SH7268/SH7269 グループ

SPI マルチ I/O バスコントローラ シリアルフラッシュメモリ接続例

R01AN0671JJ0101
Rev. 1.01
2012.02.16

要旨

SH7268/SH7269 の SPI マルチ I/O バスコントローラ（SPIBSC）は、シリアルフラッシュメモリを任意にリードライトする機能（SPI 動作モード）に加えて、プログラムデータを直接フェッチして実行する機能（外部アドレス空間リードモード）を備えています。本アプリケーションノートは、SPIBSC の使用例とシリアルフラッシュメモリの接続例について説明します。

動作確認デバイス

SH7268/SH7269

以下、総称して「SH7269」として説明します。

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. はじめに 2
2. 応用例の説明 3
3. 参考プログラムリスト 22
4. 参考ドキュメント 52

1. はじめに

1.1 仕様

本アプリケーションノートは、SPI マルチ I/O バスコントローラ（SPIBSC）の応用例についての説明と参考プログラムで構成しています。応用例の説明では、シリアルフラッシュメモリの接続例と、2 つの制御方法（SPI 動作モード、外部アドレス空間リードモード）について説明します。

1.2 使用機能

- SPI マルチ I/O バスコントローラ（SPIBSC）
- ルネサスシリアルペリフェラルインタフェース（RSPI）
- ブートモード（シリアルフラッシュメモリブート）
- 汎用入出力ポート

1.3 適用条件

マイコン	SH7268/SH7269
動作周波数	CPU 内部クロック（Iφ）：266.67 MHz
	内部バスクロック（Bφ）：133.33 MHz
	周辺クロック 1（P1φ）：66.67 MHz
	周辺クロック 0（P0φ）：33.33 MHz
統合開発環境	ルネサスエレクトロニクス製
	High-performance Embedded Workshop Ver.4.07.00
C コンパイラ	ルネサスエレクトロニクス製 SuperH RISC engine ファミリ
	C/C++コンパイラパッケージ Ver.9.03 Release02
コンパイルオプション	High-performance Embedded Workshop でのデフォルト設定 (-cpu=sh2afpu -fpu=single -object="$(CONFIGDIR)¥$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 –nologo)
シリアルフラッシュメモリ	Spansion 製 S25FL032P

1.4 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。合わせて参照してください。

- SH7268/SH7269 グループ　SPI マルチ I/O バスコントローラを使用したシリアルフラッシュメモリからのブート例

1.5 "L"アクティブ端子（信号）の表記について

端子名（信号名）末尾の＃は"L"アクティブ端子（信号）であることを示します。

2. 応用例の説明

本応用例は、SPI マルチ I/O バスコントローラ（SPIBSC）にシリアルフラッシュメモリを 1 個接続して、任意のリードライトアクセスおよびプログラムフェッチを行います。前者のアクセスには SPI 動作モードを使用し、後者のアクセスには外部アドレス空間リードモードを使用します。

ここではシリアルフラッシュメモリの端子接続例について説明した後、それぞれのモードの動作概要と制御方法について説明します。

2.1 SPIBSC の特長

以下に SPIBSC の特長を示します。

- シリアルフラッシュメモリを 2 個まで接続可能
- 1 つのシリアルフラッシュメモリに対し、データバス幅を 1 ビット、2 ビット、4 ビットから選択可能
- SPI マルチ I/O バス空間に配置したシリアルフラッシュメモリを直接フェッチすることが可能（外部アドレス空間リードモード時）
- リードキャッシュを使用したバーストリードが可能（外部アドレス空間リードモード時）
- シリアルフラッシュメモリに対し、任意のリードライト動作が可能（SPI 動作モード時）

2.2 シリアルフラッシュメモリの端子接続例

表 1に本応用例で使用するSPIマルチI/Oバス対応のシリアルフラッシュメモリ（Spansion製 S25FL032P）の仕様を示します。

表1　本応用例で使用するシリアルフラッシュメモリの仕様

項目	仕様
バス入出力	シリアル入出力（全二重）、デュアル入出力（半二重）、クワッド入出力（半二重）
SPI モード	SPI モード 0 およびモード 3 に対応可能
クロック周波数	シリアル入出力時：104MHz(max)、デュアル／クワッド入出力時：80MHz(max)
容量	4M バイト
セクタサイズ	64K バイト
ページサイズ	256 バイト
イレースサイズ	全領域／64K バイト／8K バイト／4K バイト
プログラムサイズ	Page Program（1～256 バイト）
プロテクトモード	ライトイネーブルコマンド（コマンド単位） ソフトウェア／ハードウェアプロテクトモード（ブロック単位）

図 1にシリアルフラッシュメモリの接続回路例を示します。本応用例ではシリアルフラッシュメモリを 1 個接続し、データバス幅 4 ビットでアクセスします。SH7269 の端子機能については、表 2のマルチプレクス出力端子に従い設定してください。

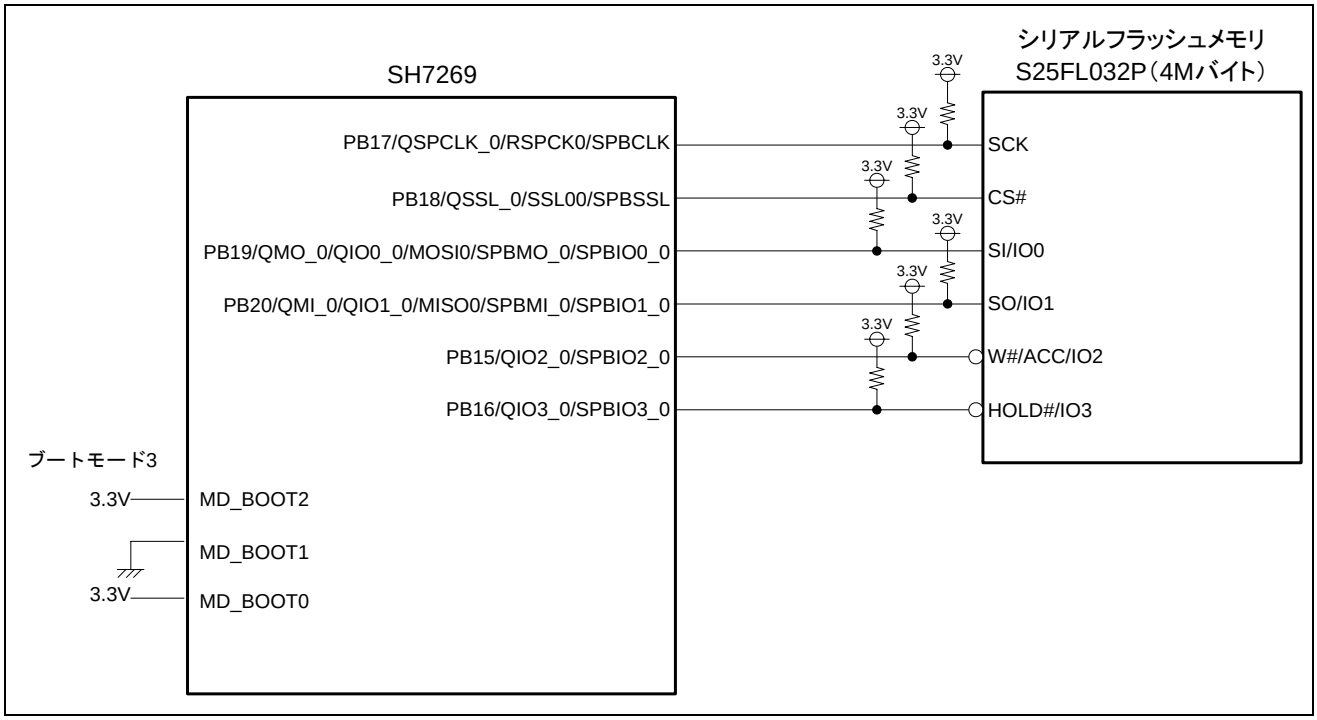


図1　シリアルフラッシュメモリの接続回路例

【注】　制御信号端子の外付け抵抗によるプルアップ/プルダウン処理について
制御信号に対するプルアップ/プルダウン処理は、マイコンの端子状態がハイインピーダンスの場合でも、外部デバイスが誤動作しないように信号線のレベルを決定します。外付け抵抗でプルアップ処理を行ってください。

