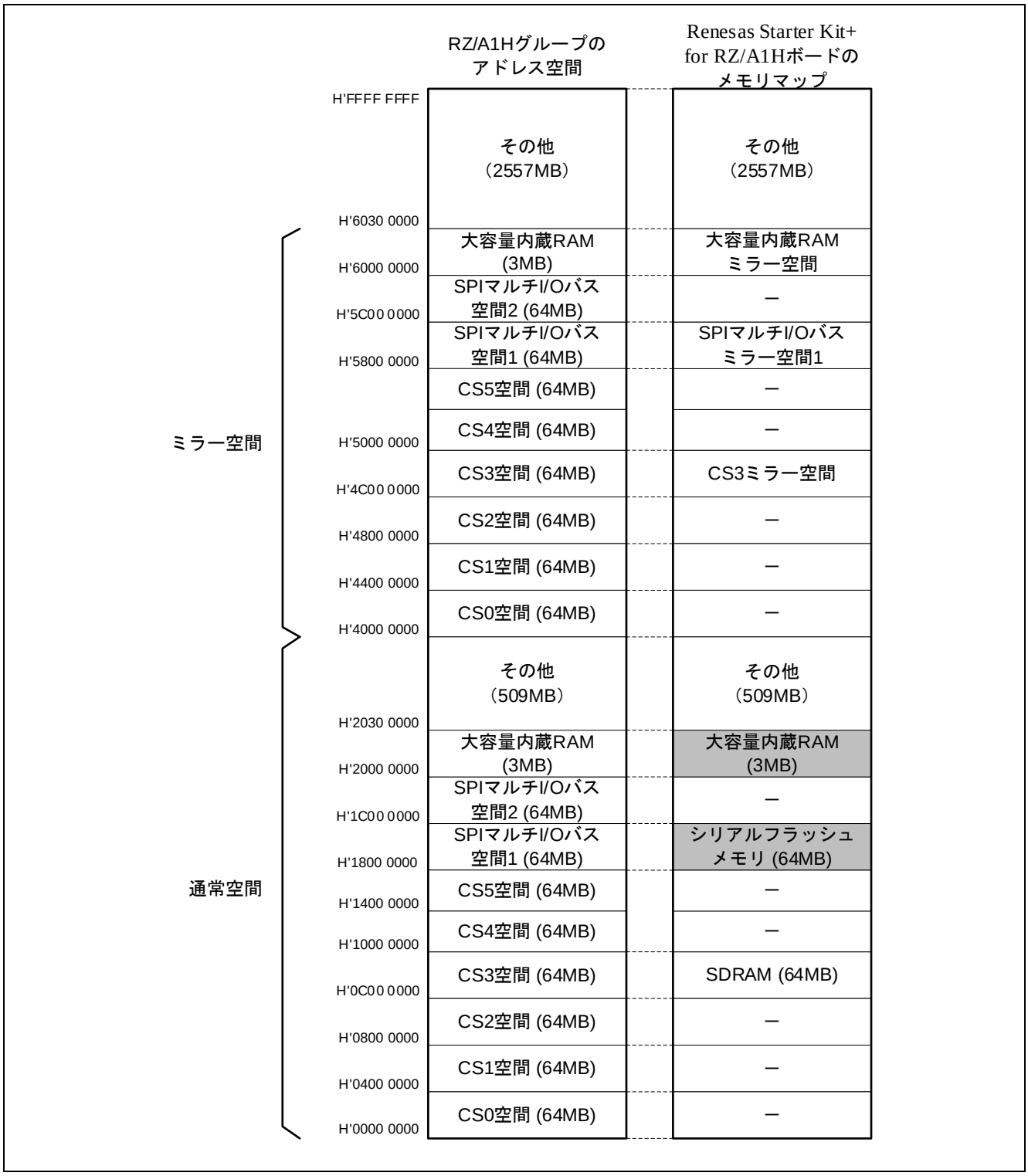


図5.3　RZ/A1Hグループのアドレス空間とRenesas Starter Kit+ for RZ/A1Hメモリマップ

5.2.3　サンプルコードのセクション配置

表5.6にローダプログラムで使用するセクションを、表5.7および表5.8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5.6　ローダ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	ローダプログラム 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	ローダプログラム 2 用プログラムコード領域	Code	S-FLASH	LRAM
DATA_SPIBSC_INIT2	ローダプログラム 2 用初期値ありデータ領域	RW Data	S-FLASH	LRAM
BSS_SPIBSC_INIT2	ローダプログラム 2 用初期値なし領域	ZI Data	－	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	－	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5.7　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5.8　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
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　使用割り込み一覧

表5.9にサンプルコード（アプリケーションプログラム）で使用する割り込みを示します。

表5.9　サンプルコード（アプリケーションプログラム）で使用する割り込み

割り込み要因（要因 ID）	優先度	処理概要
OSTM0（134）	5	500ms ごとに割り込みを発生

5.4　ローダプログラムの定数一覧

表5.10～表5.12にサンプルコードのローダプログラムで使用する定数を示します。

表5.10　サンプルコードで使用する定数（1/3）

定数名	設定値	内容
DEF_USER_PROGRAM_TOP	0x18080000	アプリケーションプログラムの先頭アドレス
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	SPIBSC 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	SPIBSC 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_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_DISABLE	0	コマンド、オプショナルコマンド、アドレス、オプションデータを出力しない設定
SPIBSC_OUTPUT_ENABLE	1	コマンド、オプショナルコマンド、アドレス、オプションデータを出力する設定
SPIBSC_OUTPUT_OPD_3	0x08	リードコマンド発行時のオプションデータイネーブル 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をリード/ライトしない設定
SPIBSC_SPIDATA_ENABLE	1	SPI 動作モード時にデータをリード/ライトする設定
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

表5.11　サンプルコードで使用する定数（2/3）

定数名	設定値	内容
SPIBSC_SDR_TRANS	0	シリアルフラッシュメモリからのリードを SDR モードで行う。
SPIBSC_DDR_TRANS	1	シリアルフラッシュメモリからのリードを DDR モードで行う。
SF_REQ_SERIALMODE	2	シリアルフラッシュメモリのレジスタを「Single モード」に設定
SF_REQ_QUADMODE	3	シリアルフラッシュメモリのレジスタを「Quad モード」に設定
SPIBSC_CKDLY_DEFAULT	0x0000A504	CKDLY レジスタへの設定値の定義 CKDLY[3:0]に B'0100 を設定（初期値）
SPIBSC_CKDLY_TUNING	0x0000A508	CKDLY レジスタへの設定値の定義 CKDLY[3:0]に B'1000 を設定
SPIBSC_SPODLY_DEFAULT	0xA5000000	SPODLY レジスタへの設定値の定義 SPODLY[15:0]に H'0000 を設定（初期値）
SPIBSC_SPODLY_TUNING	0xA5006363	SPODLY レジスタへの設定値の定義 SPODLY[15:0]に H'6363 を設定
SFLASH_MODECYC	 2 1	S25FL512Sの High Performance Read Mode Indicator の期間を示します。 SPIBSC_TRANS_MODE が SPIBSC_SDR_TRANS のとき SPIBSC_TRANS_MODE が SPIBSC_DDR_TRANS のとき

表5.12　サンプルコードで使用する定数（3/3）

定数名	設定値	内容
SFLASHCMD_READ_STATUS	0x05	S25FL512Sの Status Register リードコマンド
SFLASHCMD_READ_CONFIG	0x35	S25FL512Sの Configuration Register リードコマンド
SFLASHCMD_WRITE_STATUS	0x01	S25FL512Sのライトステータスコマンド
SFLASHCMD_WRITE_ENABLE	0x06	S25FL512Sのライトする設定
SFLASHCMD_WRITE_DISABLE	0x04	S25FL512S のライトしないコマンド
STREG_SRWD_BIT	0x80	S25FL512Sの Status Register の SRWD ビット
STREG_BPROTECT_BIT	0x1c	S25FL512Sの Status Register の BP ビット
STREG_WEL_BIT	0x02	S25FL512Sの Status Register の WEL ビット
STREG_WIP_BIT	0x01	S25FL512Sの Status Register の WIP ビット
CFREG_LC0_BIT	0x80	S25FL512Sの Configuration Register の LC ビット
CFREG_LC1_BIT	0x40	S25FL512S の Configuration Register の LC ビット
CFREG_QUAD_BIT	0x02	S25FL512Sの Configuration Register の QUAD ビット
CFREG_FREEZE_BIT	0x01	S25FL512Sの Configuration Register の FREEZE ビット

5.5　ローダプログラムの構造体/共用体一覧

表5.13～表5.21にサンプルコードのローダプログラムで使用する構造体を示します。

表5.13　SPIBSC 外部アドレス空間リードモード設定構造体（st_spibsc_cfg_t）（1/4）

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	オプションデータ • SPI マルチ I/O バス空間へのリードを 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ン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	オプションデータイネーブル • オプション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	オプションデータビット幅 • オプション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5.14　SPIBSC 外部アドレス空間リードモード設定構造体（st_spibsc_cfg_t）（2/4）

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_DMYCYC_SETTING"で定義した値をもとに、データリードダミーサイクル設定レジスタ（DRDMCR）の DMCYC[2:0]に設定する値を算出し、その結果を本メンバに設定します。結果が 0～7 となる値を設定可能です。計算の内容については、「表6.2　サンプルコードカスタマイズ用マクロ一覧（1/2）」を参照してください。
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5.15　SPIBSC 外部アドレス空間リードモード設定構造体（st_spibsc_cfg_t）（3/4）

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5.16　SPIBSC 外部アドレス空間リードモード設定構造体（st_spibsc_cfg_t）（4/4）

メンバ名	内容
uint8_t　udef_drdrenr_addre	アドレス DDR イネーブル • 外部アドレス空間リード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 SPIBSC_DDR_TRANS：DDR 転送 • 本メンバに設定した値をデータリード DDR イネーブルレジスタ（DRDRENR）の ADDRE に設定します。
uint8_t　udef_drdrenr_opdre	オプションデータ DDR イネーブル • 外部アドレス空間リード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 SPIBSC_DDR_TRANS：DDR 転送 • 本メンバに設定した値をデータリード DDR イネーブルレジスタ（DRDRENR）の OPDRE に設定します。
uint8_t　udef_drdrenr_drdre	転送データ DDR イネーブル • 外部アドレス空間リード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 SPIBSC_DDR_TRANS：DDR 転送 • 本メンバに設定した値をデータリード DDR イネーブルレジスタ（DRDRENR）の DRDRE に設定します。

