

## RZ/A2Mグループ

### SPIマルチI/Oバスコントローラを使用したOctaFlash™ からのブート例

---

#### 要旨

本アプリケーションノートは、RZ/A2M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 4（Octal-SPI フラッシュブート 1.8V 品）によって OctaFlash 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/A2M

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	4
1.1 OctaFlashからのブート	4
1.2 使用する周辺機能	6
2. 動作確認条件	8
3. 関連アプリケーションノート	9
4. ハードウェア説明	10
4.1 ハードウェア構成例	10
4.2 使用端子一覧	11
5. ソフトウェア説明	12
5.1 動作概要	12
5.1.1 OctaFlashブートに関する用語	12
5.1.2 サンプルコード全体の動作概要	13
5.1.3 ローダプログラムの動作概要	14
5.1.4 アプリケーションプログラム	18
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	20
5.2.1 周辺機能の設定	20
5.2.2 メモリマップ	21
5.2.3 サンプルコードのセクション配置	22
5.3 使用割り込み一覧	23
5.4 データ型一覧	23
5.5 定数一覧	24
5.6 構造体/共用体一覧	26
5.7 変数一覧	39
5.8 関数一覧	40
5.9 関数仕様	42
5.10 ローダプログラムのフローチャート	53
5.10.1 ローダプログラム（全体）	53
5.10.2 ブートに使用するハードウェアの初期設定	54
5.10.3 SPIBSCとOctaFlashの初期設定	55
5.10.4 SPIBSC初期設定	56
5.10.5 SPIBSC動作モード設定	58
5.10.6 OctaFlashへのSPIコマンド発行	61
6. 応用例	67
6.1 サンプルコードを初期状態で使用する場合の動作	67
6.2 OctaFlashを変更する場合のサンプルコード変更方法	70
6.2.1 外部アドレス空間リードモードでのリードコマンド発行時の出力信号	72
6.2.2 手動モードでのコマンド発行時の出力信号	75
6.2.3 OctaFlashのレジスタの設定	77
6.2.4 OctaFlashのライト完了待ち	78

## RZ/A2Mグループ SPIマルチI/Oバスコントローラを使用したOctaFlash™ からのブート例

6.2.5	OctaFlashのステータスレジスタのリード	79
6.2.6	OctaFlashのコンフィグレーションレジスタのリード	82
6.2.7	OctaFlashのコンフィグレーションレジスタ2のリード	85
6.2.8	OctaFlashのライト許可	88
6.2.9	OctaFlashのステータス/コンフィグレーションレジスタのライト	91
6.2.10	OctaFlashのコンフィグレーションレジスタ2のライト	95
6.2.11	OctaFlashのID情報のリード	98
6.2.12	OctaFlashのデータリード	101
6.2.13	OctaFlashのセクタイレース	103
6.2.14	OctaFlashのデータライト	105
7.	サンプルコードの注意事項	107
7.1	外部アドレス空間リードモードでアクセス可能な領域	107
7.2	OctaFlashに発行するコマンド	107
8.	サンプルコード	108
9.	参考ドキュメント	108
	改訂記録	109

## 1. 仕様

### 1.1 OctaFlash からのブート

RZ/A2Mは、ブートモード4の場合、SPIマルチI/Oバス空間に配置されたOctal-SPIフラッシュメモリからブートします。本アプリケーションノートでは、Octal-SPIフラッシュメモリにOctaFlashを使用してブートする例（以下、OctaFlashブートとします）について説明します。

図1.1にOctaFlashブートの動作イメージを示します。

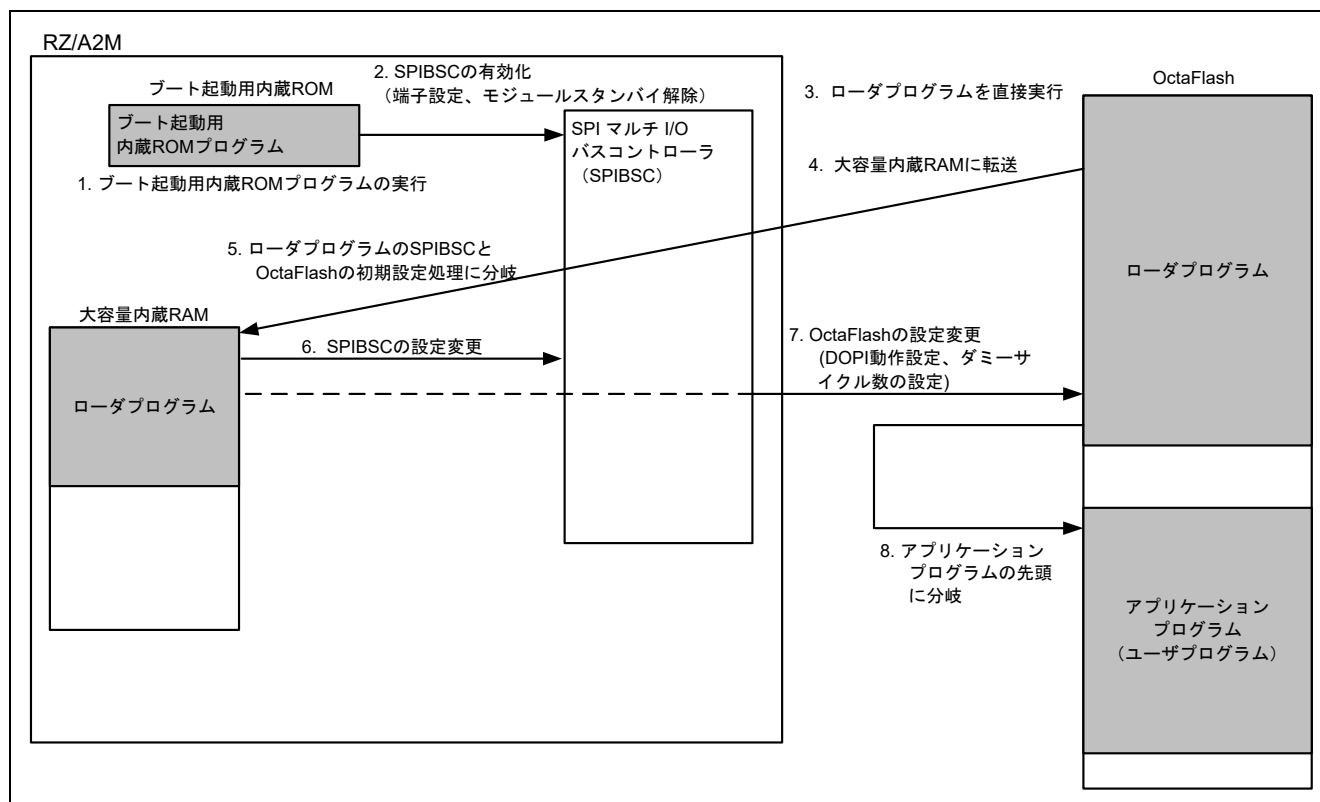


図1.1 OctaFlash ブートの動作イメージ

OctaFlash ブートの動作イメージについて説明します。

- 1 RZ/A2Mは、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、OctaFlash ブートで起動した場合、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 OctaFlash 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 OctaFlash 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 OctaFlash 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、OctaFlash 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 Octal-SPI 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 Octal-SPI 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'2000\_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 OctaFlash 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。

## 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、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、SPI マルチ I/O バスコントローラを SPIBSC、クロックパルス発振器を CPG、割り込みコントローラを INTC、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 SCIFA、汎用入出力ポートを GPIO、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 OctaFlash から、直接リードするための信号を生成。
クロックパルス発振器 (CPG)	RZ/A2Mの動作周波数の生成
割り込みコントローラ (INTC)	OSTM チャンネル 0、OSTM チャンネル 2 および SCIFA チャンネル 4 の割り込み制御に使用
OS タイマ (OSTM)	OSTM チャンネル 0 およびチャンネル 2 を使用 <ul style="list-style-type: none"> <li>OSTM チャンネル 0 LED 点灯および消灯の周期を生成</li> <li>OSTM チャンネル 2 OS Abstraction Layer による時間管理に使用</li> </ul>
FIFO 内蔵シリアルコミュニケーションインタフェース (SCIFA)	SCIFA チャンネル 4 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 (GPIO)	SCIFA チャンネル 4 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/A2M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/A2M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

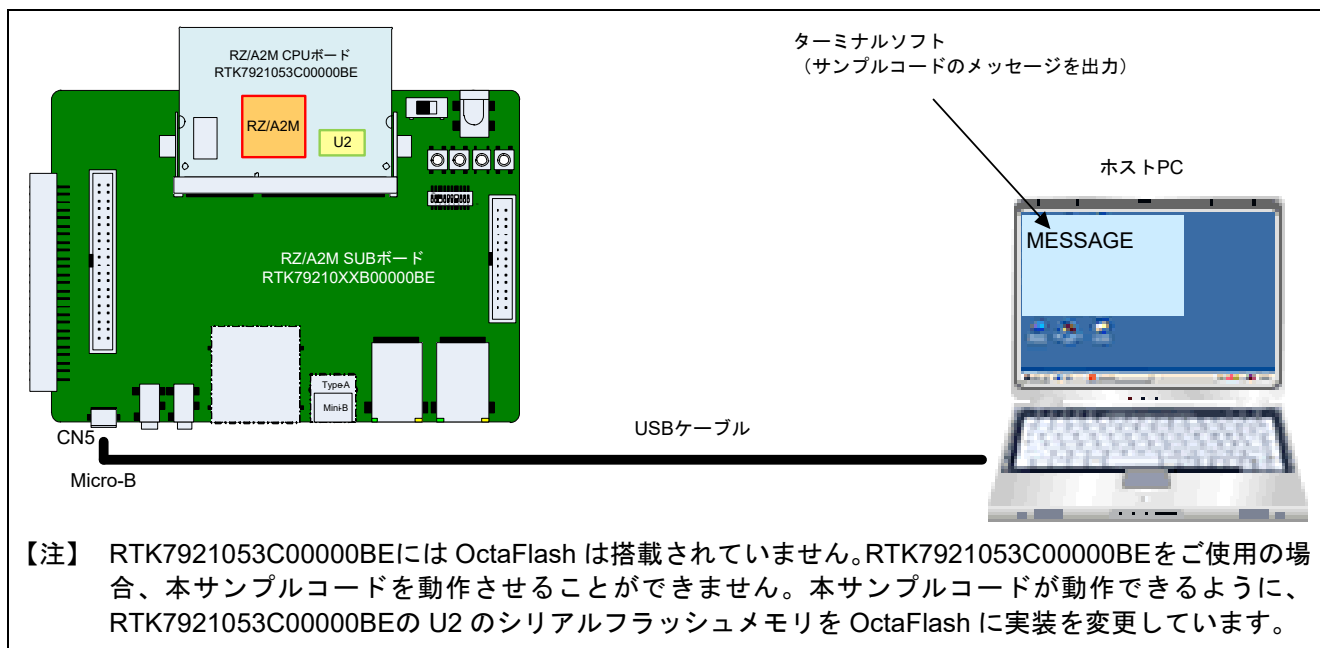


図1.2 動作環境

## 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 (1/2)

項目	内容
使用 MCU	RZ/A2M
動作周波数 (注)	CPU クロック (Iφ) : 528MHz 画像処理クロック (Gφ) : 264MHz 内部バスクロック (Bφ) : 132MHz 周辺クロック 1 (P1φ) : 66MHz 周辺クロック 0 (P0φ) : 33MHz QSPI0_SPCLK : 132MHz CKIO : 132MHz
動作電圧	電源電圧 (I/O) : 3.3V 電源電圧 (1.8/3.3V 切替 I/O (PVcc_SPI) ) : 1.8V 電源電圧 (内部) : 1.2V
統合開発環境	e2 studio V7.7.0
C コンパイラ	GNU Arm Embedded Toolchain 6-2017-q2-update コンパイラオプション (ディレクトリパスの追加は除く) Release コンフィグレーション : -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -Wstack-usage=100 -fabi-version=0 Hardware Debug コンフィグレーション : -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -g3 -Wstack-usage=100 -fabi-version=0

【注】 クロックモード 1 (EXTAL 端子からの 24MHz のクロック入力) で使用時の動作周波数です。



表2.2 動作確認条件 (2/2)

項目	内容
動作モード	ブートモード 4 (Octal-SPI フラッシュブート 1.8V 品)
ターミナルソフトの通信設定	<ul style="list-style-type: none"> <li>通信速度：115200bps</li> <li>データ長：8 ビット</li> <li>パリティ：なし</li> <li>ストップビット長：1 ビット</li> <li>フロー制御：なし</li> </ul>
使用ボード	RZ/A2M CPUボード RTK7921053C00000BE (注) RZ/A2M SUBボード RTK79210XXB00000BE
使用デバイス (ボード上で使用する機能)	<ul style="list-style-type: none"> <li>OctaFlash (SPI マルチ I/O バス空間に接続) メーカー名：Macronix社、型名：MX25UM51245G</li> <li>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)</li> <li>LED1</li> </ul>

【注】 RTK7921053C00000BEには OctaFlash は搭載されていません。RTK7921053C00000BEをご使用の場合、本サンプルコードを動作させることができません。本サンプルコードが動作できるように、RTK7921053C00000BEの U2 のシリアルフラッシュメモリを OctaFlash に実装を変更しています。

### 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/A2M グループ 初期設定例 (R01AN4321JJ)

## 4. ハードウェア説明

### 4.1 ハードウェア構成例

本アプリケーションノートで紹介する OctaFlash ブート例は、ブートモード 4 を使用して、SPI マルチ I/O バス空間に接続された OctaFlash に格納されたプログラムにより処理を行います。図4.1にブートモード4で OctaFlashからブートする場合の接続例を示します。

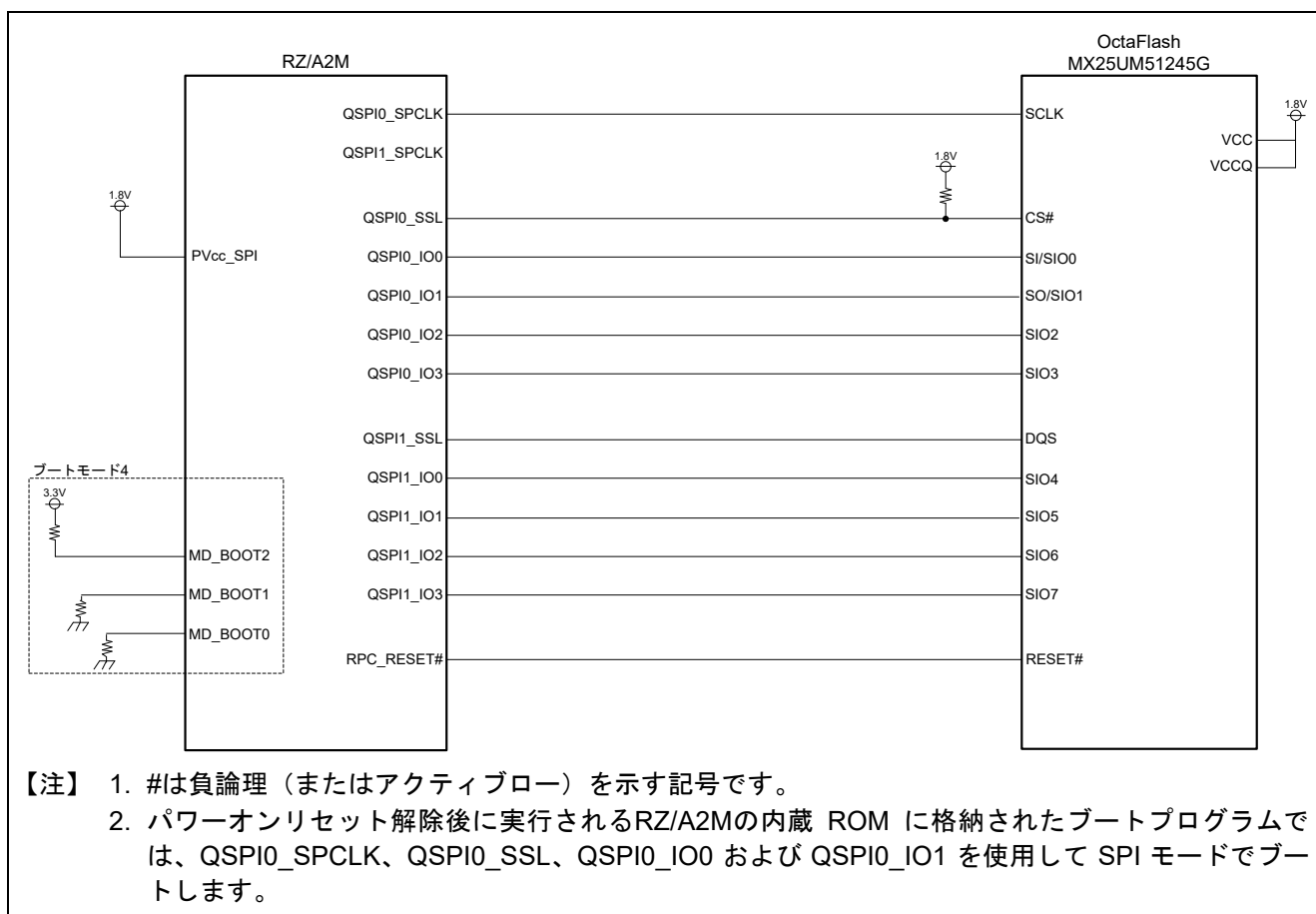


図4.1 ブートモード 4 で OctaFlash からブートする場合の接続例

## 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
MD_BOOT2	入力	ブートモードの選択（ブートモード4に設定） MD_BOOT2 : "H"、MD_BOOT1 : "L"、MD_BOOT0 : "L"
MD_BOOT1	入力	
MD_BOOT0	入力	
QSPI0_SPCLK	出力	OctaFlash のクロック
QSPI0_SSL	出力	OctaFlash のスレーブセレクト
QSPI0_IO0	入出力	OctaFlash のデータ 0
QSPI0_IO1	入出力	OctaFlash のデータ 1
QSPI0_IO2	入出力	OctaFlash のデータ 2
QSPI0_IO3	入出力	OctaFlash のデータ 3
QSPI1_SSL	入力	OctaFlash のデータストローク
QSPI1_IO0	入出力	OctaFlash のデータ 4
QSPI1_IO1	入出力	OctaFlash のデータ 5
QSPI1_IO2	入出力	OctaFlash のデータ 6
QSPI1_IO3	入出力	OctaFlash のデータ 7
RPC_RESET#	出力	OctaFlash のリセット
P6_0	出力	LED の点灯および消灯
RxD4(P9_1)	入力	シリアル受信データ信号
TxD4(P9_0)	出力	シリアル送信データ信号

【注】 #は負論理（またはアクティブロー）を示す記号です。

## 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 OctaFlash 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 OctaFlash ブート動作に関する用語を示します。

表5.1 OctaFlash 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	<p>ブート起動用内蔵 ROM プログラムは、ブートモード 4 (Octal-SPI フラッシュブート 1.8V 品) で起動した場合に、SPI マルチ I/O バス空間に接続された Octal-SPI 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。このとき、Octal-SPI フラッシュメモリには、SPI モードでアクセスを行います。</p> <p>RZ/A2M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'2000_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 Octal-SPI 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。</p> <p>RZ/A2Mの内蔵 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。</p>
ローダプログラム	<p>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。ローダプログラムでは、Octal-SPI フラッシュメモリとして OctaFlash を使用して、OctaFlash に最適にアクセスする処理を行います。</p> <p>ローダプログラムは、お客様が使用する OctaFlash に合わせて、SPIBSC および OctaFlash 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。</p> <p>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 OctaFlash 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、Macronix社製 OctaFlash (MX25UM51245G) を使用する場合に最適な設定を行っています。</p>
アプリケーションプログラム (ユーザプログラム)	<p>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。</p>

### 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 プログラムの処理完了後に実行す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（プロジェクト名：rza2m\_spibsc\_octaflash\_boot\_loader\_gcc）

ローダプログラムは、使用する OctaFlash（Macronix社製 OctaFlash（MX25UM51245G））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'2000\_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。

アプリケーションプログラムの先頭アドレスは、「linker\_script.ld」のリンクスクリプトのシンボル定義 `"__application_base_address"`により指定しています。

#### 2 アプリケーションプログラム（プロジェクト名：rza2m\_spibsc\_octaflash\_boot\_sample\_osless\_gcc）

アプリケーションプログラムは、ローダプログラムにて OctaFlash に最適な設定後に実行するプログラムです。アプリケーションプログラムの"VECTOR\_TABLE"のセクションが、`"__application_base_address"`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。

サンプルコードでは、アプリケーションプログラムを H'2001\_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。

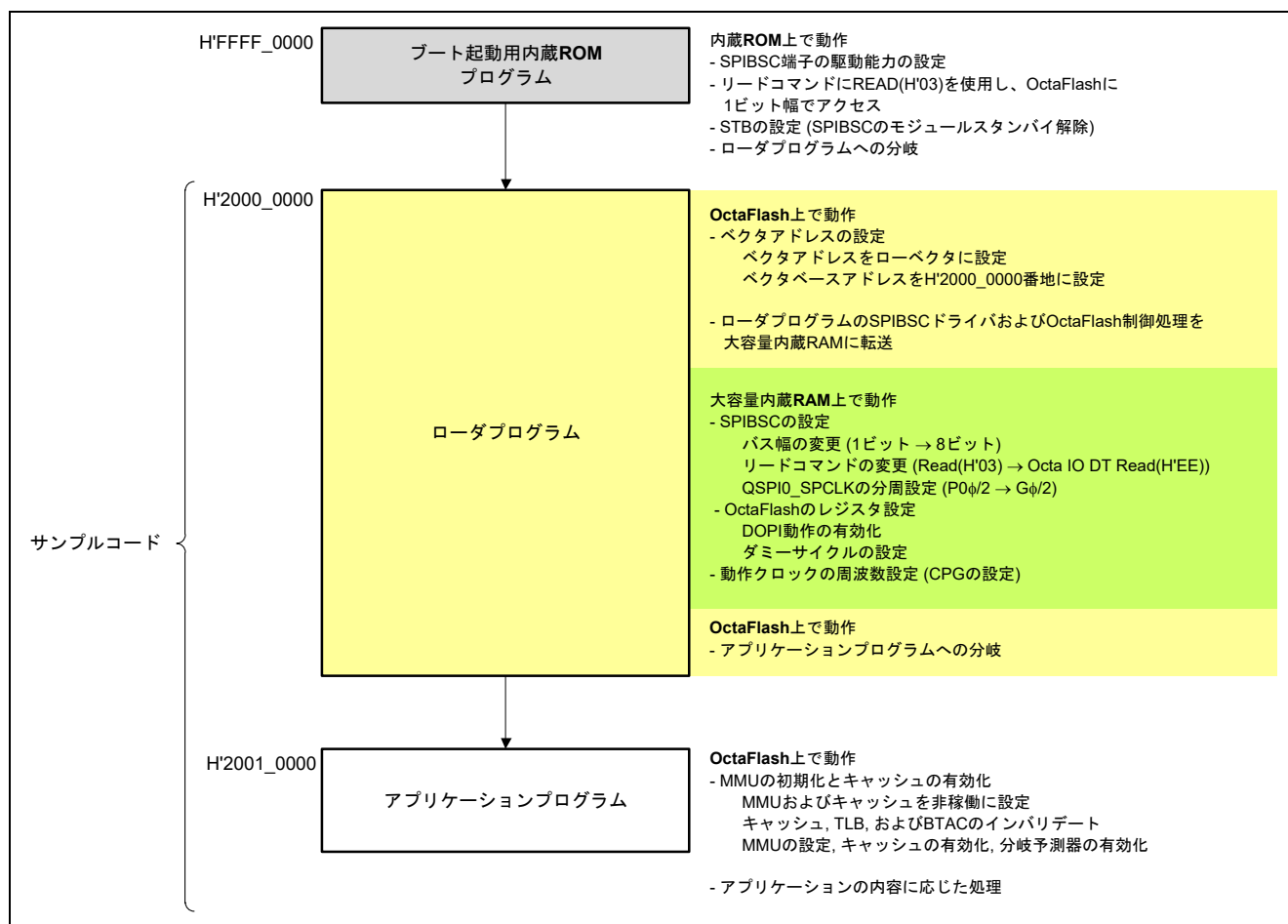


図5.1 サンプルコードの動作概要

### 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 プログラムの処理完了後に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'2000\_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/A2M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 Octal-SPI 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 通信への変換に使用する Octal-SPI フラッシュメモリへのコマンドの設定は、一般的な Octal-SPI フラッシュメモリに共通でアクセスできる設定 (SPI モードでアクセスする設定) 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 Octal-SPI フラッシュメモリに最適な設定を行う必要があります。

ローダプログラムでは、Octal-SPI フラッシュメモリとして OctaFlash を使用して、OctaFlash に最適にアクセスする処理を行います。OctaFlash に最適にアクセスするためには、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 OctaFlash のレジスタ設定 (以下、OctaFlash 設定とします) を行う必要があります。

ローダプログラムでは、データバス幅を 8 ビットにし、使用する DOPI モードのリードコマンドに合わせて転送ビットレートを最適化するために、SPIBSC のレジスタを設定します。また、OctaFlash のダミーサイクル数、DOPI 動作の有効化を行うために、Macronix社製 OctaFlash (MX25UM51245G) のレジスタを設定し、MX25UM51245Gにアクセスするための最適な設定を行っています。

ローダプログラムの SPIBSC 設定および OctaFlash 設定を行う処理は、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、SPI マルチ I/O バス空間以外の領域で実行する必要があり、サンプルコードでは SPIBSC ドライバの処理や OctaFlash の制御処理を大容量内蔵 RAM に転送して実行しています。

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。

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'2001\_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	ローダプログラム実行後
SPIBSC 設定	遅延設定 次アクセス遅延設定： SSLDR.SPNDL[2:0] QSPIn_SSL ネゲート遅延設定： SSLDR.SLNDL[2:0] クロック遅延設定： SSLDR.SCKDL[2:0]	B'000 (1QSPIn_SPCLK) B'000 (1QSPIn_SPCLK) B'000 (1.5QSPIn_SPCLK)	B'000 (1 QSPIn_SPCLK) B'000 (1 QSPIn_SPCLK) B'000 (1.5 QSPIn_SPCLK)
	シリアルクロック： (QSPiO_SPCLK)	P0φ/2=16.5 [MHz] (注 1)	Gφ/2=132 [MHz] (注 1)
	QSPIn_SSL 出カアイドル値固定：  QSPIn_IO3 の設定 QSPIn_IO2 の設定 QSPIn_IO1 の設定 QSPIn_IO0 の設定	QSPIn_SSL ネゲート期間の出力値を、前 回転送の最終ビットに設定 CMNCR.MOIO3[1:0]=B'10 CMNCR.MOIO2[1:0]=B'10 CMNCR.MOIO1[1:0]=B'10 CMNCR.MOIO0[1:0]=B'10	Octal-SPI フラッシュメモリを接続する設 定 CMNCR.MOIO3[1:0]=B'01 CMNCR.MOIO2[1:0]=B'01 CMNCR.MOIO1[1:0]=B'01 CMNCR.MOIO0[1:0]=B'01
	1 ビット幅時の端子出力値の設定：  QSPIn_IO3 の設定 QSPIn_IO2 の設定 QSPIn_IO0 の設定	1 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 CMNCR.IO3FV[1:0]=B'10 CMNCR.IO2FV[1:0]=B'10 CMNCR.IO0FV[1:0]=B'10	1 ビット幅の端子の出力値を、Hi-Z に設 定 CMNCR.IO3FV[1:0]=B'11 CMNCR.IO2FV[1:0]=B'11 CMNCR.IO0FV[1:0]=B'11 (注 2)
	データバス幅： CMNCR.BSZ[1:0]	4 ビット B'00	8 ビット (Octal-SPI フラッシュメモリ) B'01
	リードキャッシュ：DRCR.RBE	1 (有効)	1 (有効)
	QSPIn_SSL ネゲート設定： DRCR.SSLE	転送終了毎に QSPIn_SSL ネゲート 0	アドレスが連続な限り QSPIn_SSL ア サート保持し、前回転送のアドレスから 不連続の場合は、QSPIn_SSL ネゲート 1
	リードデータバースト長： DRCR.RBURST[4:0]	4 データ長 (32 バイト) B'00011	4 データ長 (32 バイト) B'00011
	データリードビット幅： DRENDR.DRDB[1:0]	1 ビット B'00	4 ビット (8 ビット幅) B'10
	リードコマンド： DRCMR.CMD[7:0] DRCMR.OCMD[7:0]	Read H'03 -	Octa IO DT Read H'EE H'11
	コマンドイネーブル： DRENDR.CDE	出力する 1	出力する 1
	コマンドビット幅： DRENDR.CDB[1:0]	1 ビット B'00	4 ビット (8 ビット幅) B'10
	オプションコマンドイネーブル： DRENDR.OCDE	出力しない 0	出力しない 0
	オプションコマンドビット幅： DRENDR.OCDB[1:0]	-	4 ビット (8 ビット幅) B'10

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	ローダプログラム実行後
SPIBSC 設定	アドレスイネーブル： DREN.R.ADE[3:0]	Address[23:0]を出力 B'0111	Octal-SPI フラッシュメモリ出力 B'1100
	アドレスビット幅： DREN.R.ADB[1:0]	1 ビット B'00	4 ビット (8 ビット幅) B'10
	オプションデータイネーブル： DREN.R.OPDE[3:0]	出力しない B'0000	出力しない B'0000
	オプションデータビット幅： DREN.R.OPDB[1:0]	—	—
	オプションデータ： DROPR.OPD3[7:0] DROPR.OPD2[7:0] DROPR.OPD1[7:0] DROPR.OPD0[7:0]	— — — —	— — — —
	ダミーサイクルイネーブル： DREN.R.DME	挿入しない 0	挿入する 1
	ダミーサイクル数： DRDMCR.DMCYC[4:0]	—	14 サイクル B'01101
	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 [27:0]が有効 256MB の空間に直接アクセス可能 B'011 H'00
	転送フォーマット： DRDREN.R.HYPE[2:0] DRDREN.R.ADDRE DRDREN.R.OPDRE DRDREN.R.DRDRE PHYOFFSET1.DDRTMG[1:0] PHYOFFSET2.OCTTMG[2:0]	アドレス、オプションデータ、データは SDR 転送、SPI フラッシュモード B'000 0 0 0 B'11 (SDR) B'100 (シリアルフラッシュ)	アドレス、オプションデータ、データは DDR 転送、Octal-SPI フラッシュメモリ にアクセスするように動作 B'101 1 1 1 B'10 (DDR) B'011 (Octal-SPI フラッシュメモリ動作)
	Octal-SPI フラッシュメモリ代替 アライメント： PHYCNT.ALT_ALIGN PHYCNT.OCTA[1:0]	Octal-SPI フラッシュメモリ接続時の代替 アライメントをサポートしない。 0 B'00	Octal-SPI フラッシュメモリ接続時の代替 アライメントをサポートする。 1 B'01
	Octal-SPI フラッシュメモリプロ トコルモード： PHYCNT.OCT	Octal-SPI フラッシュメモリプロトコル モードを使用しない 0	Octal-SPI フラッシュメモリプロトコル モードを使用する 1
	外部データストローブ： PHYCNT.EXDS	外部データストローブ信号を使用しない 0	外部データストローブ信号を使用する 1
	デバイス選択： PHYCNT.PHYMEM[1:0]	SDR モードのシリアルフラッシュ B'00	DDR モードの Octal-SPI フラッシュメモリ B'01
	ハイスピード応答モード： PHYCNT.HS	ハイスピード応答モードを使用しない 0	ハイスピード応答モードを使用しない 0
	クロックタイミング切り替え： PHYCNT.CKSEL[1:0]	— B'00	Octal-SPI フラッシュメモリ接続時の設定 B'11



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	ローダプログラム実行後
端子の設定	端子電圧  PPOC.POCSEL0 PPOC.POC0	SPIBSC の端子電圧が 1.8V で動作する設定 1 0	SPIBSC の端子電圧が 1.8V で動作する設定 0 0
	駆動能力  PSPIBSC[31:0]	H'0FFF_FFFF (SPIBSC 関連端子の駆動能力が 12mA)	H'0FFF_FFFF (SPIBSC 関連端子の駆動能力が 12mA)
Octa Flash 設定	Configuration Register 2 (注 3)		
	Address: H'0000_0000	変更なし	DTR OPI Enable DOPI=1, SOPI=0
	Address: H'0000_0300	変更なし	ダミーサイクル数 14 DC[2:0] = B'011
その他	動作クロックの設定 クロックモード 1 で EXTAL から 24MHz 入力時	Iφ=132[MHz] Gφ=264[MHz] Bφ=132[MHz] P1φ=66[MHz] P0φ=33[MHz]	Iφ=528[MHz] Gφ=264[MHz] Bφ=132[MHz] P1φ=66[MHz] P0φ=33[MHz]
	SPIBSC クロックの選択 SCLKSEL.SPICR[1:0]	P0φを選択 B'00	Gφを選択 B'11
	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~)
	CP15 ベクタベース アドレスレジスタ (VBAR)	—	H'2000_0000