表2　マルチプレクス出力

周辺機能	使用端子名	SH7269 ポートコントロールレジスタ		SH7269 マルチプレクス端子名
		レジスタ名	MD ビット設定値	
SPIBSC	SPBCLK	PBCR4	PB17MD[2:0]=B'110	PB17 / A17 / QSPCLK / RSPCK0 / SPBCLK
	SPBSSL	PBCR4	PB18MD[2:0]=B'110	PB18 / A18 / QSSL_0 / SSL00 / SPBSSL
	SPBIO0_0	PBCR4	PB19MD[2:0]=B'110	PB19 / A19 / QMO_0 / QIO0_0 / MOSI0 / SPBMO_0 / SPBIO0_0
	SPBIO1_0	PBCR5	PB20MD[2:0]=B'110	PB20 / A20 / QMI_0 / QIO1_0 / MISO0 / SPBMI_0 / SPBIO1_0
	SPBIO2_0	PBCR3	PB15MD[2:0]=B'110	PB15 / A15 / QIO2_0 / SPBIO2_0
	SPBIO3_0	PBCR4	PB16MD[2:0]=B'110	PB16 / A16 / QIO3_0 / SPBIO3_0

【注】　SH7269 のマルチプレクス端子について
SPIBSC で使用する各端子はマルチプレクス端子であり、初期状態が汎用入出力ポートの端子もあります。そのためシリアルフラッシュメモリにアクセスする前に、汎用入出力ポートのコントロールレジスタによって SPIBSC 端子機能に設定する必要があります。また、ブートモード 0 および 1（CS0 空間に接続したメモリからのブート）使用時は、これらの端子を SPIBSC 機能に設定できません。ブートモード 3（シリアルフラッシュブート）を使用してください。

2.3 インタフェースタイミング例

図 2および表 3、表 4に本応用例におけるインタフェースタイミング図とタイミング条件を示します。SPIBSCの設定値はこれらのタイミング条件を満たす必要があります。

表 5には本応用例におけるSPIBSCのインタフェース設定値を示します。本応用例で使用するシリアルフラッシュメモリはSPIモード 0（クロックネゲートレベル'L'、立ち上がり受信、立ち下がり送信）で動作する仕様のためSPIBSCもSPIモード 0 に準じた設定にしています。ただし、SH7269 のデータ入力セットアップ時間を満たすためにデータ受信タイミングだけは 1/2 サイクル遅らせています。

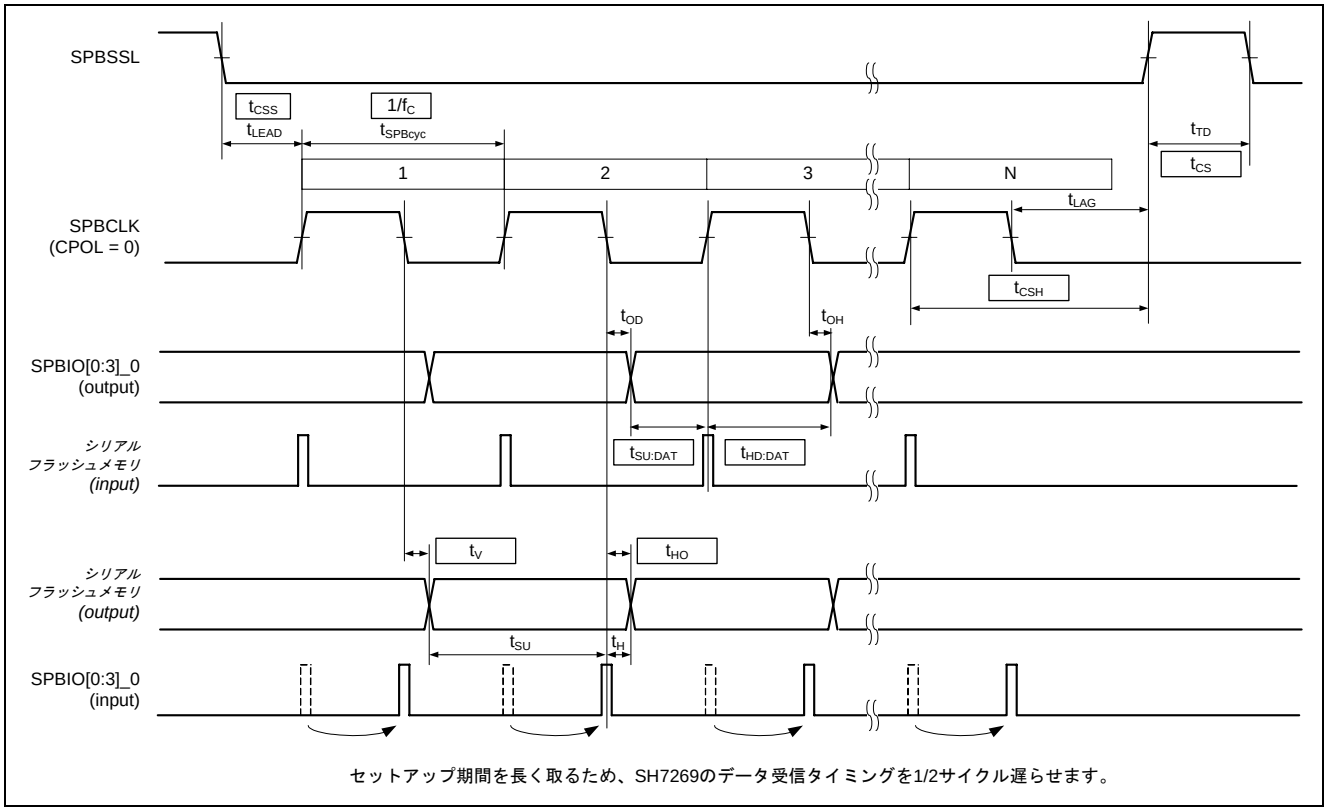


図2　本応用例におけるインタフェースタイミング図

表3　シリアルフラッシュメモリのタイミング条件

シンボル	項目	説明
t_{CSS}	チップセレクト'L' セットアップ時間	SSL のアサートからシリアルフラッシュメモリのデータ受信までに必要な時間です。以下の条件を満たす必要があります。 t_{LEAD}(=クロック遅延) ≧ t_{CSS} (min)
t_{CS}	チップセレクト'H'時間	SSL のネゲート期間として必要な時間です。以下の条件を満たす必要があります。 t_{TD}(=次アクセス遅延) ≧ t_{CS} (min)
f_C	シリアルクロック周波数	シリアルフラッシュメモリが対応可能な最大動作周波数です。以下の条件を満たす必要があります。 f_C(max) ≧ 1 / t_{SPBcyc}
t_{CSH}	チップセレクト'L' ホールド時間	SPBCLK の立ち上がりから SSL のネゲートまでに必要な時間です。以下の条件を満たす必要があります。 t_{LAG}(=SPBSSL ネゲート遅延) + (t_{SPBcyc} × 1/2) ≧ t_{CSH} (min)
$t_{SU:DAT}$	データ入力 セットアップ時間	データ入力に必要なセットアップ時間です。以下の条件を満たす必要があります。 (t_{SPBcyc} × 1/2) - t_{OD}(max) ≧ $t_{SU:DAT}$ (min)
$t_{HD:DAT}$	データ入力 ホールド時間	データ入力に必要なホールド時間です。以下の条件を満たす必要があります。 t_{OH}(min) + (t_{SPBcyc} × 1/2) ≧ $t_{HD:DAT}$ (min)