表5.17　SPIBSC SPI 動作モード設定構造体（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	オプションデータビット幅 • SPI 動作モード時のオプション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5.18　SPIBSC SPI 動作モード設定構造体（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	オプションデータイネーブル • SPI 動作モード時にオプション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5.19　SPIBSC SPI 動作モード設定構造体（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　addre	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 SPIBSC_DDR_TRANS：DDR 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 SPIBSC_DDR_TRANS：DDR 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 SPIBSC_DDR_TRANS：DDR 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5.20　SPIBSC SPI 動作モード設定構造体（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 • 設定可能な値： "SPIBSC_DMYCYC_SETTING"で定義した値をもとに、SPI モードダミーサイクル設定レジスタ（SMDMCR）の DMCYC[2:0]に設定する値を算出し、その結果を本メンバに設定します。結果が 0～7 となる値を設定可能です。計算の内容については、「表6.2　サンプルコードカスタマイズ用マクロ一覧（1/2）」を参照してください。
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5.21　SPIBSC SPI 動作モード設定構造体（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]

5.6　ローダプログラムの変数一覧

表5.22にグローバル変数を示します。

表5.22　グローバル変数

型	変数名	内容
st_spibsc_cfg_t	g_spibsc_cfg	SPIBSC 外部アドレス空間リードモードの設定内容格納変数 • SPIBSC 外部アドレス空間リードモードで使用するためのレジスタ設定情報を格納します。
st_spibsc_spimd_reg_t	g_spibsc_spimd_reg	SPIBSC SPI 動作モードの設定内容格納変数 • SPIBSC SPI 動作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。

5.7　ローダプログラムの関数一覧

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。

サンプルコードのローダプログラムの関数について、表5.23にサンプル関数一覧を、表5.24にAPI関数一覧を、表5.25にユーザ定義関数一覧を示します。

表5.23　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	ローダプログラム 1 展開関数 ローダログラム 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	ローダプログラム 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	ローダプログラム 2 展開関数 ローダプログラム 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	ローダプログラム 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5.24　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 を外部アドレス空間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、SPIBSC 関連レジスタの初期設定の内容に合わせて、シリアルフラッシュメモリのレジスタ設定を行います。初期設定後、外部アドレス空間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 外部アドレス空間リードモード切り替え関数 SPIBSC を SPI 動作モードから外部アドレス空間リードモードに切り替えます。
R_SFLASH_Set_Config	SPIBSC 外部アドレス空間リードモード設定関数 SPIBSC を外部アドレス空間リードモードで使用するためのレジスタ設定情報を g_spibsc_cfg に設定します。
R_SFLASH_SpibscStop	SPIBSC 停止関数 SSL をネゲートして、シリアルフラッシュメモリへのアクセスを停止します。
R_SFLASH_Exmode_Init	SPIBSC 外部アドレス空間リードモード初期設定関数 SPIBSC を外部アドレス空間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、外部アドレス空間リードモードに設定します。
R_SFLASH_Spibsc_Transfer	シリアルフラッシュ制御関数 SPI 動作モードで、引数の内容にしたがってシリアルフラッシュメモリにコマンドを発行します。

表5.25　ユーザ定義関数一覧

関数名	概要
Userdef_SPIBSC_Set_Config	SPIBSC 外部アドレス空間リードモードのレジスタ情報格納関数 使用するシリアルフラッシュメモリに合わせて、SPIBSC 外部アドレス空間リードモードで SPIBSC 関連レジスタに設定するためのレジスタ設定情報を引数で指定された領域に格納してください。 サンプルコードでは、CYPRESS 社製シリアルフラッシュメモリ（S25FL512S）を使用する場合に、SPIBSC 関連レジスタに設定するためのレジスタ設定情報を格納します。
Userdef_SFLASH_Set_Mode	シリアルフラッシュメモリのレジスタ設定関数 使用するシリアルフラッシュメモリに合わせて、SPIBSC を外部アドレス空間リードモードで使用するために必要なシリアルフラッシュメモリのレジスタを設定する処理を実装してください。 サンプルコードでは、CYPRESS社製シリアルフラッシュメモリ（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に合わせて、シリアルフラッシュメモリのレジスタへのライトを許可する処理を実装してください。 サンプルコードでは、CYPRESS社製シリアルフラッシュメモリ（S25FL512S）に"Write Status Register(WRSR)"コマンドを発行する処理を行っています。
Userdef_SFLASH_Busy_Wait	シリアルフラッシュメモリライト完了待ち関数 使用するシリアルフラッシュメモリに合わせて、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のライト完了を待つ処理を実装してください。 サンプルコードでは、CYPRESS社製シリアルフラッシュメモリ（S25FL512S）に"Read Status Register(RDSR)"コマンドを発行し、Status Register の内容を参照することでライト完了を待つ処理を行っています。

5.8　ローダプログラムの関数仕様

サンプルコードのローダプログラムの関数仕様を示します。

reset_handler	
概　要	ローダプログラムのリセットハンドラ
宣　言	reset_handler
説　明	ローダ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	ローダプログラム 1 展開関数
宣　言	void init_spibsc_init1_section (void);
説　明	ローダプログラム 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	ローダプログラム 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	ローダプログラム 2 のセクション初期化関数
宣　言	void init_spibsc_init2_section (void);
説　明	ローダプログラム 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	ローダプログラム 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：H'03→H'EC ● シリアルフラッシュメモリのレジスタの設定 コンフィグレーションレジスタ：QUAD ビットに 1 を設定 コンフィグレーションレジスタ：LC[1:0] （LC[1:0]ビットの設定値はリードコマンドに依存して異なります。「表6.6 S25FL512Sの動作周波数に対して必要なダミーサイクル数の一覧」を参照してください。）
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_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 を外部アドレス空間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、SPIBSC 関連レジスタの初期設定の内容に合わせて、シリアルフラッシュメモリのレジスタ設定を行います。シリアルフラッシュメモリのレジスタを設定する前に、SPIBSC を外部アドレス空間リードモードから SPI 動作モードに切り替え、シリアルフラッシュメモリのレジスタ設定後に、SPIBSC を SPI 動作モードから外部アドレス空間リードモードに切り替えます。 本関数内で SPIBSC 外部アドレス空間リード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 のみ指定可能）
	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 外部アドレス空間リードモード設定 設定内容は表5.13～表5.16を参照してください。
リターン値	0：正常終了 -1: エラー	

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 のみ指定可能）
リターン値	なし	

R_SFLASH_Exmode		
概　要	SPIBSC 外部アドレス空間リードモード切り替え関数	
宣　言	int32_t R_SFLASH_Exmode(uint32_t ch_no);	
説　明	SPIBSC を SPI 動作モードから外部アドレス空間リードモードに切り替えます。外部アドレス空間リードモードに切り替え後、SPI マルチ I/O バス空間をリードする前に、リードキャッシュの全エントリをクリアします。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 のみ指定可能）
リターン値	0：設定成功	

R_SFLASH_Set_Config		
概　要	SPIBSC 外部アドレス空間リードモード設定関数	
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合わせて、SPIBSC を外部アドレス空間リードモードで使用するためのレジスタ設定情報を g_spibsc_cfg に設定します。 本関数内で、ユーザ定義関数（SPIBSC 外部アドレス空間リードモードのレジスタ情報格納関数 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 のみ指定可能）
	st_spibsc_cfg_t *spibsccfg	SPIBSC 外部アドレス空間リードモード設定 設定内容は表5.13～表5.16を参照してください。
リターン値	0：正常終了	
	-1: エラー	

R_SFLASH_SpibscStop		
概　要	SPIBSC 停止関数	
宣　言	int32_t R_SFLASH_SpibscStop(uint32_t ch_no);	
説　明	SSL をネゲートして、シリアルフラッシュメモリへのアクセスを停止します。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 のみ指定可能）
リターン値	なし	

R_SFLASH_Exmode_Init		
概　要	SPIBSC 外部アドレス空間リード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 を外部アドレス空間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、外部アドレス空間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 のみ指定可能）
	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 外部アドレス空間リードモード設定 設定内容は表5.13～表5.16を参照してください。
リターン値	0：正常終了	
	-1: エラー	

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	SPI 動作モードで、引数 regset の内容にしたがってシリアルフラッシュメモリにコマンドを発行します。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 のみ指定可能）
	st_spibsc_spimd_reg_t * regset	SPIBSC SPI 動作モード設定 設定内容は表5.17～表5.21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	外部アドレス空間リードモードで本関数をコールした場合は、SPI 動作モードに切り替えて、シリアルフラッシュメモリにコマンドを発行します。	

Userdef_SPIBSC_Set_Config		
概　要	SPIBSC 外部アドレス空間リードモードのレジスタ情報格納関数	
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合わせて、SPIBSC 外部アドレス空間リードモードで SPIBSC 関連レジスタに設定するためのレジスタ設定情報を引数 spibsccfg で指定された領域に格納してください。 引数 spibsccfg に設定する内容については、表5.13～表5.16および「6.3.1　リードコマンド発行時の出力信号」を参照してください。	
引　数	uint32_t ch_no	SPIBSC のチャネル番号（0 のみ指定可能）
	st_spibsc_cfg_t *spibsccfg	SPIBSC 外部アドレス空間リード設定 設定内容は表5.13～表5.16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、CYPRESS社製シリアルフラッシュメモリ（S25FL512S）を使用する場合に、SPIBSC 関連レジスタに設定するためのレジスタ設定情報を格納します。	

Userdef_SFLASH_Set_Mode		
概　要	シリアルフラッシュメモリのレジスタ設定関数	
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	使用するシリアルフラッシュメモリに合わせて、SPIBSC を外部アドレス空間リードモードで使用するために必要なシリアルフラッシュメモリの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 のみ指定可能）
	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	SPIBSC とシリアルフラッシュメモリのデータ転送のバス幅 SPIBSC_1BIT：1 ビット SPIBSC_4BIT：4 ビット
	uint8_t addr_mode	シリアルフラッシュメモリに発行するアドレスのビット幅 リードコマンド発行時に出力するアドレスのビット幅を指定します 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、CYPRESS社製シリアルフラッシュメモリ（S25FL512S）へのレジスタ設定を行っています。	