- 【注】
- 「動作クロックの設定」および「SPIBSC クロックの選択」によって、QSPIO\_SPCLK の動作クロック周波数を設定します。
  - 1 ビット幅以外のデータリード転送を行う場合には、CMNCR の IO0FV は B'11 (QSPIn\_IO0 の出力値を Hi-Z) に設定する必要があります。
  - RZ/A2M のブートモード 4 (Octal-SPI フラッシュブート 1.8V 品) では、OctaFlash 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、OctaFlash のレジスタ設定値が、Octal-SPI 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。
  - PHYCNT の CAL ビットは、外部アドレス空間リードモードで OctaFlash にアクセスする場合には設定する必要がないため上記の表には載せていません。手動モードで OctaFlash をアクセスする場合には、OctaFlash にコマンドを送信する前ごとに必ず CAL ビットを "1" に設定する必要があります。

### 5.1.4 アプリケーションプログラム

#### (1) アプリケーションプログラムの動作

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'2001\_0000 番地に配置されているアプリケーションプログラムのスタートアップ処理に分岐します。

スタートアップ処理では、スタックポインタ、MMU、FPU の設定を行い、セクションの初期化を行い、resetprg 関数に分岐します。

resetprg 関数では、RTC と USB 未使用チャンネルの初期化処理を実行後、L1 キャッシュおよび L2 キャッシュの有効化と INTC の初期化を行い、割り込み処理高速化のために VBAR に大容量内蔵 RAM のアドレスを設定し、IRQ 割り込みおよび FIQ 割り込みを許可にし、main 関数をコールしています。

main 関数では、CPG、OSTM チャンネル 0、SCIFA チャンネル 4、GPIO の初期設定処理を行います。これら初期化処理により、シリアルインタフェースに接続されたホスト PC 上のターミナルに文字列（起動メッセージ）を出力し、OSTM のチャンネル 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 チャンネル 0 の割り込みを発生させ、CPU ボード上の 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。

アプリケーションプログラムで実施している初期設定の詳細は、「RZ/A2M 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。

(2) アプリケーション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 OctaFlash のセクタに配置してください。

Macronix社製 OctaFlash (MX25UM51245G) 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'2001\_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。

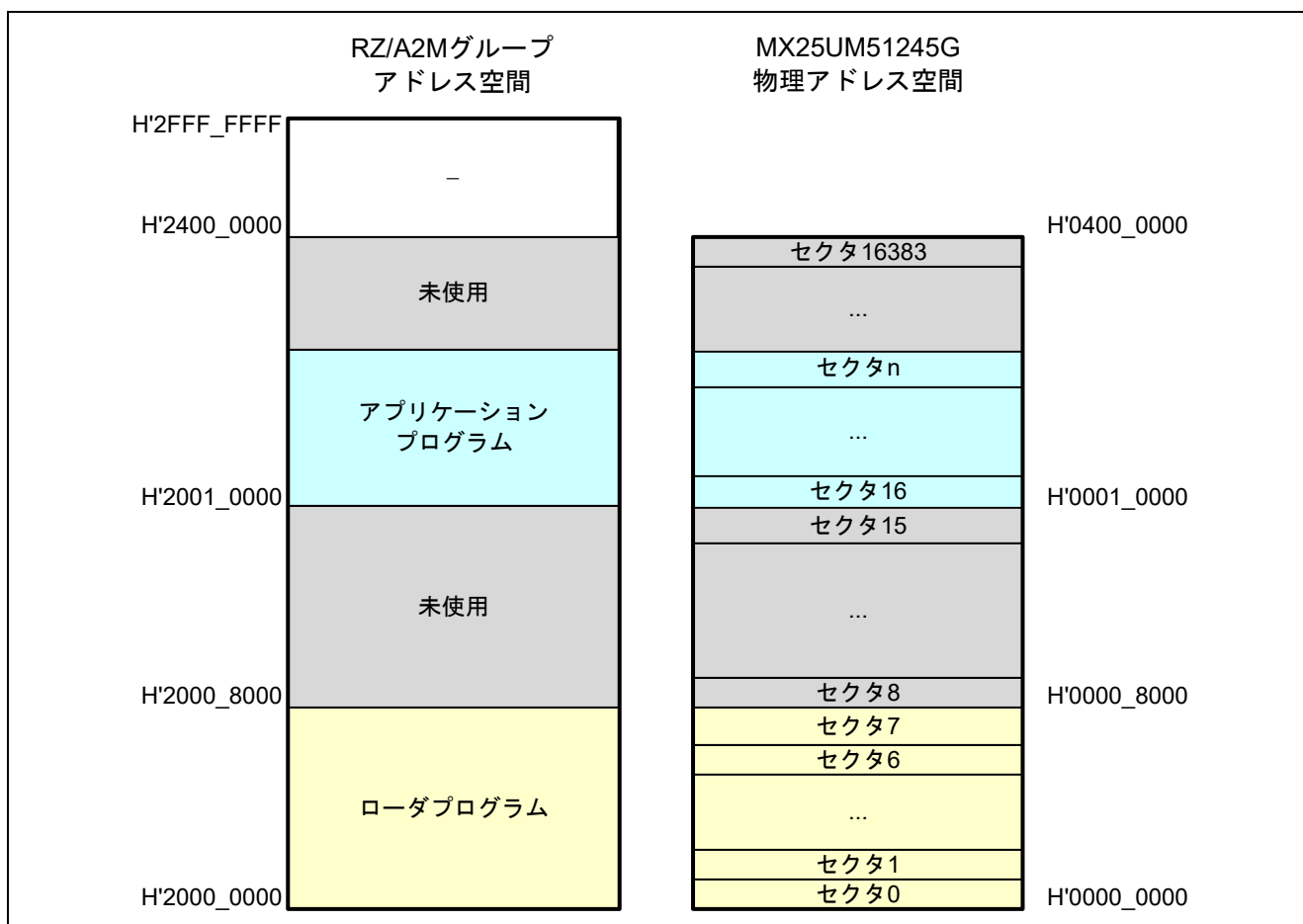


図5.2 サンプルコードのプログラム配置

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 (reset\_handler.asm) にて行っています。「linker\_script.ld」のリンクスクリプトのシンボル定義"\_\_application\_base\_address"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  
アプリケーションプログラムの"VECTOR\_TABLE"のセクションが、"\_\_application\_base\_address"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	<p>CPU クロック : PLL 回路の周波数に対して×1/2 倍に設定            内部バスクロック : PLL 回路の周波数に対して×1/8 倍に設定            周辺クロック 1 (P1φ) : PLL 回路の周波数に対して×1/16 倍に設定</p> <p>クロックモード 1 (分周器 1 : ×1/2 倍、PLL 回路 : ×88 倍) で、入力クロックが 24MHz の場合に以下の周波数となるように設定</p> <ul style="list-style-type: none"> <li>● CPU クロック (Iφ) : 528MHz</li> <li>● 画像処理クロック (Gφ) : 264MHz</li> <li>● 内部バスクロック (Bφ) : 132MHz</li> <li>● 周辺クロック 1 (P1φ) : 66MHz</li> <li>● 周辺クロック 0 (P0φ) : 33MHz</li> <li>● QSPIO_SPCLK : 132MHz (Gφ 選択時)</li> <li>● CKIO クロック : 132MHz (Bφ 選択時)</li> </ul>
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 OctaFlash から、直接リードするための信号を生成するための設定
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 STBCR3, STBCR4, STBCR8 で OSTM0, OSTM2, SCIFA4, SPIBSC にクロックを供給
GPIO	<p>PORT6、PORT9 の兼用端子機能を設定</p> <ul style="list-style-type: none"> <li>● P6_0 : LED の点灯および消灯</li> <li>● P9_1 : Rx/D4、P9_0 : Tx/D4</li> </ul>
OSTM	<p>チャンネル 0、チャンネル 2 をインターバルタイマモードに設定</p> <ul style="list-style-type: none"> <li>● チャンネル 0 P1φ=66MHz 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。LED 点灯および消灯処理で使用。</li> <li>● チャンネル 2 P1φ=66MHz の時に 1ms 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。OS Abstraction による時間管理に使用。</li> </ul>
INTC	INTC の初期設定および OSTM チャンネル 0 割り込み (割り込み ID が 88) ハンドラの登録、OSTM チャンネル 2 割り込み (割り込み ID が 90) ハンドラの登録、SCIFA チャンネル 4 割り込み (割り込み ID が 322, 323) ハンドラの登録と実行
SCIFA	<p>チャンネル 4 を調歩同期式モードに設定</p> <ul style="list-style-type: none"> <li>● データ長 : 8 ビット</li> <li>● ストップビット長 : 1 ビット</li> <li>● パリティ : なし</li> <li>● データ転送方向 : LSB ファースト転送</li> </ul> <p>P1φ=66MHz の時に、クロックソースを分周なし、ポーレートジェネレータは倍速モード、ビットレートの 8 倍の基本クロックで動作するように設定。ビットレートが 115200bps となるように、ビットレート値に 71 を設定 (誤差 : -0.53%)</p>

### 5.2.2 メモリマップ

図5.3にRZ/A2Mグループのアドレス空間とRZ/A2M CPUボードのメモリマップを示します。

サンプルコードでは、ROM 領域を使用するコードおよびデータを SPI マルチ I/O バス空間に接続した OctaFlash に配置し、RAM 領域を使用するコードおよびデータを大容量内蔵 RAM に配置するようにしています。

	RZ/A2Mグループの アドレス空間	RZ/A2M CPUボード メモリマップ
H'FFFF FFFF	内蔵IO領域 および 予約領域 (2044MB)	内蔵IO領域 および 予約領域 (2044MB)
H'8040 0000	大容量内蔵RAM (4MB)	大容量内蔵RAM (4MB)
H'8000 0000	予約領域 (256MB)	予約領域 (256MB)
H'7000 0000		
H'6100 0000	OctaRAM™空間 (256MB)	-
H'6000 0000		
H'5400 0000	OctaFlash™空間 (256MB)	-
H'5000 0000		
H'4080 0000	HyperRAM™空間 (256MB)	-
H'4000 0000		HyperRAM™ (8MB)
H'3400 0000	HyperFlash™空間 (256MB)	-
H'3000 0000		HyperFlash™ (64MB)
H'2400 0000	SPIマルチI/Oバス 空間 (256MB)	-
H'2000 0000		OctaFlash™ (注) (64MB)
H'1800 0000	内蔵IO領域および 予約領域 (128MB)	内蔵IO領域および 予約領域 (128MB)
H'1400 0000	CS5空間 (64MB)	-
H'1000 0000	CS4空間 (64MB)	-
H'0C00 0000	CS3空間 (64MB)	-
H'0800 0000	CS2空間 (64MB)	-
H'0400 0000	CS1空間 (64MB)	-
H'0000 0000	CS0空間 (64MB)	-

【注】 RZ/A2M CPU ボードには OctaFlash は搭載されていません。本サンプルコードが動作できるように、RZ/A2M CPU ボードの U2 のシリアルフラッシュメモリを OctaFlash に実装を変更しています。

図5.3 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名とオブジェクト名一覧を示します。

アプリケーションプログラムで使用するセクション配置については、「RZ/A2M グループ 初期設定例」のアプリケーションノートをご参照ください。

表5.6 ローダプログラムで使用するセクション名とオブジェクト名一覧

出力セクション名	入力セクション名 入力オブジェクト名	内容	ロード 領域	実行 領域
LOAD_MODULE1	VECTOR_TABLE	例外処理ベクタテーブル	FLASH	FLASH
LOAD_MODULE2	*r_cpg/*.o (.text .rodata)	CPG の設定処理	FLASH	LRAM
	*rza_io_regrw.o (.text .rodata)	IO レジスタアクセス処理		
	*r_spibsc/*.o (.text .rodata)	SPIBSC の設定処理		
	*hwsetup.o (.text .rodata)	HardwareSetup の設定処理		
	* (.data .data.*)	デフォルトの初期値ありデータ領域		
LOAD_MODULE3	RESET_HANDLER	リセット処理	FLASH	FLASH
	INIT_SECTION */sections.o	セクション初期化処理		
	* (.text .text.*)	デフォルトのコード領域		
	* (.rodata .rodata.*)	デフォルトの定数データ領域		
.data.memclk_setup	*r_memclk_setup.o (.text .rodata .data)	メモリクロックの設定処理	FLASH	LRAM
	*r_spibsc_setup.o (.text .rodata .data)	SPIBSC 用メモリクロックの設定処理		
	*r_*_memclk_setup.o (.text .rodata .data)	各ドライバ用メモリクロックの設定処理		
.bss.memclk_setup	*r_memclk_setup.o (.bss COMMON)	メモリクロックの設定処理の初期値なしデータ領域	-	LRAM
	*r_spibsc_setup.o (.bss COMMON)	SPIBSC 用メモリクロックの設定処理の初期値なしデータ領域		
	*r_*_memclk_setup.o (.bss COMMON)	各ドライバ用メモリクロックの設定処理の初期値なしデータ領域		
.stack	なし	SVC モードのスタック領域	-	LRAM
.bss	* (.bss .bss.*) * (COMMON)	デフォルトの初期値なしデータ領域	-	LRAM
.heap	なし	ヒープ領域	-	LRAM

【注】 表中のロード領域および実行領域において、FLASH は OctaFlash の領域を、LRAM は大容量内蔵 RAM の領域を表します。

### 5.3 使用割り込み一覧

ローダプログラムでは割り込みは使用していません。

アプリケーションプログラムで使用する割り込みについては、「RZ/A2M グループ 初期設定例」のアプリケーションノートをご参照ください。

### 5.4 データ型一覧

表5.7にサンプルコードで使用するデータ型一覧を示します。

表5.7 サンプルコードで使用するデータ型一覧

シンボル	内容
char_t	8ビット文字
bool_t	論理型。値は true (1)、false (0)
int_t	高速な整数、符号あり、本サンプルコードでは 32 ビット整数。
int8_t	8 ビット整数、符号あり (標準ライブラリ stdint.h にて定義)
int16_t	16 ビット整数、符号あり (標準ライブラリ stdint.h にて定義)
int32_t	32 ビット整数、符号あり (標準ライブラリ stdint.h にて定義)
int64_t	64 ビット整数、符号あり (標準ライブラリ stdint.h にて定義)
uint8_t	8 ビット整数、符号なし (標準ライブラリ stdint.h にて定義)
uint16_t	16 ビット整数、符号なし (標準ライブラリ stdint.h にて定義)
uint32_t	32 ビット整数、符号なし (標準ライブラリ stdint.h にて定義)
uint64_t	64 ビット整数、符号なし (標準ライブラリ stdint.h にて定義)
float32_t	32 ビット浮動小数
float64_t	64 ビット浮動小数
float128_t	128 ビット浮動小数

## 5.5 定数一覧

表5.8および表5.9にローダプログラムで使用する定数を示します。

アプリケーションプログラムで使用する定数については、「RZ/A2M グループ 初期設定例」のアプリケーションノートをご参照ください。

表5.8 ローダプログラムで使用する定数 (1/2)

定数名	設定値	内容
STARTUP_CFG_BOOT_MODE	(4)	Octal-SPI フラッシュブートとして動作するように設定
STARTUP_CFG_SPIBSC_CONNECT_DEVICE_TYPE	(2)	SPIBSC に接続するフラッシュメモリを Octal-SPI フラッシュメモリに設定
STARTUP_CFG_PROJECT_TYPE	(0)	ローダプログラムであることを示すための情報
SPIBSC_SUCCESS	(0)	正常終了
SPIBSC_ERR_INVALID	(-1)	エラー終了
SPIBSC_PORT_VOLTAGE_3_3V	(0)	SPIBSC が使用する専用端子の動作電圧を 3.3V に設定
SPIBSC_PORT_VOLTAGE_1_8V	(1)	SPIBSC が使用する専用端子の動作電圧を 1.8V に設定
SPIBSC_FLASH_OCTA	(1)	フラッシュメモリの種別を OctaFlash に設定
SPIBSC_MODE_MANUAL	(0)	SPIBSC 動作モードを手動モードに設定
SPIBSC_MODE_XIP	(1)	SPIBSC 動作モードを外部アドレス空間リードモードに設定
SPIBSC_CMNCR_BSZ_SINGLE	(0)	SPIBSC に接続するフラッシュメモリのデータバス幅を 4 ビットに設定
SPIBSC_CMNCR_BSZ_DUAL	(1)	SPIBSC に接続するフラッシュメモリのデータバス幅を 8 ビットに設定
SPIBSC_OCTA_OPI_DISABLE	(0)	OctaFlash (MX25UM51245G) の Configuration Register 2 で DOPI モードを無効にするための情報として使用
SPIBSC_OCTA_OPI_ENABLE	(1)	OctaFlash (MX25UM51245G) の Configuration Register 2 で DOPI モードを有効にするための情報として使用
SPIBSC_RDSR_WEL	(0x02)	OctaFlash (MX25UM51245G) の Status Register の WEL ビットを参照するための情報として使用
SPIBSC_RDSR_WIP	(0x01)	OctaFlash (MX25UM51245G) の Status Register の WIP ビットを参照するための情報として使用
SPIBSC_1BIT_WIDTH	(0)	コマンド、オプションコマンド、アドレス、オプションデータ、転送データのビット幅を 1 ビットに設定
SPIBSC_8BIT_WIDTH	(2)	コマンド、オプションコマンド、アドレス、オプションデータ、転送データのビット幅を 8 ビットに設定
SPIBSC_OUTPUT_DISABLE	(0)	コマンド、オプションコマンド、アドレス、オプションデータ、ダミーサイクルを出力しない設定
SPIBSC_OUTPUT_ENABLE	(1)	コマンド、オプションコマンド、ダミーサイクルを出力する設定
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_ADDR_OCTA	(0x0c)	Octal-SPI フラッシュメモリ用のアドレスを出力
SPIBSC_OUTPUT_OPD_3	(0x08)	オプションデータ OPD3 を出力
SPIBSC_OUTPUT_OPD_32	(0x0c)	オプションデータ OPD3, OPD2 を出力
SPIBSC_OUTPUT_OPD_321	(0x0e)	オプションデータ OPD3, OPD2, OPD1 を出力
SPIBSC_OUTPUT_OPD_3210	(0x0f)	オプションデータ OPD3, OPD2, OPD1, OPD0 を出力



表5.9 ローダプログラムで使用する定数 (2/2)

定数名	設定値	内容
SPIBSC_DUMMY_02CYC	(1)	ダミーサイクル数を 2 に設定
SPIBSC_DUMMY_03CYC	(2)	ダミーサイクル数を 3 に設定
SPIBSC_DUMMY_04CYC	(3)	ダミーサイクル数を 4 に設定
SPIBSC_DUMMY_05CYC	(4)	ダミーサイクル数を 5 に設定
SPIBSC_DUMMY_06CYC	(5)	ダミーサイクル数を 6 に設定
SPIBSC_DUMMY_07CYC	(6)	ダミーサイクル数を 7 に設定
SPIBSC_DUMMY_08CYC	(7)	ダミーサイクル数を 8 に設定
SPIBSC_DUMMY_09CYC	(8)	ダミーサイクル数を 9 に設定
SPIBSC_DUMMY_10CYC	(9)	ダミーサイクル数を 10 に設定
SPIBSC_DUMMY_11CYC	(10)	ダミーサイクル数を 11 に設定
SPIBSC_DUMMY_12CYC	(11)	ダミーサイクル数を 12 に設定
SPIBSC_DUMMY_13CYC	(12)	ダミーサイクル数を 13 に設定
SPIBSC_DUMMY_14CYC	(13)	ダミーサイクル数を 14 に設定
SPIBSC_DUMMY_15CYC	(14)	ダミーサイクル数を 15 に設定
SPIBSC_DUMMY_16CYC	(15)	ダミーサイクル数を 16 に設定
SPIBSC_DUMMY_17CYC	(16)	ダミーサイクル数を 17 に設定
SPIBSC_DUMMY_18CYC	(17)	ダミーサイクル数を 18 に設定
SPIBSC_DUMMY_19CYC	(18)	ダミーサイクル数を 19 に設定
SPIBSC_DUMMY_20CYC	(19)	ダミーサイクル数を 20 に設定
SPIBSC_DDR_TRANSFER	(1)	アドレス、オプションデータ、データの転送を DDR 転送に設定
SIPBSC_SDR_TRANSFER	(0)	アドレス、オプションデータ、データの転送を SDR 転送に設定
SPIBSC_QSPI_IO_OUTPUT_0	(0x00)	QSPIn_IO 端子に出力する値を 0 にする
SPIBSC_QSPI_IO_OUTPUT_1	(0x01)	QSPIn_IO 端子に出力する値を 1 にする
SPIBSC_QSPI_IO_OUTPUT_PREVIOUS	(0x02)	QSPIn_IO 端子に出力する値を前回転送の最終ビットにする
SPIBSC_QSPI_IO_OUTPUT_HI_Z	(0x03)	QSPIn_IO 端子に出力する値を Hi-Z にする
SPIBSC_MANUAL_8BIT_TRANSFERRED	(0x8)	手動モードのデータ転送時の転送ビットを 8 ビットに設定
SPIBSC_MANUAL_16BIT_TRANSFERRED	(0x8)	手動モードのデータ転送時の転送ビットを 16 ビットに設定
SPIBSC_MANUAL_32BIT_TRANSFERRED	(0xf)	手動モードのデータ転送時の転送ビットを 32 ビットに設定
SPIBSC_MANUAL_64BIT_TRANSFERRED	(0xf)	手動モードのデータ転送時の転送ビットを 64 ビットに設定

## 5.6 構造体/共用体一覧

表5.10～表5.22にローダプログラムで使用する構造体を示します。

表5.10 SPIBSC レジスタ設定構造体 (st\_spibsc\_config\_t)

メンバ名	内容
uint8_t flash_type	接続するフラッシュメモリの種別を指定します。 SPIBSC_FLASH_OCTA : OctaFlash
uint8_t flash_num	フラッシュメモリのデータバス幅を指定します。 SPIBSC_CMNCR_BSZ_SINGLE : 4ビット幅 (未サポート) SPIBSC_CMNCR_BSZ_DUAL : 8ビット幅
uint8_t flash_port_voltage	SPIBSCの専用端子の電圧設定を指定します。 SPIBSC_PORT_VOLTAGE_3_3V : 3.3V SPIBSC_PORT_VOLTAGE_1_8V : 1.8V

表5.11 SPIBSC 外部アドレス空間リードモード設定構造体 (st\_spibsc\_xip\_config\_t) (1/6)

メンバ名	内容
uint8_t command_name[20]	リードコマンドを識別する文字列 <ul style="list-style-type: none"> <li>本メンバはレジスタ設定に影響しません。</li> </ul>
uint8_t cmd	リードコマンド <ul style="list-style-type: none"> <li>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 OctaFlash に出力するリードコマンドを設定します。</li> <li>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。</li> </ul>
uint8_t cmd_width	リードコマンドビット幅 <ul style="list-style-type: none"> <li>リードコマンド発行時のビット幅を設定します。</li> <li>設定可能な値 :  SPIBSC_1BIT_WIDTH : 1 ビット幅  SPIBSC_8BIT_WIDTH : 8 ビット幅</li> <li>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。</li> </ul>
uint8_t cmd_output_enable	リードコマンドイネーブル <ul style="list-style-type: none"> <li>リードコマンドを発行するかどうかを選択します。</li> <li>設定可能な値 :  SPIBSC_OUTPUT_DISABLE : 発行しない  SPIBSC_OUTPUT_ENABLE : 発行する</li> <li>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 CDE に設定します。</li> </ul>
uint8_t ocmd	オプションコマンド <ul style="list-style-type: none"> <li>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 OctaFlash に出力するオプションコマンドを設定します。</li> <li>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 OCMD[7:0]に設定します。</li> </ul>
uint8_t ocmd_width	オプションコマンドビット幅 <ul style="list-style-type: none"> <li>オプションコマンド発行時のビット幅を設定します。</li> <li>設定可能な値 :  SPIBSC_1BIT_WIDTH : 1 ビット幅  SPIBSC_8BIT_WIDTH : 8 ビット幅</li> <li>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 OCDB[1:0]に設定します。</li> </ul>
uint8_t ocmd_output_enable	オプションコマンドイネーブル <ul style="list-style-type: none"> <li>オプションコマンドを発行するかどうかを選択します。</li> <li>設定可能な値 :  SPIBSC_OUTPUT_DISABLE : 発行しない  SPIBSC_OUTPUT_ENABLE : 発行する</li> <li>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 OCDE に設定します。</li> </ul>

表5.12 SPIBSC 外部アドレス空間リードモード設定構造体 (st\_spibsc\_xip\_config\_t) (2/6)

メンバ名	内容
uint8_t addr_width	<p>アドレスビット幅</p> <ul style="list-style-type: none"> <li>• 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 OctaFlash に出力するアドレスのビット幅を設定します。</li> <li>• 設定可能な値 : SPIBSC_1BIT_WIDTH : 1 ビット幅 SPIBSC_8BIT_WIDTH : 8 ビット幅</li> <li>• 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 (DRENDR) の ADB[1:0]に設定します。</li> </ul>
uint8_t addr_output_enable	<p>アドレスイネーブル</p> <ul style="list-style-type: none"> <li>• 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 OctaFlash に出力するアドレスを設定します。</li> <li>• 設定可能な値 : SPIBSC_OUTPUT_DISABLE : 出力しない 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 SPIBSC_OUTPUT_ADDR_OCTA : Octal-SPI フラッシュメモリ用のアドレスを出力</li> <li>• 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 (DRENDR) の ADE[3:0]に設定します。</li> </ul>
uint8_t addr_ddr_enable	<p>アドレス DDR イネーブル</p> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 設定可能な値 : SPIBSC_SDR_TRANSFER : SDR 転送 SPIBSC_DDR_TRANSFER : DDR 転送</li> <li>• 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 (DRDRENDR) の ADDRE に設定します。</li> </ul>
uint8_t reserve1	<p>リザーブデータ</p> <ul style="list-style-type: none"> <li>• サンプルコードでは、本メンバを参照しません。</li> </ul>
uint8_t reserve2	<p>リザーブデータ</p> <ul style="list-style-type: none"> <li>• サンプルコードでは、本メンバを参照しません。</li> </ul>

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\_xip\_config\_t) (3/6)

メンバ名	内容
uint8_t opdata_width	<p>オプションデータビット幅</p> <ul style="list-style-type: none"> <li>オプションデータ発行時のビット幅を設定します。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_1BIT_WIDTH : 1 ビット幅</li> <li>SPIBSC_8BIT_WIDTH : 8 ビット幅</li> </ul> </li> <li>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 (DRENDR) の OPDB[1:0]に設定します。</li> </ul>
uint8_t opdata_output_enable	<p>オプションデータイネーブル</p> <ul style="list-style-type: none"> <li>オプションデータを発行するかどうかを選択します。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_OUTPUT_DISABLE : 出力しない</li> <li>SPIBSC_OUTPUT_OPD_3 : OPD3 を出力</li> <li>SPIBSC_OUTPUT_OPD_32 : OPD3,OPD2 を出力</li> <li>SPIBSC_OUTPUT_OPD_321 : OPD3,OPD2,OPD1 を出力</li> <li>SPIBSC_OUTPUT_OPD_3210 : OPD3,OPD2,OPD1,OPD0 を出力</li> </ul> </li> <li>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 (DRENDR) の OPDE[3:0]に設定します。</li> </ul>
uint8_t opdata_ddr_enable	<p>オプションデータ DDR イネーブル</p> <ul style="list-style-type: none"> <li>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。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_SDR_TRANSFER : SDR 転送</li> <li>SPIBSC_DDR_TRANSFER : DDR 転送</li> </ul> </li> <li>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 (DRDRENDR) の OPDRE に設定します。</li> </ul>
uint8_t opd3 uint8_t opd2 uint8_t opd1 uint8_t opd0	<p>オプションデータ</p> <ul style="list-style-type: none"> <li>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 OctaFlash に出力するオプションデータを設定します。</li> <li>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。</li> </ul>

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\_xip\_config\_t) (4/6)

メンバ名	内容
uint8_t reserve3	リザーブデータ サンプルコードでは、本メンバを参照しません。
uint8_t dummy_cycle_enable	ダミーサイクルイネーブル <ul style="list-style-type: none"> <li>ダミーサイクルを挿入するかを選択します。</li> <li>設定可能な値 : SPIBSC_OUTPUT_DISABLE : 挿入しない SPIBSC_OUTPUT_ENABLE : 挿入する</li> <li>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 (DRENDR) の DME に設定します。</li> </ul>
uint8_t dummy_cycle_count	ダミーサイクル数 <ul style="list-style-type: none"> <li>挿入するダミーサイクル数を設定します。</li> <li>設定可能な値 : SPIBSC_DUMMY_02CYC~SPIBSC_DUMMY_20CYC</li> <li>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[4:0]に設定します。</li> </ul>
uint8_t data_width	データリードビット幅 <ul style="list-style-type: none"> <li>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 OctaFlash のデータリードビット幅を設定します。</li> <li>設定可能な値 : SPIBSC_1BIT_WIDTH : 1 ビット幅 SPIBSC_8BIT_WIDTH : 8 ビット幅</li> <li>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 (DRENDR) の DRDB[1:0]に設定します。</li> </ul>
uint8_t data_ddr_enable	転送データ DDR イネーブル <ul style="list-style-type: none"> <li>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。</li> <li>設定可能な値 : SPIBSC_SDR_TRANSFER : SDR 転送 SPIBSC_DDR_TRANSFER : DDR 転送</li> <li>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 (DRDRENDR) の DRDRE に設定します。</li> </ul>

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\_xip\_config\_t) (5/6)

メンバ名	内容
uint8_t cmncr_moiio3	SSL ネゲート時の QSPIn_IO3 端子状態 <ul style="list-style-type: none"> <li>データ転送完了後のネゲート期間に出力する値を選択します。</li> <li>設定可能な値 :               <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の MOIIO3[1:0]に設定します。</li> </ul>
uint8_t cmncr_moiio2	SSL ネゲート時の QSPIn_IO2 端子状態 <ul style="list-style-type: none"> <li>データ転送完了後のネゲート期間に出力する値を選択します。</li> <li>設定可能な値 :               <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の MOIIO2[1:0]に設定します。</li> </ul>
uint8_t cmncr_moiio1	SSL ネゲート時の QSPIn_IO1 端子状態 <ul style="list-style-type: none"> <li>データ転送完了後のネゲート期間に出力する値を選択します。</li> <li>設定可能な値 :               <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の MOIIO1[1:0]に設定します。</li> </ul>
uint8_t cmncr_moiio0	SSL ネゲート時の QSPIn_IO0 端子状態 <ul style="list-style-type: none"> <li>データ転送完了後のネゲート期間に出力する値を選択します。</li> <li>設定可能な値 :               <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の MOIIO0[1:0]に設定します。</li> </ul>

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\_xip\_config\_t) (6/6)

メンバ名	内容
uint8_t cmnncr_io3fv	<p>1 ビット幅時の QSPIn_IO3 端子状態</p> <ul style="list-style-type: none"> <li>転送を 1 ビット幅で行っている間の値を選択します。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の IO3FV[1:0]に設定します。</li> </ul>
uint8_t cmnncr_io2fv	<p>1 ビット幅時の QSPIn_IO2 端子状態</p> <ul style="list-style-type: none"> <li>転送を 1 ビット幅で行っている間の値を選択します。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の IO2FV[1:0]に設定します。</li> </ul>
uint8_t cmnncr_io0fv	<p>1 ビット幅入力時の QSPIn_IO0 端子状態</p> <ul style="list-style-type: none"> <li>入力を 1 ビット幅で行っている間の値を選択します。</li> <li>設定可能な値 (注) : <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の IO0FV[1:0]に設定します。</li> </ul>

【注】 1 ビット幅以外 (data\_width に "SPIBSC\_1BIT\_WIDTH"以外を設定) のデータリード転送を行う場合には、cmnncr\_io0fv には "SPIBSC\_QSPI\_IO\_OUTPUT\_HI\_Z"を設定する必要があります。



表5.17 SPIBSC 手動モード設定構造体 (st\_spibsc\_manual\_mode\_command\_config\_t) (1/6)

メンバ名	内容
uint8_t command_name[20]	SPI コマンドを識別する文字列 <ul style="list-style-type: none"> <li>本メンバはレジスタ設定に影響しません。</li> </ul>
uint8_t cmd	コマンド <ul style="list-style-type: none"> <li>手動モード時に出力するコマンドを設定します。</li> <li>本メンバに設定した値を手動モードコマンド設定レジスタ (SMCMR) の CMD[7:0] に設定します。</li> </ul>
uint8_t cmd_width	コマンドビット幅 <ul style="list-style-type: none"> <li>コマンド発行時のビット幅を設定します。</li> <li>設定可能な値 :  SPIBSC_1BIT_WIDTH : 1 ビット幅  SPIBSC_8BIT_WIDTH : 8 ビット幅</li> <li>本メンバに設定した値を手動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。</li> </ul>
uint8_t cmd_output_enable	コマンドイネーブル <ul style="list-style-type: none"> <li>コマンドを発行するかどうかを選択します。</li> <li>設定可能な値 :  SPIBSC_OUTPUT_DISABLE : 発行しない  SPIBSC_OUTPUT_ENABLE : 発行する</li> <li>本メンバに設定した値を手動モードイネーブル設定レジスタ (SMENR) の CDE に設定します。</li> </ul>
uint8_t ocmd	オプションコマンド <ul style="list-style-type: none"> <li>手動モード時に出力するオプションコマンドを設定します。</li> <li>本メンバに設定した値を手動モードコマンド設定レジスタ (SMCMR) の OCMD[7:0] に設定します。</li> </ul>
uint8_t ocmd_width	オプションコマンドビット幅 <ul style="list-style-type: none"> <li>手動モード時のオプションコマンドビット幅を指定します。</li> <li>設定可能な値 :  SPIBSC_1BIT_WIDTH : 1 ビット幅  SPIBSC_8BIT_WIDTH : 8 ビット幅</li> <li>本メンバに設定した値を手動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。</li> </ul>
uint8_t ocmd_enable	オプションコマンドイネーブル <ul style="list-style-type: none"> <li>手動モード時にオプションコマンドを出力するかを設定します。</li> <li>設定可能な値 :  SPIBSC_OUTPUT_DISABLE : 出力しない  SPIBSC_OUTPUT_ENABLE : 出力する</li> <li>本メンバに設定した値を手動モードイネーブル設定レジスタ (SMENR) の OCDE に設定します。</li> </ul>