【注】t_{SPBcyc} は 2 t_{cyc} 固定です。また t_{cyc} はバスクロック（Bφ）の 1 サイクル時間を示します。

表4　SPIBSC のタイミング条件

シンボル	項目	説明
t_{SU}	データ入力 セットアップ時間	データ入力に必要なセットアップ時間です。以下の条件を満たす必要があります。 t_{SPBcyc} - t_V(max) ≧ t_{SU}(min)
t_H	データ入力 ホールド時間	データ入力に必要なホールド時間です。以下の条件を満たす必要があります。 t_{HO}(min) ≧ t_H(min)

【注】t_{SPBcyc} は 2 t_{cyc} 固定です。また t_{cyc} はバスクロック（Bφ）の 1 サイクル時間を示します。

表5　本応用例におけるインタフェースタイミングの設定値

レジスタ名	ビット名	設定値	機能
ビットレート設定レジスタ（SPBCR）	-	H'0000 0100	SPBCLK のビットレートを Bφの 2 分周に設定（66.67Mbps）
共通コントロールレジスタ（CMNCR）	CPOL ビット	B'0	SPBCLK のネゲートレベルを 'L' に設定
	CPAHT ビット	B'0	偶数エッジでデータ送信
	CPAHR ビット	B'1	偶数エッジでデータ受信
SSL 遅延レジスタ（SSLDR）	SPND[2:0]	B'000	次アクセス遅延設定を 1SPBCLK に設定
	SLNDL[2:0]	B'000	SPBSSL ネゲート遅延設定を 1.5SPBCLK に設定
	SCKDL[2:0]	B'000	クロック遅延設定を 1SPBCLK に設定

2.4　初期設定フロー

図 3に本応用例におけるSPIBSCの初期設定フローを示します。

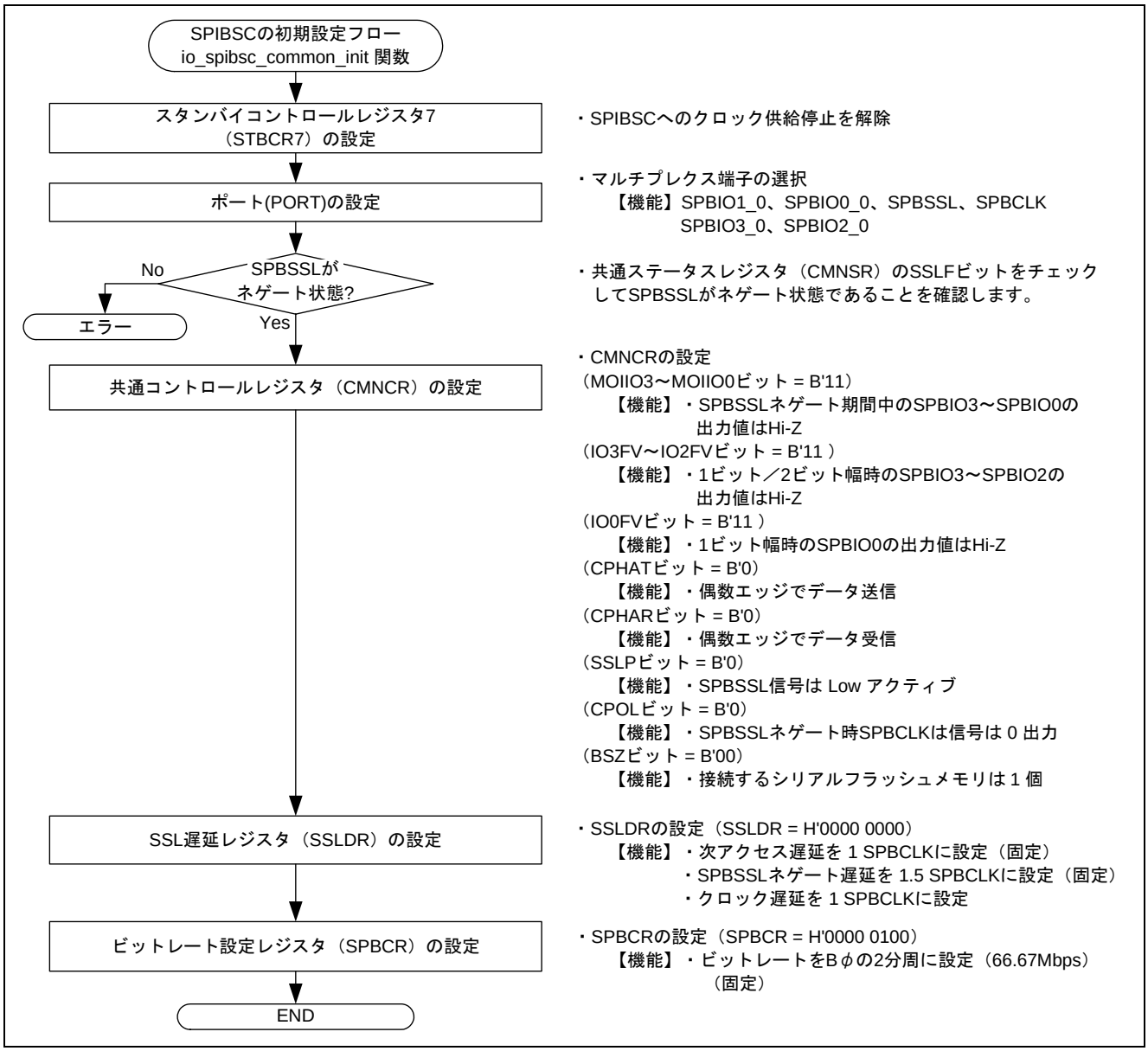


図3　本応用例における SPIBSC の初期設定フロー

2.5　SPI動作モード

2.5.1　　動作概要

SPI 動作モードは、シリアルフラッシュメモリに対して任意のリードライト動作を行うことができるモードです。SPIBSC を使用してシリアルフラッシュメモリに書き込みを行う場合は、本モードを使用する必要があります。

SPI動作モードを使用する場合は、「2.4 初期設定フロー」の 図 3で示した設定に加えて、図 6および 図 7、図 9に示す設定を行ってください。

2.5.2　　データフォーマットと関連レジスタ

シリアルフラッシュメモリに対するリードライト動作にはコマンドを使用します。コマンドのデータフォーマットはSPIBSCのレジスタで設定します。発行するコマンドに応じて設定してください。表 6にSPI動作モードのデータフォーマットと関連レジスタを示します。

表6　SPI 動作モードのデータフォーマットと関連レジスタ

項目	コマンド	オプショナルコマンド	アドレス	オプションデータ	転送データ
データ	SMCMR.CMD[7:0]ビット	SMCMR.OCMD[7:0]ビット	32 ビット時：SMADR.ADR[31:0]ビット 24 ビット時：SMADR.ADR[23:0]ビット	SMOPR.OPDn[7:0]ビット（n = 0～3）	<リード用> 32 ビット時：SMRDR0.RDATA0[31:0]ビット 16 ビット時：SMRDR0.RDATA0[31:16]ビット 8 ビット時：SMRDR0.RDATA0[31:24]ビット <ライト用> 32 ビット時：SMWDR0.WDATA0[31:0]ビット 16 ビット時：SMWDR0.WDATA0[31:16]ビット 8 ビット時：SMWDR0.WDATA0[31:24]ビット
ビット幅の設定（Single/Dual/Quad）	SMENR.CDB[1:0]ビット	SMENR.OCDB[1:0]ビット	SMENR.ADB[1:0]ビット	SMENR.OPDB[1:0]ビット	SMENR.SPIDB[1:0]ビット
データ入出力の許可	（常に出力）				SMCR.SPIRE ビット SMCR.SPIWE ビット
転送許可	SMENR.CDE ビット	SMENR.OCDE ビット	SMENR.ADE[3:0]ビット（ビット長も設定）	SMENR.OPDE[3:0]ビット	SMENR.SPIDE[3:0]ビット（ビット長も設定）