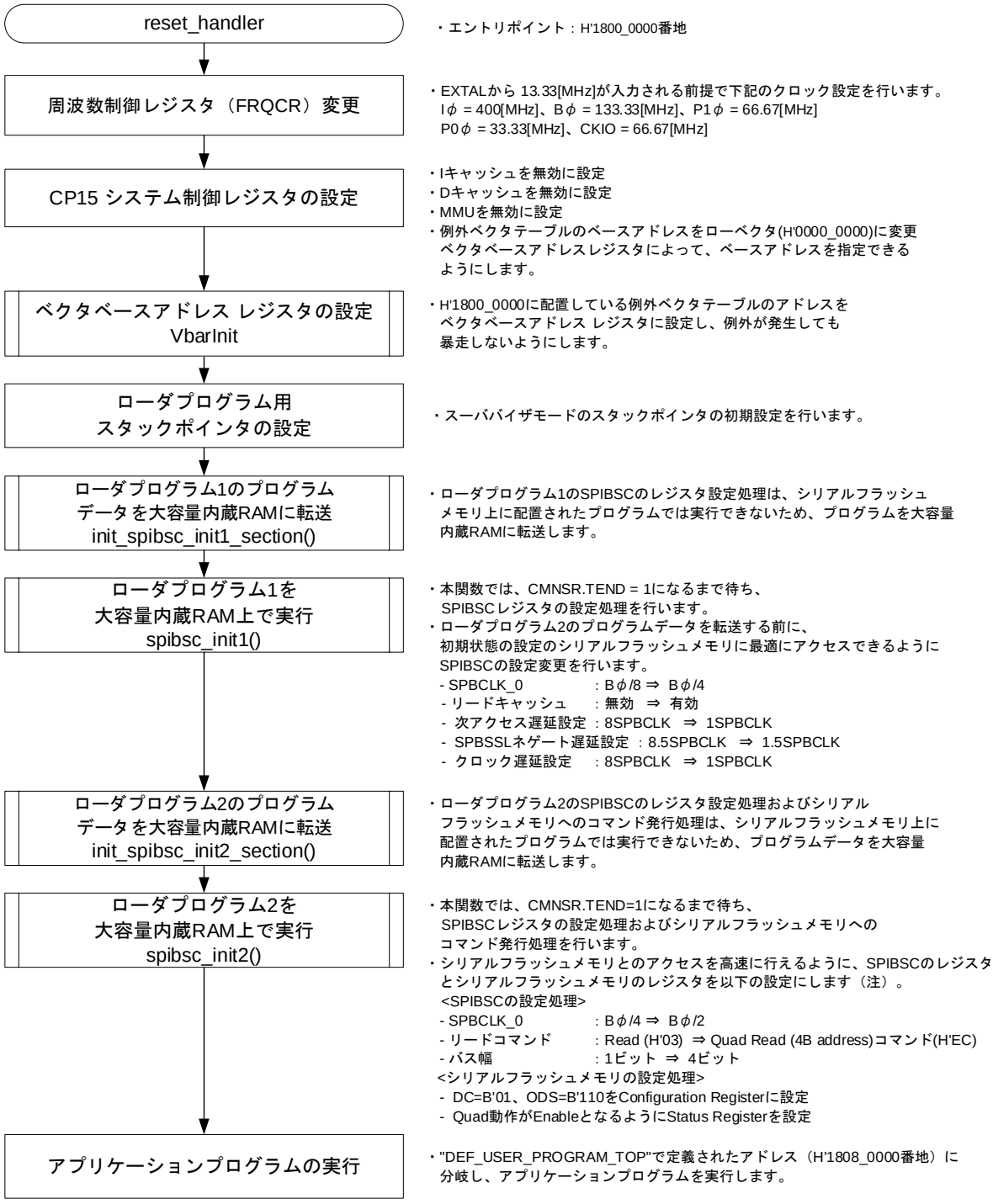
Userdef_SFLASH_Write_Enable	
概　要	シリアルフラッシュメモリライト許可関数
宣　言	int32_t Userdef_SFLASH_Write_Enable(uint32_t ch_no);
説　明	使用するシリアルフラッシュメモリに合わせて、シリアルフラッシュメモリのレジスタへのライトを許可する処理を実装してください。
引　数	uint32_t ch_no　SPIBSC のチャネル番号（0 のみ指定可能）
リターン値	0：設定成功 -1：設定失敗
備　考	サンプルコードでは、CYPRESS社製シリアルフラッシュメモリ（S25FL512S）に"Write Status Register(WRSR)"コマンドを発行する処理を行っています。

Userdef_SFLASH_Busy_Wait	
概　要	シリアルフラッシュメモリライト完了待ち関数
宣　言	int32_t Userdef_SFLASH_Busy_Wait(uint32_t ch_no, uint32_t dual, uint8_t data_width);
説　明	使用するシリアルフラッシュメモリに合わせて、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のライト完了を待つ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 のみ指定可能） 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　SPIBSC とシリアルフラッシュメモリのデータ転送のバス幅 SPIBSC_1BIT：1 ビット SPIBSC_4BIT：4 ビット
リターン値	なし
備　考	サンプルコードでは、CYPRESS社製シリアルフラッシュメモリ（S25FL512S）に"Read Status Register(RDSR)"コマンドを発行し、Status Register の内容を参照することでライト完了を待つ処理を行っています。

5.9　ローダプログラムのフローチャート

5.9.1　ローダプログラム（全体）

図5.4にローダプログラム（全体）のフローチャートを示します。



【注】 シリアルフラッシュメモリのレジスタ設定時に、SPIBSC を外部アドレス空間リードモードから SPI 動作モードに変更し、レジスタ設定後に SPI 動作モードから外部アドレス空間リードモードに戻す処理を行います。

図5.4　ローダプログラム（全体）のフローチャート

5.9.2　ローダプログラム 1（STEP1）

ローダプログラム 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にする処理を行います。

ローダプログラム 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にローダプログラム1のフローチャートを示します。