表5.18 SPIBSC 手動モード設定構造体 (st\_spibsc\_manual\_mode\_command\_config\_t) (2/6)

メンバ名	内容
uint8_t addr_width	<p>アドレスビット幅</p> <ul style="list-style-type: none"> <li>手動モード時のアドレスビット幅を指定します。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_1BIT_WIDTH : 1 ビット幅</li> <li>SPIBSC_8BIT_WIDTH : 8 ビット幅</li> </ul> </li> <li>本メンバに設定した値を手動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。</li> </ul>
uint8_t addr_output_enable	<p>アドレスイネーブル</p> <ul style="list-style-type: none"> <li>手動モード時にアドレスを出力するかを設定します。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_OUTPUT_DISABLE : 出力しない</li> <li>SPIBSC_OUTPUT_ADDR_24 : ADR[23:0] を出力</li> <li>SPIBSC_OUTPUT_ADDR_32 : ADR[31:0] を出力</li> <li>SPIBSC_OUTPUT_ADDR_OCTA : <ul style="list-style-type: none"> <li>Octal-SPI フラッシュメモリ用のアドレスを出力</li> </ul> </li> </ul> </li> <li>本メンバに設定した値を手動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。</li> </ul>
uint8_t addr_sdr_ddr	<p>アドレス DDR イネーブル</p> <ul style="list-style-type: none"> <li>手動モード時に出力するアドレスの SDR/DDR 転送を選択します。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_SDR_TRANSFER : SDR 転送</li> <li>SPIBSC_DDR_TRANSFER : DDR 転送</li> </ul> </li> <li>本メンバに設定した値を手動モード DDR イネーブルレジスタ (SMDREN) の ADDRE に設定します。</li> </ul>

表5.19 SPIBSC 手動モード設定構造体 (st\_spibsc\_manual\_mode\_command\_config\_t) (3/6)

メンバ名	内容
uint8_t opdata_width	<p>オプションデータビット幅</p> <ul style="list-style-type: none"> <li>手動モード時のオプションデータビット幅を指定します。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_1BIT_WIDTH : 1 ビット幅</li> <li>SPIBSC_8BIT_WIDTH : 8 ビット幅</li> </ul> </li> <li>本メンバに設定した値を手動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。</li> </ul>
uint8_t opdata_output_enable	<p>オプションデータイネーブル</p> <ul style="list-style-type: none"> <li>手動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。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_OUTPUT_DISABLE : 出力しない</li> <li>SPIBSC_OUTPUT_OPD_3 : OPD3 を出力</li> <li>SPIBSC_OUTPUT_OPD_32 : OPD3,OPD2 を出力</li> <li>SPIBSC_OUTPUT_OPD_321 : OPD3,OPD2,OPD1 を出力</li> <li>SPIBSC_OUTPUT_OPD_3210 : OPD3,OPD2,OPD1,OPD0 を出力</li> </ul> </li> <li>本メンバに設定した値を手動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。</li> </ul>
uint8_t opdata_ddr_enable	<p>オプションデータ DDR イネーブル</p> <ul style="list-style-type: none"> <li>手動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。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_SDR_TRANSFER : SDR 転送</li> <li>SPIBSC_DDR_TRANSFER : DDR 転送</li> </ul> </li> <li>本メンバに設定した値を手動モード DDR イネーブルレジスタ (SMDREN) の OPDRE に設定します。</li> </ul>
uint8_t opd3 uint8_t opd2 uint8_t opd1 uint8_t opd0	<p>オプションデータ</p> <ul style="list-style-type: none"> <li>手動モード時に出力するオプションデータを設定します。</li> <li>本メンバに設定した値を手動モードオプション設定レジスタ (SMOPR) の OPD3[7:0]、OPD2[7:0]、OPD1[7:0]、OPD0[7:0]に設定します。</li> </ul>

表5.20 SPIBSC 手動モード設定構造体 (st\_spibsc\_manual\_mode\_command\_config\_t) (4/6)

メンバ名	内容
uint8_t reserve3	リザーブデータ サンプルコードでは、本メンバを参照しません。
uint8_t dummy_cycle_output_enable	ダミーサイクルイネーブル <ul style="list-style-type: none"> <li>手動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。</li> <li>設定可能な値： <ul style="list-style-type: none"> <li>SPIBSC_OUTPUT_DISABLE：挿入しない</li> <li>SPIBSC_OUTPUT_ENABLE：挿入する</li> </ul> </li> <li>本メンバに設定した値を手動モードイネーブル設定レジスタ (SMENR) の DME に設定します。</li> </ul>
uint8_t dummy_cycle_count	ダミーサイクル数 <ul style="list-style-type: none"> <li>挿入するダミーサイクル数を設定します。</li> <li>設定可能な値： <ul style="list-style-type: none"> <li>SPIBSC_DUMMY_02CYC~SPIBSC_DUMMY_20CYC</li> </ul> </li> <li>本メンバに設定した値を手動モードダミーサイクル設定レジスタ (SMDMCR) の DMCYC[4:0] に設定します。</li> </ul>
uint8_t transfer_data_width	転送データビット幅 <ul style="list-style-type: none"> <li>手動モード時の転送データビット幅を指定します。</li> <li>設定可能な値： <ul style="list-style-type: none"> <li>SPIBSC_1BIT_WIDTH：1 ビット幅</li> <li>SPIBSC_8BIT_WIDTH：8 ビット幅</li> </ul> </li> <li>本メンバに設定した値を手動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。</li> </ul>
uint8_t transfer_data_sdr_ddr	転送データ DDR イネーブル <ul style="list-style-type: none"> <li>手動モード時に転送するデータの SDR/DDR 転送を選択します。</li> <li>設定可能な値： <ul style="list-style-type: none"> <li>SPIBSC_SDR_TRANSFER：SDR 転送</li> <li>SPIBSC_DDR_TRANSFER：DDR 転送</li> </ul> </li> <li>本メンバに設定した値を手動モード DDR イネーブルレジスタ (SMDRENr) の SPIDRE に設定します。</li> </ul>
uint8_t reserve1	リザーブデータ サンプルコードでは、本メンバを参照しません。
uint8_t reserve2	リザーブデータ サンプルコードでは、本メンバを参照しません。

表5.21 SPIBSC 手動モード設定構造体 (st\_spibsc\_manual\_mode\_command\_config\_t) (5/6)

メンバ名	内容
uint8_t cmncr_moiio3	SSL ネゲート時の QSPIn_IO3 端子状態 <ul style="list-style-type: none"> <li>データ転送完了後のネゲート期間に出力する値を選択します。</li> <li>設定可能な値 :               <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の MOIIO3[1:0]に設定します。</li> </ul>
uint8_t cmncr_moiio2	SSL ネゲート時の QSPIn_IO2 端子状態 <ul style="list-style-type: none"> <li>データ転送完了後のネゲート期間に出力する値を選択します。</li> <li>設定可能な値 :               <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の MOIIO2[1:0]に設定します。</li> </ul>
uint8_t cmncr_moiio1	SSL ネゲート時の QSPIn_IO1 端子状態 <ul style="list-style-type: none"> <li>データ転送完了後のネゲート期間に出力する値を選択します。</li> <li>設定可能な値 :               <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の MOIIO1[1:0]に設定します。</li> </ul>
uint8_t cmncr_moiio0	SSL ネゲート時の QSPIn_IO0 端子状態 <ul style="list-style-type: none"> <li>データ転送完了後のネゲート期間に出力する値を選択します。</li> <li>設定可能な値 :               <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の MOIIO0[1:0]に設定します。</li> </ul>

表5.22 SPIBSC 手動モード設定構造体 (st\_spibsc\_manual\_mode\_command\_config\_t) (6/6)

メンバ名	内容
uint8_t cmnncr_io3fv	<p>1 ビット幅時の QSPIn_IO3 端子状態</p> <ul style="list-style-type: none"> <li>転送を 1 ビット幅で行っている間の値を選択します。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の IO3FV[1:0]に設定します。</li> </ul>
uint8_t cmnncr_io2fv	<p>1 ビット幅時の QSPIn_IO2 端子状態</p> <ul style="list-style-type: none"> <li>転送を 1 ビット幅で行っている間の値を選択します。</li> <li>設定可能な値 : <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の IO2FV[1:0]に設定します。</li> </ul>
uint8_t cmnncr_io0fv	<p>1 ビット幅入力時の QSPIn_IO0 端子状態</p> <ul style="list-style-type: none"> <li>入力を 1 ビット幅で行っている間の値を選択します。</li> <li>設定可能な値 (注) : <ul style="list-style-type: none"> <li>SPIBSC_QSPI_IO_OUTPUT_0 : 出力値は 0</li> <li>SPIBSC_QSPI_IO_OUTPUT_1 : 出力値は 1</li> <li>SPIBSC_QSPI_IO_OUTPUT_PREVIOUS : 前回転送の最終ビット (前回は Hi-Z のときは Hi-Z)</li> <li>SPIBSC_QSPI_IO_OUTPUT_HI_Z : 出力値は Hi-Z</li> </ul> </li> <li>本メンバに設定した値を共通コントロールレジスタ (CMNCR) の IO0FV[1:0]に設定します。</li> </ul>

【注】 1 ビット幅以外 (data\_width に "SPIBSC\_1BIT\_WIDTH"以外を設定) のデータリード転送を行う場合には、cmnncr\_io0fv には "SPIBSC\_QSPI\_IO\_OUTPUT\_HI\_Z"を設定する必要があります。

## 5.7 変数一覧

表5.23にローダプログラムで使用する変数一覧を示します。

表5.23 ローダプログラムで使用する変数一覧

変数名	説明	備考
st_spibsc_xip_config_t gs_read_table[0]	外部アドレス空間リードモード用 DOPI モードの 8DTRD コマンド用の設定テーブルデータ	表6.7参照
st_spibsc_manual_mode_command_config_t gs_command_table[0]	手動モード用 SPI モードの RDSR コマンド設定テーブルデータ	表6.8参照
st_spibsc_manual_mode_command_config_t gs_command_table[1]	手動モード用 SPI モードの RDCR コマンド設定テーブルデータ	表6.10参照
st_spibsc_manual_mode_command_config_t gs_command_table[2]	手動モード用 SPI モードの RDCR2 コマンド設定テーブルデータ	表6.12参照
st_spibsc_manual_mode_command_config_t gs_command_table[3]	手動モード用 SPI モードの WREN コマンド設定テーブルデータ	表6.14参照
st_spibsc_manual_mode_command_config_t gs_command_table[4]	手動モード用 SPI モードの WRSR コマンド設定テーブルデータ	表6.16参照
st_spibsc_manual_mode_command_config_t gs_command_table[5]	手動モード用 SPI モードの WRCR2 コマンド設定テーブルデータ	表6.19参照
st_spibsc_manual_mode_command_config_t gs_command_table[6]	手動モード用 SPI モードの RDID コマンド設定テーブルデータ	表6.21参照
st_spibsc_manual_mode_command_config_t gs_command_table[7]	手動モード用 DOPI モードの RDSR コマンド設定テーブルデータ	表6.9参照
st_spibsc_manual_mode_command_config_t gs_command_table[8]	手動モード用 DOPI モードの RDCR コマンド設定テーブルデータ	表6.11参照
st_spibsc_manual_mode_command_config_t gs_command_table[9]	手動モード用 DOPI モードの RDCR2 コマンド設定テーブルデータ	表6.13参照
st_spibsc_manual_mode_command_config_t gs_command_table[10]	手動モード用 DOPI モードの WREN コマンド設定テーブルデータ	表6.15参照
st_spibsc_manual_mode_command_config_t gs_command_table[11]	手動モード用 DOPI モードの WRSR コマンド設定テーブルデータ	表6.17参照
st_spibsc_manual_mode_command_config_t gs_command_table[12]	手動モード用 DOPI モードの WRCR コマンド設定テーブルデータ	表6.18参照
st_spibsc_manual_mode_command_config_t gs_command_table[13]	手動モード用 DOPI モードの WRCR2 コマンド設定テーブルデータ	表6.20参照
st_spibsc_manual_mode_command_config_t gs_command_table[14]	手動モード用 DOPI モードの RDID コマンド設定テーブルデータ	表6.22参照
st_spibsc_manual_mode_command_config_t gs_command_table[15]	手動モード用 DOPI モードの 8DTRD コマンド設定テーブルデータ	表6.23参照
st_spibsc_manual_mode_command_config_t gs_command_table[16]	手動モード用 DOPI モードの SE コマンド設定テーブルデータ	表6.24参照
st_spibsc_manual_mode_command_config_t gs_command_table[17]	手動モード用 DOPI モードの PP コマンド設定テーブルデータ	表6.25参照

## 5.8 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。

ローダプログラムの関数について、表5.24にサンプル関数一覧を、表5.25にAPI関数一覧を、表5.26にユーザ定義関数一覧を示します。

表5.24 サンプル関数一覧

関数名	概要
reset_handler	リセットハンドラ処理（アセンブラ関数）
INITSCT	ローダプログラムのセクション初期化（アセンブラ関数）
R_SC_HardwareSetup	ブートに使用するハードウェアの初期設定
r_memclk_setup	メモリクロックの設定処理

表5.25 API 関数一覧

関数名	概要
R_SPIBSC_Setup	SPIBSC と OctaFlash の初期設定
R_SPIBSC_Init	SPIBSC 初期設定
R_SPIBSC_ChangeMode	SPIBSC 動作モード設定
R_SPIBSC_SPICMDIssue	OctaFlash への SPI コマンド発行（手動モード用）
R_SPIBSC_XipStopAccess	OctaFlash へのアクセス停止
R_SPIBSC_FlushReadCache	SPIBSC リードキャッシュのクリア
R_CPG_InitialiseHwlf	CPG の初期化処理



表5.26 ユーザ定義関数一覧

関数名	概要
Userdef_SPIBSC_OCTAFLASH_SetMode	OctaFlash のレジスタの設定
Userdef_SPIBSC_OCTAFLASH_ReadStatus	OctaFlash のステータスレジスタのリード
Userdef_SPIBSC_OCTAFLASH_ReadConfig	OctaFlash のコンフィグレーションレジスタのリード
Userdef_SPIBSC_OCTAFLASH_WriteStatus	OctaFlash のステータスレジスタおよびコンフィグレーションレジスタのライト
Userdef_SPIBSC_OCTAFLASH_WriteEnable	OctaFlash のライト許可
Userdef_SPIBSC_OCTAFLASH_WaitReady	OctaFlash のライト完了待ち
Userdef_SPIBSC_OCTAFLASH_ReadConfig2	OctaFlash のコンフィグレーションレジスタ 2 のリード
Userdef_SPIBSC_OCTAFLASH_WriteConfig2	OctaFlash のコンフィグレーションレジスタ 2 のライト
Userdef_SPIBSC_OCTAFLASH_ReadId	OctaFlash の ID 情報のリード
Userdef_SPIBSC_OCTAFLASH_Read	OctaFlash のデータリード
Userdef_SPIBSC_OCTAFLASH_Erase	OctaFlash のセクタイレース
Userdef_SPIBSC_OCTAFLASH_Write	OctaFlash のデータライト
Userdef_PreHardwareSetup	SPIBSC 初期化前に必要なハードウェア初期化処理
Userdef_PostHardwareSetup	SPIBSC 初期化後に必要なハードウェア初期化処理

## 5.9 関数仕様

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

INITSCT	
概要	ローダプログラムのセクション初期化
宣言	void INITSCT(void)
説明	RAM 領域に転送する必要がある初期値ありデータ（RAM 領域で実行する必要があるコードおよび定数データを含む）を ROM 領域から転送し、RAM 領域の初期値なしデータの初期化を行います。
引数	p_dtbl : 初期値ありデータのセクション情報が格納された領域へのポインタ p_btbl : 初期値なしデータのセクション情報が格納された領域へのポインタ
リターン値	なし

R_SC_HardwareSetup	
概要	ブートに使用するハードウェアの初期設定
宣言	void R_SC_HardwareSetup(void)
説明	使用する OctaFlash に最適な設定を行い、SPIBSC を外部アドレス空間リードモードに設定して OctaFlash へのアクセスを行います。 サンプルコードでは、R_SPIBSC_Setup 関数をコールして、OctaFlash（MX25UM51245G）のレジスタ設定とMX25UM51245Gの仕様に合わせた SPIBSC の初期設定を行います。
引数	なし
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

r_memclk_setup	
概要	メモリクロックの設定処理
宣言	void r_memclk_setup (void)
説明	R_SC_HardwareSetup 関数の実行前に、メモリクロックの設定を行う関数です。 本サンプルコードでは、何も処理を行いません。
引数	なし
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

R_SPIBSC_Setup	
概要	SPIBSC と OctaFlash の初期設定
宣言	void R_SPIBSC_Setup (uint32_t ddrhdr)
説明	使用する OctaFlash に最適な設定を行い、SPIBSC を外部アドレス空間リードモードに設定して OctaFlash へのアクセスを行います。 サンプルコードでは、以下の設定を行っています。 <ul style="list-style-type: none"> <li>● リードコマンドの変更 : H'03→H'EE</li> <li>● OctaFlash のレジスタの設定 <ul style="list-style-type: none"> <li>コンフィグレーションレジスタ 2 (アドレス H'0000_0000) : DOPI ビットに 1 を設定</li> <li>コンフィグレーションレジスタ 2 (アドレス H'0000_0300) : DC[2:0]ビットの設定 (DC[2:0]ビットの設定値は「表6.5 MX25UM51245Gの最大動作周波数に対して必要なタミーサイクル数の一覧」を参照してください。)</li> </ul> </li> <li>● QSPIn_SPCLK 動作周波数を変更 : P0φ/2→Gφ/2</li> </ul>
引数	ddrhdr : DDR モード/SDR モードの指定 HWSETUP_SPIBSC_USE_DDR : DDR モード HWSETUP_SPIBSC_USE_SDR : SDR モード
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

R_SPIBSC_Init	
概要	SPIBSC 初期設定
宣言	e_spibsc_err_t R_SPIBSC_Init(const spibsc_config_t *p_spibsc_config_tbl)
説明	引数 spibsc_config_tbl により、SPIBSC 関連レジスタの初期設定を行います。本関数終了時は、外部アドレス空間リードモードの設定になっています。 <ul style="list-style-type: none"> <li>• SPIBSC 関連端子の駆動能力の設定</li> <li>• SPIBSC のモジュールスタンバイの解除</li> <li>• SPIBSC のクロックの設定</li> <li>• SPIBSC 関連レジスタの設定</li> </ul>
引数	const spibsc_config_t : SPIBSC の初期設定テーブルへのポインタ *p_spibsc_config_tbl
リターン値	SPIBSC_SUCCESS : 正常終了
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

R_SPIBSC_ChangeMode	
概要	SPIBSC 動作モード設定
宣言	void R_SPIBSC_ChangeMode(uint8_t mode, uint8_t sdr_ddr, uint8_t table_no)
説明	引数 mode で指定された動作モードにより、引数 sdr_ddr および table_no で指定された転送フォーマットで OctaFlash にアクセスするように設定します。
引数	uint8_t mode : 動作モード SPIBSC_MODE_MANUAL : 手動モード SPIBSC_MODE_XIP : 外部アドレス空間リードモード uint8_t sdr_ddr : 転送フォーマット SPIBSC_DDR_TRANSFER : DDR 転送 SPIBSC_SDR_TRANSFER : SDR 転送 uint8_t table_no : 外部アドレス空間リードモード用のコマンド設定テーブル番号 0 : コマンド設定テーブル 0 を選択 (DDR リードコマンド) 1 : コマンド設定テーブル 1 を選択 (SDR リードコマンド)
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

R_SPIBSC_SPICMDIssue	
概要	OctaFlash への SPI コマンド発行（手動モード用）
宣言	spibsc_err_t R_SPIBSC_SPICMDIssue(uint8_t table_no, uint32_t addr, uint8_t *write_buff, uint32_t write_size, uint8_t *read_buff, uint32_t read_size)
説明	引数 table_no で指定した SPI コマンド発行用のコンフィグレーションテーブルを使用して、SPI コマンドを発行します。 table_no の内容にしたがって、ライトコマンド発行時は、引数 addr で指定されたアドレスから引数 write_size で指定されたバイト数分、引数*write_buff に格納されたデータをライトします。リードコマンド発行時は、引数 addr で指定されたアドレスから引数 read_size で指定されたバイト数分のデータをリードし、引数*read_buff で指定された領域に格納します。 なお、リードコマンド発行時に、read_size に 8 バイト以上のバイト数が指定された場合は、SPI コマンドを繰り返し発行してリードの処理を行います。
引数	uint8_t table_no : 使用するコマンドに対する設定情報が格納されたテーブル uint32_t addr : アドレス uint8_t * write_buff : ライトバッファのポインタ uint32_t write_size : ライトするバイト数 uint8_t * read_buff : リードバッファのポインタ uint32_t read_size : リードするバイト数
リターン値	SPIBSC_SUCCESS : 正常終了
注意事項	ライトコマンド発行時は read_size に"0"を、リードコマンド発行時は write_size に"0"を指定してください。 本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

R_SPIBSC_XipStopAccess	
概要	OctaFlash へのアクセス停止
宣言	void R_SPIBSC_XipStopAccess( void )
説明	外部アドレス空間リードモードで QSPIn_SSL をネゲートし、OctaFlash へのアクセスを停止します。 SPIBSC 関連レジスタの設定時に、OctaFlash へのアクセスが発生することがないように本関数をコールします。
引数	なし
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

R_SPIBSC_FlushReadCache	
概要	SPIBSC リードキャッシュのクリア
宣言	void R_SPIBSC_FlushReadCache( void )
説明	SPIBSC のリードキャッシュをクリアします。
引数	なし
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。



---

<b>Userdef_SPIBSC_OCTAFLASH_SetMode</b>	
概要	OctaFlash のレジスタの設定
宣言	void Userdef_SPIBSC_OCTAFLASH_SetMode(uint8_t mode, uint8_t sdr_dds)
説明	使用する OctaFlash の仕様に合わせて、引数 mode で指定されたビット幅のアクセスで、引数 sdr_dds で指定された転送フォーマットでアクセスできるように、OctaFlash 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、MX25UM51245Gのレジスタを設定し、8 ビット幅で DDR 転送でアクセスする設定を行っています。
引数	uint8_t mode : 動作モード SPIBSC_OCTA_OPI_DISABLE: single モード (ビット幅 1 ビット) SPIBSC_OCTA_OPI_ENABLE: OPI モード (ビット幅 8 ビット) uint8_t sdr_dds : 転送フォーマット SPIBSC_DDR_TRANSFER : DDR 転送 SPIBSC_SDR_TRANSFER : SDR 転送 (未サポート)
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

<b>Userdef_SPIBSC_OCTAFLASH_ReadStatus</b>	
概要	OctaFlash のステータスレジスタのリード
宣言	void Userdef_SPIBSC_OCTAFLASH_ReadStatus(uint8_t *p_status)
説明	使用する OctaFlash の仕様に合わせて、OctaFlash のステータスレジスタをリードして、引数*p_status にリードしたデータを格納する処理を実装してください。 サンプルコードでは、MX25UM51245Gの Status Register をリードする処理を行っています。
引数	uint8_t *p_status : ステータスレジスタからリードした値
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

<b>Userdef_SPIBSC_OCTAFLASH_ReadConfig</b>	
概要	OctaFlash のコンフィグレーションレジスタのリード
宣言	void Userdef_SPIBSC_OCTAFLASH_ReadConfig(uint8_t *p_config)
説明	使用する OctaFlash の仕様に合わせて、OctaFlash のコンフィグレーションレジスタをリードして、引数*p_config にリードしたデータを格納する処理を実装してください。 サンプルコードでは、MX25UM51245Gの Configuration Register をリードする処理を行っています。
引数	uint8_t *p_config : コンフィグレーションレジスタからリードした値
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

<b>Userdef_SPIBSC_OCTAFLASH_WriteStatus</b>	
概要	OctaFlash のステータスレジスタおよびコンフィグレーションレジスタのライト
宣言	void Userdef_SPIBSC_OCTAFLASH_WriteStatus(uint8_t *p_status, uint8_t *p_config)
説明	使用する OctaFlash の仕様に合わせて、OctaFlash のステータスレジスタおよびコンフィグレーションレジスタに、引数*p_status および引数*p_config で指定した値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、MX25UM51245Gの Status Register および Configuration Register にライトする処理を行っています。
引数	uint8_t *p_status : status レジスタへの設定値 uint8_t *p_config : config レジスタへの設定値
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

<b>Userdef_SPIBSC_OCTAFLASH_WriteEnable</b>	
概要	OctaFlash のライト許可
宣言	void Userdef_SPIBSC_OCTAFLASH_WriteEnable(void)
説明	使用する OctaFlash の仕様に合わせて、OctaFlash 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、MX25UM51245Gに"Write Enable(WREN)"コマンドを発行する処理を行っています。
引数	なし
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

<b>Userdef_SPIBSC_OCTAFLASH_WaitReady</b>	
概要	OctaFlash のライト完了待ち
宣言	void Userdef_SPIBSC_OCTAFLASH_WaitReady(void)
説明	使用する OctaFlash の仕様に合わせて、OctaFlash 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、MX25UM51245G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。
引数	なし
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。



---

Userdef\_SPIBSC\_OCTAFLASH\_ReadConfig2

---

概要	OctaFlash のコンフィグレーションレジスタ 2 のリード
宣言	void Userdef_SPIBSC_OCTAFLASH_ReadConfig2(uint32_t addr, uint8_t *p_config)
説明	使用する OctaFlash の仕様に合わせて、OctaFlash のコンフィグレーションレジスタ 2 をリードして、引数*p_config にリードしたデータを格納する処理を実装してください。 サンプルコードでは、MX25UM51245G の Configuration Register2 をリードする処理を行っています。
引数	uint32_t addr : コンフィグレーションレジスタ 2 のアドレス uint8_t *p_config : コンフィグレーションレジスタ 2 からリードした値
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

Userdef\_SPIBSC\_OCTAFLASH\_WriteConfig2

---

概要	OctaFlash のコンフィグレーションレジスタ 2 のライト
宣言	void Userdef_SPIBSC_OCTAFLASH_WriteConfig2(uint32_t addr, uint8_t config)
説明	使用する OctaFlash の仕様に合わせて、OctaFlash のコンフィグレーションレジスタ 2 に、引数 config で指定した値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、MX25UM51245G の Configuration Register2 にライトする処理を行っています。
引数	uint32_t addr : コンフィグレーションレジスタ 2 のアドレス uint8_t *p_config : コンフィグレーションレジスタ 2 への設定値
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

<b>Userdef_SPIBSC_OCTAFLASH_ReadId</b>	
概要	OctaFlash の ID 情報のリード
宣言	void Userdef_SPIBSC_OCTAFLASH_ReadId(uint8_t *p_id)
説明	使用する OctaFlash の仕様に合わせて、OctaFlash の ID 情報をリードして、引数 p_id にリードした値を格納する処理を実装してください。 サンプルコードでは、MX25UM51245G の ID 情報をリードする処理を行っています。
引数	uint8_t *p_id : ID 情報のリード値を格納するポインタ
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

<b>Userdef_SPIBSC_OCTAFLASH_Read</b>	
概要	OctaFlash のデータリード
宣言	void Userdef_SPIBSC_OCTAFLASH_Read(uint32_t addr, uint8_t *p_read_buff, int32_t read_size, uint8_t sdr_ddr)
説明	使用する OctaFlash の仕様に合わせて、OctaFlash のデータをリードして、引数 p_read_buff にリードした値を格納する処理を実装してください。 サンプルコードでは、MX25UM51245G からデータをリードする処理を行っています。
引数	uint32_t addr : リードするデータのアドレス uint8_t *p_read_buff : リードするデータを格納する領域へのポインタ int32_t read_size : リードするバイト数 uint8_t sdr_ddr : 転送フォーマット 本関数では使用しません。
リターン値	なし
注意事項	本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

Userdef_SPIBSC_OCTAFLASH_Erase	
概要	OctaFlash のセクタイレーズ
宣言	void Userdef_SPIBSC_OCTAFLASH_Erase(uint32_t addr)
説明	使用する OctaFlash の仕様に合わせて、引数 addr が属する OctaFlash のセクタをイレーズする処理を実装してください。 サンプルコードでは、MX25UM51245Gのセクタをイレーズする処理を行っています。
引数	uint32_t addr : イレーズするセクタのアドレス
リターン値	なし
注意事項	本関数を実行する前に、OctaFlash をライト許可状態に設定する必要があります。 本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。
Userdef_SPIBSC_OCTAFLASH_Write	
概要	OctaFlash のデータライト
宣言	void Userdef_SPIBSC_OCTAFLASH_Write(uint32_t addr, uint8_t *p_write_buff, int32_t write_size, uint8_t sdr_ddr)
説明	使用する OctaFlash の仕様に合わせて、引数 addr が示す OctaFlash のアドレスへ、引数 p_write_buff のデータをプログラムする処理を実装してください。 サンプルコードでは、MX25UM51245Gへデータをライトする処理を行っています。
引数	uint32_t addr : ライトするアドレス uint8_t *p_write_buff : ライトするデータを格納する領域へのポインタ int32_t write_size : ライトするバイト数 uint8_t sdr_ddr : 転送フォーマット 本関数では使用しません。
リターン値	なし
注意事項	本関数を実行する前に、OctaFlash をライト許可状態に設定する必要があります。 サンプルコードでは、Page Program コマンドを発行してライト処理を行うため、引数 addr は 256 バイトの境界のアドレス、引数 write_size は最大で 256 バイトのバイト数を指定してください。 本関数は、OctaFlash に配置して実行することができません。 本関数は、OctaFlash 以外の領域に配置する必要があります。

---

Userdef_PreHardwareSetup	
概要	SPIBSC 初期化前に必要なハードウェア初期化処理
宣言	void Userdef_PretHardwareSetup (void)
説明	SPIBSC の初期化前に実行する必要がある、ハードウェア初期化処理を記述するためのユーザ定義可能関数です。 本サンプルコードでは、何も処理を行いません。
引数	なし
リターン値	なし
注意事項	本関数に OctaFlash で実行することができない処理を記述する場合、本関数は OctaFlash 以外の領域に配置する必要があります。

---

Userdef_PostHardwareSetup	
概要	SPIBSC 初期化後に必要なハードウェア初期化処理
宣言	void Userdef_PostHardwareSetup (void)
説明	SPIBSC の初期化後に実行する必要がある、ハードウェア初期化処理を記述するためのユーザ定義可能関数です。R_SC_HardwareSetup 関数の終端でコールされます。 ローダプログラムでは、R_CPG_InitialiseHwIf 関数をコールして、EXTAL から 24[MHz]が入力される前提で、以下の CPG の設定を行います。 <ul style="list-style-type: none"> <li>• I<math>\phi</math> = 528[MHz], G<math>\phi</math> = 264[MHz], B<math>\phi</math> = 132[MHz], P1<math>\phi</math> = 66[MHz], P0<math>\phi</math> = 33[MHz], QSPI0_SPCLK = 132[MHz]</li> </ul>
引数	なし
リターン値	なし
注意事項	本関数に OctaFlash で実行することができない処理を記述する場合、本関数は OctaFlash 以外の領域に配置する必要があります。