2.5.3 SPBSSL 端子アサート保持機能

図 4にSPI動作モードのSPBSSLアサート保持機能を示します。SPI動作モードの場合、SPIモードコントロールレジスタ（SMCR）のSSLKPビットを 1 に設定すると、転送終了後から次アクセス開始までSPBSSL信号レベルを保持します。この機能を使用すると連続転送が可能になりますが、転送データのビット幅を 2 ビット以上に設定したデータリード処理では、本機能を使用できません。

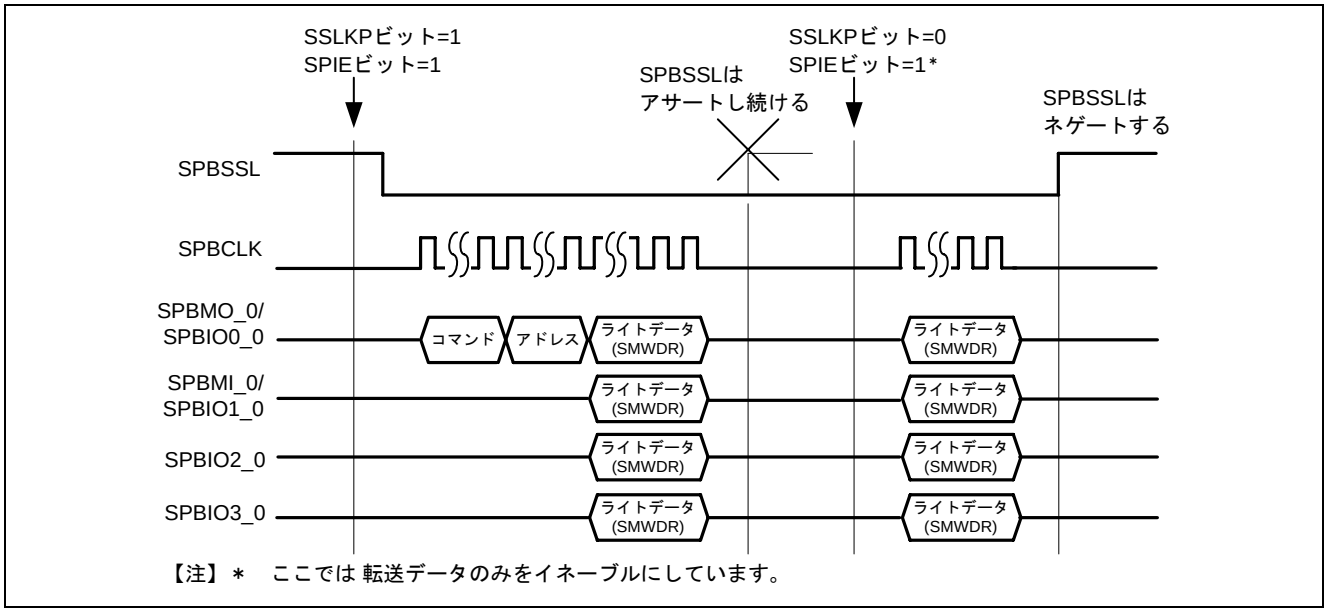


図4　SPI 動作モードの SPBSSL アサート保持機能

2.5.4　データリード手順

(1)　リードコマンド

表 7と 図 5にSPI動作モードで使用するS25FL032Pのリードコマンドを示します。なお、ここでは参考プログラムで使用しているコマンドのみ記載しています。

表7　SPI 動作モードで使用する S25FL032P のリードコマンド

コマンド名	コマンドコード	アドレスバイト数	ダミーバイト数	データバイト数	機能
Quad Output Read	H'6B	3	1	1 以上*	データのリード（Quad-SPI）