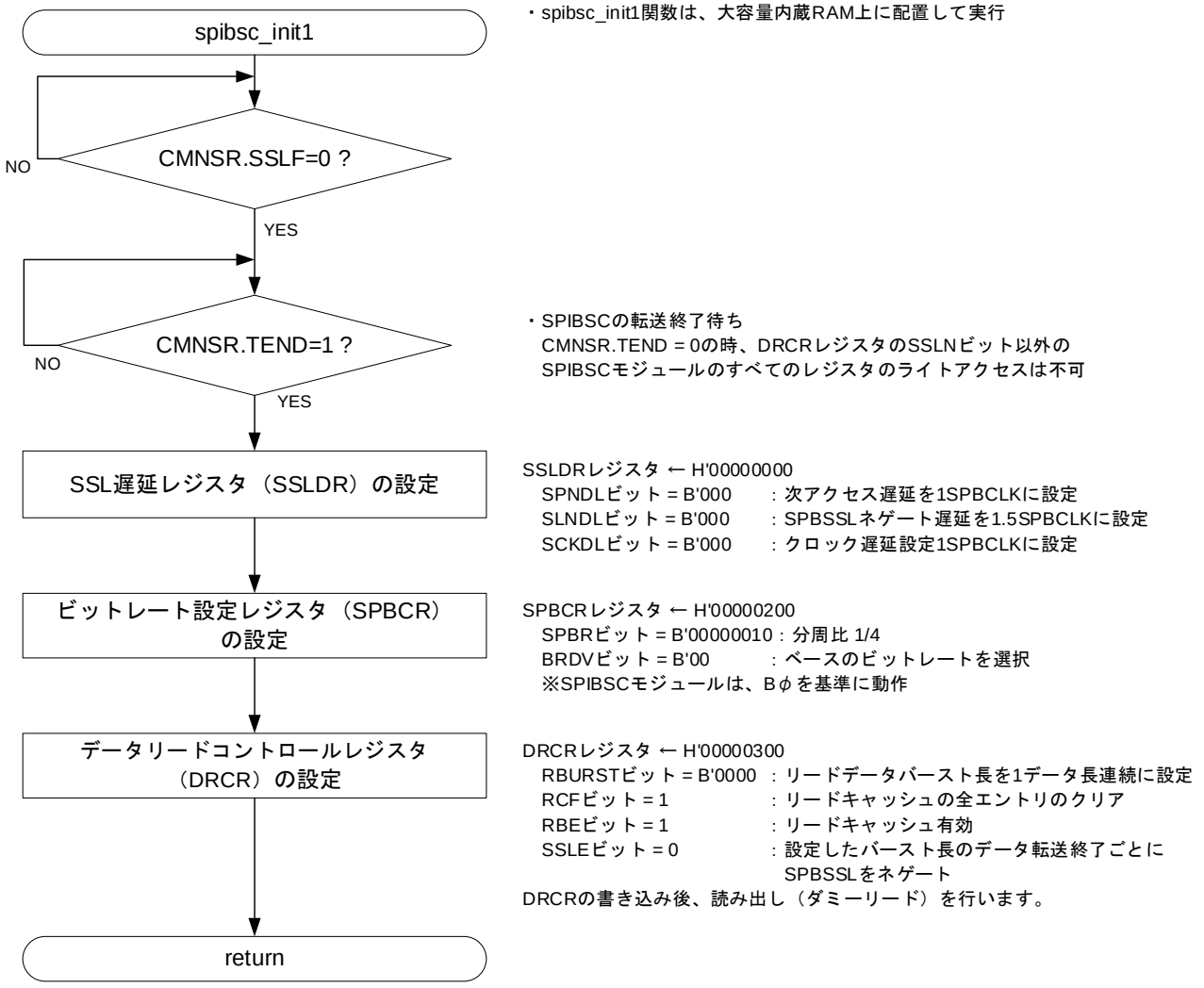


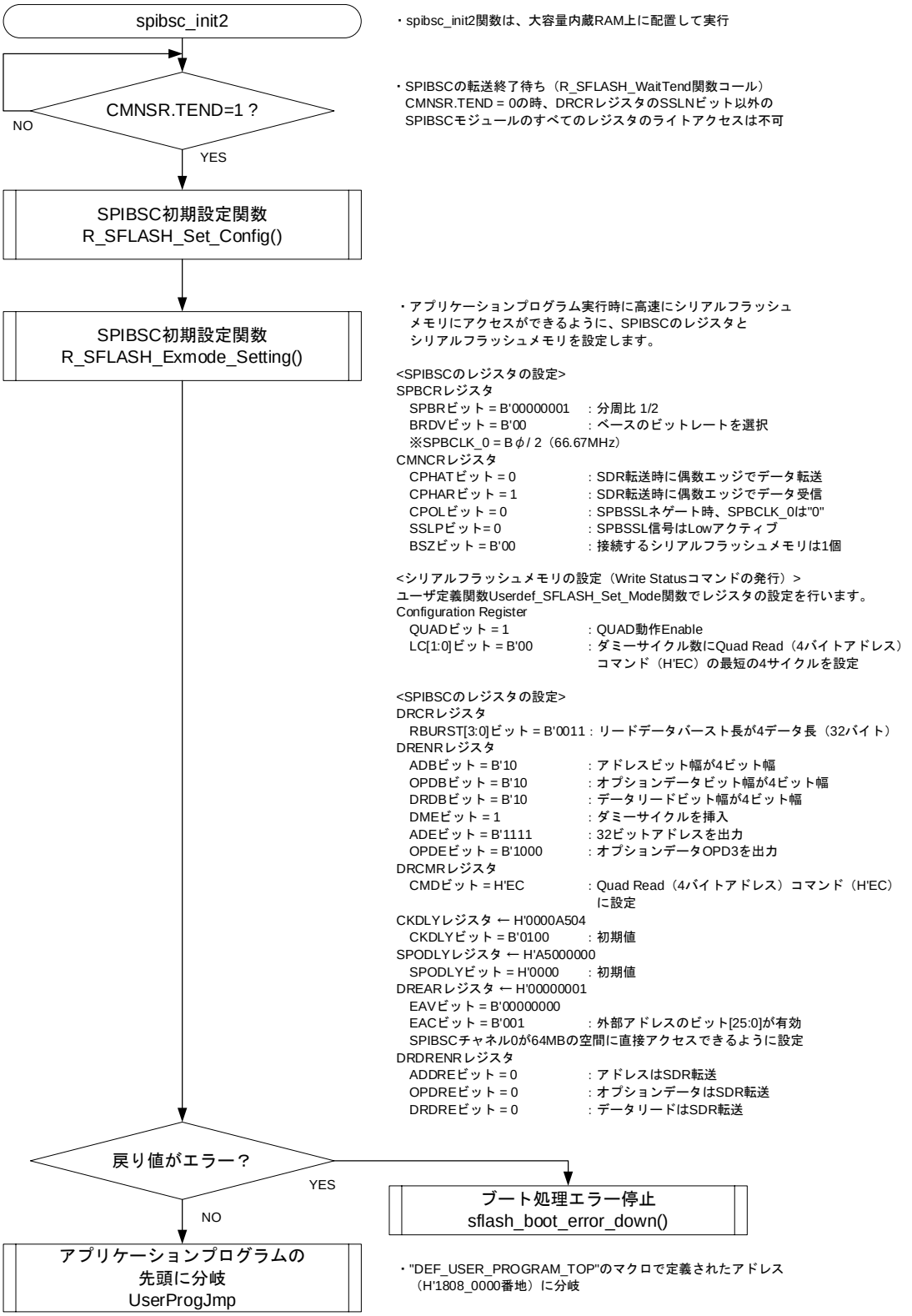
図5.5　ローダプログラム 1 のフローチャート

5.9.3　ローダプログラム 2（STEP2）

ローダプログラム 2 では、さらに高速にシリアルフラッシュメモリにアクセスできるように、シリアルフラッシュメモリのレジスタ（Configuration Register の QUAD ビット、Configuration Register の LC[1:0]ビット）を設定し、バス幅を 1 ビットから 4 ビットに変更します。シリアルフラッシュメモリのレジスタ設定後、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 Read (4B address)"(H'EC)に設定し、SPI クロック周波数を Bφ/2 に変更します。

ローダプログラム 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5.6にローダプログラム2のフローチャートを示します。



【注】 シリアルフラッシュメモリのアクセスを 1 ビット幅から 4 ビット幅に変更するために、Configuration Register に設定を行います。これらのレジスタの設定は、SPIBSC を外部アドレス空間リードモードから SPI 動作モードに切り替えて、WRSR コマンドをシリアルフラッシュメモリに発行する必要があります。WRSR コマンドを発行してレジスタを設定後、SPIBSC を SPI 動作モードから外部アドレス空間リードモードに切り替える処理を行っています。

図5.6　ローダプログラム 2 のフローチャート

6. 応用例

6.1 サンプルコードを初期状態で使用する場合の動作

サンプルコードの初期状態では、CYPRESS社製シリアルフラッシュメモリ（S25FL512S）に対して表6.1に示す設定内容でアクセスしています。

表6.1　サンプルコードのアクセス設定

項目	設定内容
シリアルフラッシュメモリ	・CYPRESS社製シリアルフラッシュメモリ ・型名：S25FL512S ・使用するリードコマンド：H'EC バス幅 4 ビット、SDR 転送 最大動作周波数 80MHz 時に、必要なダミーサイクル数 4
SPI マルチ I/O バス空間に接続するシリアルフラッシュメモリ数	1 個
データバス幅	4 ビット
アドレスバイト数 （アドレス指定時に発行するバイト数）	4 バイト
リード時転送フォーマット	SDR 転送
アイドル時の SPBCLK 端子の出力レベル	SPBSSL ネゲート時に SPBCLK は"0"を出力 （CMNCR.CPOL = 0）
SPIBSC のデータ送信タイミング	SDR 転送時、偶数エッジでデータ送信 （CMNCR.CPHAT = 0）
SPIBSC のデータ受信タイミング	SDR 転送時、偶数エッジでデータ受信 （CMNCR.CPHAR = 1）

図6.1にSDR転送（サンプルコード初期状態）のリード動作を示します。リード動作では、SPIBSC からのコマンド出力、アドレス出力、およびダミーサイクルの出力に続いて、データサイクルが開始し、S25FL512S からリードデータが出力されます。

S25FL512Sは、SDR 転送時に、クロックの立ち上がりエッジで入力データをサンプリングします。SPIBSC はクロックの立ち下がりエッジを基準にデータ出力を開始し、立ち上がりエッジではデータ出力を継続します。S25FL512Sが SPIBSC からの出力データの MSB を、最初の SPBCLK の立ち上がりエッジでサンプリングできるように、CPOL=0、CPHAT=0 の設定としています。

また、S25FL512Sは、SDR 転送時に、クロックの立ち下がりエッジを基準にデータの出力を開始します。データサイクルでは、ダミーサイクルの最終クロックの立ち下がりエッジを基準に、次のクロックの立ち下がりエッジまでデータを出力するため、SPIBSC では CPHAR=1 に設定して、偶数エッジ（CPOL=0 の場合は立ち下がりエッジ）で入力データをサンプリングするようにしています。

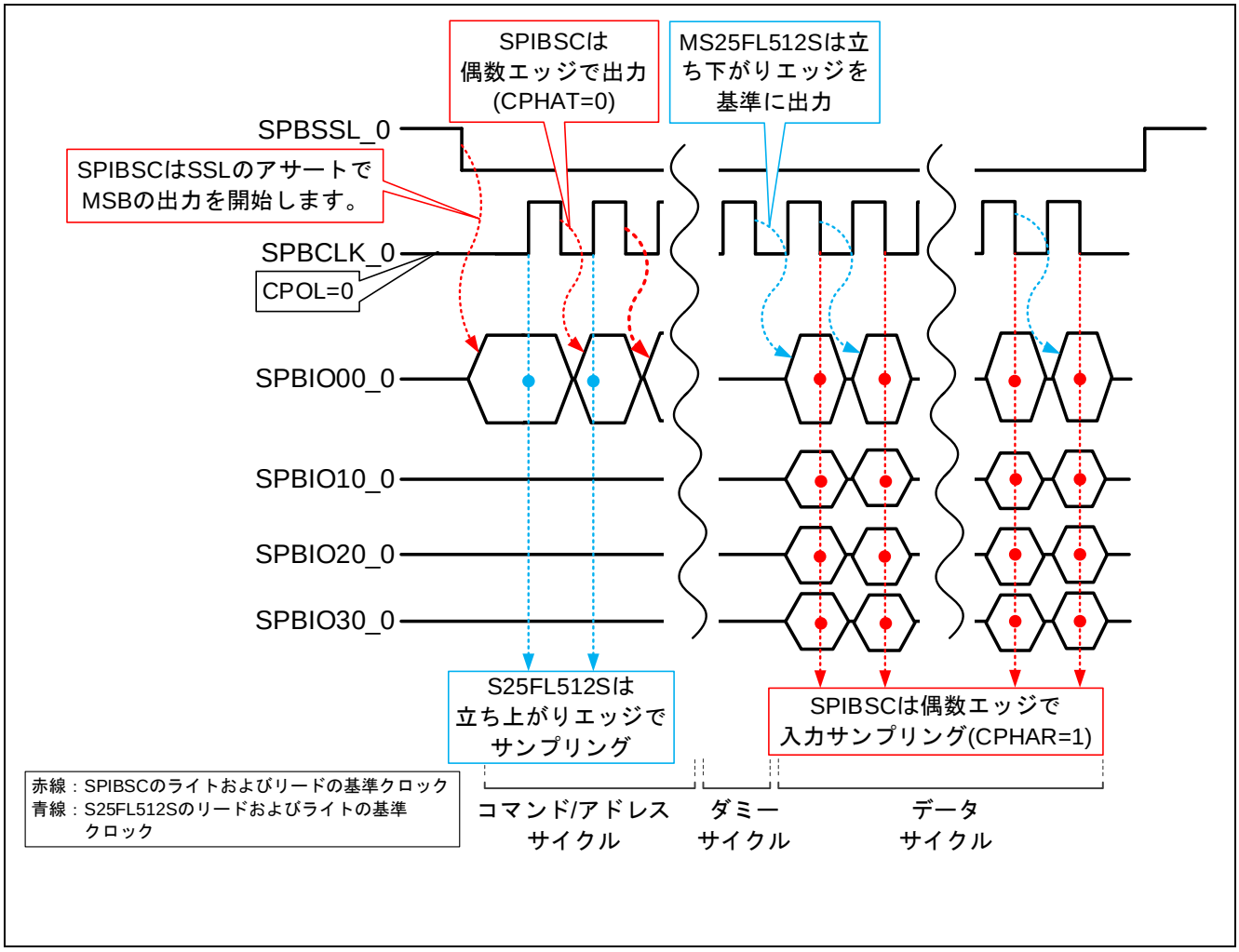


図6.1　SDR 転送（サンプルコード初期状態）のリード動作

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

サンプルコードは、表6.2および表6.3に示すマクロ定義（spibsc_config.h で定義）を変更することで各条件をカスタマイズすることができます。表中の「　　」は、サンプルコードを初期状態で使用する場合の設定です。