## 5.10 ロードプログラムのフローチャート

### 5.10.1 ロードプログラム（全体）

図5.4にロードプログラム（全体）のフローチャートを示します。

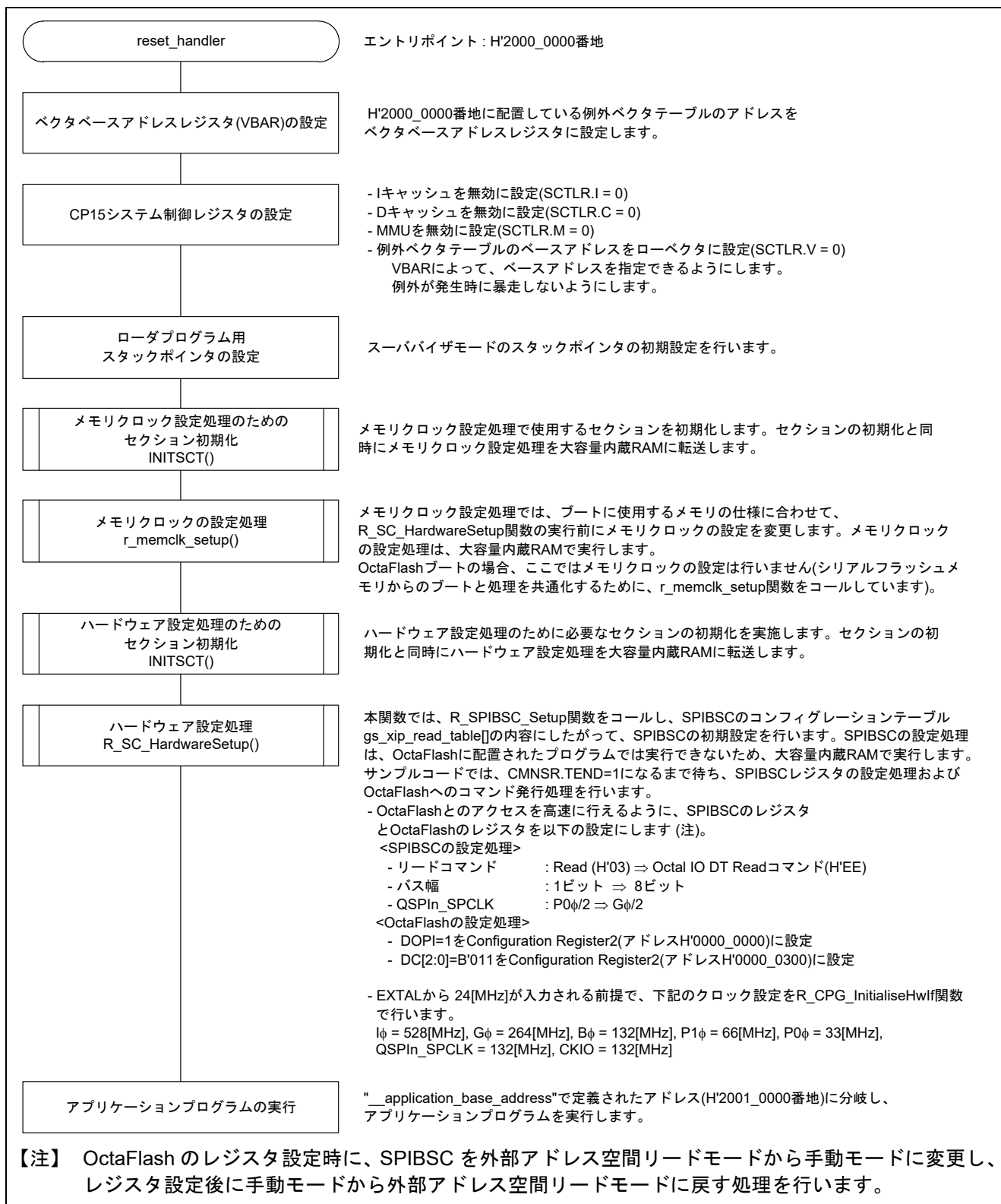


図5.4 ロードプログラム（全体）のフローチャート

### 5.10.2 ブートに使用するハードウェアの初期設定

ハードウェアの初期設定を行うための関数です。ローダプログラムでは高速に OctaFlash にアクセスができるように、R\_SPIBSC\_Setup 関数をコールして、SPIBSC および OctaFlash の設定を行います。

SPIBSC のレジスタおよび OctaFlash のレジスタを設定する処理は、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 で実行します。

図5.5にハードウェア初期設定処理のフローチャートを示します。

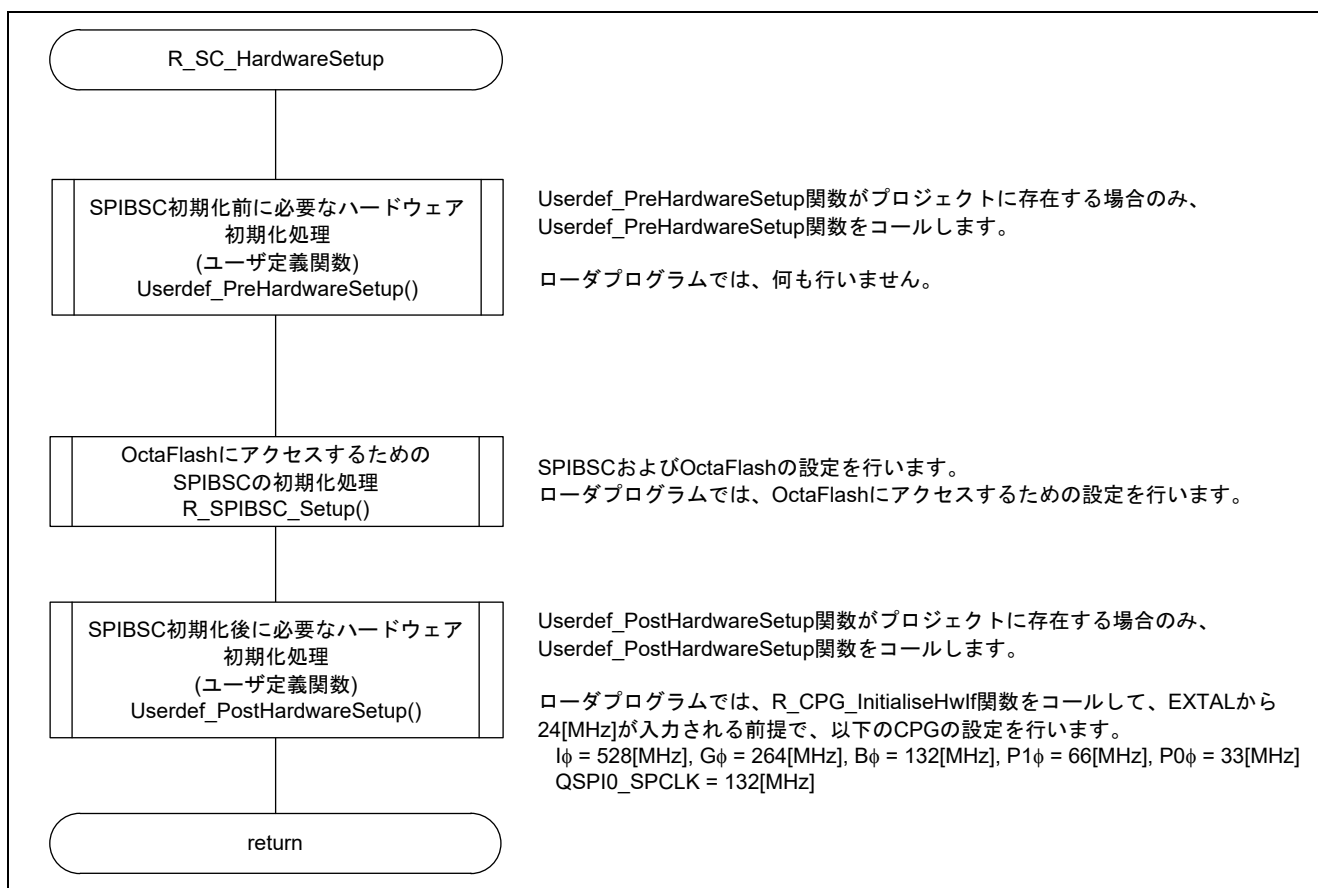


図5.5 ハードウェア初期設定処理のフローチャート

### 5.10.3 SPIBSC と OctaFlash の初期設定

ローダプログラムでは、高速に OctaFlash にアクセスできるように、OctaFlash のレジスタ (Configuration Register2(アドレス H'0000\_0000)の DOPI ビット、および Configuration Register2(アドレス H'0000\_0300)の DC[2:0] ビット) を設定し、DTR OPI モードを有効にしてダミーサイクル数を 14 に変更します。OctaFlash 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 OctaFlash に発行するリードコマンドを"Octa IO DT Read "(H'EE)に設定し、QSPIn\_SPCLK の動作周波数を  $G\phi/2$  に変更します。

ローダプログラムの処理は、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SPIBSCとOctaFlashの初期設定のフローチャートを、図5.7～図5.17に使用する関数のフローチャートを示します。

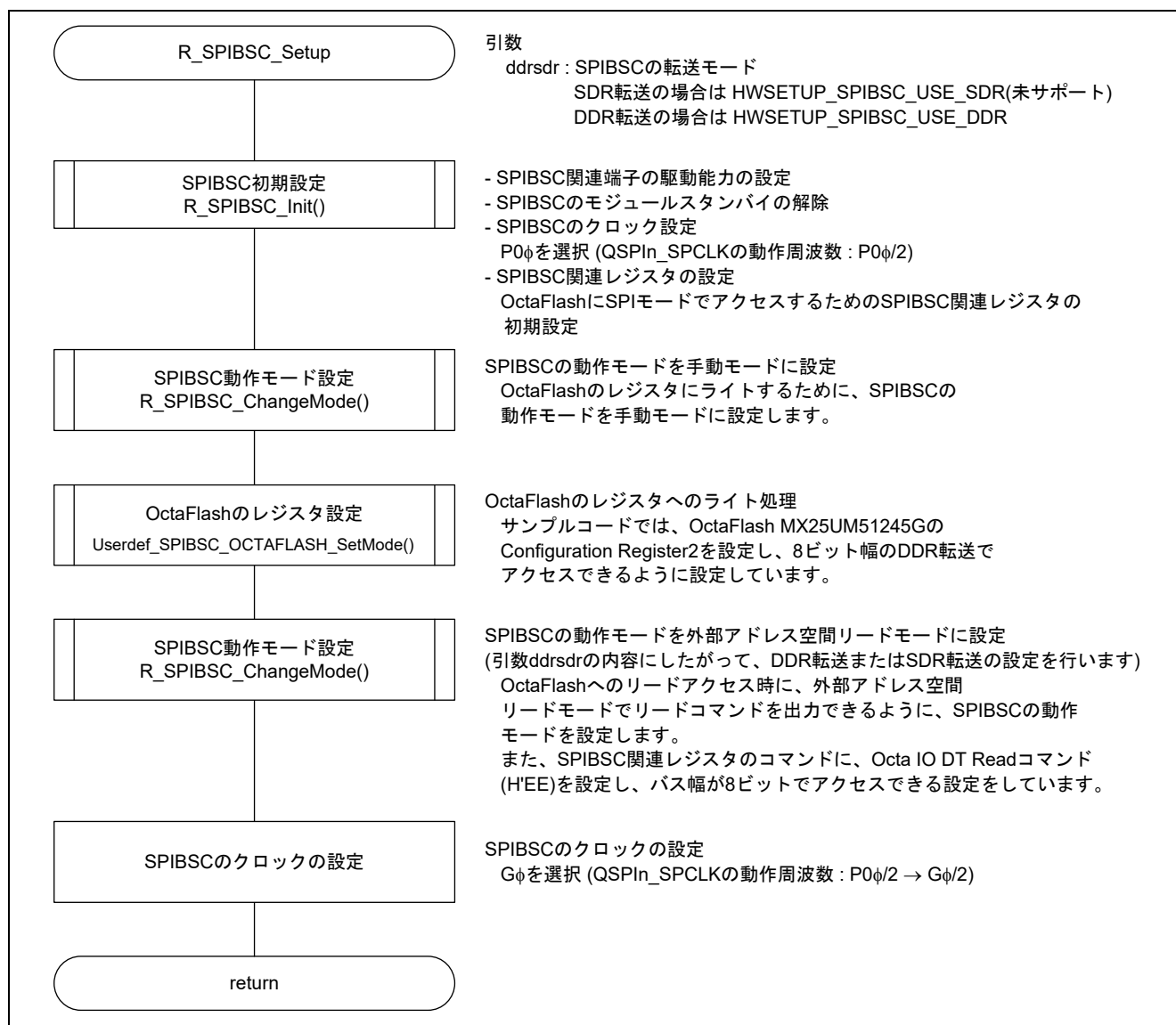


図5.6 SPIBSC と OctaFlash の初期設定のフローチャート

### 5.10.4 SPIBSC 初期設定

図5.7および図5.8に SPIBSC 初期設定のフローチャートを示します。ローダプログラムでは、OctaFlash に SPI モードでアクセスするための SPIBSC の初期設定を行います。

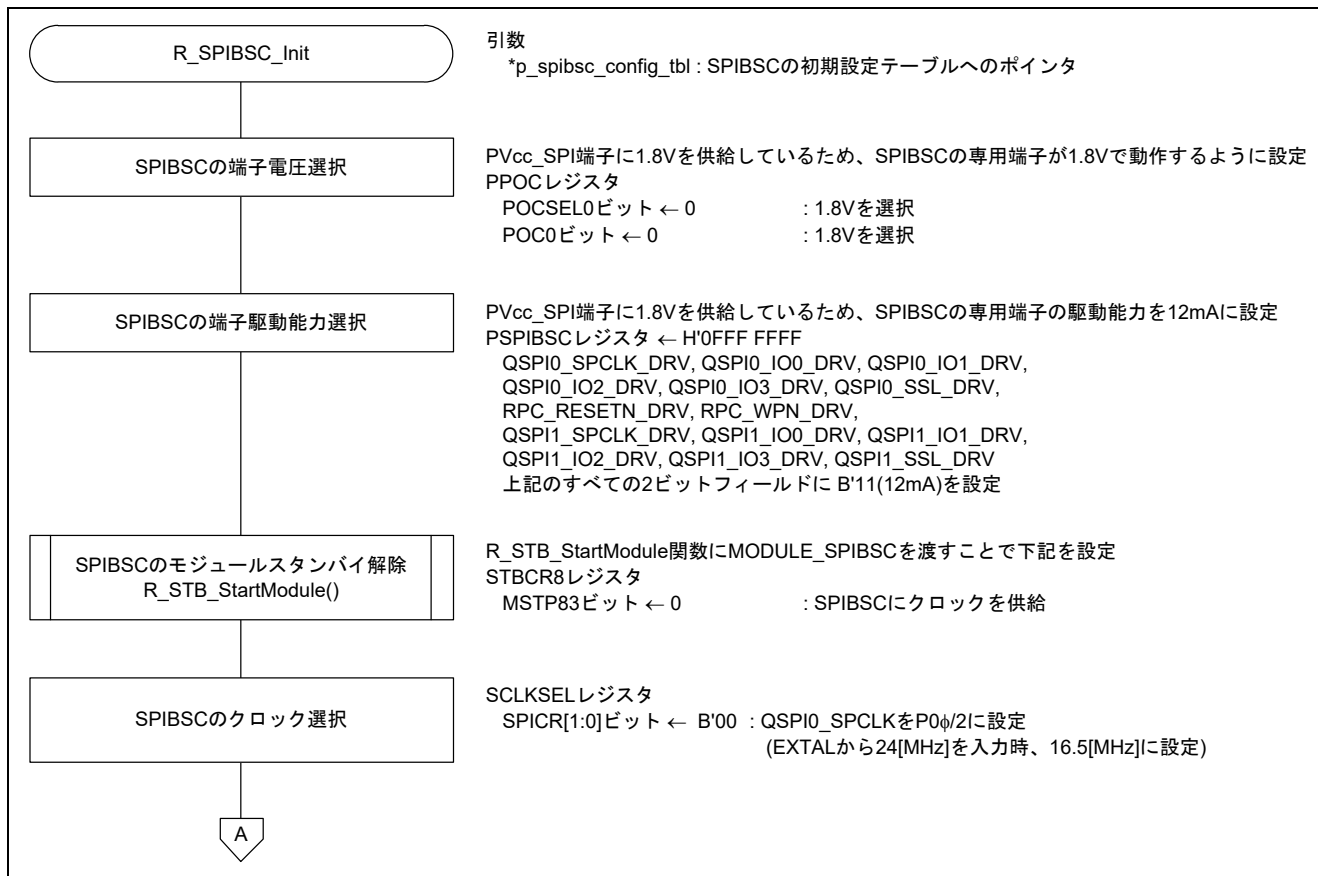


図5.7 SPIBSC 初期設定のフローチャート (1/2)



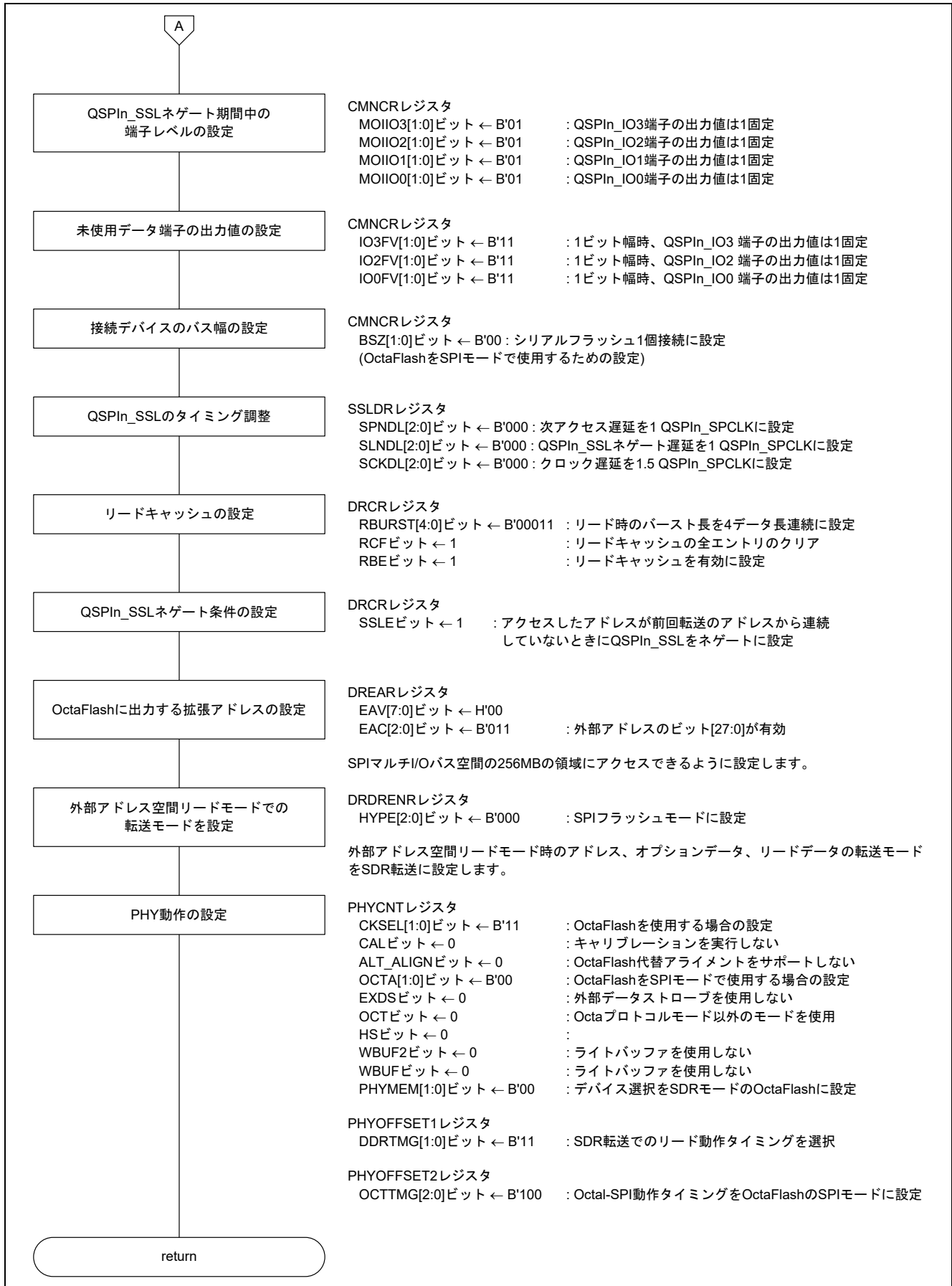


図5.8 SPIBSC 初期設定のフローチャート (2/2)

5.10.5 SPIBSC 動作モード設定

図5.9～図5.11に SPIBSC 動作モード設定を示します。

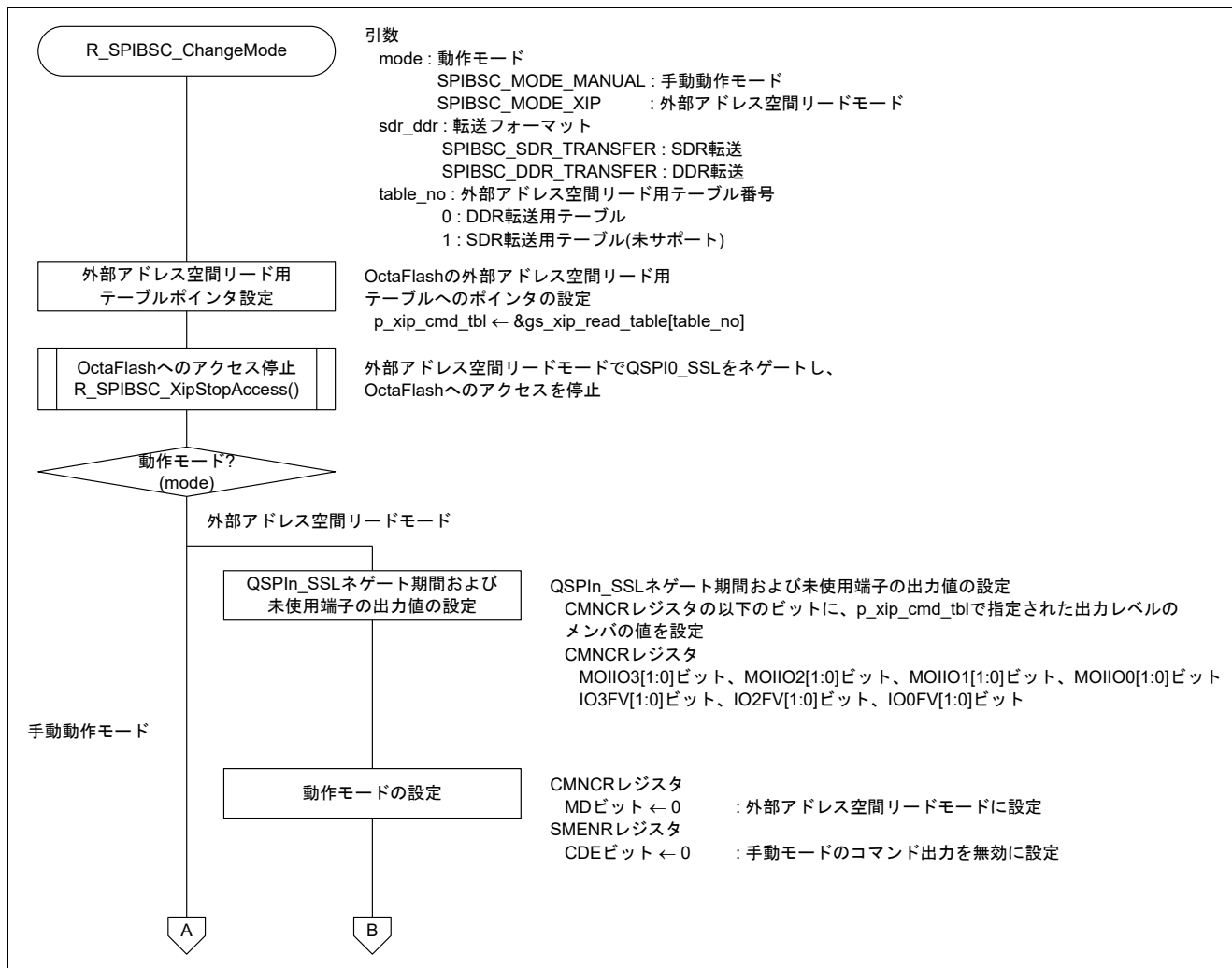


図5.9 SPIBSC 動作モード設定のフローチャート (1/3)

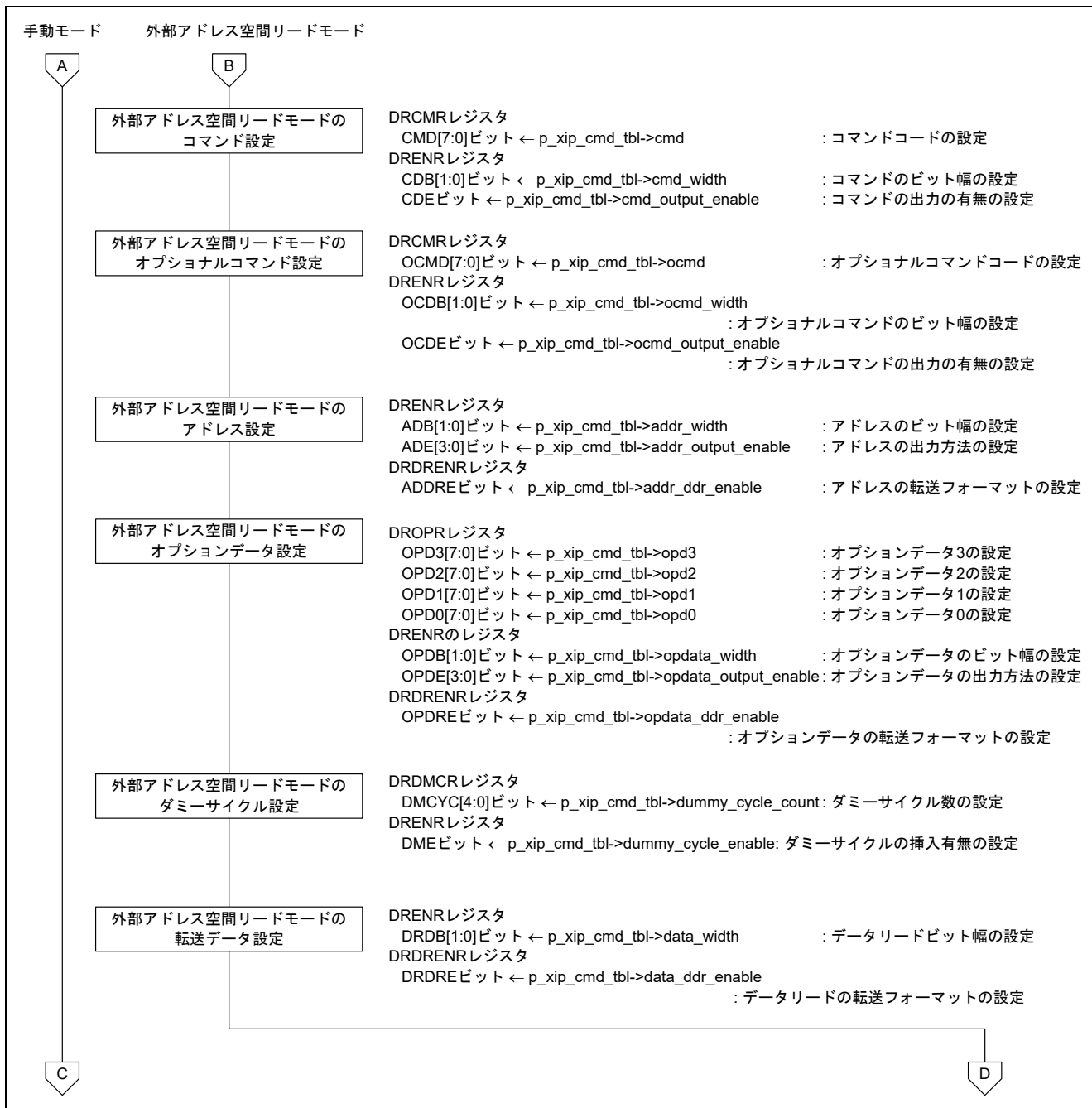


図5.10 SPIBSC 動作モード設定のフローチャート (2/3)

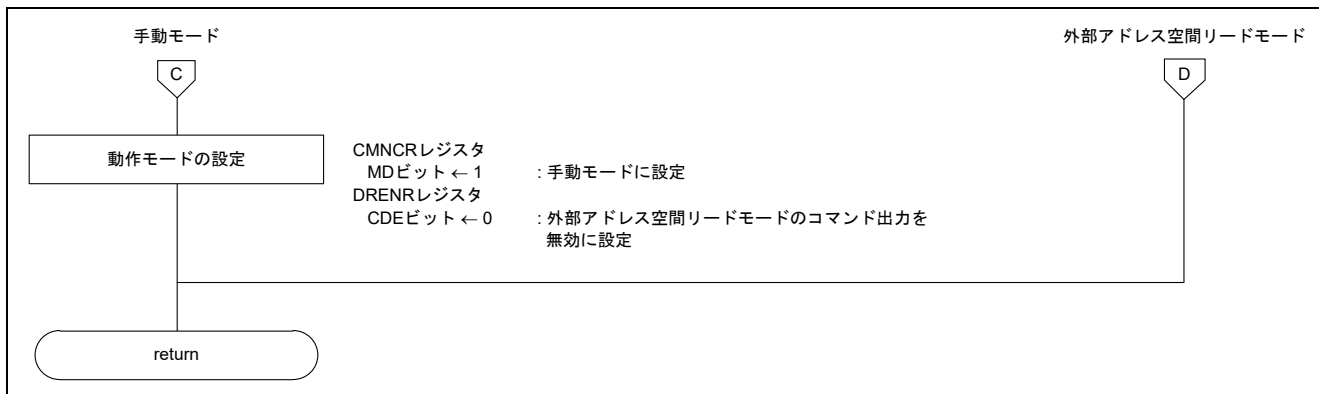


図5.11 SPIBSC 動作モード設定のフローチャート (3/3)

5.10.6 OctaFlash への SPI コマンド発行

図5.12~図5.17に OctaFlash への SPI コマンド発行のフローチャートを示します。本関数は手動モード時に使用してください。

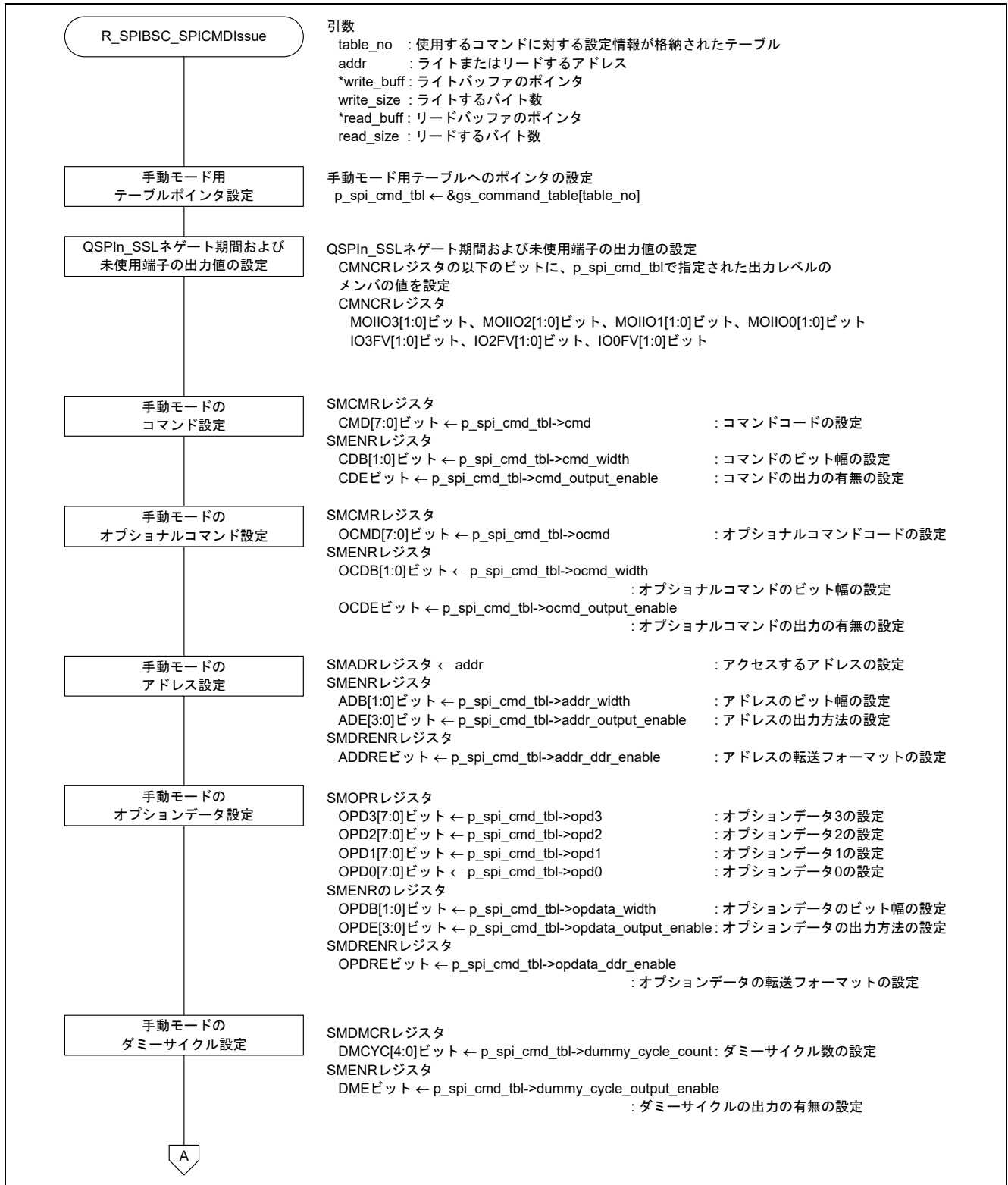


図5.12 OctaFlash への SPI コマンド発行のフローチャート (1/6)