【注】*　指定アドレスからインクリメントされた領域をリードします。（最終番地を超えた場合は 0 番地に戻ります。）

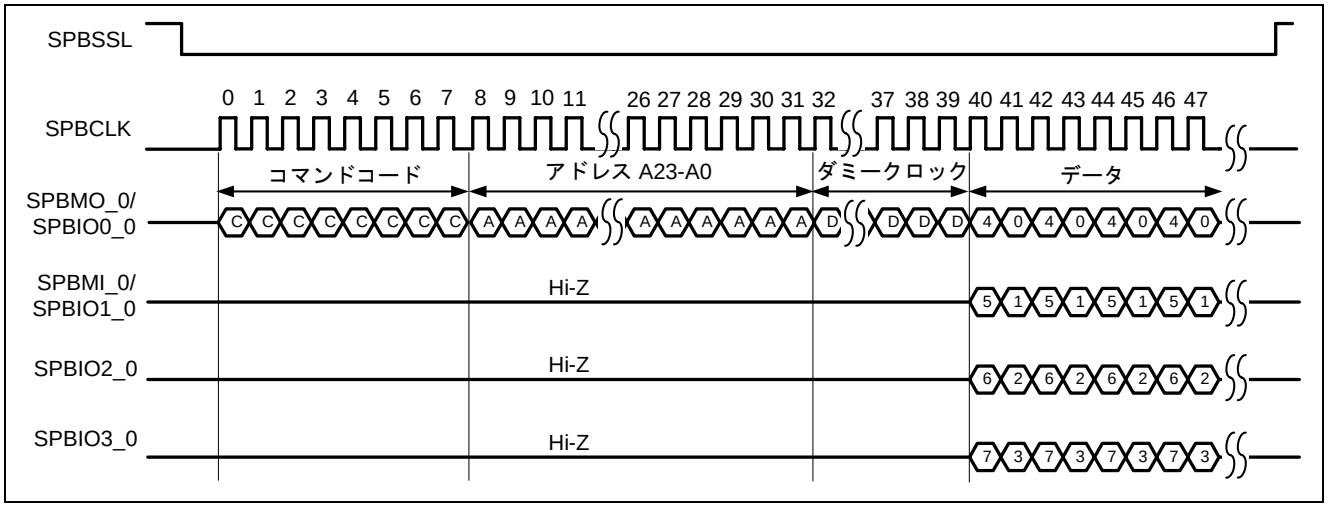


図5　Quad Output Read のコマンドシーケンス

【注】 Quad Output Read コマンドは、転送データのビット幅が 4 ビットのため一回のコマンド発行で 4 バイトしかリードできません。

(2) SPI 動作モード設定フロー（リード）

図 6および図 7に本応用例におけるSPI動作モードのリードコマンド転送フローを示します。

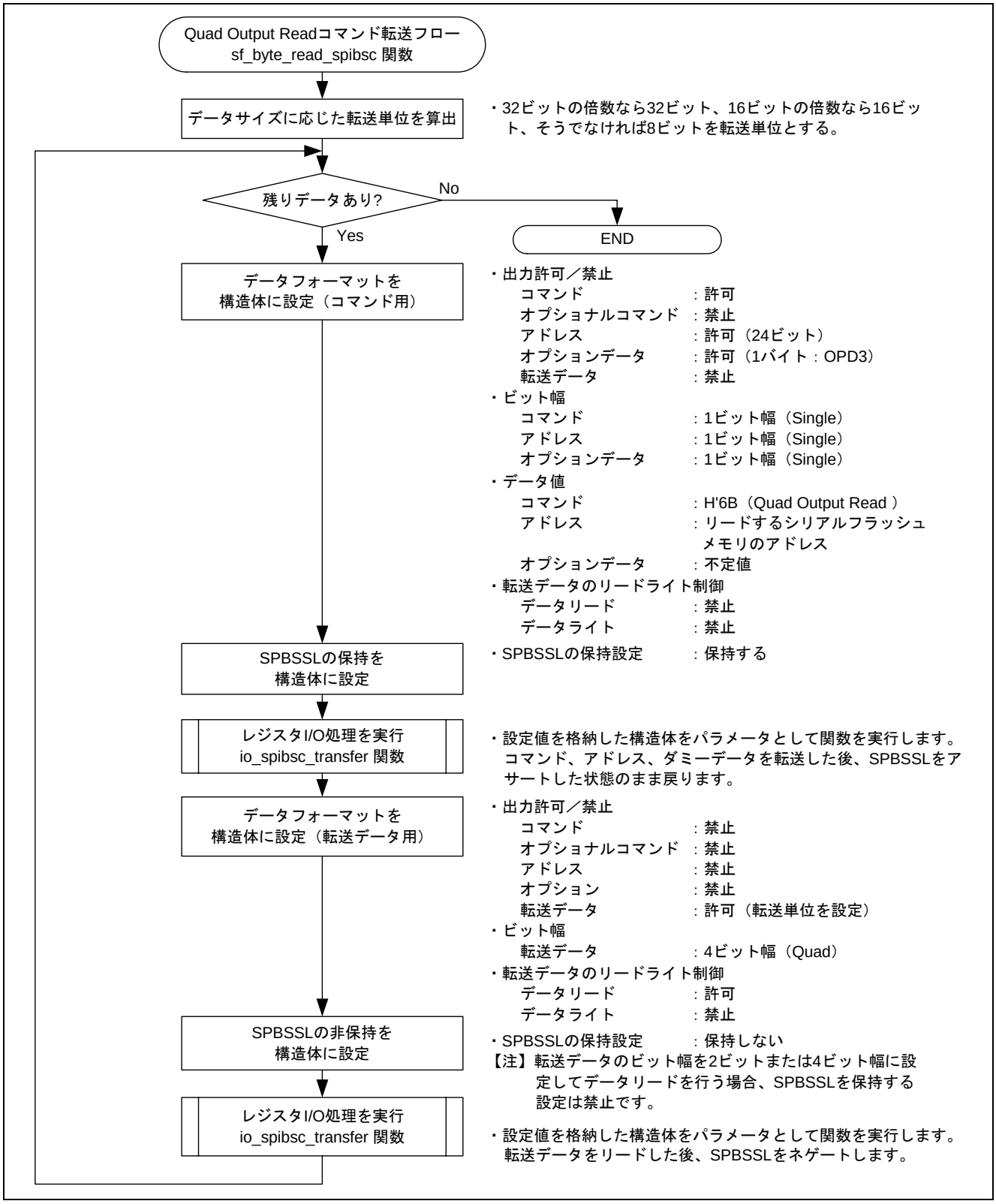
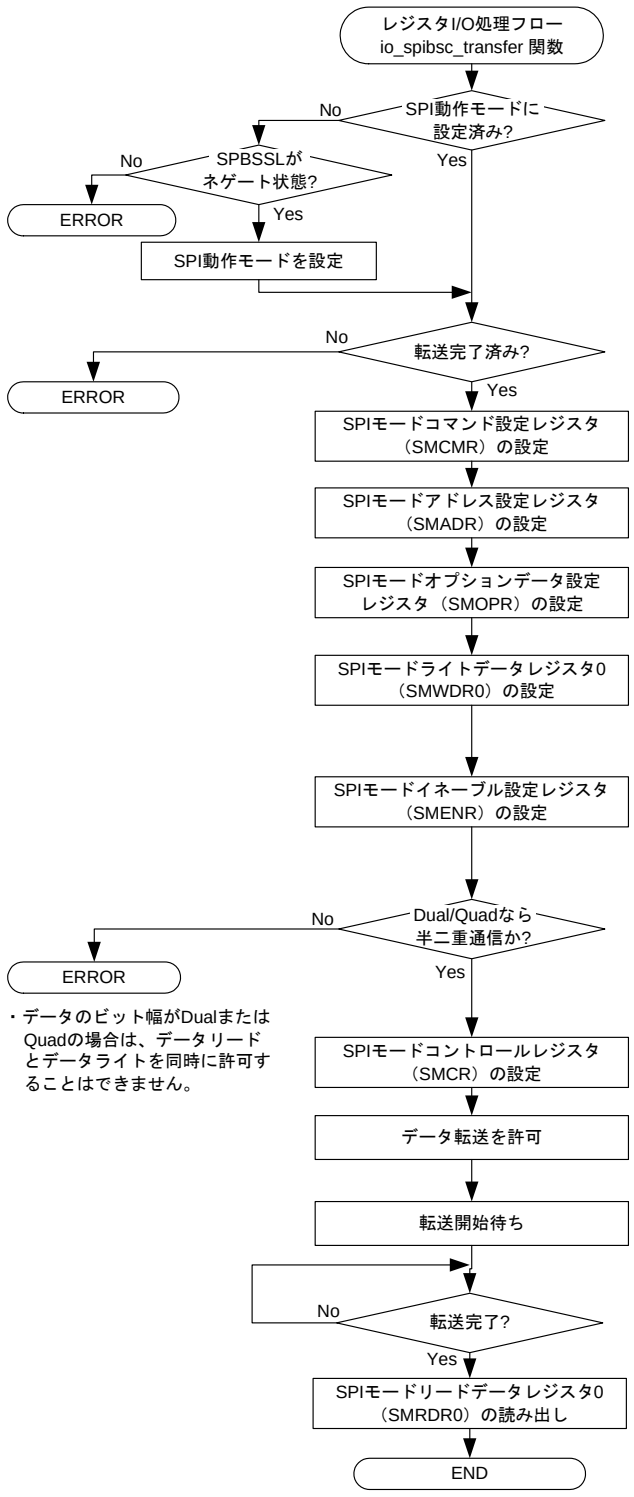


図6 SPI 動作モードのリードコマンド転送フロー



・共通コントロールレジスタ（CMNCR）のMDビットの値がSPI動作モードでない場合は、共通ステータスレジスタ（CMNSR）のSSLFビットの値がネゲート状態であることを確認した上でSPI動作モードに変更します。

・転送が完了していることを確認します。CMNSRレジスタのTENDビットが B'0 の場合は転送中です。

・SMCMRレジスタの設定
【機能】 ・CMD[7:0]ビット ：コマンドの値を設定
・OCMD[7:0]ビット ：オプショナルコマンドの値を設定

・SMADRレジスタの設定
【機能】 ・ADR[31:0]ビット ：アドレスの値を設定

・SMOPRレジスタの設定
【機能】 ・OPD3[7:0]ビット ：オプションデータの値を設定（最初の出力）
・OPD2[7:0]ビット ：オプションデータの値を設定（2番目の出力）
・OPD1[7:0]ビット ：オプションデータの値を設定（3番目の出力）
・OPD0[7:0]ビット ：オプションデータの値を設定（4番目の出力）

・SMWDR0レジスタの設定
【機能】 ・WDATA0[31:0]ビット：ライトする転送データの値を設定
8ビット転送時 WDATA0[31:24]が有効
16ビット転送時 WDATA0[31:16]が有効
32ビット転送時 WDATA0[31:0]が有効

・SMENRレジスタの設定
【機能】 ・CDB[1:0]ビット ：コマンドのビット幅を設定
・OCDB[1:0]ビット ：オプショナルコマンドのビット幅を設定
・ADB[1:0]ビット ：アドレスのビット幅を設定
・OPDB[1:0]ビット ：オプションデータのビット幅を設定
・SPIDB[1:0]ビット ：転送データのビット幅を設定
・CDEビット ：コマンド出力の許可/禁止を設定
・OCDEビット ：オプショナルコマンド出力の許可/禁止を設定
・ADE[3:0]ビット ：アドレスの出力許可、データ長を設定
・OPDE[3:0]ビット ：オプションデータの出力許可、データ長を設定
・SPIDE[3:0]ビット ：転送データの出力許可、データ長を設定

・データのビット幅がDualまたはQuadの場合は、データリードとデータライトを同時に許可することはできません。

・SMCRレジスタの設定
【機能】 ・SSLKPビット ：SPBSSL信号レベル保持／非保持を設定
・SPIREビット ：データリードの許可／禁止を設定
・SPIWEビット ：データライトの許可／禁止を設定