初期化時に実行される R_SFLASH_Exmode_Setting 関数で、シリアルフラッシュメモリのレジスタ設定を行うためのユーザ定義関数 Userdef_SFLASH_Set_Mode をコールします。Userdef_SFLASH_Set_Mode 関数では、使用するリードコマンドの仕様に合わせて、表6.2および表6.3で定義する SPIBSC のマクロ値の内容と合うように、表6.4および表6.5に示すS25FL512Sの Configuration Register を設定する処理を行うことで、S25FL512S へのリードアクセスを行うことができます。

表6.2　サンプルコードカスタマイズ用マクロ一覧（1/2）

マクロ名	定義する値	設定内容
DEF_SPIBSC_DUAL_MODE	SPIBSC_CMNCR_BSZ_SINGLE：1 個	CMNCR.BSZ[1:0] = B'00
	SPIBSC_CMNCR_BSZ_DUAL：2 個	CMNCR.BSZ[1:0] = B'01
SPIBSC_BUS_WITDH （注 2）	SPIBSC_1BIT：　1 ビット幅	DRENR.ADB[1:0] = B'00 DRENR.OPDB[1:0] = B'00 DRENR.DRDB[1:0] = B'00
	SPIBSC_4BIT：4 ビット幅	DRENR.ADB[1:0] = B'10 DRENR.OPDB[1:0] = B'10 DRENR.DRDB[1:0] = B'10
SPIBSC_OUTPUT_ADDR （注 2）	SPIBSC_OUTPUT_ADDR_24：3 バイト	DRENR.ADE[3:0] = B'0111
	SPIBSC_OUTPUT_ADDR_32：4 バイト	DRENR.ADE[3:0] = B'1111
SPIBSC_TRANS_MODE （注 2）	SPIBSC_SDR_TRANS：SDR 転送	DRDRENR.ADDRE = 0 DRDRENR.OPDRE = 0 DRDRENR.DRDRE = 0
	SPIBSC_DDR_TRANS：DDR 転送	DRDRENR.ADDRE = 1 DRDRENR.OPDRE = 1 DRDRENR.DRDRE = 1
SPIBSC_READCMD_SETTING （注 1）（注 2）	DRCMR.CMD に設定するシリアルフラッシュメモリのリードコマンド （初期状態では"0xEC"）	DRCMR.CMD[7:0] = H'EC
SPIBSC_DMYCYC_SETTING （注 2）	ダミーサイクル数（初期状態では"4"）	DRDMCR.DMCYC[2:0] = B'001（注 3）

【注】 1. サンプルコードでは、S25FL512Sに発行するコマンドとして、H'0B（FAST READ）、H'0C（FAST READ4B）、H'EB（4READ）、H'EC（4READ4B）のコマンドをサポートしています。

2. SPIBSC_DMYCYC_SETTING　、　SPIBSC_BUS_WITDH　、　SPIBSC_TRANS_MODE　、SPIBSC_OUTPUT_ADDR の内容は、SPIBSC_READCMD_SETTING に設定するリードコマンドの内容に合わせて設定する必要があります。

3. DRDMCR.DMYCYC[2:0]の設定値は、使用するリードコマンドに依存して、S25FL512Sに必要なダミーサイクル数と"High Performance Read Mode indicator"から求めます。サンプルコードでは、ダミーサイクル数は SPIBSC_DMYCYC_SETTING で定義します

表6.3　サンプルコードカスタマイズ用マクロ一覧（2/2）

マクロ名	定義する値	設定内容
SPIBSC_OUTPUT_OPTION_SETTING	出力するオプションデータを OPD3、OPD3/OPD2、OPD3/OPD2/OPD1、OPD3/OPD2/OPD1/OPD0、出力しないより指定	
	SPIBSC_OUTPUT_DISABLE	DRENR.OPDE[3:0] = B'0000
	SPIBSC_OUTPUT_OPD_3	DRENR.OPDE[3:0] = B'1000
	SPIBSC_OUTPUT_OPD_32	DRENR.OPDE[3:0] = B'1100
	SPIBSC_OUTPUT_OPD_321	DRENR.OPDE[3:0] = B'1110
	SPIBSC_OUTPUT_OPD_3210	DRENR.OPDE[3:0] = B'1111
SPIBSC_OUTPUT_OPTION_DATA_OPD3	0x00	DROPR.OPD3[7:0] = H'00
SPIBSC_OUTPUT_OPTION_DATA_OPD2	0x00	DROPR.OPD2[7:0] = H'00
SPIBSC_OUTPUT_OPTION_DATA_OPD1	0x00	DROPR.OPD1[7:0] = H'00
SPIBSC_OUTPUT_OPTION_DATA_OPD0	0x00	DROPR.OPD0[7:0] = H'00
SPIBSC_CPOL_SETTING	SPIBSC_CMNCR_CPOL_LOW： SPBSSL ネゲート時に SPBCLK は"0"を出力	CMNCR.CPOL = 0
	SPIBSC_CMNCR_CPOL_HIGH： SPBSSL ネゲート時に SPBCLK は"1"を出力	CMNCR.CPOL = 1
SPIBSC_CPHAT_SETTING	SPIBSC_CMNCR_CPHAT_EVEN： SDR 転送時、偶数エッジでデータ送信 DDR 転送時、偶数エッジからデータ送信開始	CMNCR.CPHAT = 0
	SPIBSC_CMNCR_CPHAT_ODD： SDR 転送時、奇数エッジでデータ送信 DDR 転送時、奇数エッジからデータ送信開始	CMNCR.CPHAT = 1
SPIBSC_CPHAR_SETTING	SPIBSC_CMNCR_CPHAR_ODD： SDR 転送時、奇数エッジでデータ受信 DDR 転送時、奇数エッジからデータ受信開始	CMNCR.CPHAR = 0
	SPIBSC_CMNCR_CPHAR_EVEN： SDR 転送時、偶数エッジでデータ受信 DDR 転送時、偶数エッジからデータ受信開始	CMNCR.CPHAR = 1
SPIBSC_CKDLY_SETTING	SPIBSC_CKDLY_DEFAULT：初期値	CKDLY.CKDLY[3:0] = B'0100
	SPIBSC_CKDLY_TUNING： データ入力セットアップ時間を縮め、データホールド時間を長くする設定	CKDLY.CKDLY[3:0] = B'1000
SPIBSC_SPODLY_SETTING	SPIBSC_SPODLY_DEFAULT：初期値	SPODLY.SPODLY[15:0] = H'0000
	SPIBSC_SPODLY_TUNING： データ出力遅延/ホールド/バッファオン/バッファオフ時間を遅らせる設定	SPODLY.SPODLY[15:0] = H'6363

表6.4にS25FL512SのStatus Registerを、表6.5にS25FL512SのConfiguration Registerを示します。Userdef_SFLASH_Set_Mode 関数により、表中に「　　」で示したビットを設定し、その他のビットは初期値で使用しています。

表6.4　S25FL512Sの Status Register

ビット	ビット名	属性（注）	説明
7	SRWD	NV	Status register write protect 1 = Status register write disabled 0 = Status register write enabled
6	P_ERR	V, Read only	1 = Error occurred 0 = No Error
5	E_ERR	V Read only	1 = Error occurred 0 = No Error
4,3,2	BP2,BP1,BP0	NV	Level of protected block
1	WEL	V	Write enable latch 1 = Write enable 0 = Not write enable
0	WIP	V	Write in progress bit 1 = Write operation 0 = Not in write operation

【注】 属性の"NV"は"Non-volatile bit"を、"V"は"Volatile bit"を意味します。

表6.5　S25FL512Sの Configuration Register

ビット	ビット名	属性（注 1）	説明
7,6	LC1, LC0	NV	Dummy cycle 1, Dummy cycle 0 LC[1:0] = B'00（注 2）
5	TBPROT	OTP	0 = BP starts at bottom (Low address) 1 = BP starts at top(High address)
4	RFU		Reserved Futue Use
3	BPNV	OTP	1 = Volatile 0 = Nonvolatile
2	RFU		Reserved Future use
1	QUAD	NV	1 = QUAD 0 = Dual or Serial
0	FREEZE	V	1 = block Protection and OTP locked 0 = block Protection and OTP un-locked

【注】 1. 属性の"NV"は"Non-volatile bit"を、"OTP"は"One-time prgrammable bit"を意味します。
2. 表6.6に示すように、使用するリードコマンドによってダミーサイクル数は異なります。サンプルコードでは、リードコマンドに H'EC を使用しており、最適なダミーサイクル数となるように、LC[1:0]ビットに B'00 を設定しています。

表6.6にS25FL512Sの動作周波数に対して必要なダミーサイクル数の一覧を示します。リードコマンドと動作周波数によって、必要なダミーサイクル数が異なります。サンプルコードでは、リードコマンドに H'EC コマンドを使用し、SPBCLK が 66.66MHz で使用しているため、ダミーサイクル数が 4 サイクルの LC[1:0] = B'00 が最適な設定です。