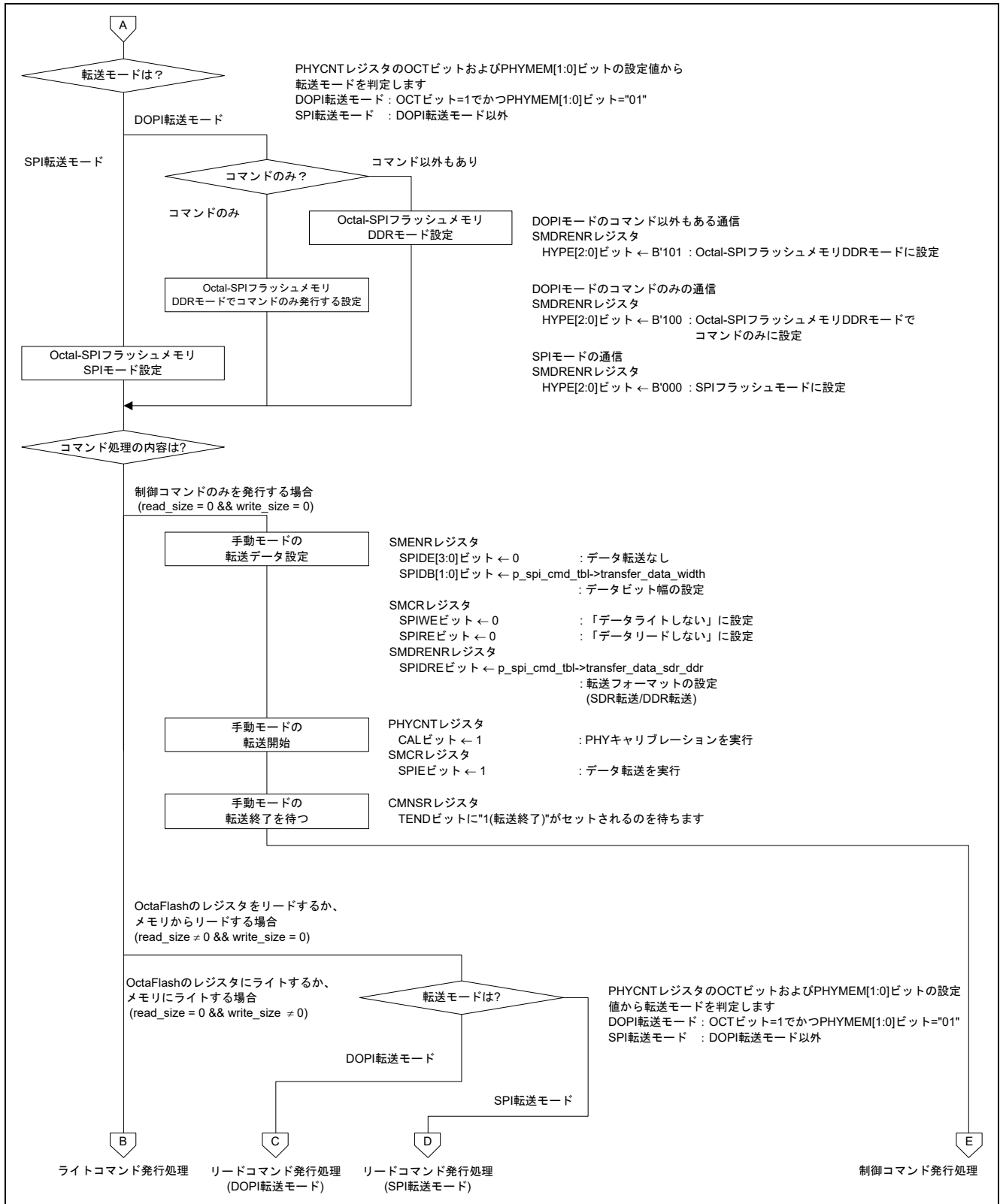


図5.13 OctaFlash への SPI コマンド発行のフローチャート (2/6)

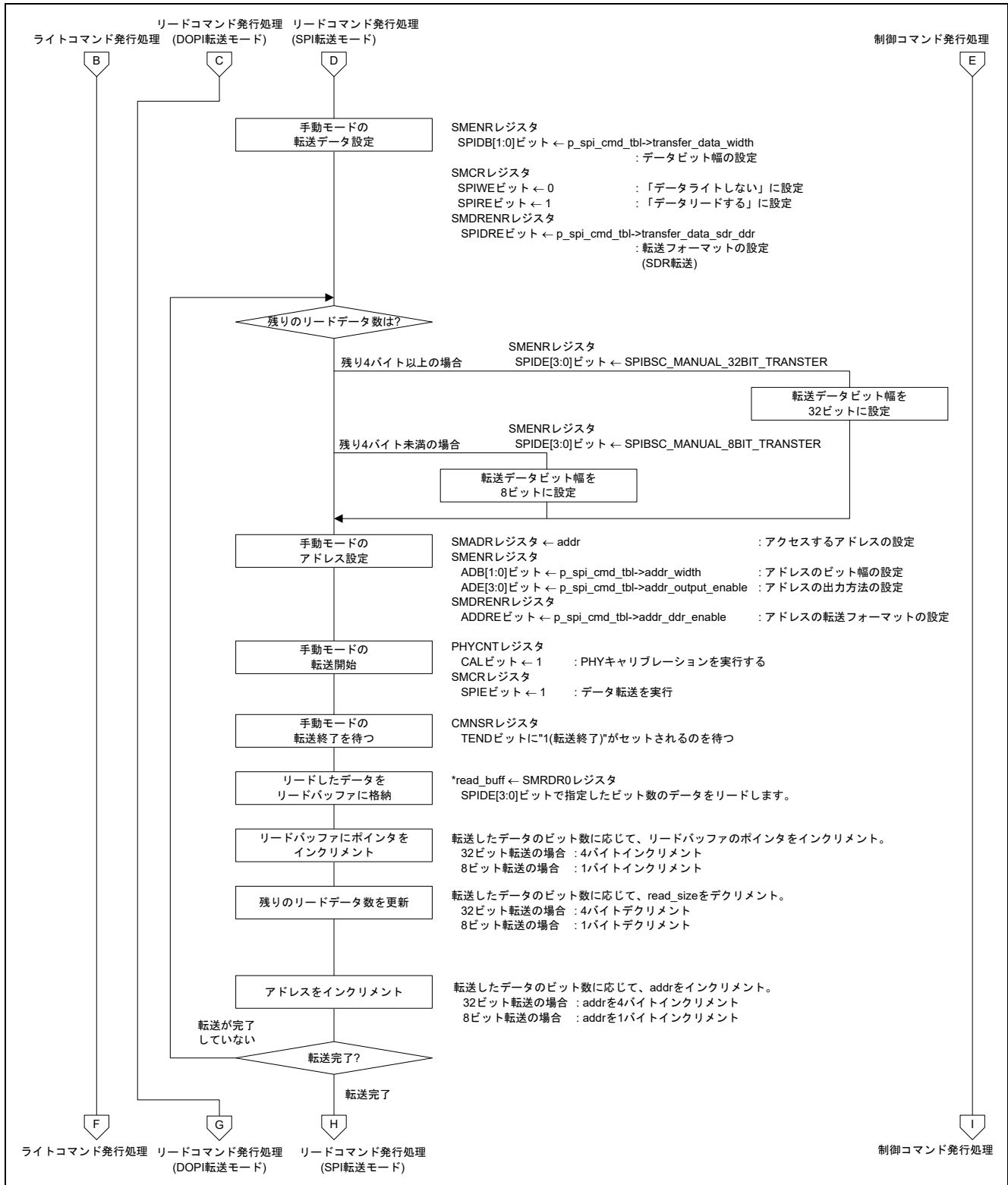


図5.14 OctaFlash への SPI コマンド発行のフローチャート (3/6)

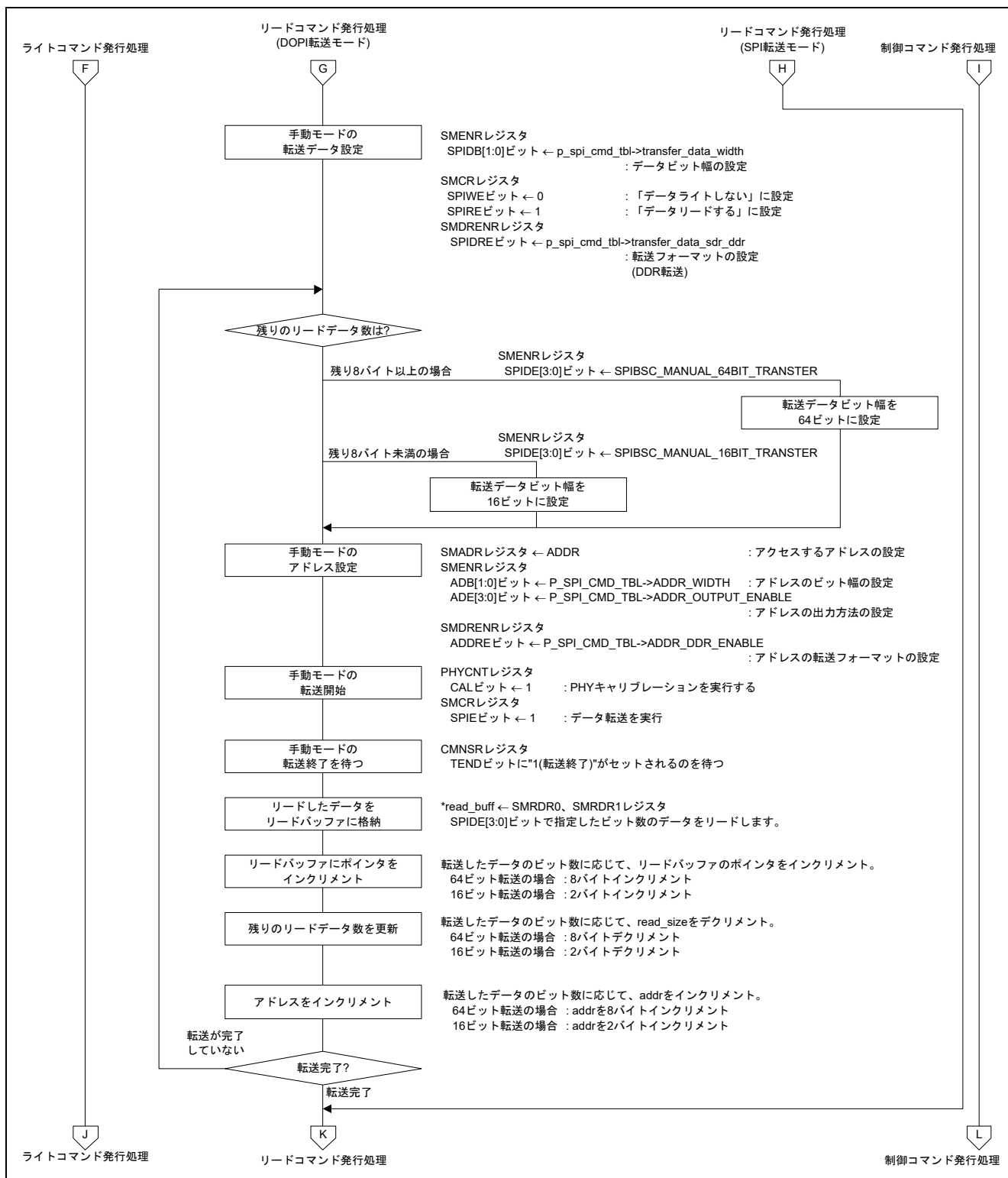


図5.15 OctaFlash への SPI コマンド発行のフローチャート (4/6)



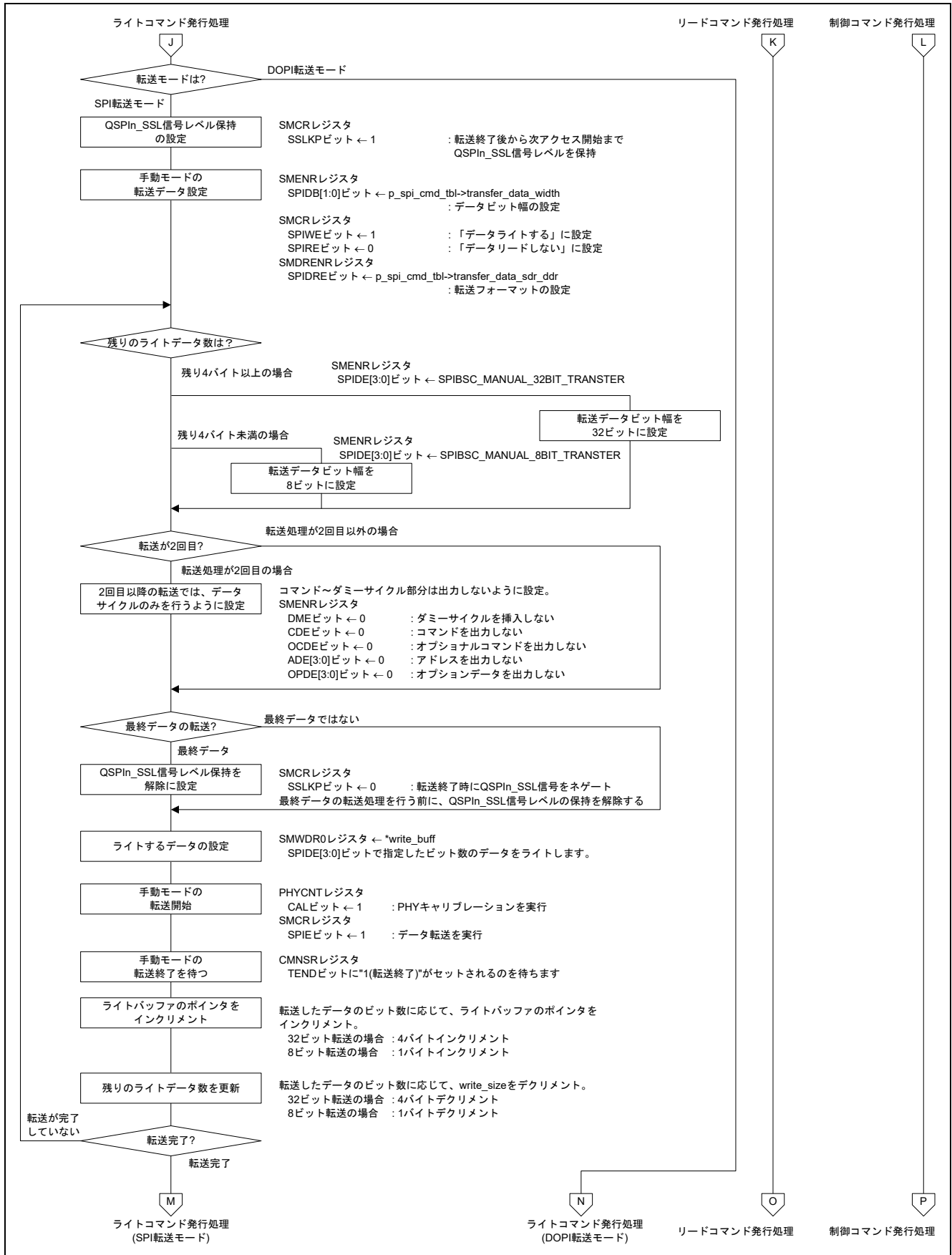


図5.16 OctaFlash への SPI コマンド発行のフローチャート (5/6)

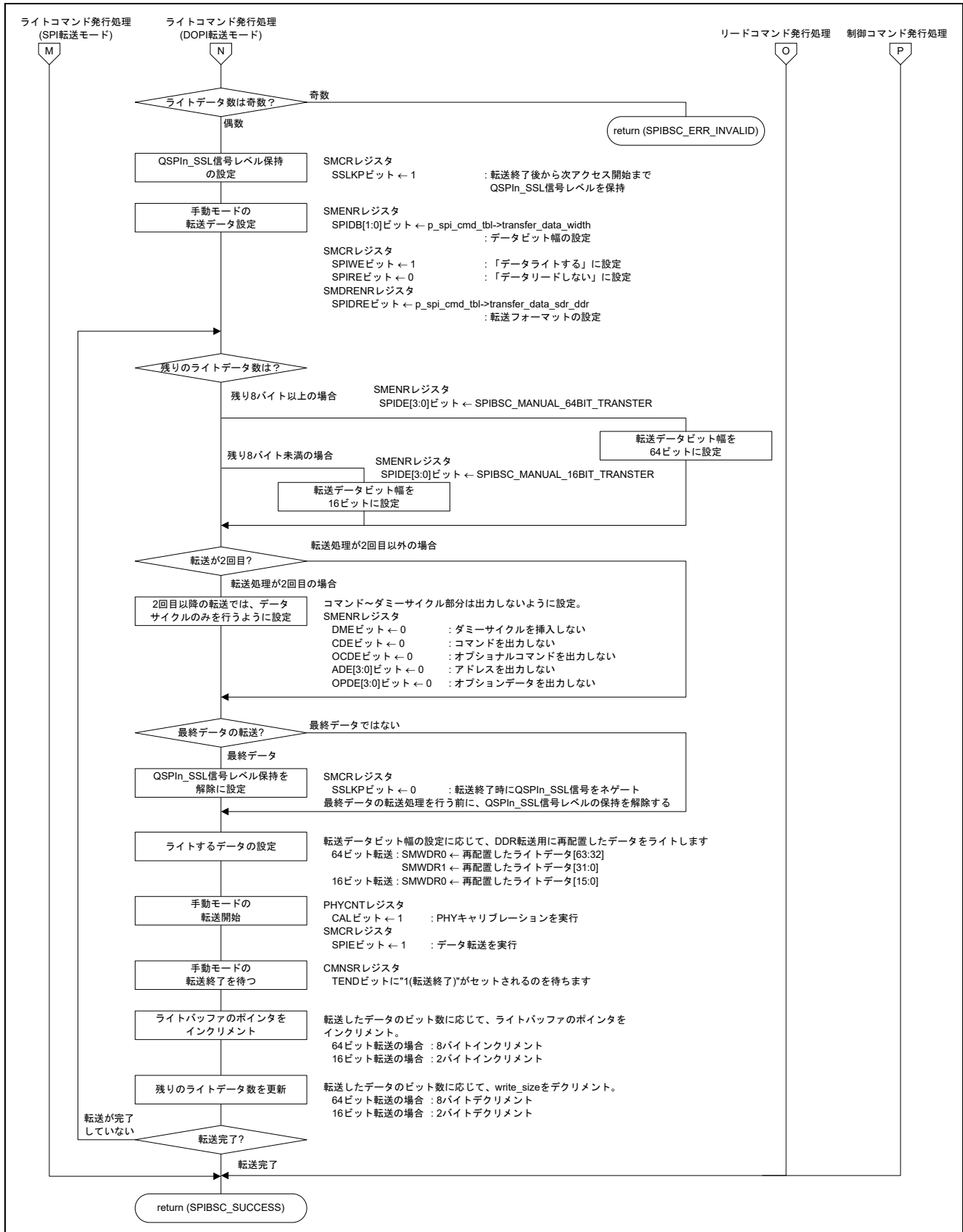


図5.17 OctaFlash への SPI コマンド発行のフローチャート (6/6)

## 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、Macronix社製 OctaFlash（MX25UM51245G）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
OctaFlash	<ul style="list-style-type: none"> <li>Macronix社製 OctaFlash</li> <li>型名：MX25UM51245G</li> <li>使用するリードコマンド：H'EE（8DTRD） バス幅 8 ビット、DTR-OPI モード 最大動作周波数 132MHz 時に、必要なダミーサイクル数 14</li> </ul>
SPIBSC	<ul style="list-style-type: none"> <li>接続するデバイス：Octal-SPI フラッシュ</li> <li>データバス幅：4 ビット（8 ビット幅）</li> <li>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）</li> <li>転送フォーマット：DDR 転送</li> </ul>

図6.1に、DDR転送のリード動作（サンプルコード初期状態）を示します。表6.2にサンプルコード初期状態でのレジスタ設定内容を示します。OctaFlash の DOPI モードでは、コマンド、オプション

コマンド、アドレス、およびデータの転送は 8 ビットバス幅で DDR 転送を行います。

SPIBSC は QSPI0\_SSL のアサートを基準にデータ出力を開始し、クロックの立ち上がりおよび立ち下がりの両方のエッジを基準にデータを出力します。MX25UM51245Gは SPIBSC から出力される QSPI0\_SSL アサート後の最初のクロックの立ち上がりエッジからデータのサンプリングを開始し、以降はクロックの両方のエッジを基準に入力を行います。

また、MX25UM51245Gからのデータ出力は、最初のデータストローブ（DQS）の立ち上がりを基準にデータ出力が開始され、以降はデータストローブの両方のエッジを基準にデータが出力されます。SPIBSC はデータストローブ（QSPI1\_SSL）の最初の立ち上がりエッジでデータ出力の開始を検出し、以降は両方のエッジでデータをサンプリングします。

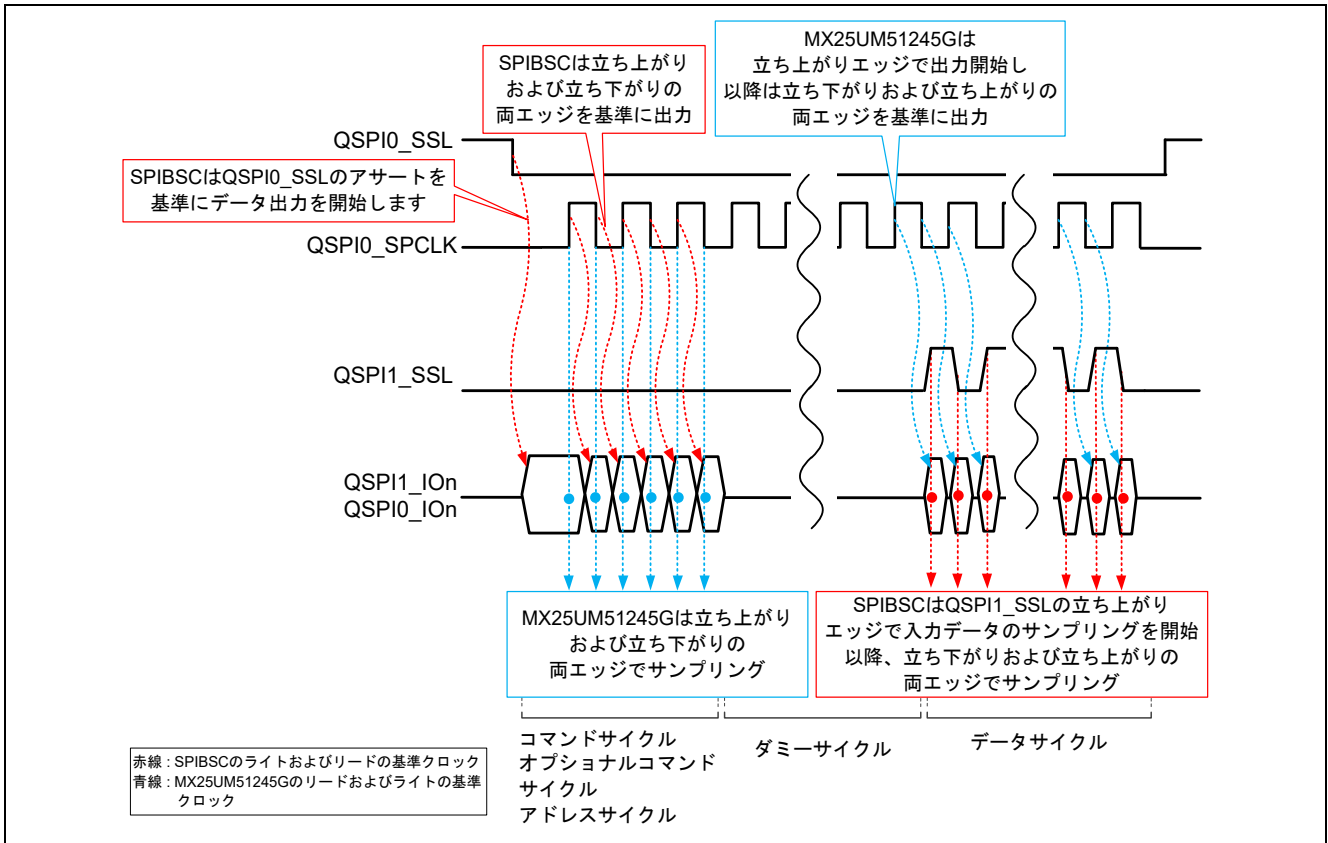


図6.1 DDR 転送のリード動作 (サンプルコード初期状態)

サンプルコードでは、初期状態で SPIBSC および OctaFlash のレジスタに、表6.2に示す内容を設定し、DOPI モードの動作を行います。

表6.2 サンプルコード初期状態でのレジスタ設定内容

設定項目	設定内容 (サンプルコードの初期状態)
リードコマンド設定	DRCMR.CMD[7:0] = 0xEE DREN.R.CDB[1:0] = SPIBSC_8BIT_WIDTH DREN.R.CDE = SPIBSC_OUTPUT_ENABLE
オプションコマンド設定	DRCMR.OCMD[7:0] = 0x11 DREN.R.OCDB[1:0] = SPIBSC_8BIT_WIDTH DREN.R.OCDE = SPIBSC_OUTPUT_DISABLE
アドレス設定	DREN.R.ADB[1:0] = SPIBSC_8BIT_WIDTH DREN.R.ADE[3:0] = SPIBSC_OUTPUT_ADDR_OCTA DRDREN.R.ADDRE = SPIBSC_DDR_TRANSFER
オプションデータ設定	DREN.R.OPDB[1:0] = SPIBSC_8BIT_WIDTH DREN.R.OPDE[3:0] = SPIBSC_OUTPUT_DISABLE DRDREN.R.OPDRE = SPIBSC_DDR_TRANSFER
ダミーサイクル設定	DREN.R.DME = SPIBSC_OUTPUT_ENABLE DRDMCR.DMCYC[4:0] = SPIBSC_DUMMY_14CYC
転送データ設定	DREN.R.DRDB[1:0] = SPIBSC_8BIT_WIDTH DRDREN.R.DRDRE = SPIBSC_DDR_TRANSFER
コンフィグレーション レジスタ 2 設定	DC[2:0]ビット = b'011 (14 サイクル) DOPI ビット = 1 (DTR OPI enable)

サンプルコードは、初期化時に実行される R\_SC\_HardwareSetup 関数内の処理で、R\_SPIBSC\_ChangeMode 関数で手動モードに切り替え後、OctaFlash のレジスタ設定を行うためのユーザ定義関数 Userdef\_SPIBSC\_OCTAFLASH\_SetMode をコールします。Userdef\_SPIBSC\_OCTAFLASH\_SetMode 関数では、使用するリードコマンドの仕様に合わせて、OctaFlash のレジスタ (Configuration Register2) の設定処理を行います。

表6.3にMX25UM51245GのConfiguration Register2 (アドレスH'0000\_0000) を、表6.4にMX25UM51245GのConfiguration Register2 (アドレスH'0000\_0300) を示します。Userdef\_SPIBSC\_SFLASH\_SetMode 関数により、表中に「          」で示した値を設定します。

表6.3 MX25UM51245Gの Configuration Register2 (アドレス H'0000\_0000)

ビット	ビット名	属性 (注)	説明
7-2	Reserved	-	Reserved
1	DOPI	V	B'00 = SPI B'01 = STR OPI enable
0	SOPI	V	B'10 = DTR OPI enable B'11 = inhibit

【注】 属性の"V"は"Volatile bit"を意味します。

表6.4 MX25UM51245Gの Configuration Register2 (アドレス H'0000\_0300)

ビット	ビット名	属性 (注 1)	説明
7-3	Reserved	-	Reserved
2,1,0	DC	V	Dummy cycle B'011(ダミーサイクル数 14) (注 2)

【注】 1. 属性の"V"は"Volatile bit"を意味します。

2. 表6.5に示すように、動作周波数によってダミーサイクル数は異なります。

表6.5にMX25UM51245Gの最大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。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'EE コマンドを使用し、QSPIn\_SPCLK を 132MHz で使用しているため、ダミーサイクル数が 14 サイクルの DC[2:0] = B'011 が最適な設定です。

表6.5 MX25UM51245Gの最大動作周波数に対して必要なダミーサイクル数の一覧

DC[2:0]ビット	ダミーサイクル数	最大動作周波数(MHz)
000(デフォルト)	20	200
001	18	166
010	16	166
011	14	133
100	12	104
101	10	104
110	8	84
111	6	66

## 6.2 OctaFlash を変更する場合のサンプルコード変更方法

OctaFlash を変更する場合、使用する OctaFlash 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6.6にサンプルコードの変更のポイントを示します。

表6.6 サンプルコードの変更のポイント

変更のポイント	内容	関連する見出し番号
外部アドレス空間リードモードでのリードコマンド発行時の出力信号	使用する OctaFlash 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 OctaFlash に出力する信号を変更します。	6.2.1
手動モードでのコマンド発行時の出力信号	使用する OctaFlash に合わせて、手動モードのコマンド発行時に OctaFlash に出力する信号を変更します。	6.2.2
OctaFlashのレジスタの設定	使用する OctaFlash 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 OctaFlash のレジスタの設定を行います。	6.2.3
OctaFlashのライト完了待ち	使用する OctaFlash に合わせて、OctaFlash のライト完了を待ちます。	6.2.4
OctaFlashのステータスレジスタのリード	使用する OctaFlash に合わせて、OctaFlash のステータスレジスタをリードします。	6.2.5
OctaFlashのコンフィグレーションレジスタのリード	使用する OctaFlash に合わせて、OctaFlash のコンフィグレーションレジスタをリードします。	6.2.6
OctaFlashのコンフィグレーションレジスタ2のリード	使用する OctaFlash に合わせて、OctaFlash のコンフィグレーションレジスタ2をリードします。	6.2.7
OctaFlashのライト許可	使用する OctaFlash に合わせて、OctaFlash 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）。	6.2.8
OctaFlashのステータス/コンフィグレーションレジスタのライト	使用する OctaFlash に合わせて、OctaFlash のステータス/コンフィグレーションレジスタにライトします。	6.2.9
OctaFlashのコンフィグレーションレジスタ2のライト	使用する OctaFlash に合わせて、OctaFlash のコンフィグレーションレジスタ2をライトします。	6.2.10
OctaFlashのID情報のリード	使用する OctaFlash に合わせて、OctaFlash の ID 情報をリードします。	6.2.11
OctaFlash のデータリード	使用する OctaFlash に合わせて、OctaFlash のデータをリードします。	6.2.12
OctaFlashのセクタイレズ	使用する OctaFlash に合わせて、OctaFlash のセクタをイレズします。	6.2.13
OctaFlashのデータライト	使用する OctaFlash に合わせて、OctaFlash をプログラムします。	6.2.14

【注】 OctaFlash によっては、OctaFlash 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6.6に示した設定の中で OctaFlash ブートに必要な設定は、SPIBSC と OctaFlash の初期設定処理 (R\_SPIBSC\_Setup 関数) を実行することにより行われます。サンプルコードのユーザ定義関数の処理を、使用する OctaFlash に合わせて変更することによって、OctaFlash のレジスタやメモリにアクセスすることができます。図6.2にSPIBSCとOctaFlashの設定処理のモジュール階層図を示し、それぞれの処理について6.2.1~6.2.14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。

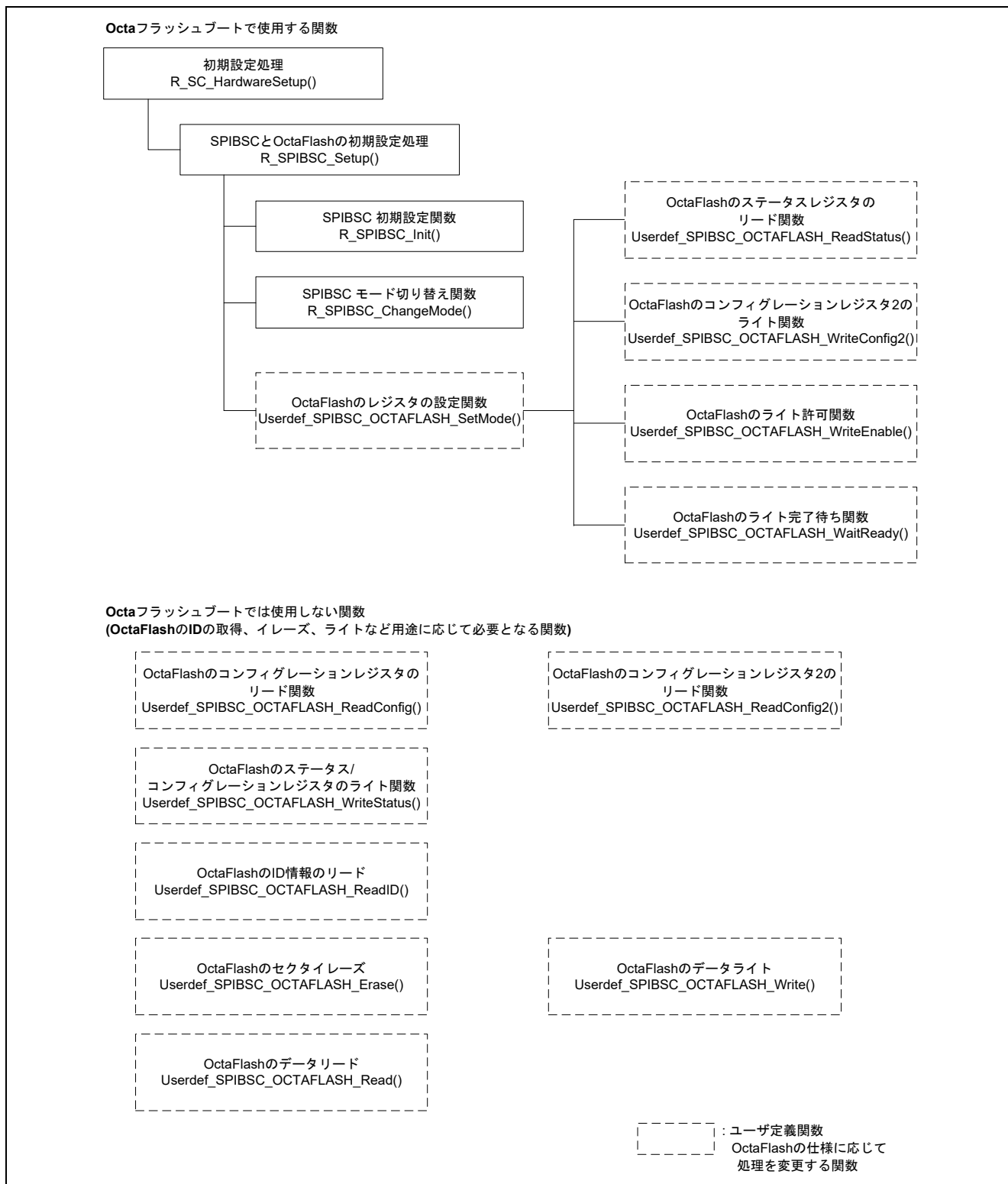


図6.2 SPIBSC と OctaFlash の設定処理のモジュール階層図

### 6.2.1 外部アドレス空間リードモードでの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 OctaFlash 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 OctaFlash を変更する場合は、OctaFlash 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する必要があります。