・SMCRレジスタのSPIEビットに B'1 を設定して、データ転送を開始します。

・CMNSRレジスタのダミーリードを4回繰り返して、転送開始状態（TENDビット = B'0）になるまで待ちます。

・CMNSRレジスタのTENDビットが B'1 になるまで待ちます。

・SMRDR0レジスタの読み出し
【機能】 ・RDATA0[31:0]ビット：リードした転送データの値を読み出し
8ビット転送時 RDATA0[31:24]が有効
16ビット転送時 RDATA0[31:16]が有効
32ビット転送時 RDATA0[31:0]が有効

図7　SPI 動作モードのレジスタ IO 処理フロー

2.5.5 データライト手順

(1) ライトコマンド

表 8と 図 8にSPI動作モードで使用するS25FL032Pのライトコマンドを示します。なお、ここでは参考プログラムで使用しているコマンドのみ記載しています。

表8 SPI 動作モードで使用する S25FL032P のライトコマンド

コマンド名	コマンドコード	アドレスバイト数	ダミーバイト数	データバイト数	機能
Quad Page Programming	H'32	3	0	1 以上※1	データのライト（Quad-SPI）

【注】 ※1 指定アドレスと同一ページ内で、インクリメントされた領域にライトします。（ページの最終番地を超えた場合はページの先頭に戻ります。）

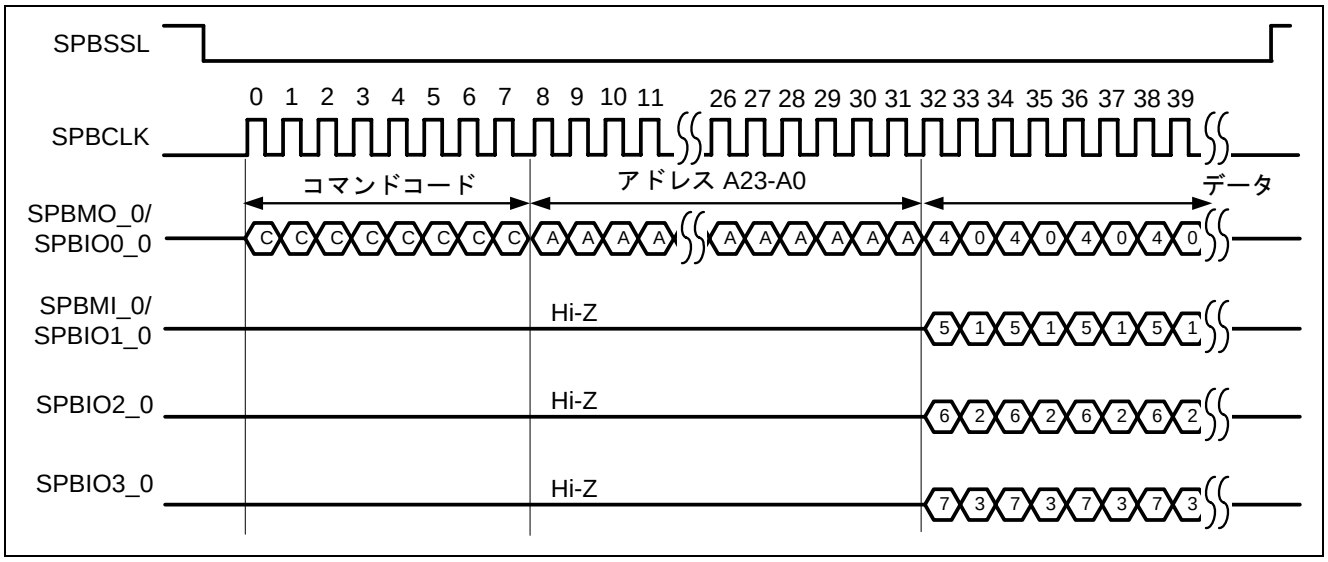


図8 Quad Page Programming コマンドシーケンス

(2) SPI 動作モード設定フロー（ライト）

図 9に本応用例におけるSPI動作モードのライトコマンド転送フローを示します。レジスタI/O処理フローについては 図 7を参照してください。

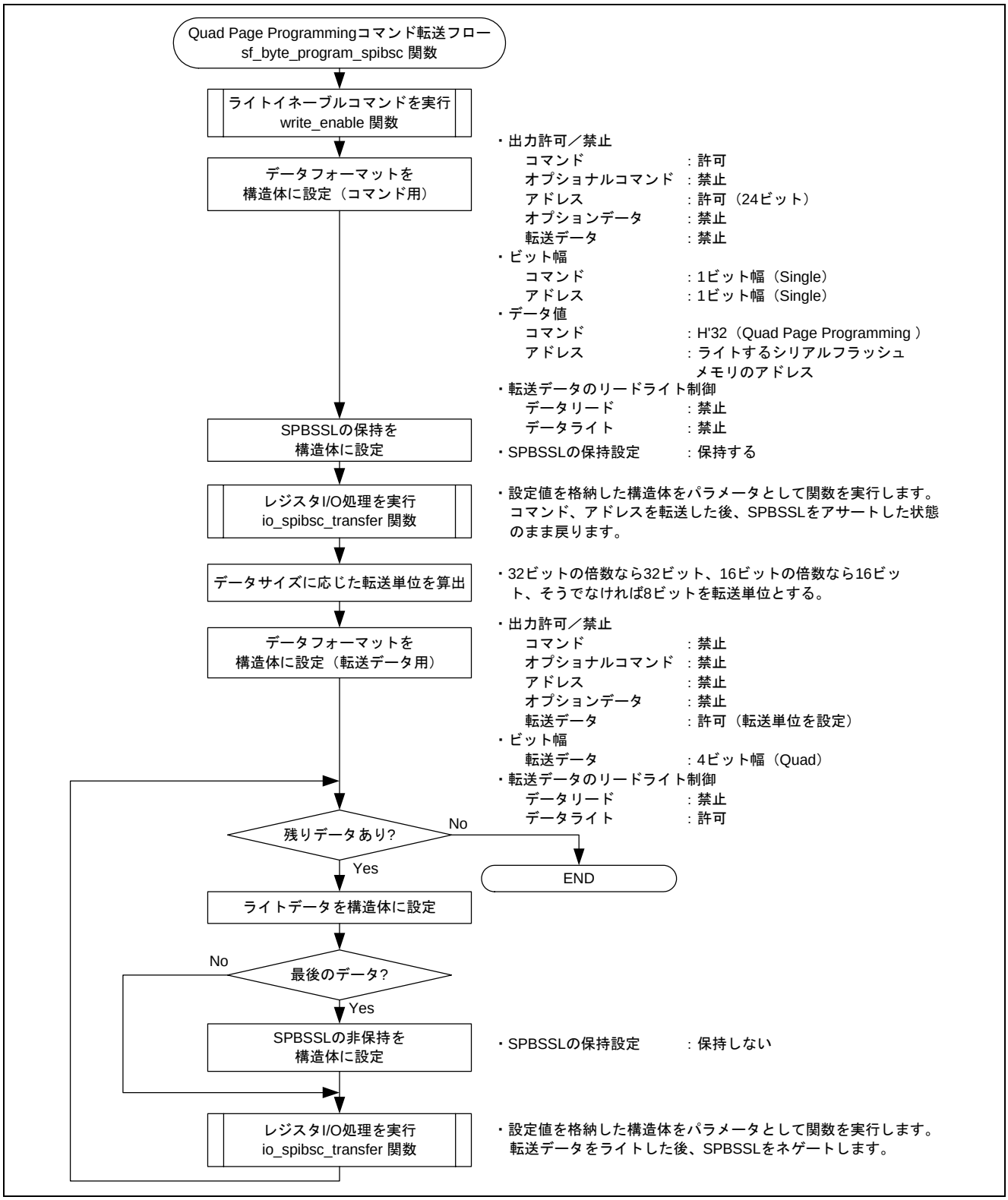


図9　SPI 動作モードのライトコマンド転送フロー

2.6 外部アドレス空間リードモード

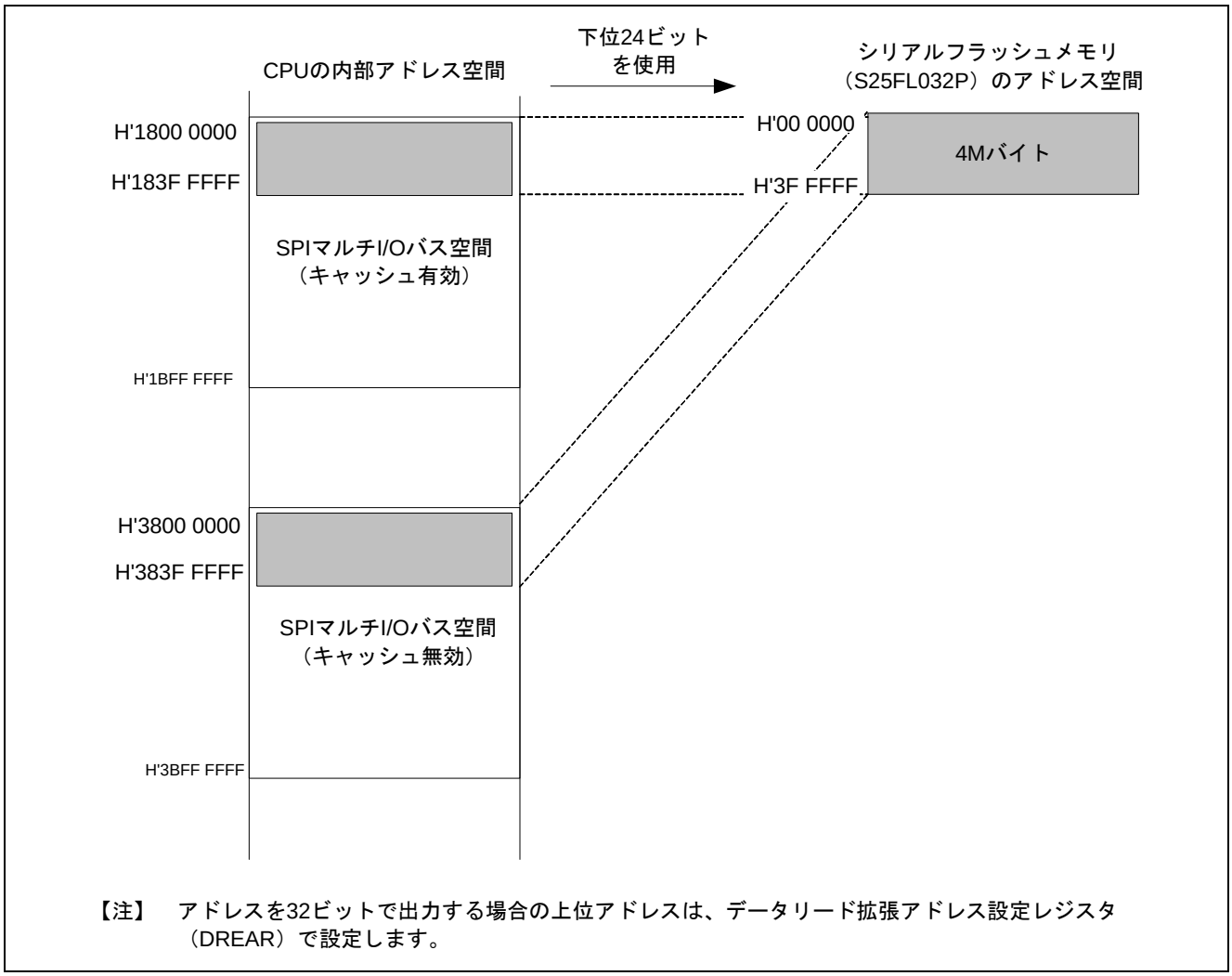
2.6.1 動作概要

外部アドレス空間リードモードは、SPI マルチ I/O バス空間へのリードアクセスを SPI 通信に自動変換するモードです。本モードを使用すると、NOR フラッシュメモリと同じように、メモリ上のプログラムを直接フェッチできるようになります。そのため、プログラムを RAM 上に展開する必要がなくなり RAM 容量を削減できます。

外部アドレス空間リードモードを使用する場合は、「2.4 初期設定フロー」の 図 3で示した設定に加えて、図 14に示す設定を行ってください。

2.6.2 アドレスの自動変換