使用するリードコマンドを変更する場合には、リードコマンドと SPBCLK の周波数に合わせて、ダミーサイクル数を設定してください。

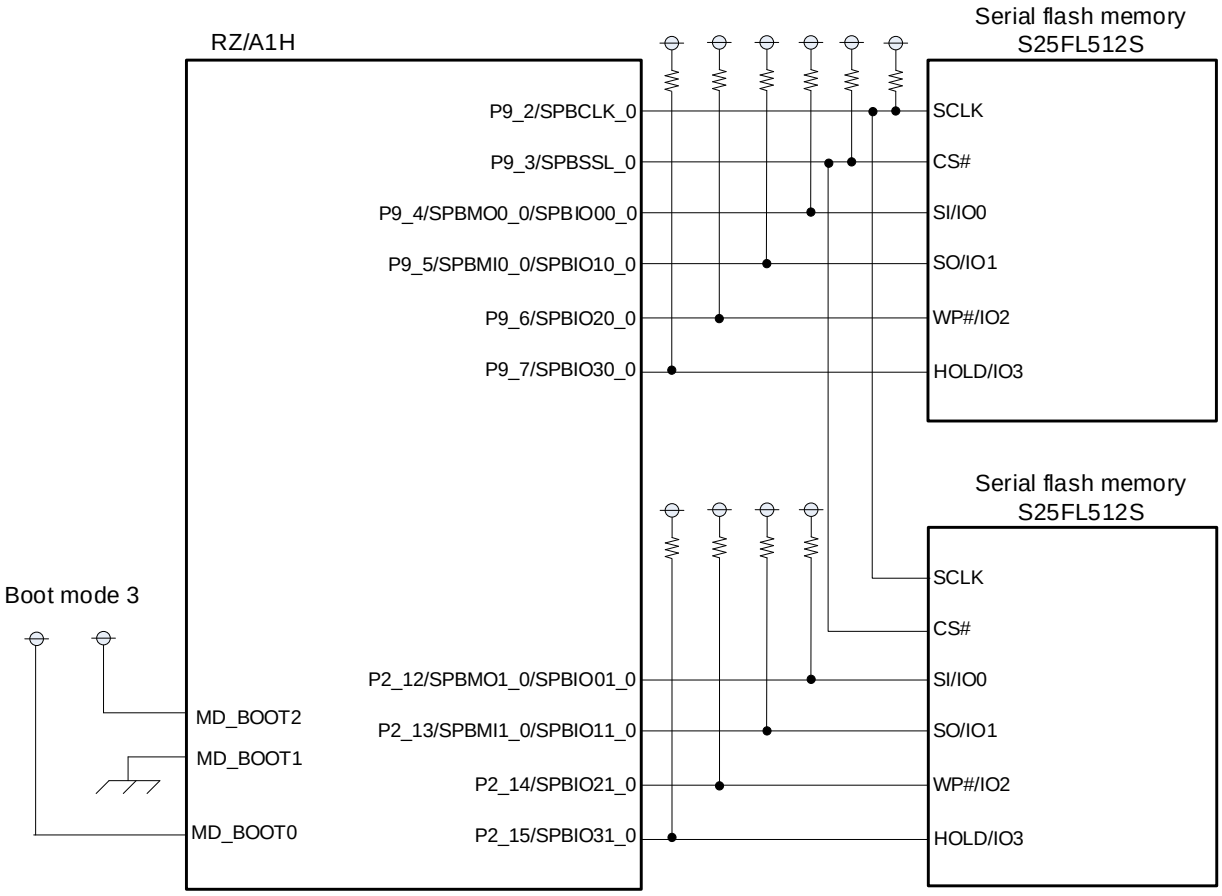
表6.6　S25FL512Sの動作周波数に対して必要なダミーサイクル数の一覧

リードコマンド	LC[1:0]			
	B'00	B'01	B'10	B'11
FAST READ（H'0B） FAST READ4B（H'0C）	8 サイクル /80MHz	8 サイクル /90MHz	8 サイクル /104MHz	0 サイクル /50MHz
4READ（H'EB） 4READ4B（H'EC）	4 サイクル /80MHz	4 サイクル /90MHz	5 サイクル /104MHz	1 サイクル /50MHz

6.2.1　シリアルフラッシュメモリを 2 個接続（8 ビット幅アクセス）に変更する方法

SPI マルチ I/O バス空間に接続するシリアルフラッシュメモリを 2 個使用することで、8 ビット幅でシリアルフラッシュメモリにアクセスすることが可能になります。

図6.2にシリアルフラッシュメモリを2個（8ビット幅）接続する場合の接続例を示します。



【注】 RZ/A1Hでは、シリアルフラッシュメモリを 2 個接続する場合、P2_12～P2_15 端子を使用します。これらの端子は、アドドレスバス（A14～A11）と共用端子になっています。Renesas Starter Kit+ for RZ/A1Hでは、CS3 空間に SDRAM（Samsung 社製 K4S561632D）が配置されていますが、シリアルフラッシュメモリを 2 個接続して使用する場合は、SDRAM 制御信号との共用端子を使用するため、SDRAM を使用することはできません。

図6.2　シリアルフラッシュメモリを 2 個（8 ビット幅）接続する場合の接続例

フラッシュメモリを 2 個接続して使用するために、P2_12～P2_15 端子を、それぞれ、SPBMO1_0/SPBIO01_0、SPBMI1_0/SPBIO11_0、SPBIO21_0、SPBIO31_0 の端子機能に設定する必要があります。サンプルコードでは、io_spibsc_port_setting 関数で兼用端子機能の設定を行っています。

SPI マルチ I/O バス空間に接続するシリアルフラッシュメモリを 2 個接続に変更するには、サンプルコードの初期状態から表6.7に示すマクロ定義を変更してください。

表6.7　サンプルコードカスタマイズ用マクロー覧（2 個接続時）

マクロ名	定義する値	設定内容
DEF_SPIBSC_DUAL_MODE	SPIBSC_CMNCR_BSZ_DUAL：2 個	CMNCR.BSZ[1:0] = B'01

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、シリアルフラッシュメモリの SPI マルチ I/O バス空間に配置されるアドレスが異なります。ただし、ブート起動用内蔵 ROM プログラム実行後の SPIBSC は、アドレスバイト数が 3 バイト（最大 16MB の領域）でシリアルフラッシュメモリを 1 個接続する設定で、H'1800_0000 番地（シリアルフラッシュメモリの H'0000_0000 番地）に分岐します。ローダプログラムは、SPBMO0_0/SPBIO00_0、SPBMI0_0/SPBIO10_0 に接続されたシリアルフラッシュメモリの H'0000_0000 番地を先頭に 16MB の領域の範囲内に配置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メモリのセクタ 0～7（H'0000_0000～H'0000_7FFF 番地）に配置しています。

アプリケーションプログラムは、ローダプログラムによってアドレスバイト数が 4 バイト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アドレスに配置することができます。

図6.3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。

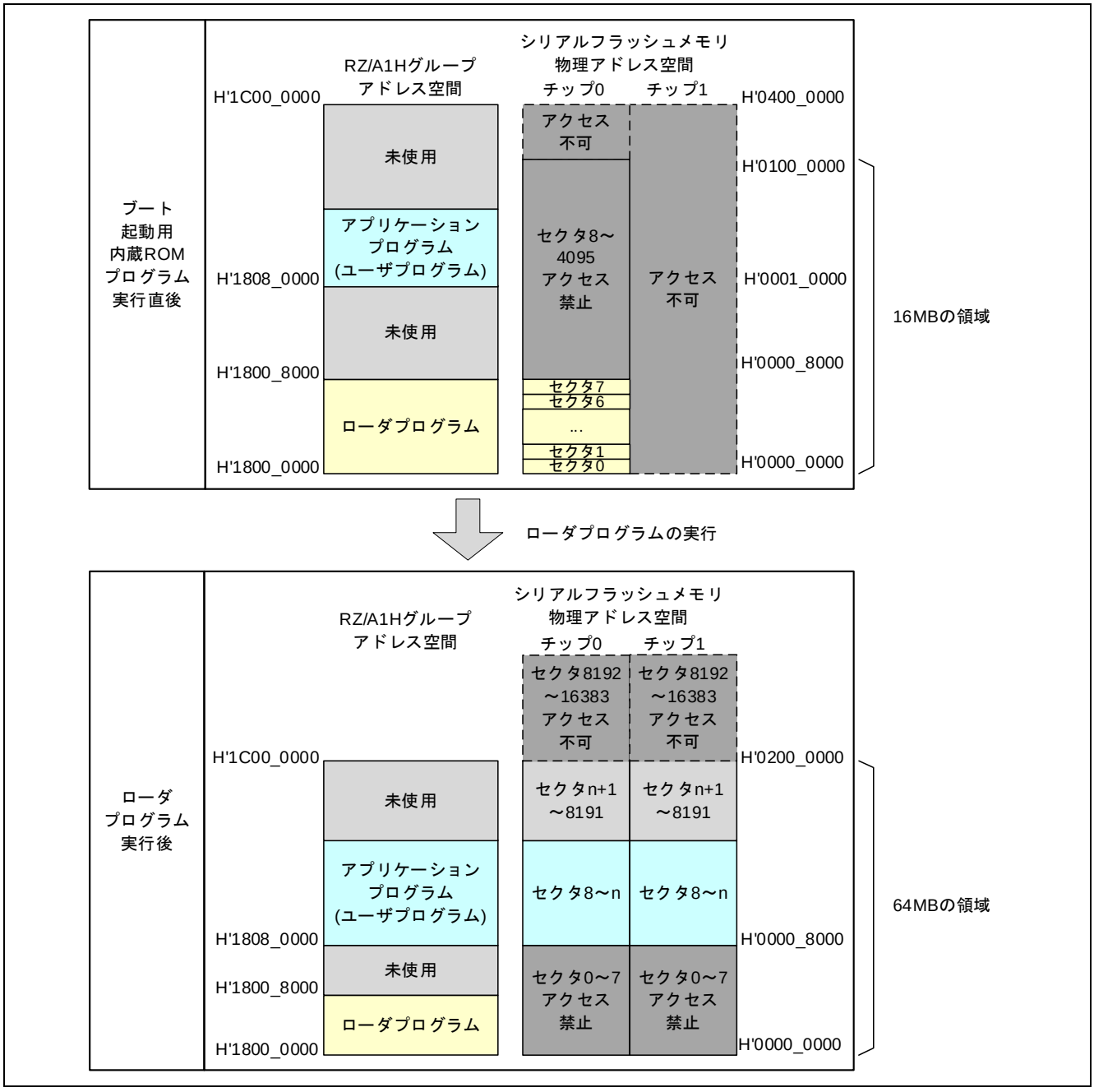


図6.3　シリアルフラッシュメモリを 2 個接続する場合のプログラム配置の例

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6.9にサンプルコードの変更のポイント示します。

表6.8　サンプルコードの変更のポイント

変更のポイント	内容
リードコマンド発行時の出力信号	使用するシリアルフラッシュメモリのリードコマンドの仕様に合わせて、外部アドレス空間リードモードのリードコマンド発行時にシリアルフラッシュメモリに出力する信号を変更します。
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に合わせて、SPIBSC を外部アドレス空間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に合わせて、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、ライト許可に設定します（注）。
シリアルフラッシュメモリライト完了待ち	使用するシリアルフラッシュメモリに合わせて、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のライト完了を待ちます。

【注】 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にライトするために、ライト許可が必要な場合があります。