SPIBSC は、SPIBSC 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時に OctaFlash に出力するリードコマンドの信号を変更することが可能です。

サンプルコードでは、SPIBSC 外部アドレス空間リードモードのリードコマンド設定テーブル（表6.7）の内容を変更することにより、SPIBSC レジスタの設定値を変更し、リードコマンド発行時の出力信号を変更することが可能です。リードコマンド設定テーブルで指定した SPIBSC レジスタの設定値は、SPIBSC 動作モード設定関数（R\_SPIBSC\_ChangeMode）を実行することにより設定しています。なお、リードコマンドの設定内容に関連する OctaFlash のレジスタ設定（ダミーサイクル数、ビット幅など）は、OctaFlash のレジスタ設定関数（Userdef\_SPIBSC\_OCTAFLASH\_SetMode）により設定しています。使用する OctaFlash の仕様に合わせて、Userdef\_SPIBSC\_OCTAFLASH\_SetMode 関数での OctaFlash へのレジスタ設定処理の実装についても、合わせて変更してください。

図6.3にSPIBSCレジスタと外部アドレス空間リードモード時にOctaFlash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 OctaFlash のリードコマンドに合わせて、外部アドレス空間リードモードのリードコマンド設定テーブル（表6.7）の設定を変更してください。



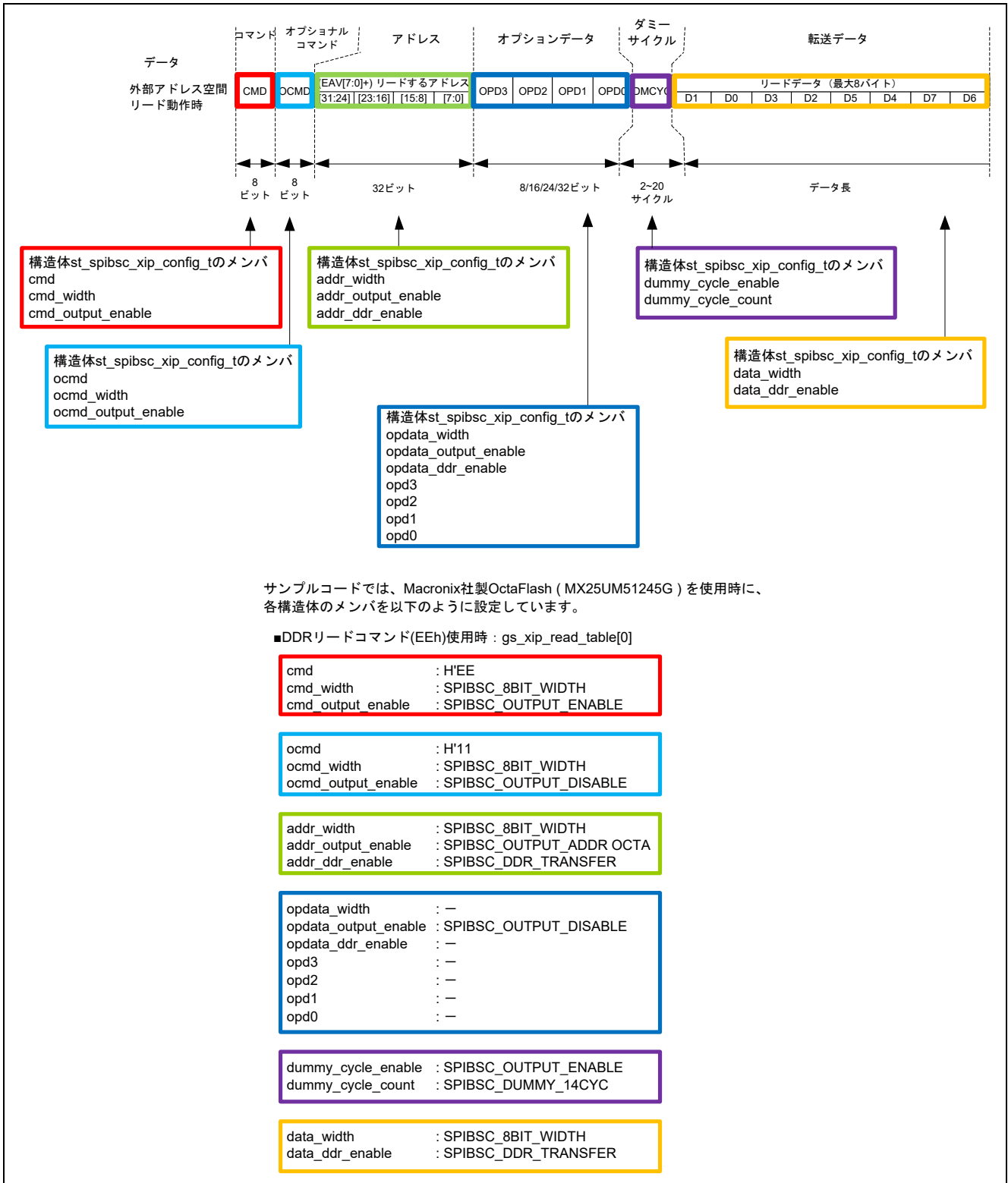


図6.3 SPIBSC レジスタと外部アドレス空間リードモード時に OctaFlash に出力される波形の関係

表6.7 外部アドレス空間リードモード用のコマンド設定テーブル gs\_xip\_read\_table[0]

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"OCTA_OPI_8DTRD"
uint8_t cmd	コマンドのコード	0xEE
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x11
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_output_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_ddr_enable	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t dummy_cycle_count	ダミーサイクル数	SPIBSC_DUMMY_14CYC
uint8_t data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t data_ddr_enable	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

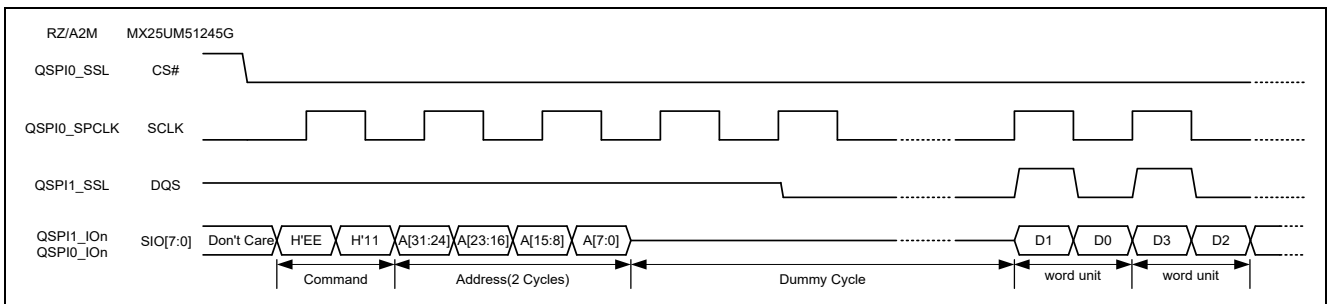


図6.4 EEH リードコマンドの波形フォーマット (参考)

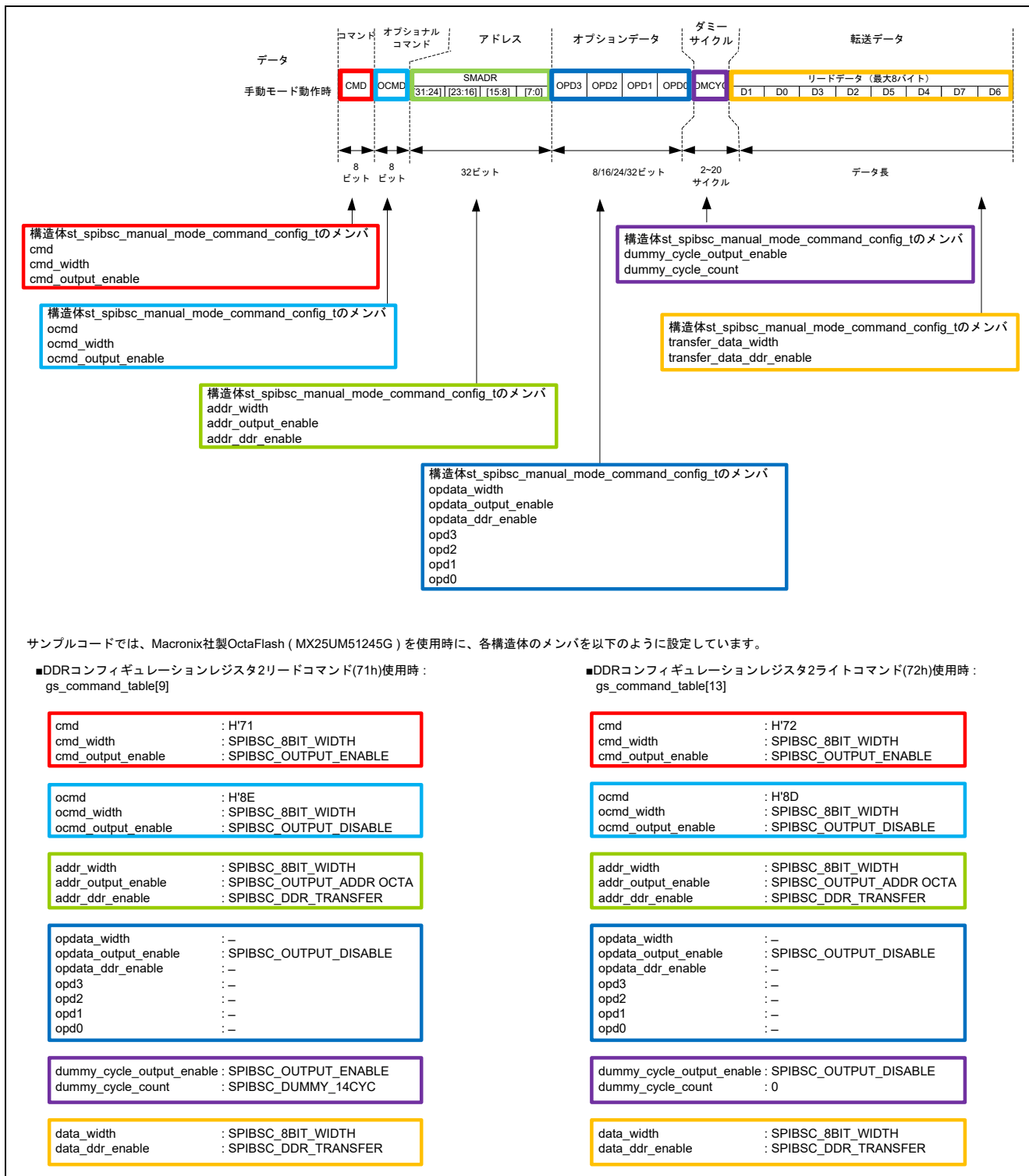
### 6.2.2 手動モードでのコマンド発行時の出力信号

手動モードでは、SPI マルチ I/O バス空間へのアクセスは、手動モードコントロールレジスタ (SMCR) の SPI データ転送イネーブルビット (SPIE) へ"1"を書き込むことにより動作を開始します。使用する OctaFlash を変更する場合は、OctaFlash のコマンド仕様に合わせて、コマンド発行時の出力信号を変更する必要があります。OctaFlash のレジスタアクセスまたはデータ書き込み時のライトアクセスを行う場合には、手動モードを使用して OctaFlash にコマンドを発行する必要があります。

SPIBSC は、SPIBSC レジスタを設定することにより、手動モード時に OctaFlash に出力するコマンドの信号を変更することが可能です。

サンプルコードでは、SPIBSC 手動モードのコマンド設定テーブル (表6.8~表6.25) の内容を変更することにより、SPIBSC レジスタの設定値を変更し、コマンド発行時の出力信号を変更することが可能です。コマンド設定テーブルで指定した SPIBSC レジスタの設定値は、OctaFlash への SPI コマンド発行関数 (R\_SPIBSC\_SPICMDIssue) を実行するたびに設定しています。なお、コマンドの設定内容に関連する OctaFlash のレジスタ設定 (ダミーサイクル数、ビット幅など) は、OctaFlash のレジスタ設定関数 (Userdef\_SPIBSC\_OCTAFLASH\_SetMode)により設定しています。使用する OctaFlash の仕様に合わせて、Userdef\_SPIBSC\_OCTAFLASH\_SetMode 関数での OctaFlash へのレジスタ設定処理の実装についても、合わせて変更してください。

図6.5にSPIBSCレジスタと手動モード時にOctaFlash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 OctaFlash のコマンドに合わせて、手動モードのコマンド設定テーブルの設定を変更してください。



### 6.2.3 OctaFlash のレジスタの設定

サンプルコードでは、使用するリードコマンド仕様に適した OctaFlash (MX25UM51245G) のレジスタ設定 (Configuration Register2 (アドレス H'0000\_0000) の DOPI ビット、およびアドレス H'0000\_0300 の DC[2:0] ビット) を行う処理を、ユーザ定義関数 `Userdef_SPIBSC_OCTAFLASH_SetMode` 関数で実現しています。

サンプルコードでは、リードコマンドとして Octa IO DT Read コマンド (H'EE) を使用しているため、ビット幅が 8 ビットの DDR 転送モードになるように Configuration Register2 (アドレス H'0000\_0000) の DOPI ビットを 1 に設定しています。また、動作周波数に応じて挿入されるダミーサイクル数が最小になるように、Configuration Register2 (アドレス H'0000\_0300) の DC[2:0] ビットの値を設定しています (詳細は表 6.5 MX25UM51245G の最大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)。ご使用の OctaFlash のリードコマンド仕様に適した、OctaFlash の制御レジスタの設定値となるように、`Userdef_SPIBSC_OCTAFLASH_SetMode` 関数の実装を変更してください。

図 6.6 に、サンプルコードの `Userdef_SPIBSC_OCTAFLASH_SetMode` 関数のフローを示します。

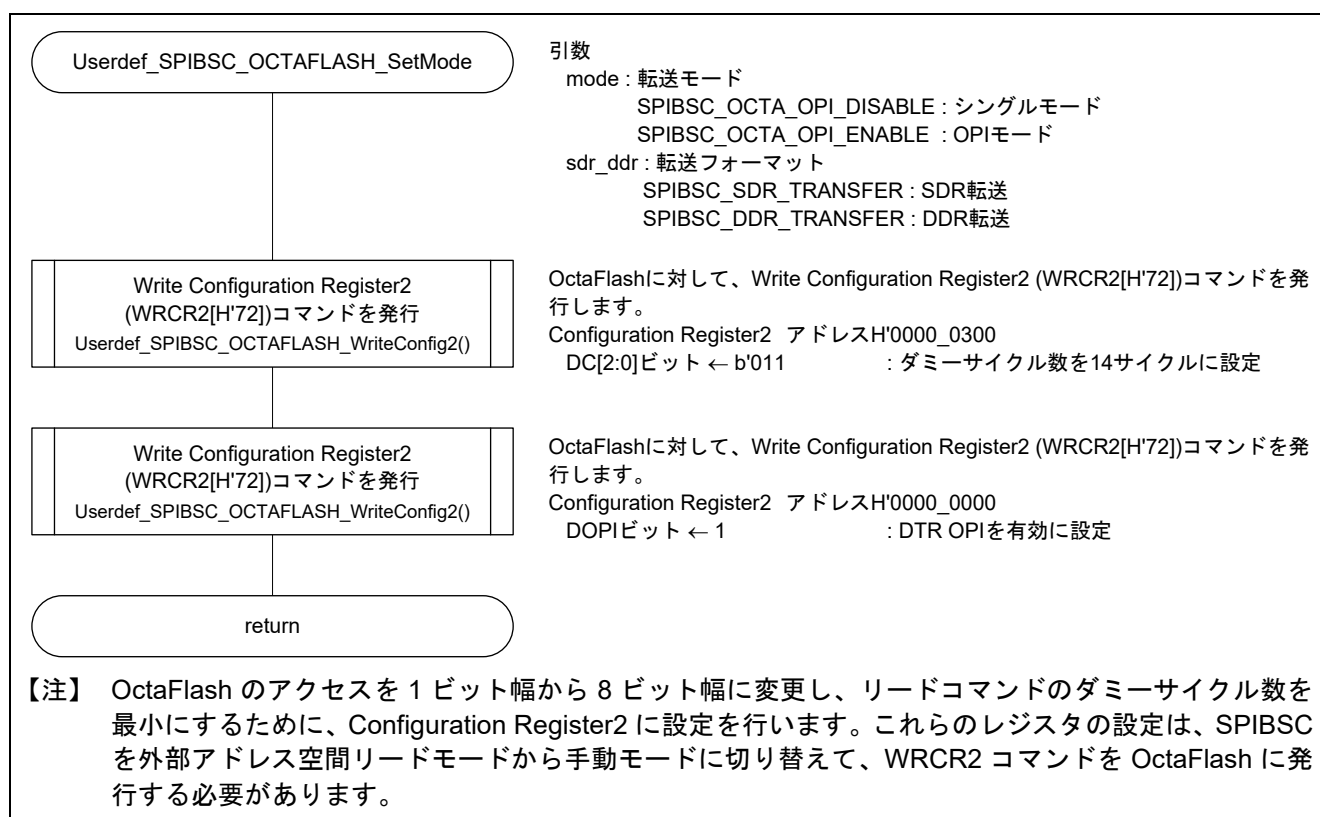


図 6.6 `Userdef_SPIBSC_OCTAFLASH_SetMode` 関数のフロー

### 6.2.4 OctaFlash のライト完了待ち

OctaFlash の Status Register やメモリにライトした場合、OctaFlash はビジー状態に遷移します。次の OctaFlash へのアクセス処理を行うためには、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。

サンプルコードでは、このウェイト処理を Userdef\_SPIBSC\_OCTAFLASH\_WaitReady 関数で実現しています。

ご使用の OctaFlash の仕様に合わせて、OctaFlash のライト完了までウェイトするように、Userdef\_SPIBSC\_OCTAFLASH\_WaitReady 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、ライトが完了するまでウェイトする処理を行います。

図6.7に、サンプルコードのUserdef\_SPIBSC\_OCTAFLASH\_WaitReady関数のフローを示します。

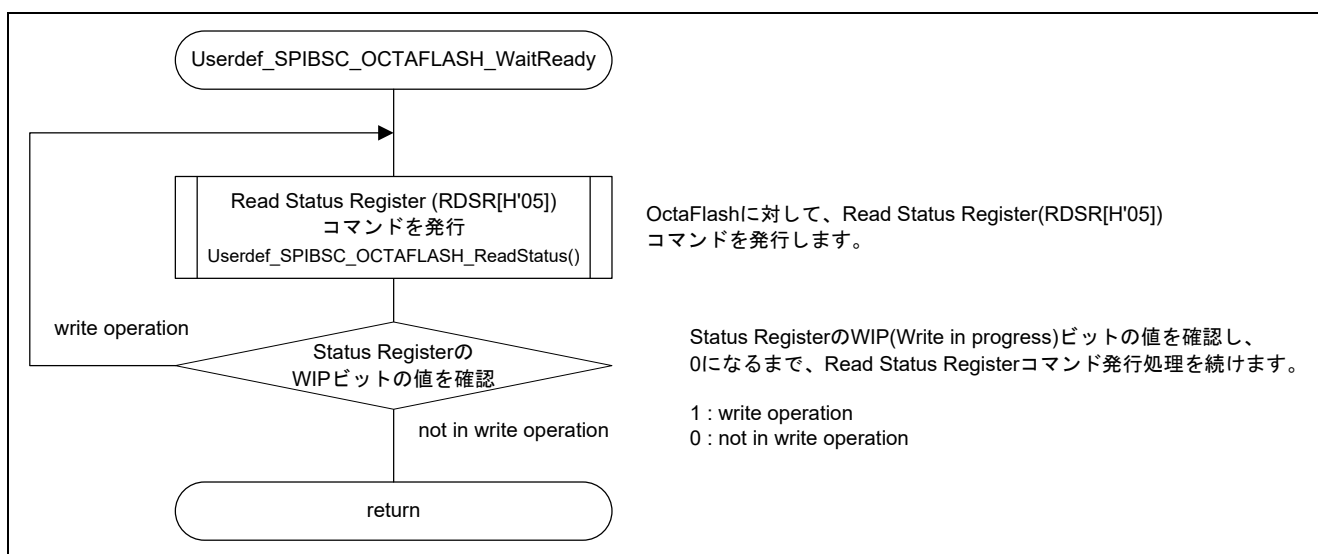


図6.7 Userdef\_SPIBSC\_OCTAFLASH\_WaitReady 関数のフロー

### 6.2.5 OctaFlash のステータスレジスタのリード

サンプルコードでは、OctaFlash の Status Register のリード処理を Userdef\_SPIBSC\_OCTAFLASH\_ReadStatus 関数で実現しています。

ご使用の OctaFlash の仕様に合わせて、OctaFlash のステータスレジスタをリードするように、Userdef\_SPIBSC\_OCTAFLASH\_ReadStatus 関数を実装してください。

図6.8に、サンプルコードの Userdef\_SPIBSC\_OCTAFLASH\_ReadStatus 関数のフローを示します。

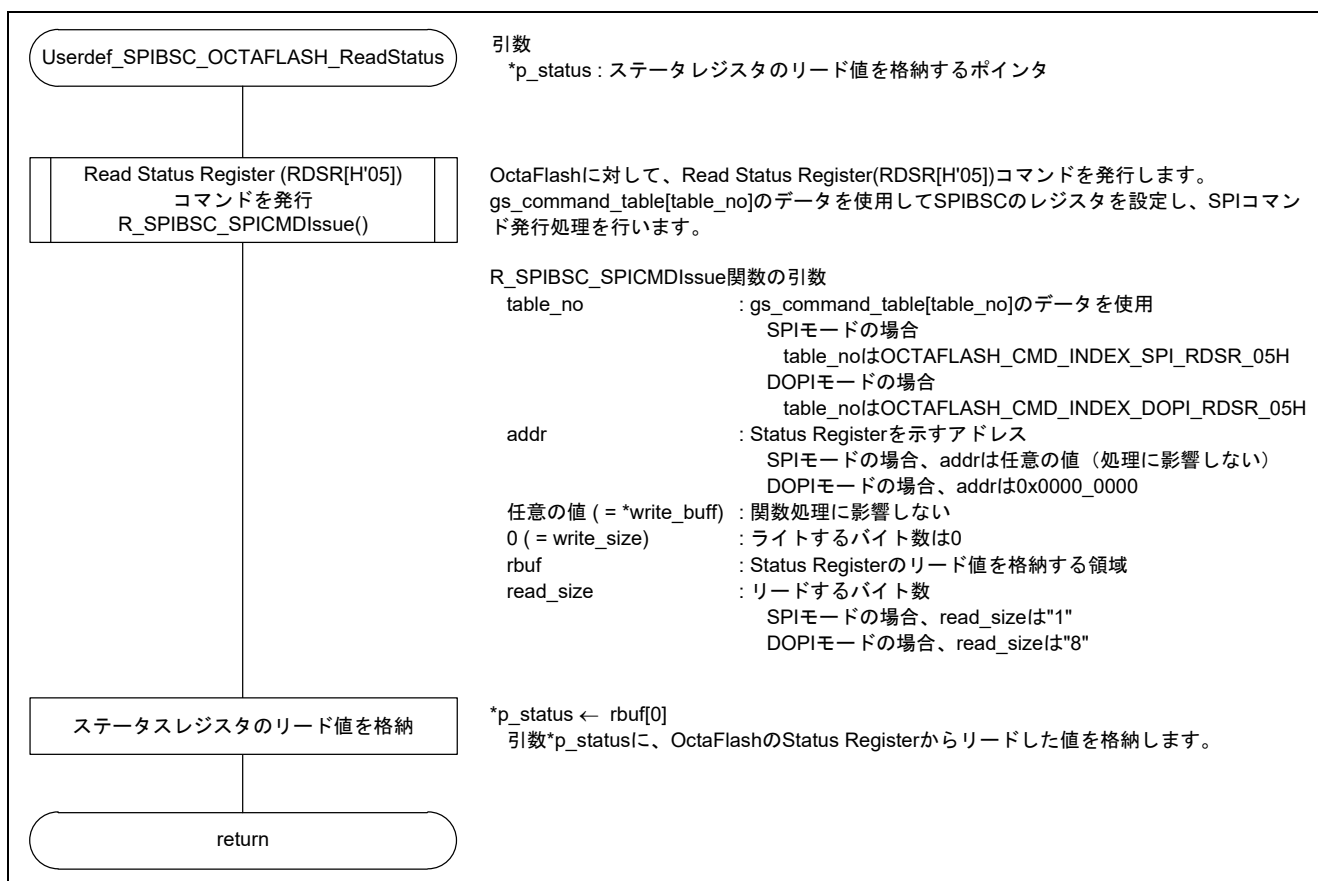


図6.8 Userdef\_SPIBSC\_OCTAFLASH\_ReadStatus 関数のフロー

表6.8 手動モード用のコマンド設定テーブル gs\_command\_table[0] : SPIモードのRDSRコマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"SPI_RDSR"
uint8_t cmd	コマンドのコード	0x05
uint8_t cmd_width	コマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x00
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_1BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_1BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_1BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_SDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

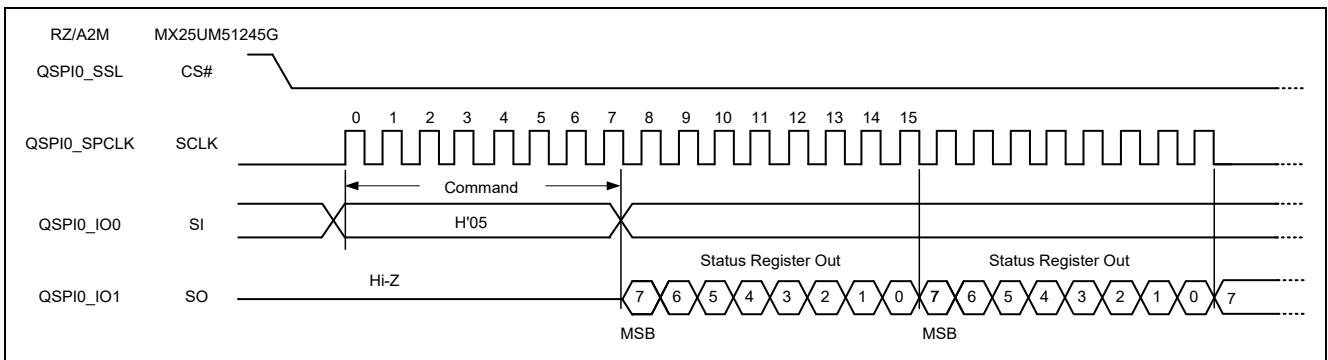


図6.9 SPIモードのRDSRコマンドの波形フォーマット (参考)



表6.9 手動モード用のコマンド設定テーブル gs\_command\_table[7] : DOPI モードの RDSR コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_RDSR"
uint8_t cmd	コマンドのコード	0x05
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0xFA
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t dummy_cycle_count	ダミーサイクル数	SPIBSC_DUMMY_14CYC
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

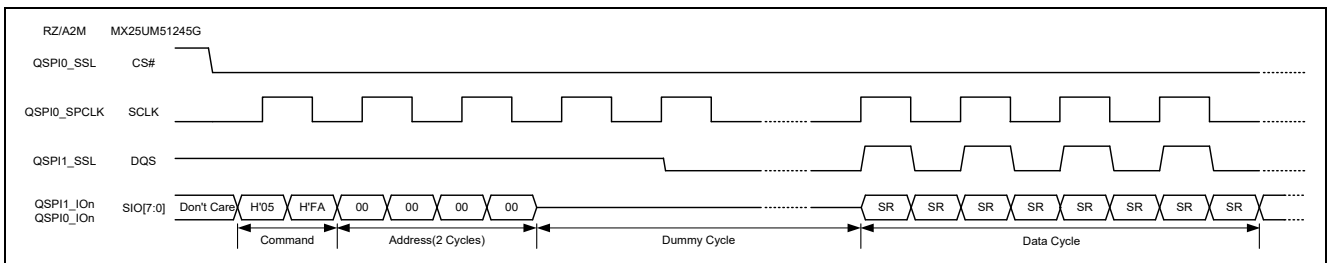


図6.10 DOPI モードの RDSR コマンドの波形フォーマット (参考)

### 6.2.6 OctaFlash のコンフィグレーションレジスタのリード

サンプルコードでは、OctaFlash の Configuration Register のリード処理を Userdef\_SPIBSC\_OCTAFLASH\_ReadConfig 関数で実現しています。

ご使用の OctaFlash の仕様に合わせて、OctaFlash のコンフィグレーションレジスタをリードするように、Userdef\_SPIBSC\_OCTAFLASH\_ReadConfig 関数を実装してください。

図6.11に、サンプルコードの Userdef\_SPIBSC\_OCTAFLASH\_ReadConfig 関数のフローを示します。

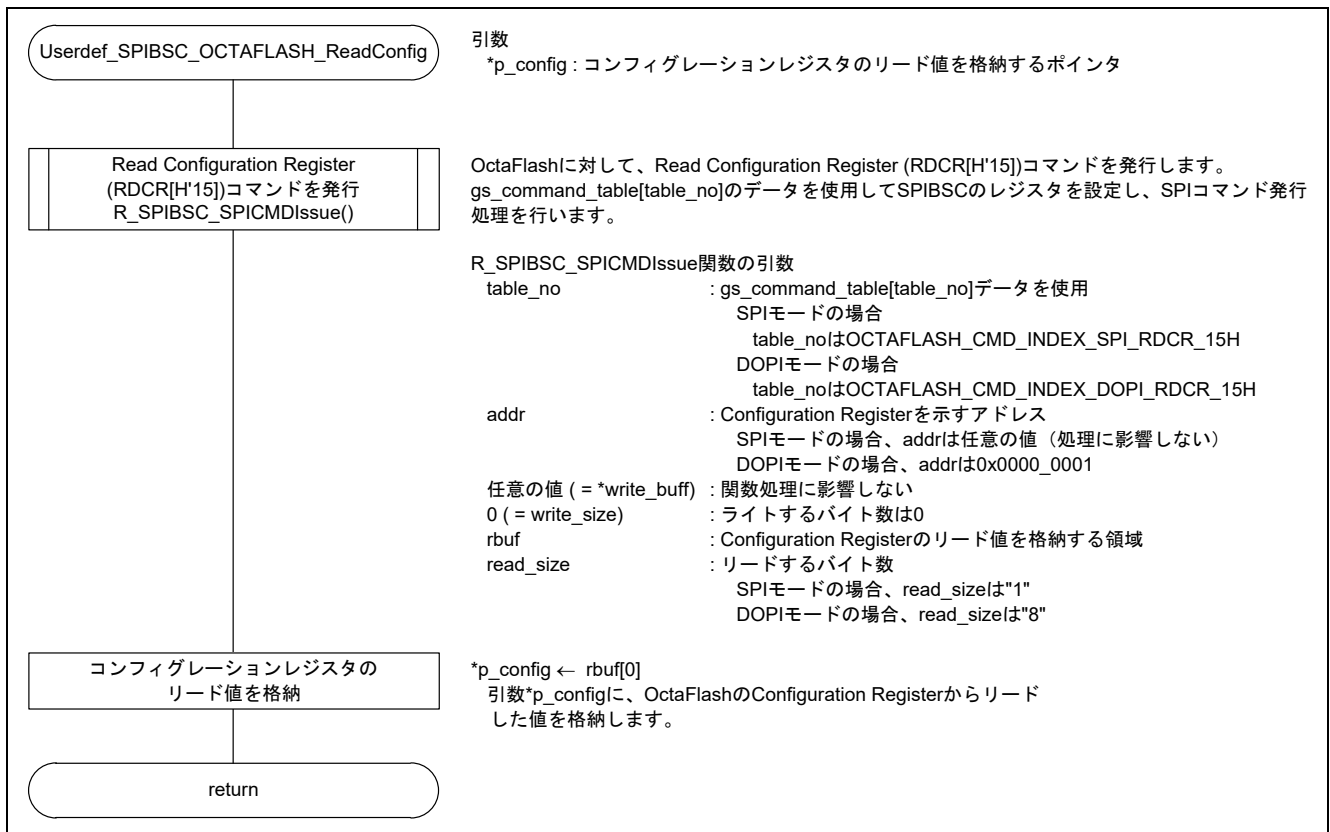


図6.11 Userdef\_SPIBSC\_OCTAFLASH\_ReadConfig 関数のフロー

表6.10 手動モード用のコマンド設定テーブル gs\_command\_table[1] : SPIモードのRDCRコマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"SPI_RDCR"
uint8_t cmd	コマンドのコード	0x15
uint8_t cmd_width	コマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x00
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_1BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_1BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_1BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_SDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