図 10に外部アドレス空間リードモードにおけるアドレス変換イメージ（シリアルフラッシュメモリが 24 ビットアドレスの場合）を示します。SPIBSCは、SPIマルチI/Oバス空間であるH'1800 0000 番地 ～ H'1BFF FFFF番地へのリードアクセスを検出すると、検出したアドレスの下位 24 ビットをシリアルフラッシュメモリへのアクセスに使用します。またキャッシュ無効空間（H'3800 0000 番地 ～ H'3BFF FFFF）に対しても同様に変換されます。



【注】 アドレスを32ビットで出力する場合の上位アドレスは、データリード拡張アドレス設定レジスタ（DREAR）で設定します。

図10 外部アドレス空間リードモードのアドレス変換イメージ

2.6.3　データフォーマットと関連レジスタ

シリアルフラッシュメモリにはコマンドを使ってアクセスします。コマンドのデータフォーマットはSPIBSCのレジスタで設定します。表 6に外部アドレス空間リードモードのデータフォーマットと関連レジスタを示します。SPI動作モードで使用するレジスタとは異なるレジスタを使用する点に注意してください。

表9　外部アドレス空間リードモードのデータフォーマットと関連レジスタ

項目	コマンド	オプショナルコマンド	アドレス	オプションデータ	転送データ
データ	DRCMR.CMD[7:0]ビット	DRCMR.OCMD[7:0]ビット	（24 ビット時）リードした下位アドレス[23:0]ビット	DROPR.OPDn[7:0]ビット（n = 0～3）	<通常リード時> アクセスサイズに応じたビット数を転送（8/16/32/64 ビット） <バーストリード時> DRCR.RBURST[3:0]ビット（RBURST × 64 ビット）
ビット幅の設定（Single/Dual/Quad）	DRENR.CDB[1:0]ビット	DRENR.OCDB[1:0]ビット	DRENR.ADB[1:0]ビット	DRENR.OPDB[1:0]ビット	DRENR.DRDB[1:0]ビット
データ入出力の許可	（常に出力）				（常に入力）
転送許可	DRENR.CDEビット	DRENR.OCDEビット	DRENR.ADE[3:0]ビット（ビット長も設定）	DRENR.OPDE[3:0]ビット	常に許可

2.6.4 リードコマンド

表 10と 図 11に外部アドレス空間リードモードで使用するS25FL032Pのリードコマンドを示します。なお、ここでは参考プログラムで使用しているコマンドのみ記載しています。

表10　外部アドレス空間リードモードで使用する S25FL032P のリードコマンド

コマンド名	コマンドコード	アドレスバイト数	モードバイト数	ダミーバイト数	データバイト数	機能
Quad I/O High Performance Read	H'EB	3	1*1	2	1 以上*2	データの高速リード（Quad-SPI）

【注】 *1 H'A0～H'AF を設定すると連続モードが設定され、次に SPBSSL をアサートした際にコマンドコードが不要になります。ただし本応用例では使用しません。