表6.2に示した処理は、ローダプログラム 2（spibsc_init2 関数）で実行しており、サンプルコードのユーザ定義関数の処理を、使用するシリアルフラッシュメモリに合わせて変更することによって対応することができます。図6.5にローダプログラム2のモジュール階層図を示し、それぞれの処理について6.3.1～6.3.4にサンプルコードで実施している処理の内容を示します。

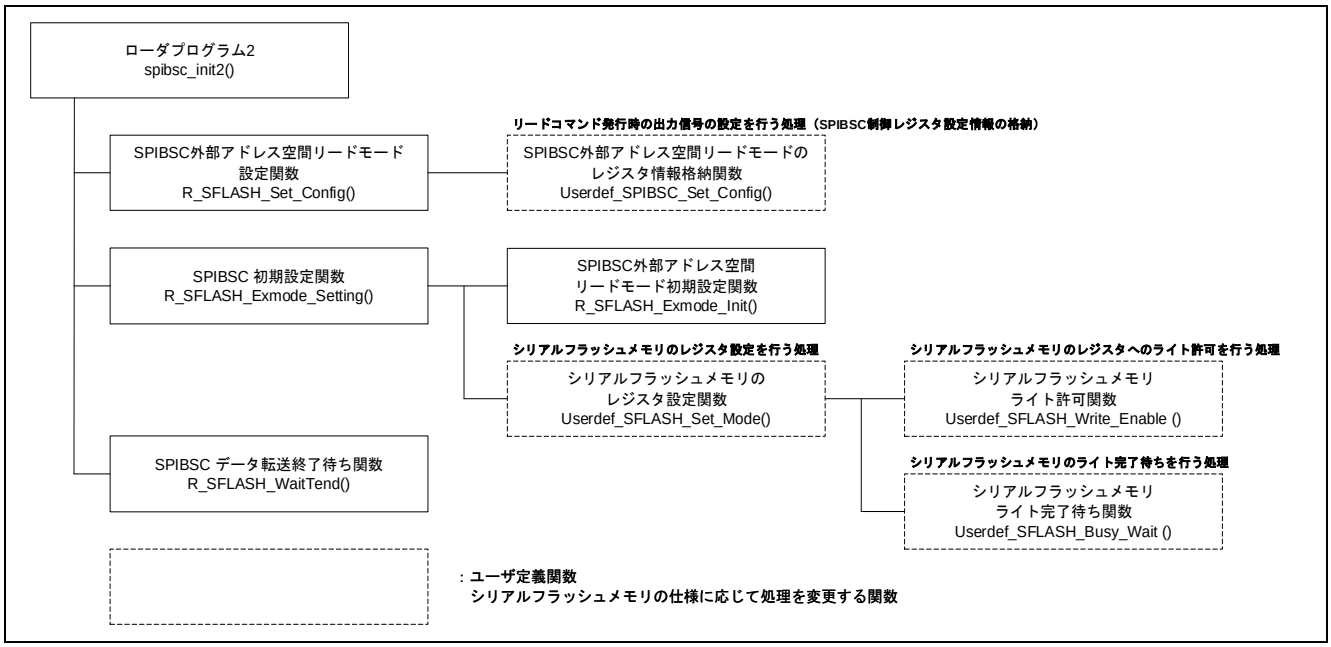


図6.4　ローダプログラム 2 のモジュール階層図

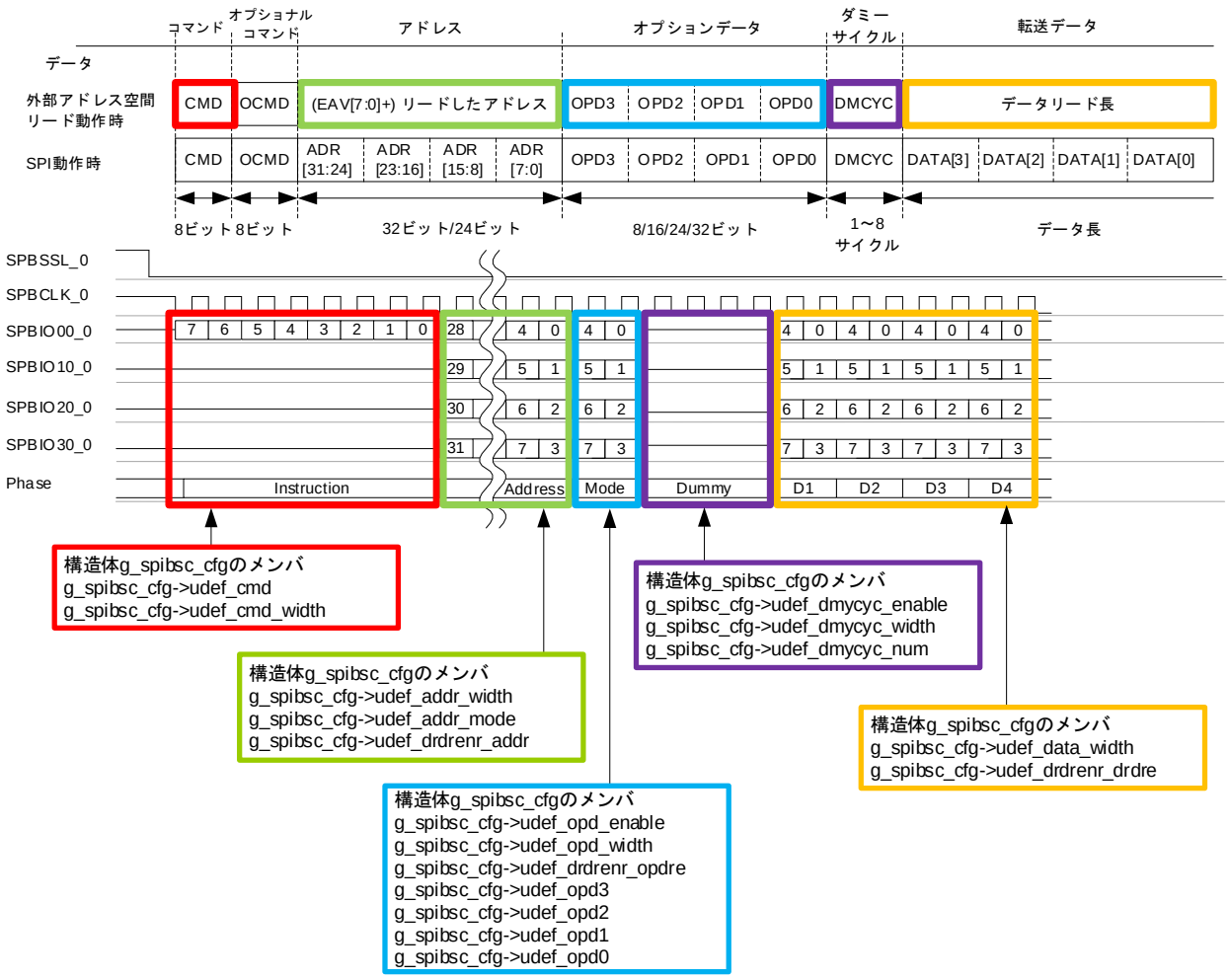
6.3.1　リードコマンド発行時の出力信号

外部アドレス空間リードモードでは、SPI マルチ I/O バス空間へのリードアクセスは、リードコマンド発行時にSPI通信に変換した信号をシリアルフラッシュメモリに出力することにより、リード動作を開始します。使用するシリアルフラッシュメモリを変更する場合は、シリアルフラッシュメモリのリードコマンド仕様に合わせて、リードコマンド発行時の出力信号を変更します。

SPIBSC は、SPIBSC 制御レジスタを設定することにより、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 外部アドレス空間リードモードの設定内容格納変数：g_spibsc_cfg）にて変更することが可能です。g_spibsc_cfgの設定は SPIBSC 外部アドレス空間リードモード設定関数（R_SFLASH_Set_Config）から実行されるユーザ定義関数（SPIBSC 外部アドレス空間リードモードのレジスタ情報格納関数：Userdef_SPIBSC_Set_Config）にて行っています。表5.13～表5.21、および使用するシリアルフラッシュメモリの仕様に合わせて、Userdef_SPIBSC_Set_Config 関数の実装を変更してください。

図6.6にSPIBSC制御レジスタとSPIBSC外部アドレス空間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示します。サンプルコードの内容を参照して、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。



サンプルコードでは、CYPRESS社製シリアルフラッシュメモリ（S25FL512S）を使用時に、各構造体のメンバを以下のように設定しています。

g_spibsc_cfg->udef_cmd	: SPIBSC_READCMD_SETTING (H'EC)
g_spibsc_cfg->udef_cmd_width	: SPIBSC_1BIT
g_spibsc_cfg->udef_addr_width	: SPIBSC_BUS_WITDH (SPIBSC_4BIT)
g_spibsc_cfg->udef_addr_mode	: SPIBSC_OUTPUT_ADDR (SPIBSC_OUTPUT_ADDR_32)
g_spibsc_cfg->udef_drdrenr_addr	: SPIBSC_TRANS_MODE (SPIBSC_SDR_TRANS)
g_spibsc_cfg->udef_opd_enable	: SPIBSC_OUTPUT_OPTION_SETTING (SPIBSC_OUTPUT_OPD_3)
g_spibsc_cfg->udef_opd_width	: SPIBSC_BUS_WITDH (SPIBSC_4BIT)
g_spibsc_cfg->udef_drdrenr_opdre	: SPIBSC_TRANS_MODE (SPIBSC_SDR_TRANS)
g_spibsc_cfg->udef_opd3	: SPIBSC_OUTPUT_OPTION_DATA_OPD3 (H'00)
g_spibsc_cfg->udef_opd2	: SPIBSC_OUTPUT_OPTION_DATA_OPD2 (OPD2は出力しない。)
g_spibsc_cfg->udef_opd1	: SPIBSC_OUTPUT_OPTION_DATA_OPD1 (OPD1は出力しない。)
g_spibsc_cfg->udef_opd0	: SPIBSC_OUTPUT_OPTION_DATA_OPD0 (OPD0は出力しない。)
g_spibsc_cfg->udef_dmycyc_enable	: SPIBSC_DUMMY_CYC_ENABLE
g_spibsc_cfg->udef_dmycyc_width	: SPIBSC_1BIT
g_spibsc_cfg->udef_dmycyc_num	: SPIBSC_DMYCYC_SETTINGに指定された値から算出。
g_spibsc_cfg->udef_data_width	: SPIBSC_BUS_WITDH (SPIBSC_4BIT)
g_spibsc_cfg->udef_drdrenr_drdre	: SPIBSC_TRANS_MODE (SPIBSC_SDR_TRANS)