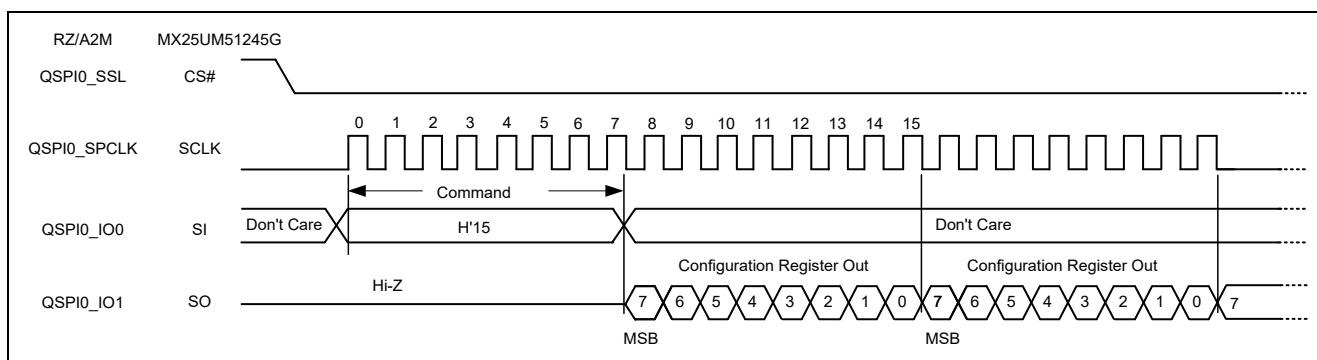


図6.12 SPIモードのRDCRコマンドの波形フォーマット (参考)

表6.11 手動モード用のコマンド設定テーブル gs\_command\_table[8] : DOPI モードの RDCR コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_RDCR"
uint8_t cmd	コマンドのコード	0x15
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0xEA
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t dummy_cycle_count	ダミーサイクル数	SPIBSC_DUMMY_14CYC
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

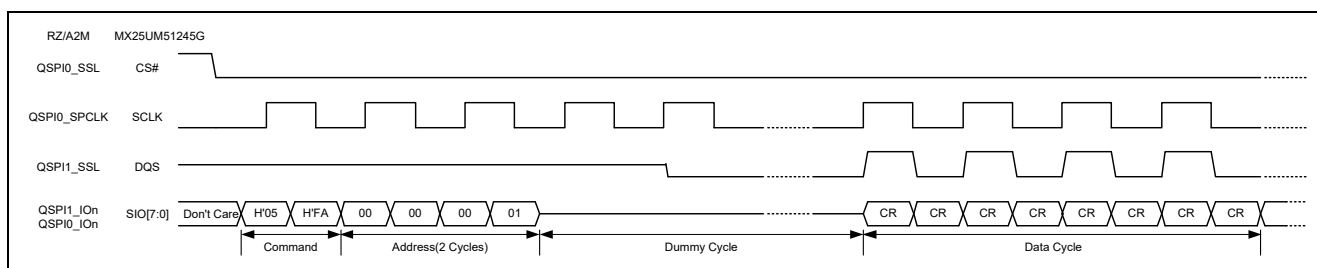


図6.13 DOPI モードの RDCR コマンドの波形フォーマット (参考)

### 6.2.7 OctaFlash のコンフィグレーションレジスタ 2 のリード

ご使用の OctaFlash の仕様に合わせて、OctaFlash のコンフィグレーションレジスタをリードするように、Userdef\_SPIBSC\_OCTAFLASH\_ReadConfig2 関数を実装してください。

図6.14に、サンプルコードのUserdef\_SPIBSC\_OCTAFLASH\_ReadConfig2関数のフローを示します。

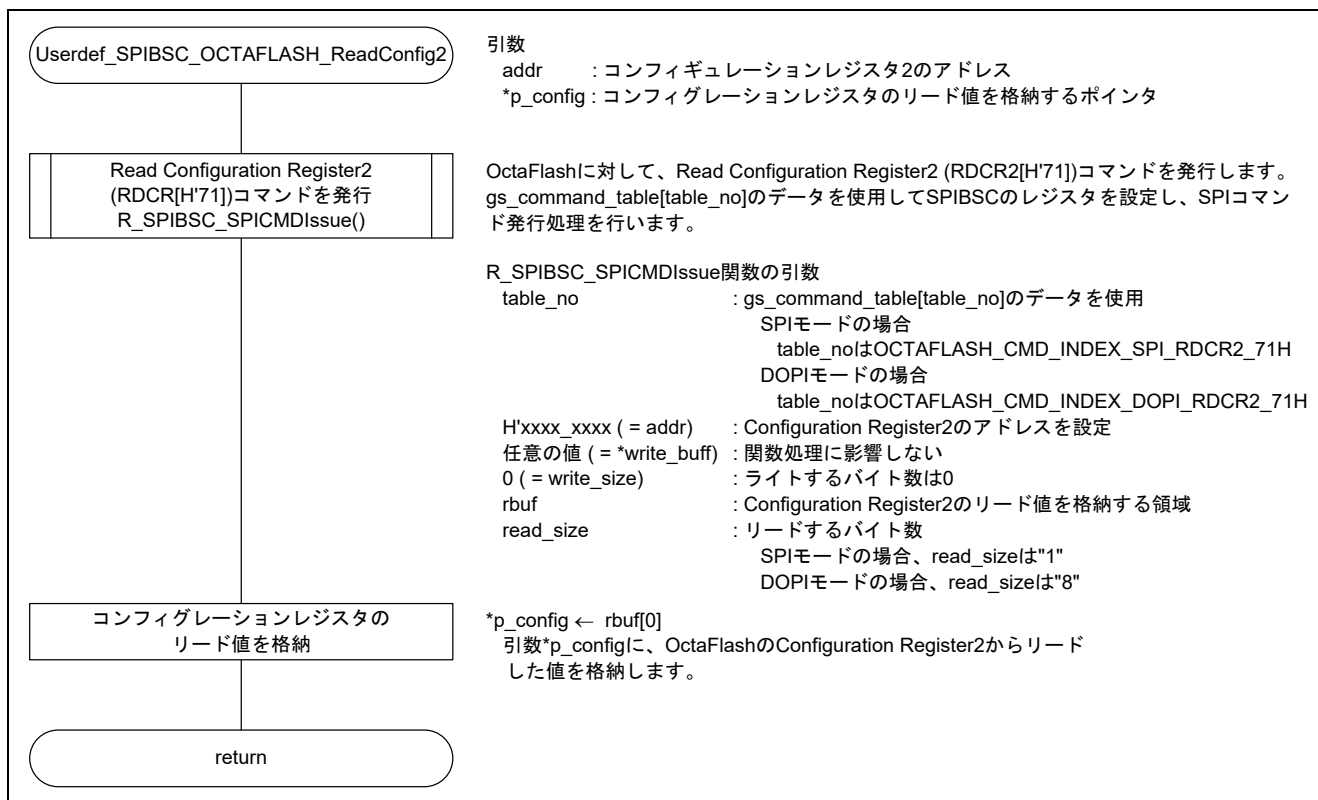


図6.14 Userdef\_SPIBSC\_OCTAFLASH\_ReadConfig2 関数のフロー

表6.12 手動モード用のコマンド設定テーブル gs\_command\_table[2] : SPIモードのRDCR2コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"SPI_RDCR2"
uint8_t cmd	コマンドのコード	0x71
uint8_t cmd_width	コマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x00
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_1BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_32
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_1BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_1BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_SDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

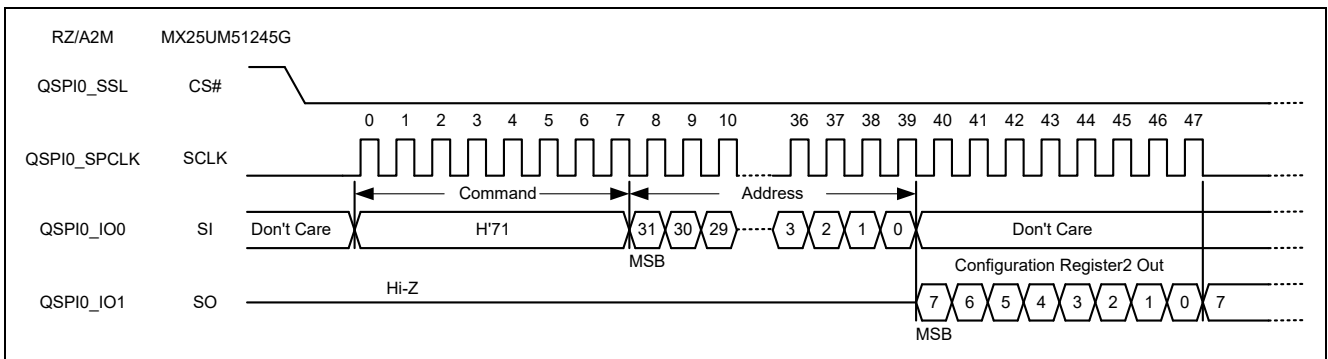


図6.15 SPIモードのRDCR2コマンドの波形フォーマット (参考)

表6.13 手動モード用のコマンド設定テーブル gs\_command\_table[9] : DOPI モードの RDCR2 コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_RDCR2"
uint8_t cmd	コマンドのコード	0x71
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x8E
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t dummy_cycle_count	ダミーサイクル数	SPIBSC_DUMMY_14CYC
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

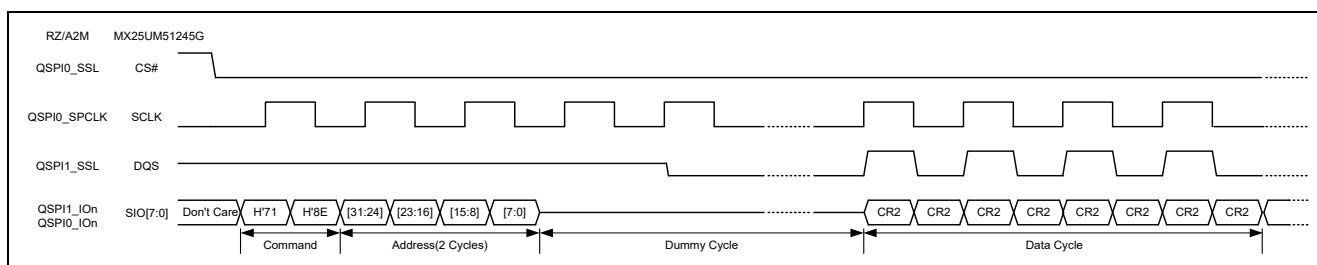


図6.16 DOPI モードの RDCR2 コマンドの波形フォーマット (参考)

### 6.2.8 OctaFlash のライト許可

OctaFlash のレジスタ (Status Register、Configuration Register、および Configuration Register2) にライトするためには、事前に OctaFlash のライトを許可にすることが必要です。サンプルコードでは、この処理を Userdef\_SPIBSC\_OCTAFLASH\_WriteEnable 関数で実現しています。

ご使用の OctaFlash の仕様に合わせて、OctaFlash のライト許可が行えるように、Userdef\_SPIBSC\_OCTAFLASH\_WriteEnable 関数を実装してください。サンプルコードでは、Write Enable コマンド (WREN [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6.17に、サンプルコードのUserdef\_SPIBSC\_OCTAFLASH\_WriteEnable関数のフローを示します。

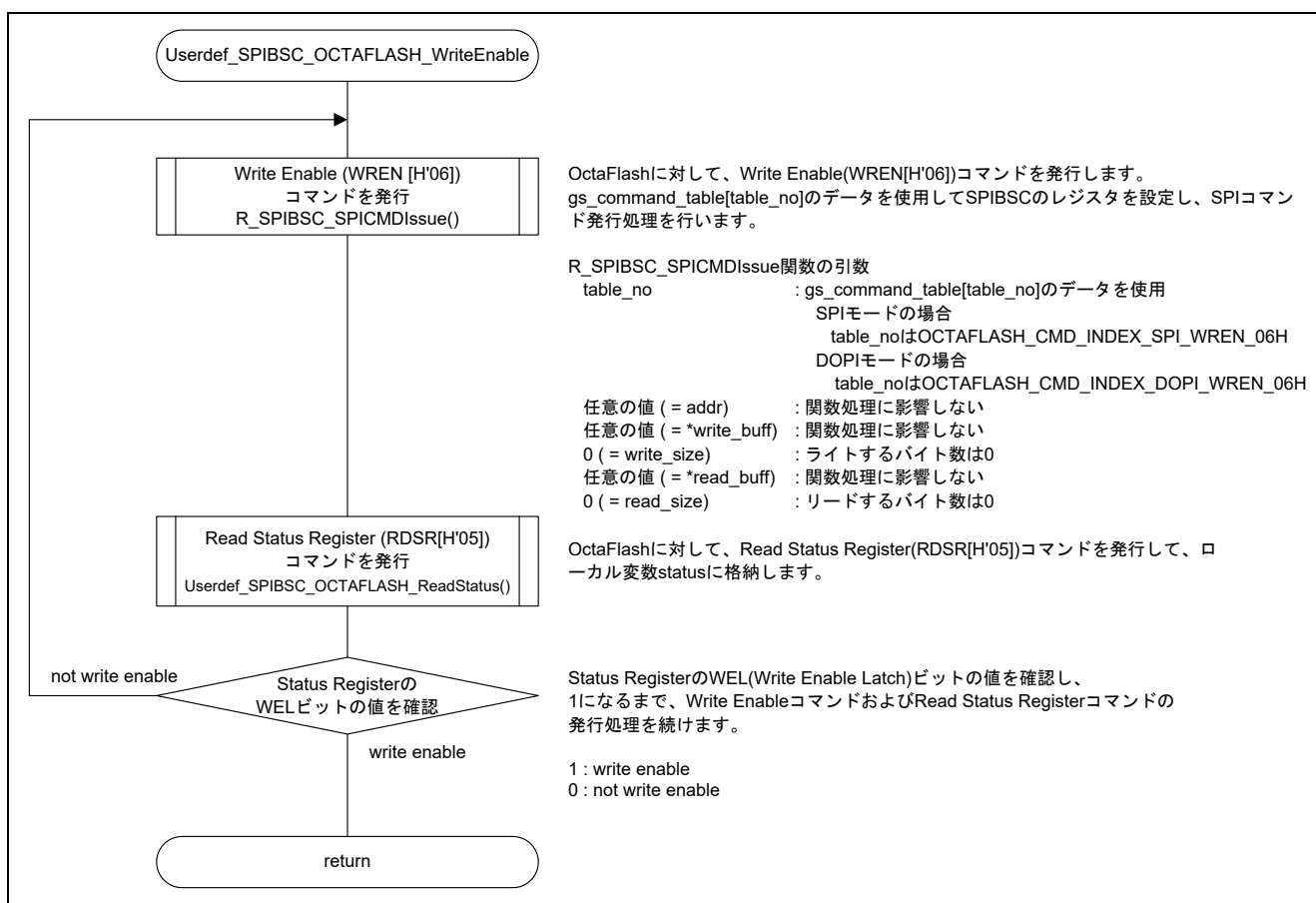


図6.17 Userdef\_SPIBSC\_OCTAFLASH\_WriteEnable 関数のフロー



表6.14 手動モード用のコマンド設定テーブル gs\_command\_table[3] : SPIモードのWRENコマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"SPI_WREN"
uint8_t cmd	コマンドのコード	0x06
uint8_t cmd_width	コマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x00
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_1BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_1BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_1BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_SDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

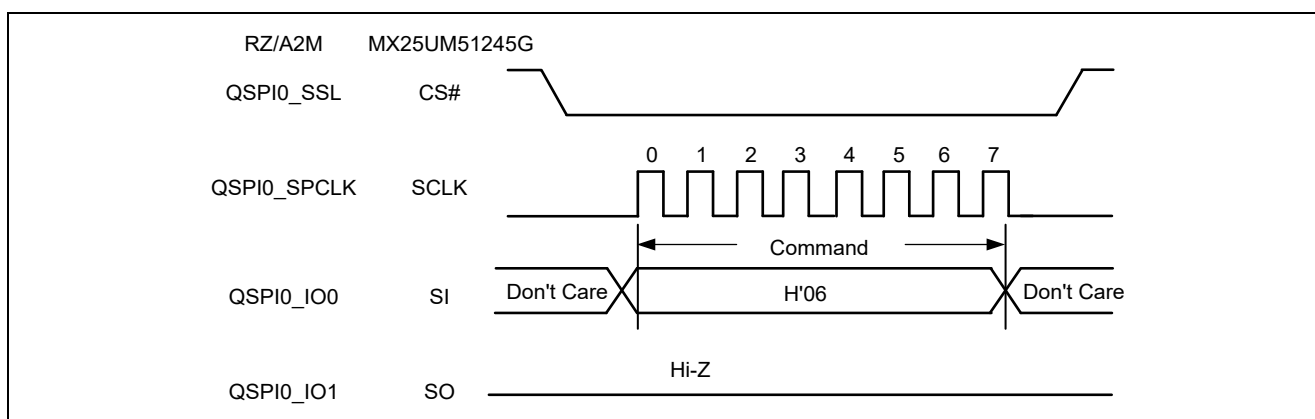


図6.18 SPIモードのWRENコマンドの波形フォーマット (参考)

表6.15 手動モード用のコマンド設定テーブル gs\_command\_table[10] : DOPI モードの WREN コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_WREN"
uint8_t cmd	コマンドのコード	0x06
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0xF9
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

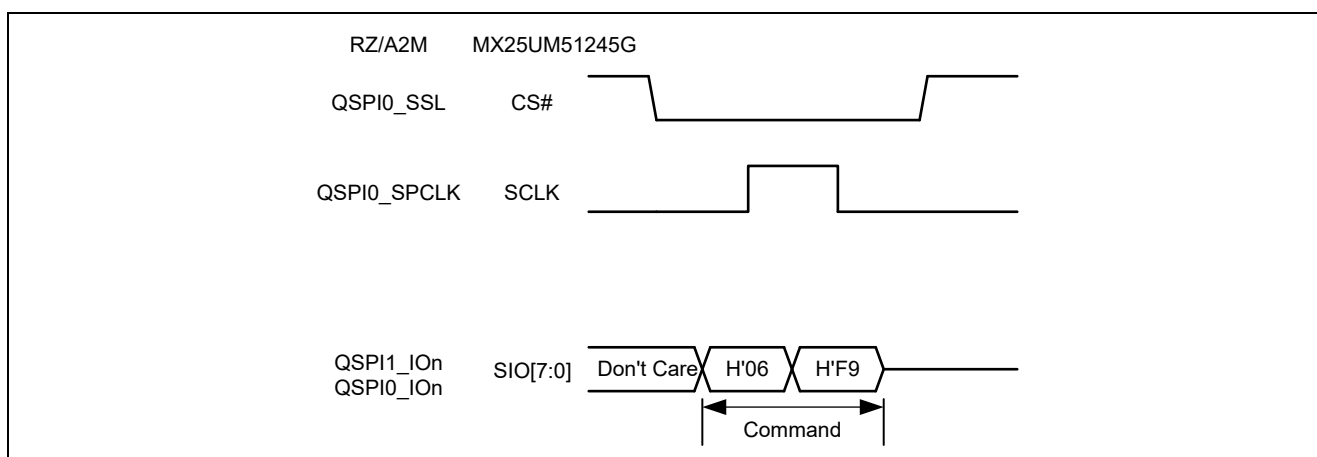


図6.19 DOPI モードの WREN コマンドの波形フォーマット (参考)

### 6.2.9 OctaFlash のステータス/コンフィグレーションレジスタのライト

ご使用の OctaFlash の仕様に合わせて、OctaFlash のステータスレジスタおよびコンフィグレーションレジスタへの設定値をライトするように、Userdef\_SPIBSC\_OCTAFLASH\_WriteStatus 関数を実装してください。

図6.20に、サンプルコードのUserdef\_SPIBSC\_OCTAFLASH\_WriteStatus関数のフローを示します。

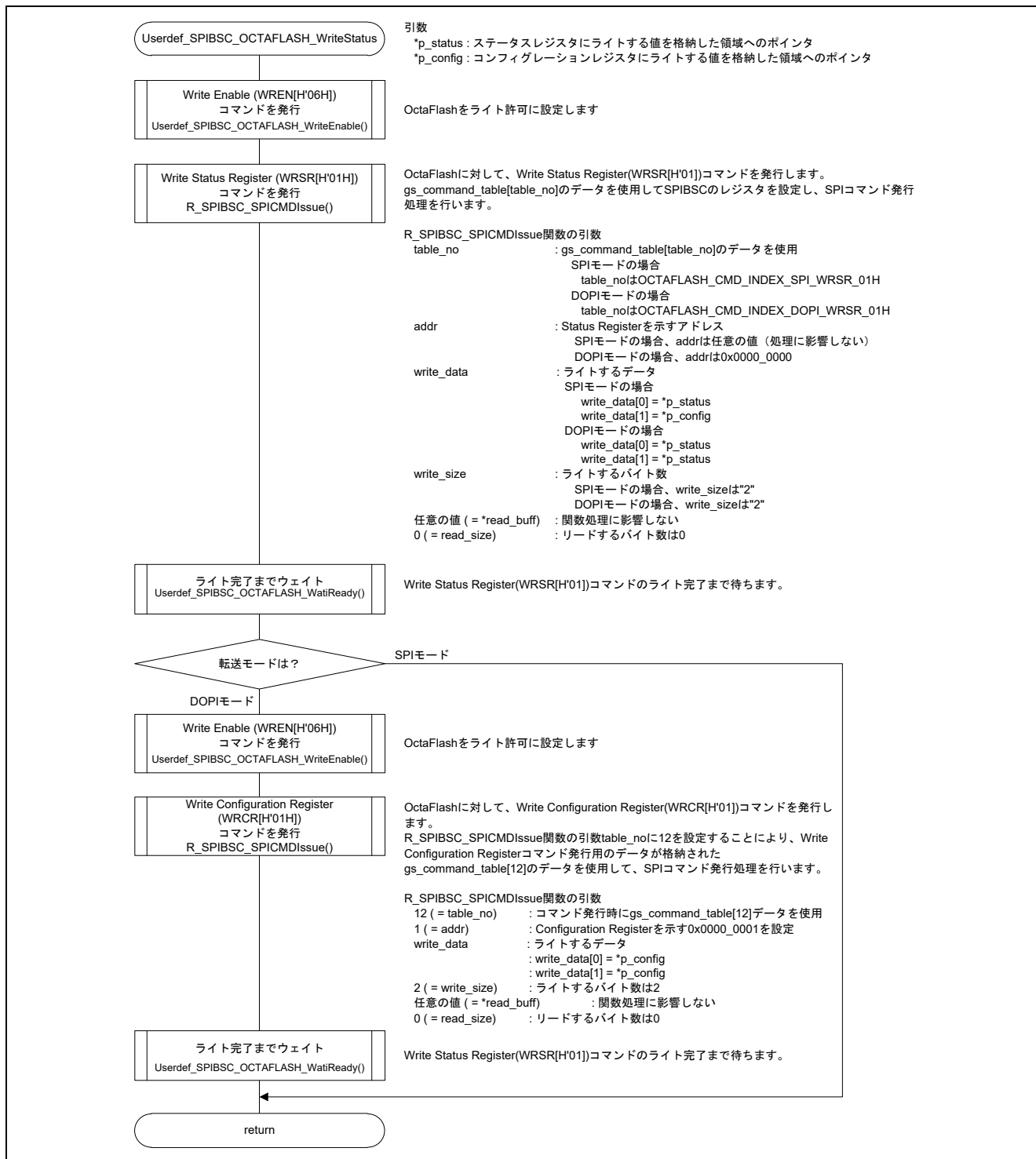


図6.20 Userdef\_SPIBSC\_OCTAFLASH\_WriteStatus 関数のフロー

表6.16 手動モード用のコマンド設定テーブル gs\_command\_table[4] : SPIモードのWRSRコマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"SPI_WRSR"
uint8_t cmd	コマンドのコード	0x01
uint8_t cmd_width	コマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x00
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_1BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_1BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_1BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_SDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

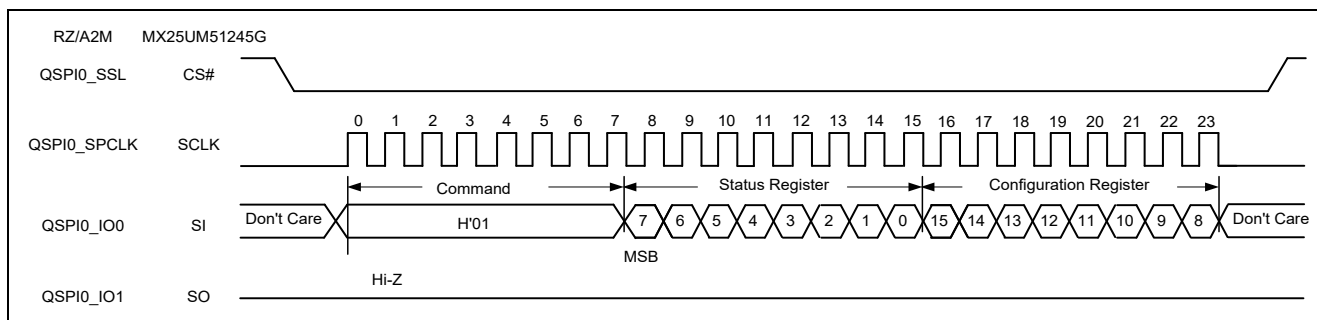


図6.21 SPIモードのWRSRコマンドの波形フォーマット (参考)

表6.17 手動モード用のコマンド設定テーブル gs\_command\_table[11] : DOPI モードの WRSR コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_WRSR"
uint8_t cmd	コマンドのコード	0x01
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0xFE
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1 番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2 番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3 番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4 番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

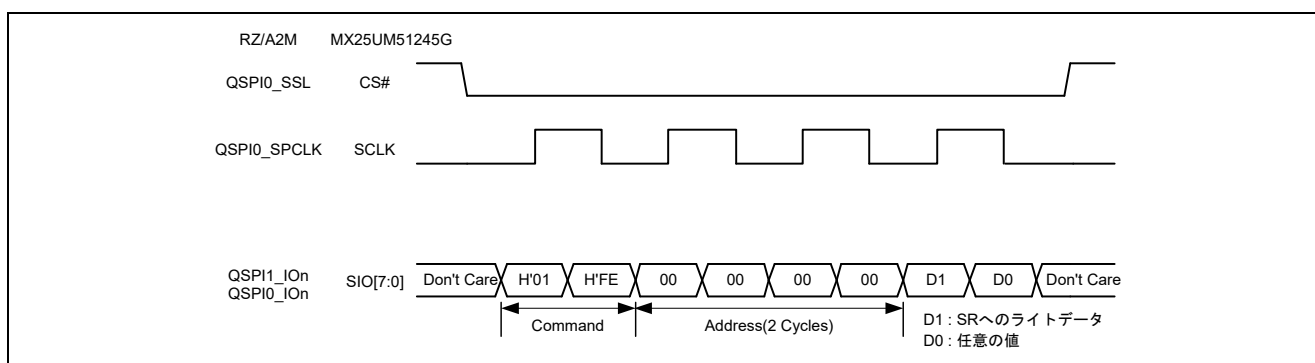


図6.22 DOPI モードの WRSR コマンドの波形フォーマット (参考)

表6.18 手動モード用のコマンド設定テーブル gs\_command\_table[12] : DOPI モードの WRCR コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_WRCR"
uint8_t cmd	コマンドのコード	0x01
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0xFE
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

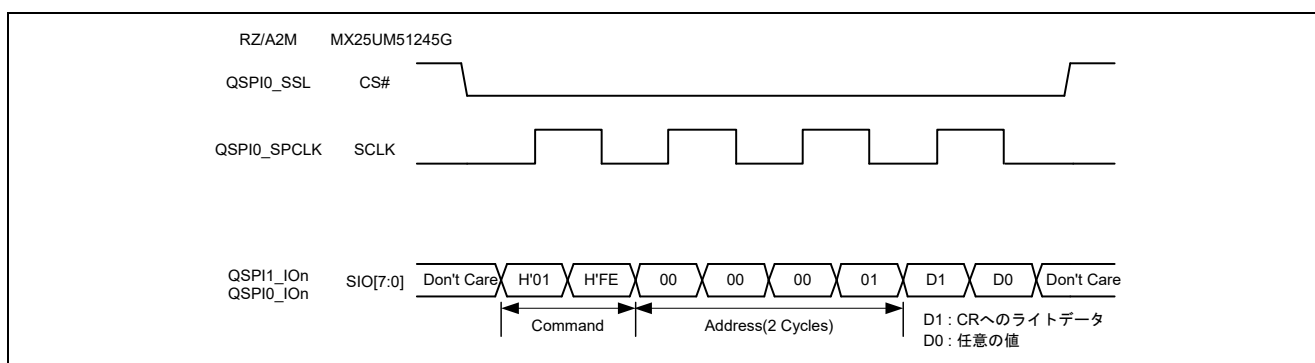


図6.23 DOPI モードの WRCR コマンドの波形フォーマット (参考)

### 6.2.10 OctaFlash のコンフィグレーションレジスタ 2 のライト

ご使用の OctaFlash の仕様に合わせて、OctaFlash のコンフィグレーションレジスタ 2 への設定値をライトするように、Userdef\_SPIBSC\_OCTAFLASH\_WriteConfig2 関数を実装してください。

図6.24に、サンプルコードのUserdef\_SPIBSC\_OCTAFLASH\_WriteConfig2関数のフローを示します。

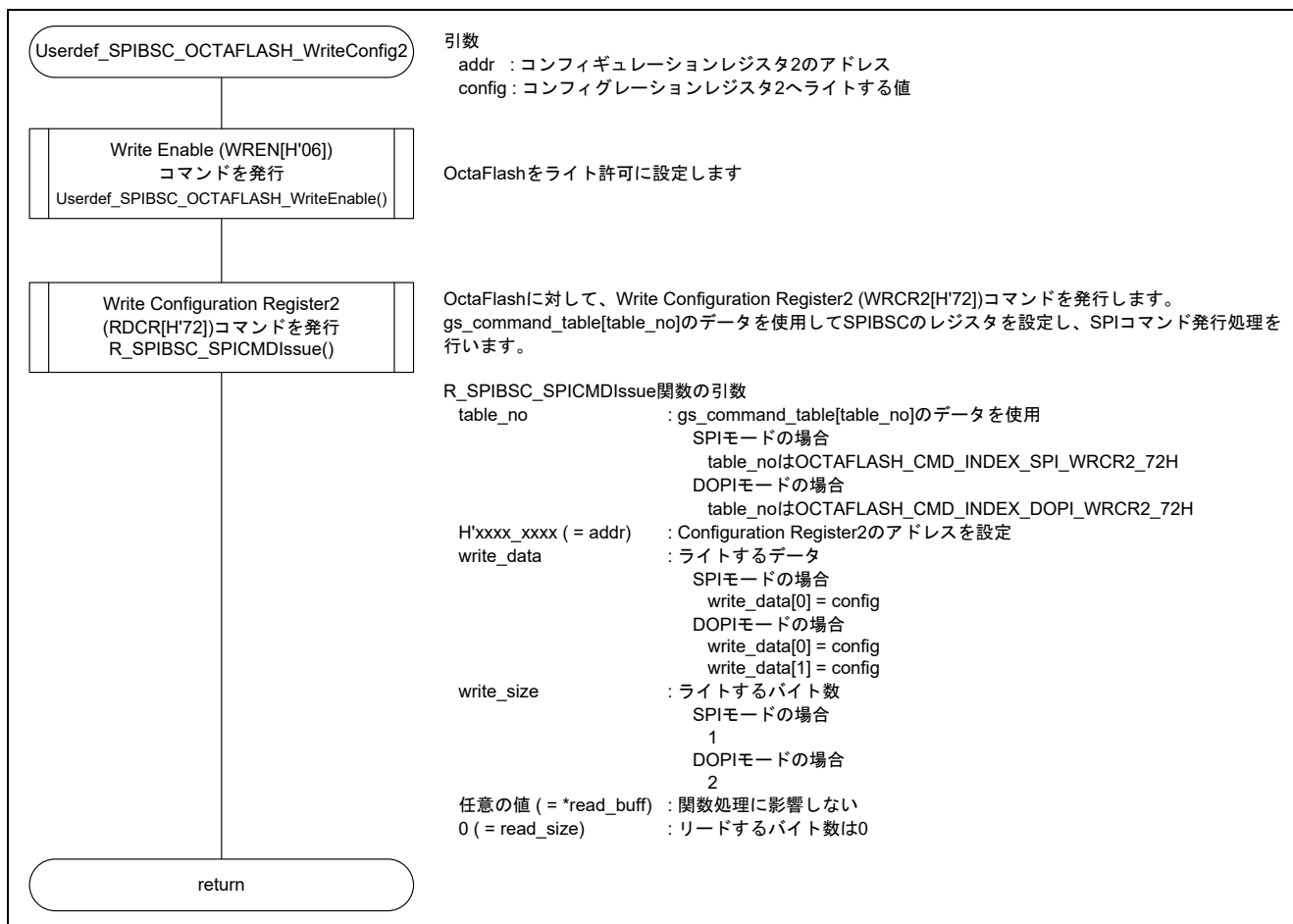


図6.24 Userdef\_SPIBSC\_OCTAFLASH\_WriteConfig2 関数のフロー

表6.19 手動モード用のコマンド設定テーブル gs\_command\_table[5] : SPIモードのWRCR2コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"SPI_WRCR2"
uint8_t cmd	コマンドのコード	0x72
uint8_t cmd_width	コマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x00
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_1BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_32
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_1BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_1BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_SDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

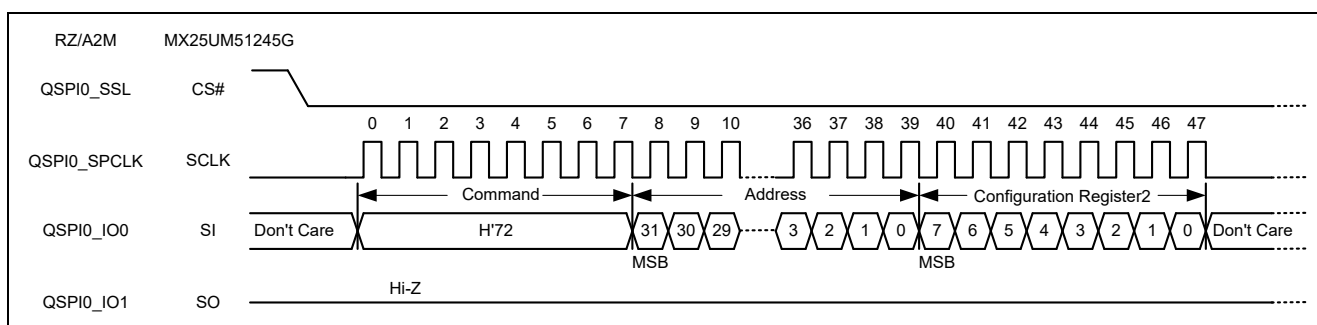


図6.25 SPIモードのWRCR2コマンドの波形フォーマット (参考)



表6.20 手動モード用のコマンド設定テーブル gs\_command\_table[13] : DOPI モードの WRCR2 コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_WRCR2"
uint8_t cmd	コマンドのコード	0x72
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x8D
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

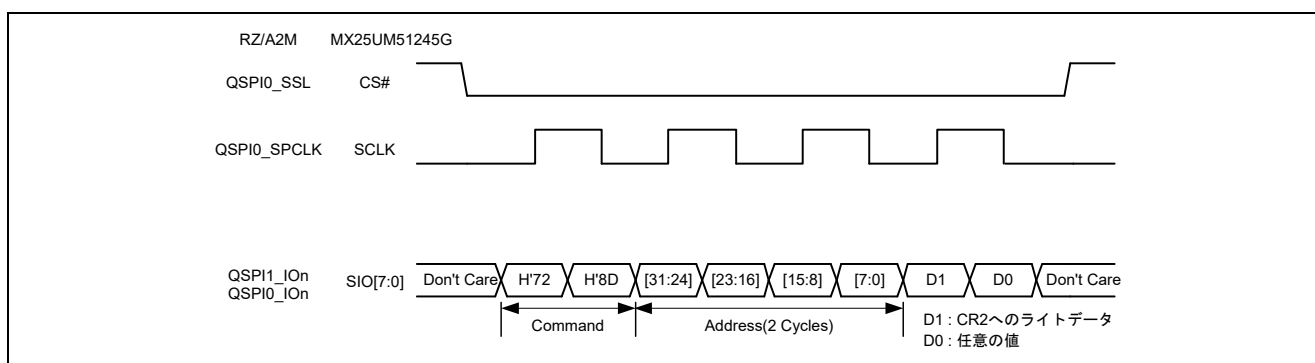


図6.26 DOPI モードの WRCR2 コマンドの波形フォーマット (参考)

### 6.2.11 OctaFlash の ID 情報のリード

サンプルコードでは、OctaFlash の ID 情報のリード処理を Userdef\_SPIBSC\_OCTAFLASH\_ReadId 関数で実現しています。

ご使用の OctaFlash の仕様に合わせて、OctaFlash の ID 情報をリードするように、Userdef\_SPIBSC\_OCTAFLASH\_ReadId 関数を実装してください。

図6.27に、サンプルコードの Userdef\_SPIBSC\_OCTAFLASH\_ReadId 関数のフローを示します。

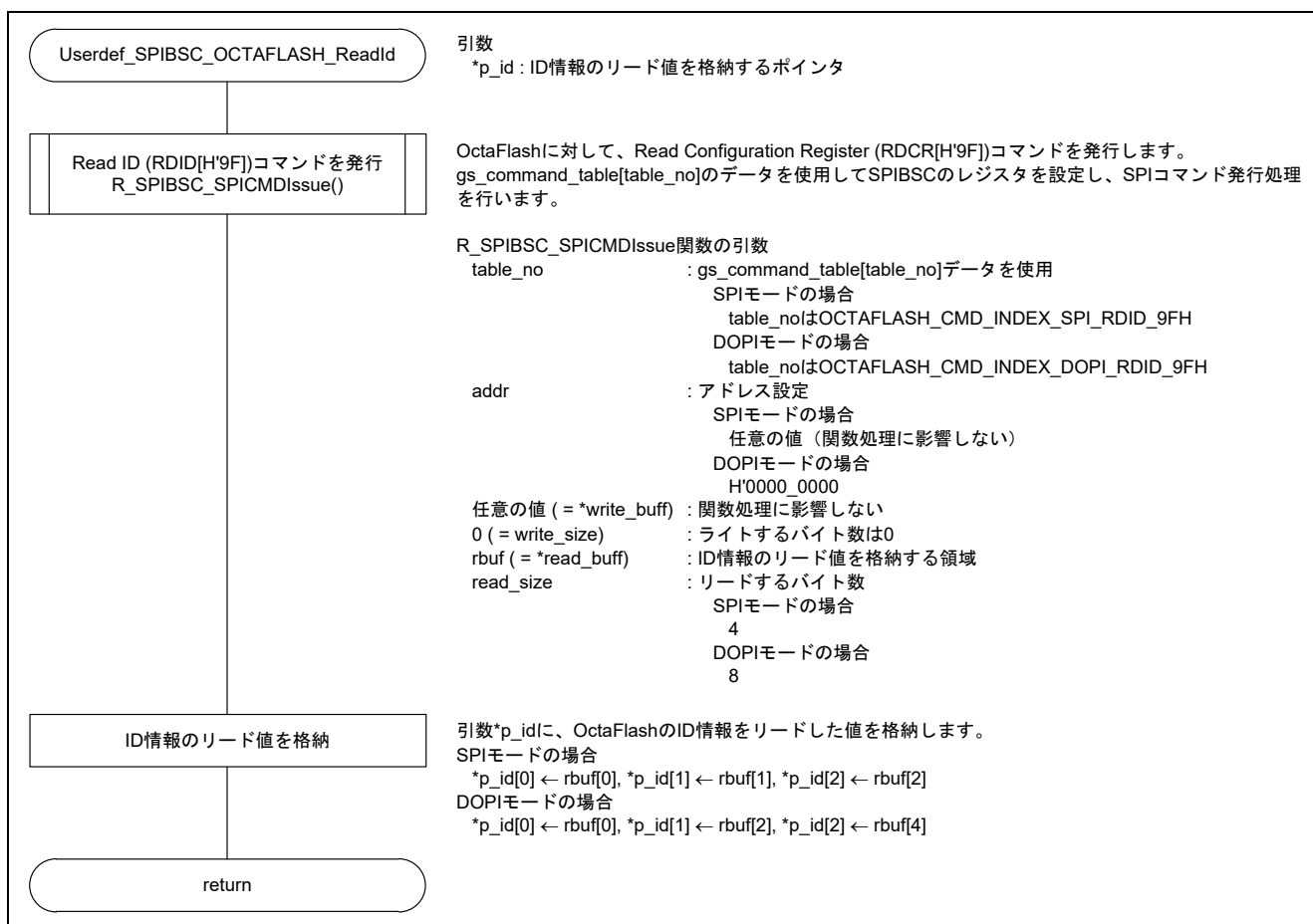


図6.27 Userdef\_SPIBSC\_OCTAFLASH\_ReadId 関数のフロー

表6.21 手動モード用のコマンド設定テーブル gs\_command\_table[6] : SPIモードのRDIDコマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"SPI_RDID"
uint8_t cmd	コマンドのコード	0x9F
uint8_t cmd_width	コマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x00
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_1BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_1BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_32
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_1BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_SDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_1BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_SDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

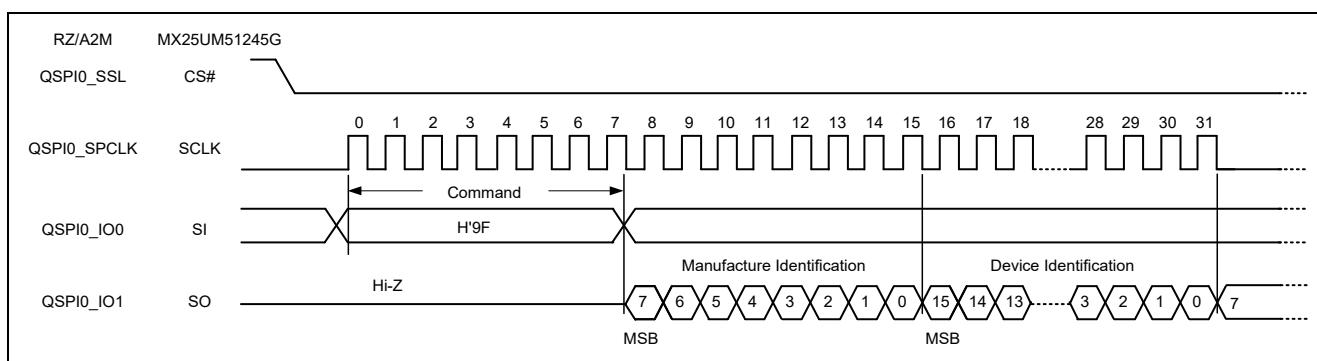


図6.28 SPIモードのRDIDコマンドの波形フォーマット (参考)

表6.22 手動モード用のコマンド設定テーブル gs\_command\_table[14] : DOPI モードの RDID コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_RDID"
uint8_t cmd	コマンドのコード	0x9F
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x60
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t dummy_cycle_count	ダミーサイクル数	SPIBSC_DUMMY_14CYC
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

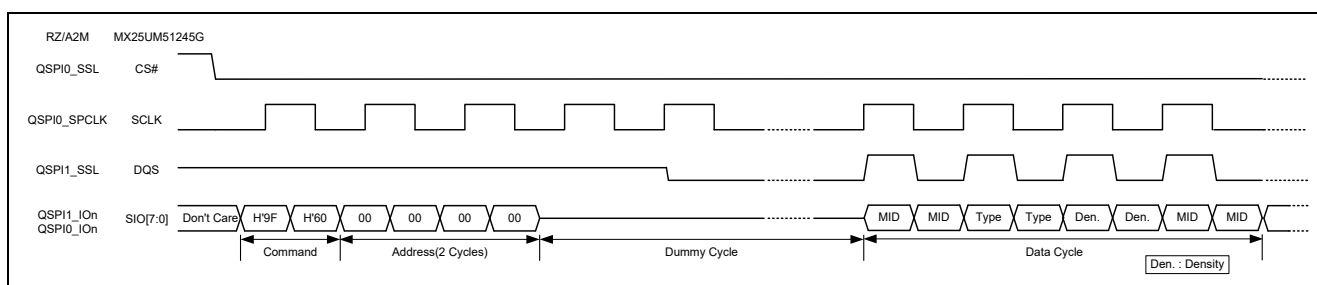


図6.29 DOPI モードの RDID コマンドの波形フォーマット (参考)

### 6.2.12 OctaFlash のデータリード

サンプルコードでは、OctaFlash のデータリード処理を Userdef\_SPIBSC\_OCTAFLASH\_Read 関数で実現しています。

ご使用の OctaFlash の仕様に合わせて、OctaFlash のデータをリードするように、Userdef\_SPIBSC\_OCTAFLASH\_Read 関数を実装してください。

図6.30に、サンプルコードのUserdef\_SPIBSC\_OCTAFLASH\_Read関数のフローを示します。

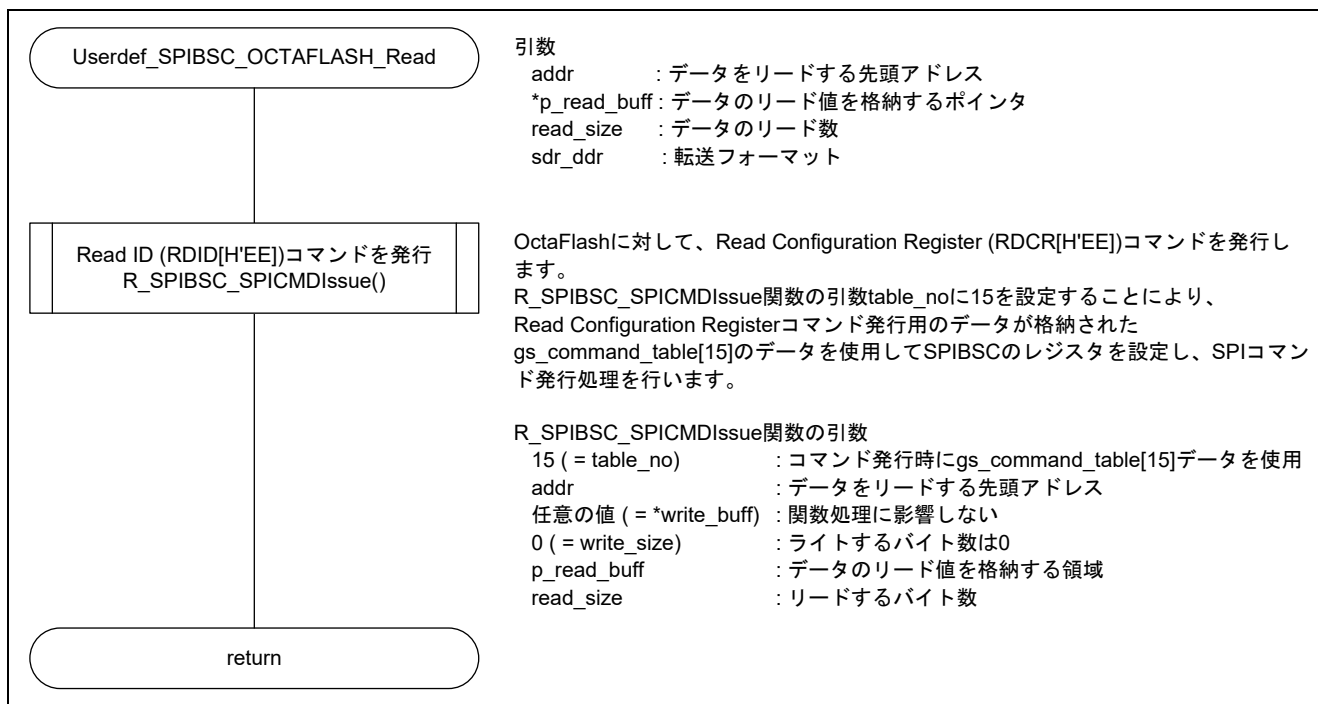


図6.30 Userdef\_SPIBSC\_OCTAFLASH\_Read 関数のフロー

表6.23 手動モード用のコマンド設定テーブル gs\_command\_table[15] : DOPI モードの 8DTRD コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_8DTRD"
uint8_t cmd	コマンドのコード	0xEE
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0x11
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t dummy_cycle_count	ダミーサイクル数	SPIBSC_DUMMY_14CYC
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

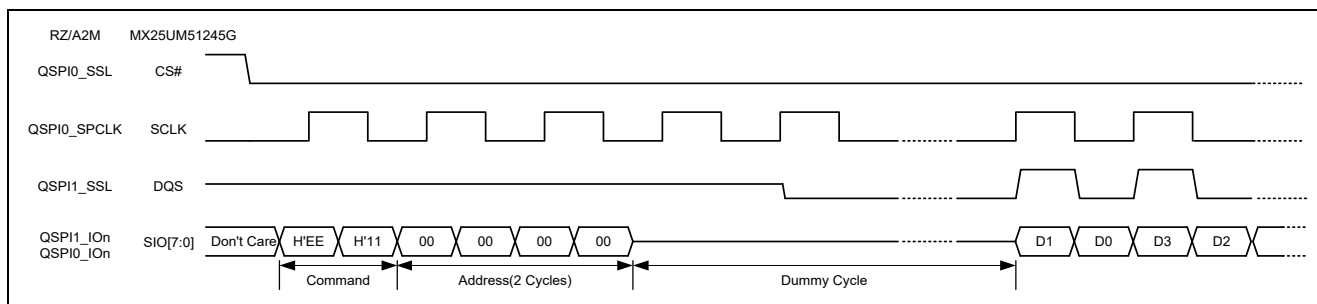


図6.31 DOPI モードの 8DTRD コマンドの波形フォーマット (参考)

### 6.2.13 OctaFlash のセクタイレーズ

サンプルコードでは、OctaFlash のセクタイレーズ処理を Userdef\_SPIBSC\_OCTAFLASH\_Erase 関数で実現しています。

ご使用の OctaFlash の仕様に合わせて、OctaFlash のセクタをイレーズするように、Userdef\_SPIBSC\_OCTAFLASH\_Erase 関数を実装してください。

図6.32に、サンプルコードのUserdef\_SPIBSC\_OCTAFLASH\_Erase関数のフローを示します。

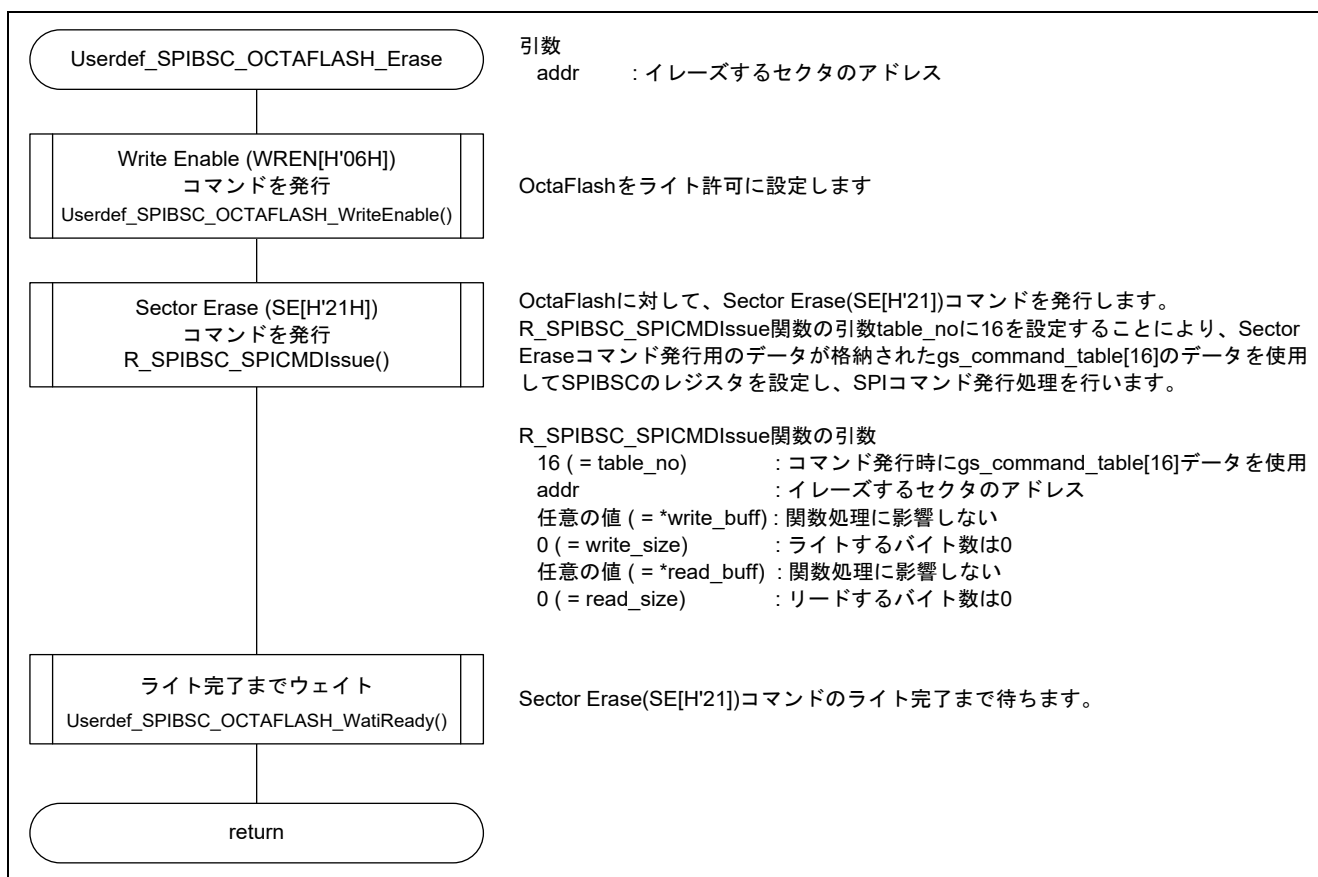


図6.32 Userdef\_SPIBSC\_OCTAFLASH\_Erase 関数のフロー

表6.24 手動モード用のコマンド設定テーブル gs\_command\_table[16] : DOPI モードの SE コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_SE"
uint8_t cmd	コマンドのコード	0x21
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0xDE
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1 番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2 番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3 番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4 番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

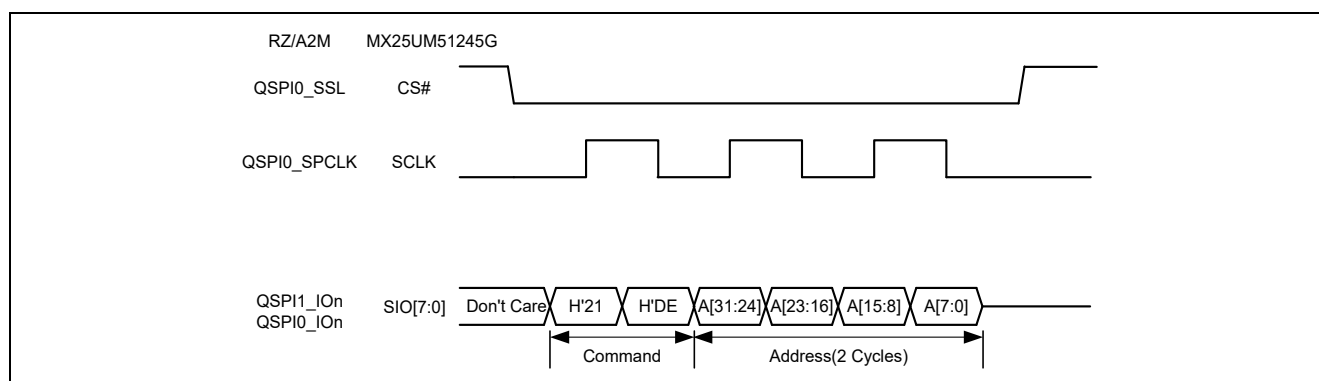


図6.33 DOPI モードの SE コマンドの波形フォーマット (参考)



### 6.2.14 OctaFlash のデータライト

サンプルコードでは、OctaFlash のプログラム処理を Userdef\_SPIBSC\_OCTAFLASH\_Write 関数で実現しています。

ご使用の OctaFlash の仕様に合わせて、OctaFlash のプログラムするように、Userdef\_SPIBSC\_OCTAFLASH\_Write 関数を実装してください。

図6.34に、サンプルコードのUserdef\_SPIBSC\_OCTAFLASH\_Write関数のフローを示します。

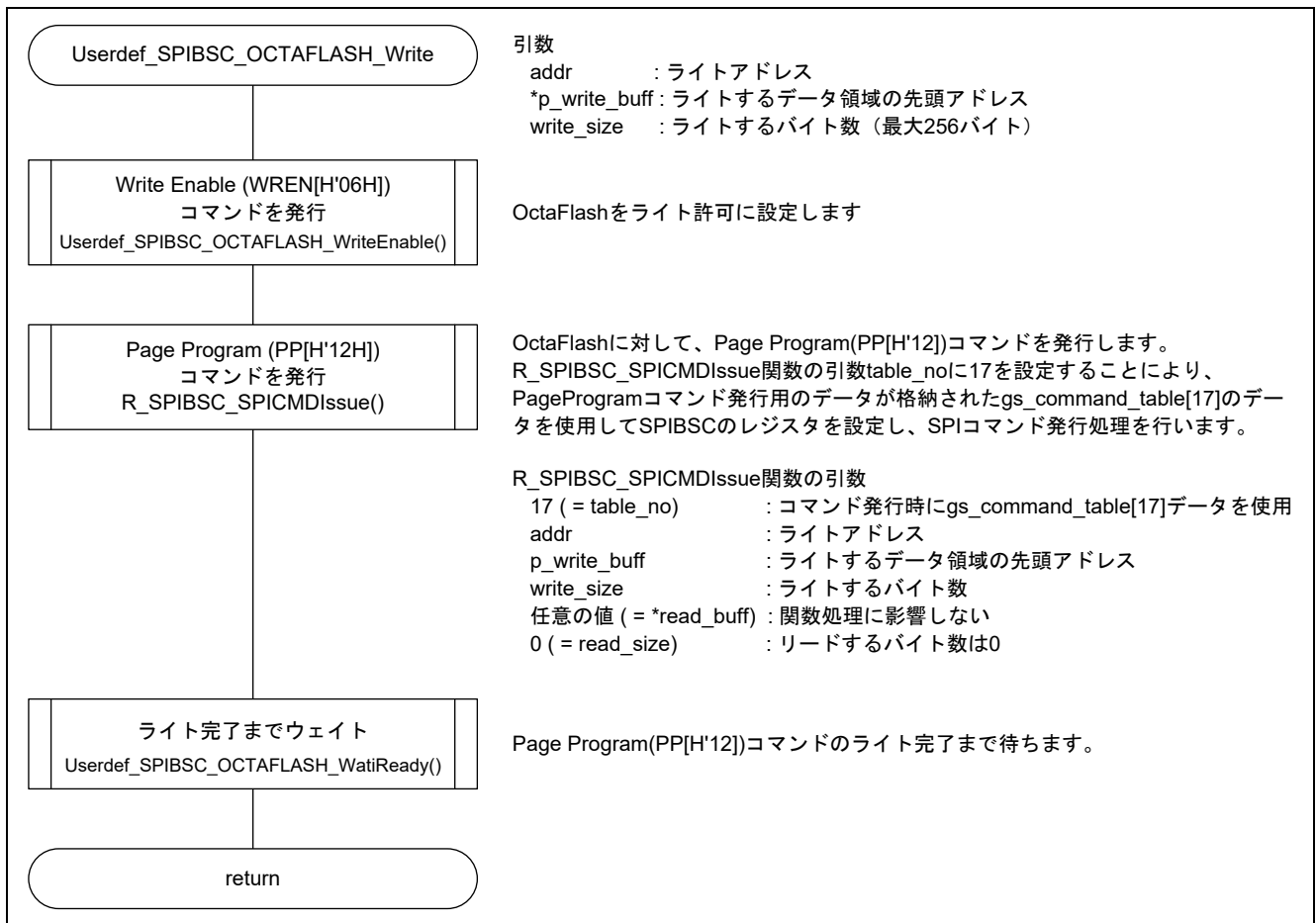


図6.34 Userdef\_SPIBSC\_OCTAFLASH\_Write 関数のフロー

表6.25 手動モード用のコマンド設定テーブル gs\_command\_table[17] : DOPI モードの PP コマンド

メンバ名	説明	設定値
uint8_t command_name[20]	コマンド識別の文字列	"DOPI_PP"
uint8_t cmd	コマンドのコード	0x12
uint8_t cmd_width	コマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t cmd_output_enable	コマンドの出力有無	SPIBSC_OUTPUT_ENABLE
uint8_t ocmd	オプションコマンドのコード	0xED
uint8_t ocmd_width	オプションコマンドのビット幅	SPIBSC_8BIT_WIDTH
uint8_t ocmd_enable	オプションコマンドの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t addr_width	アドレスのビット幅	SPIBSC_8BIT_WIDTH
uint8_t addr_output_enable	アドレスの出力設定	SPIBSC_OUTPUT_ADDR_OCTA
uint8_t addr_sdr_ddr	アドレスの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opdata_width	オプションデータのビット幅	SPIBSC_8BIT_WIDTH
uint8_t opdata_output_enable	オプションデータの出力設定	SPIBSC_OUTPUT_DISABLE
uint8_t opdata_ddr_enable	オプションデータの転送方式	SPIBSC_DDR_TRANSFER
uint8_t opd3	Option Data3(8bit)の設定(1番目に出力)	0x00
uint8_t opd2	Option Data2(8bit)の設定(2番目に出力)	0x00
uint8_t opd1	Option Data1(8bit)の設定(3番目に出力)	0x00
uint8_t opd0	Option Data0(8bit)の設定(4番目に出力)	0x00
uint8_t dummy_cycle_output_enable	ダミーサイクルの出力有無	SPIBSC_OUTPUT_DISABLE
uint8_t dummy_cycle_count	ダミーサイクル数	0x00
uint8_t transfer_data_width	転送データのビット幅	SPIBSC_8BIT_WIDTH
uint8_t transfer_data_sdr_ddr	転送データの転送方式	SPIBSC_DDR_TRANSFER

【注】 CMNCR.MOIIOn(n=0~3)には B'01 (出力値は 1)、CMNCR.IOnFV(n=0,2,3)には B'11 (出力値は Hi-Z) を設定しています。

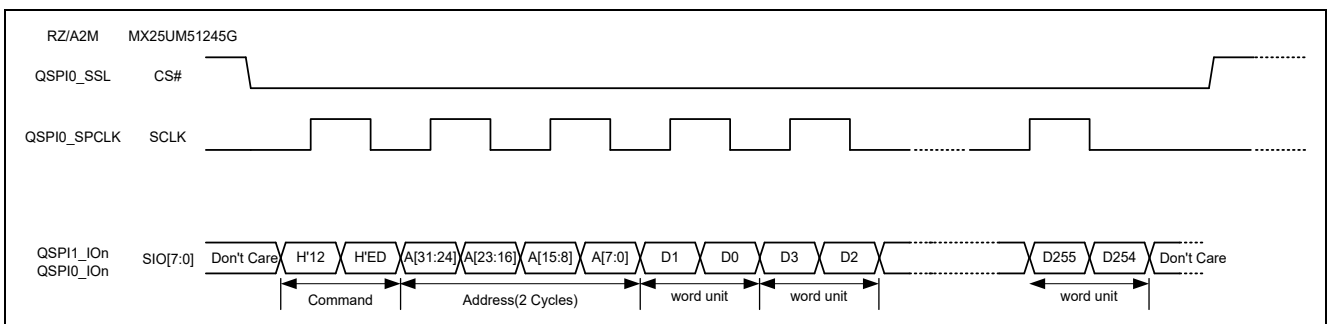


図6.35 DOPI モードの PP コマンドの波形フォーマット (参考)

## 7. サンプルコードの注意事項

### 7.1 外部アドレス空間リードモードでアクセス可能な領域

外部アドレス空間リードモードでアクセス可能な領域は、SPI マルチ I/O バス空間に割り当てられた 256MB の領域 (H'2000\_0000 番地~H'2FFF\_FFFF 番地) です。SPIBSC はこの領域へのアクセスを OctaFlash の H'0000\_0000 番地~H'0FFF\_FFFF 番地に変換してアクセスを行います。サンプルコードでは、OctaFlash の H'0000\_0000 番地~H'0FFF\_FFFF 番地の 256MB の領域に対してアクセスが可能です。RZ/A2M では SPIBSC のデータリード拡張アドレス設定レジスタ (DREAR) を制御して、SPI マルチ I/O バス空間に割り付ける OctaFlash のアドレスを変更することにより、256MB を超える領域についてもアクセスが可能です。DREAR を制御する前の領域にはアクセスができなくなるため、サンプルコードでは 256MB を超える領域へのアクセスには対応せず、OctaFlash の H'0000\_0000 番地~H'0FFF\_FFFF 番地のみアクセス可能な仕様としています。

なお、手動モードでは、4 バイトアドレスを使用したアクセスに対応しており、OctaFlash の 4G バイトの領域にアクセスが可能です。

### 7.2 OctaFlash に発行するコマンド

OctaFlash に発行可能なコマンドは、ローダプログラムでは SPI モードのコマンドのみ、アプリケーションプログラムでは DOPI モードのコマンドのみ発行可能です。アプリケーションプログラムに分岐後は、SPI モードのコマンドを発行することはできません。

## 8. 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。

## 9. 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/A2M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。)

RTK7921053C00000BE (RZ/A2M CPUボード) 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。)

RTK79210XXB00000BE (RZ/A2M SUBボード) 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 Cortex™-A9 Technical Reference Manual Revision: r4p1

(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 version2.0

(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 Technical Reference Manual Revision: r3p3

(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：統合開発

統合開発環境 e<sup>2</sup> studio のユーザーズマニュアルは、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。

(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。)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
Rev.1.00	Jun.22.20	—	初版発行

## 製品ご使用上の注意事項

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

### 1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

### 2. 電源投入時の処置

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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

### 4. 未使用端子の処理

未使用端子は、「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。

### 5. クロックについて

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。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

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、将来の拡張機能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。

### 8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違っていると、フラッシュメモリ、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、ノイズ幅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生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
  2. 当社製品、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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
  3. 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、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等  
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
  6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。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. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
  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。
  12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 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直接的、間接的に支配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。

(Rev.4.0-1 2017.11)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

[www.renesas.com](http://www.renesas.com)

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。