【注】　図中の Mode サイクルは、S25FL512Sでは High Performance Read Mode　のサイクルを示します。

図6.5　SPIBSC 制御レジスタと SPIBSC 外部アドレス空間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のレジスタ設定

サンプルコードでは、初期化時に実行される R_SFLASH_Exmode_Setting 関数で、シリアルフラッシュメモリの設定を行うためにユーザ定義関数 Userdef_SFLASH_Set_Mode を呼び出します。

「6.3.1　リードコマンド発行時の出力信号」で示したグローバル変数g_spibsc_cfgの内容に合わせて、シリアルフラッシュメモリのレジスタを設定します。

Userdef_SFLASH_Set_Mode 関数からコールされる write_status 関数では、Status Register および Configuration Register にライトする前に、シリアルフラッシュメモリのライトを許可するため、Userdef_SFLASH_Write_Enable をコールし Write Enable コマンドを発行しています。その後、ライト許可状態であることを確認するために、Userdef_SFLASH_Busy_Wait 関数をコールしています。ご使用のシリアルフラッシュメモリの仕様に合わせて、Userdef_SFLASH_Set_Mode 関数を実装してください。

図6.7に、サンプルコードのUserdef_SFLASH_Set_Mode関数のフローを示します。

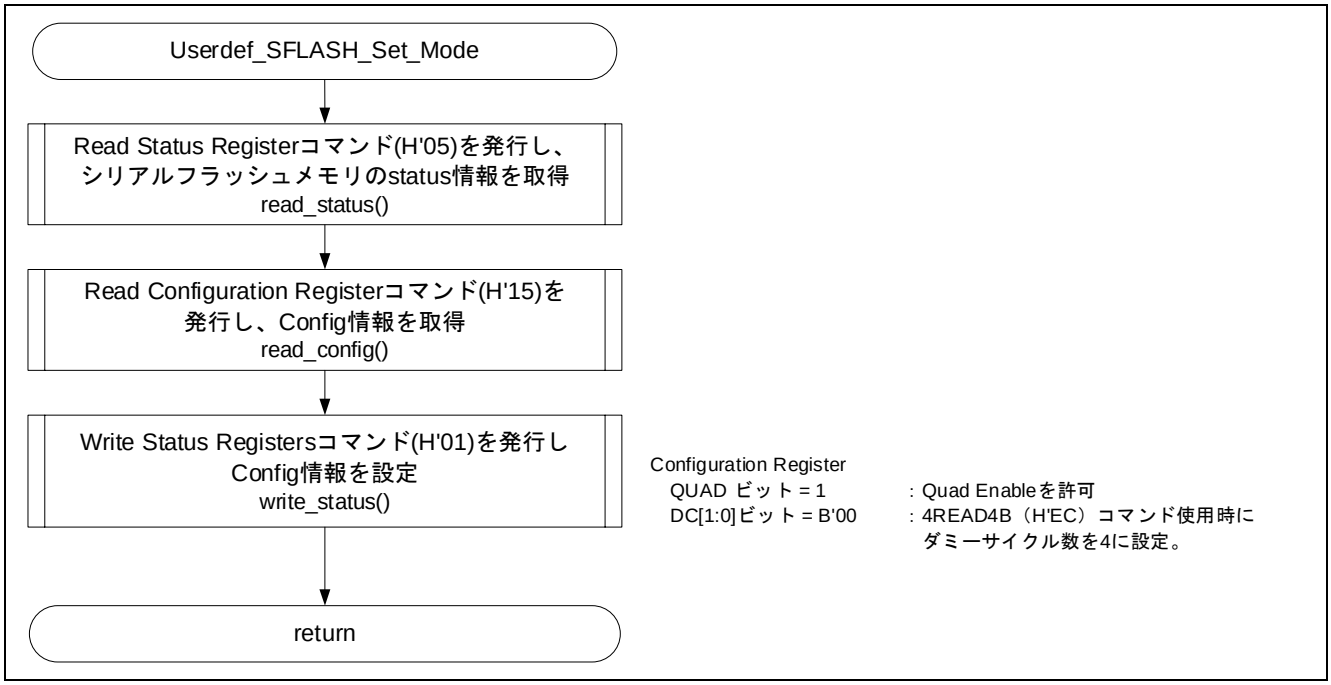


図6.6　Userdef_SFLASH_Set_Mode 関数のフロー

6.3.3　シリアルフラッシュメモリライト許可

シリアルフラッシュメモリのレジスタ（Status Register および Configuration Register）にライトするためには、事前にシリアルフラッシュメモリのライトを許可にしておく必要があります。

ご使用のシリアルフラッシュメモリの仕様に合わせて、シリアルフラッシュメモリのライト許可が行えるように、Userdef_SFLASH_Write_Enable 関数を実装してください。

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 コマンド（H'06）を発行することで、ライト許可（Status Register の WEL ビットを"1"に設定）に変更する処理を行います。

図6.8に、サンプルコードのUserdef_SFLASH_Write_Enable関数のフローを示します。

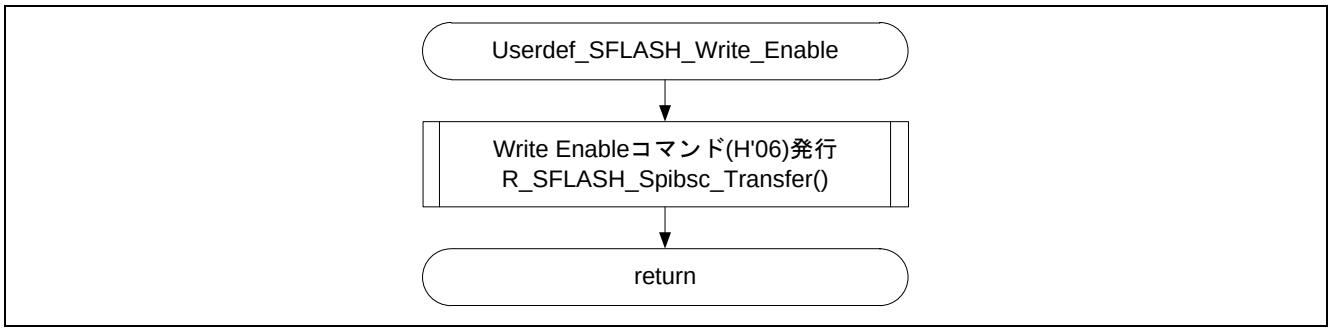


図6.7　Userdef_SFLASH_Write_Enable 関数のフロー

6.3.4　シリアルフラッシュメモリライト完了待ち

シリアルフラッシュメモリのレジスタ（Status Register および Configuration Register）にライトした場合、シリアルフラッシュメモリはビジー状態に遷移します。ビジー状態からレジスタにライトしたデータが反映されるまでウェイトする必要があります。

ご使用のシリアルフラッシュメモリの仕様に合わせて、シリアルフラッシュメモリのライト完了までウェイトするように、Userdef_SFLASH_Busy_Wait 関数を実装してください。

サンプルコードでは、Status Register の WIP ビットをリードしライト完了をウェイトする処理を行います。

図6.9に、サンプルコードのUserdef_SFLASH_Busy_Wait関数フローを示します。

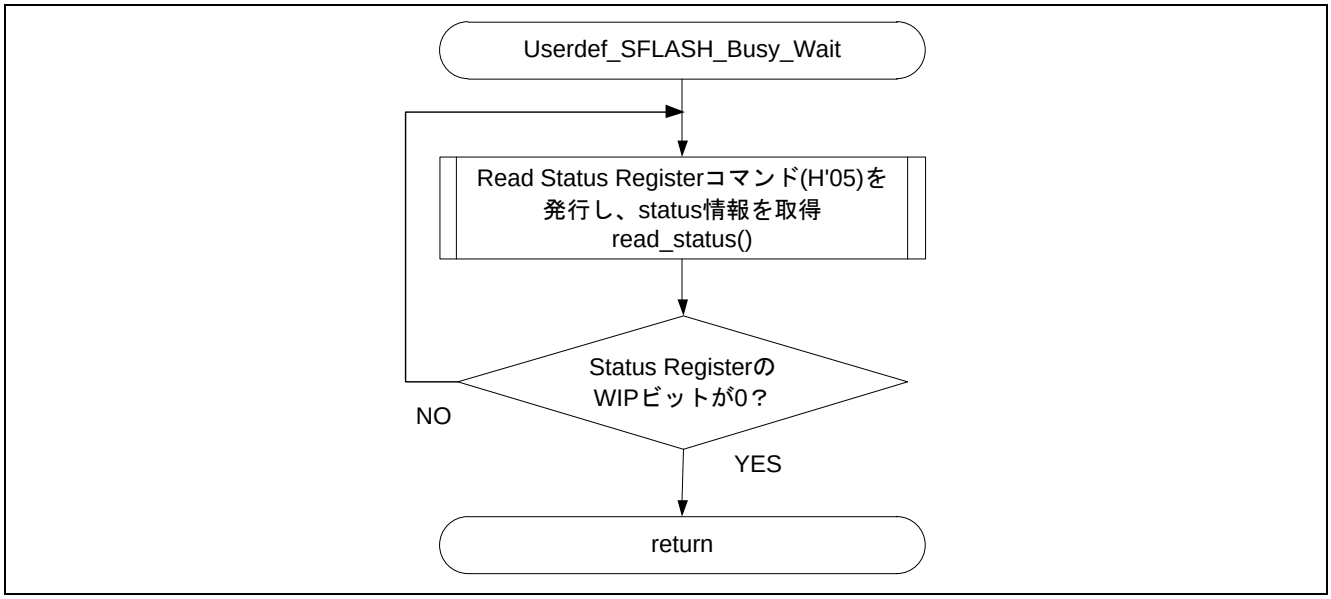


図6.8　Userdef_SFLASH_Busy_Wait 関数フロー

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/A1L グループ、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。）

RZ/A1H AVB ボード YR0K77210C000BE（Renesas Starter Kit+ for RZ/A1H）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。）

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	2020.08.19	ー	新規作成

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　〒135-0061　東京都江東区豊洲3-2-24（豊洲フォレシア）

■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/