*2 指定アドレスからインクリメントされた領域をリードします。（最終番地を超えた場合は 0 番地に戻ります。）

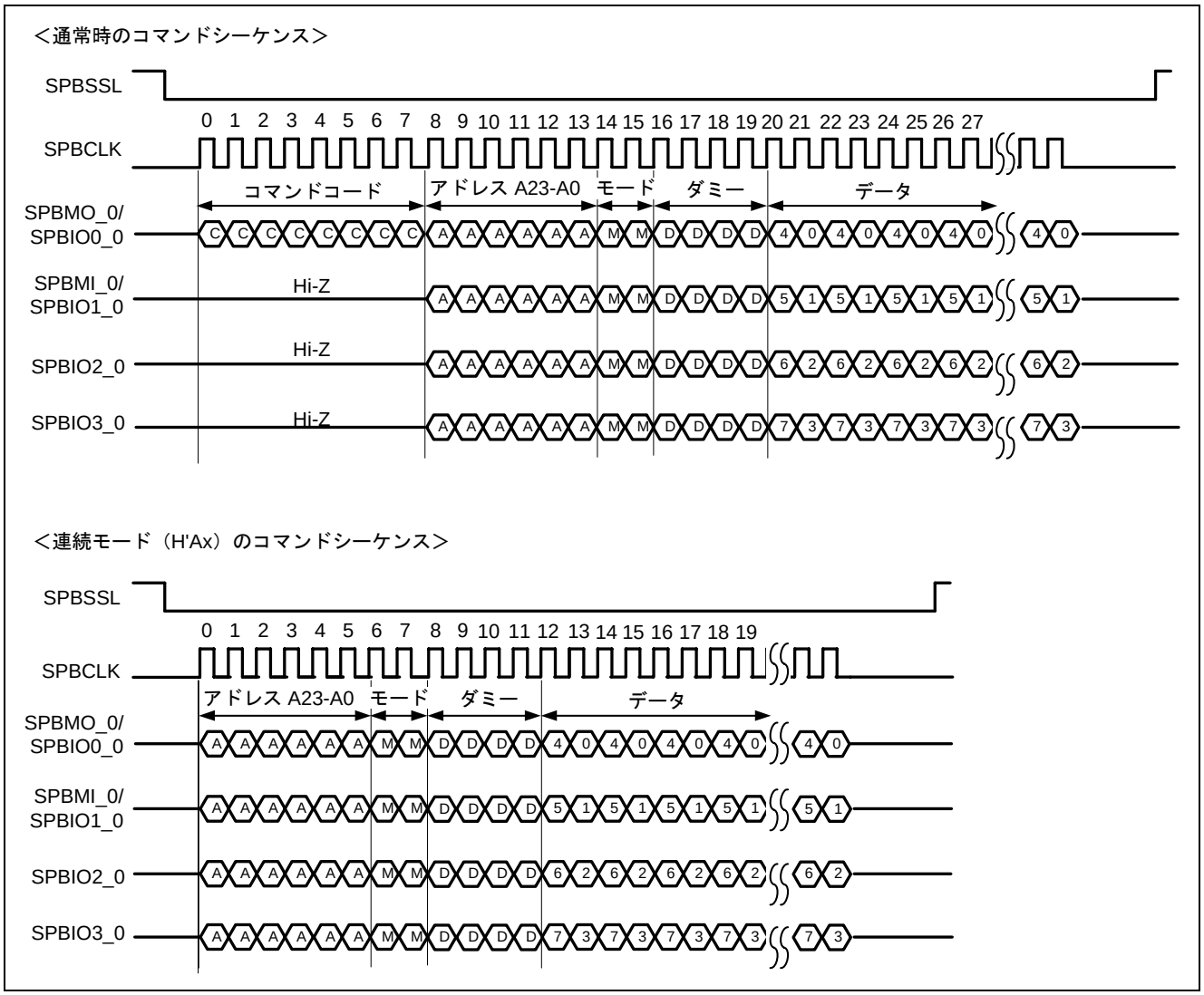


図11　Quad I/O High Performance Read のコマンドシーケンス

2.6.5 バーストリード動作

データリードコントロールレジスタ（DRCR）の RBE ビットを 1 に設定すると、バーストリード動作となります。また同時にリードキャッシュも有効になります。以下にバーストリードおよびリードキャッシュの動作概要を示します。

(1) バーストリードとリードキャッシュ

図 12にバーストリードとリードキャッシュの動作を示します。

SPI マルチ I/O バス空間へのリードアクセスを検出すると、SPIBSC はまずリードキャッシュからデータを参照します。リードキャッシュにデータがある場合は、シリアルフラッシュメモリへはアクセスせず、リードキャッシュからデータを読み出します。リードキャッシュにデータが無い場合は、シリアルフラッシュメモリをバーストリードし読み出したデータをリードキャッシュに格納します。このときのデータ転送長は 64 ビット×RBURST[3:0]ビットであり、必ず 64 ビット境界からリードします。

また、リードキャッシュをフラッシュしたい場合は DRCR レジスタの RCF ビットをセットします。

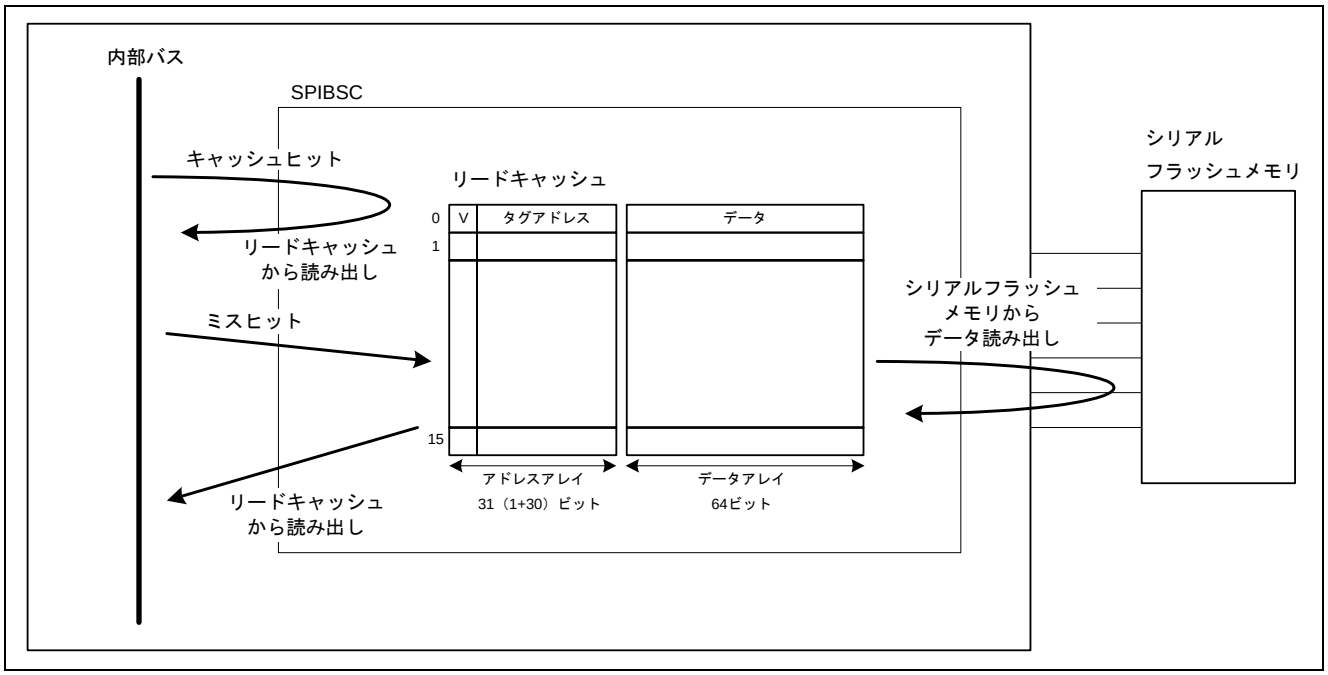


図12　バーストリードとリードキャッシュの動作

(2) SPBSSL 自動ネゲート

図 13にバーストリード動作のSPBSSL自動ネゲート機能を示します。DRCRレジスタのSSLEビットを 1 に設定すると、バーストリード転送後にSPBSSL端子をネゲートしません。次回アクセス時、前回リードアドレスに対してアドレスが連続している場合、コマンド／オプショナルコマンド／アドレス／オプションデータを発行することなしにバーストリードを行います。また、アドレスが連続していない場合はSPBSSL端子を一度ネゲートし、コマンド／オプショナルコマンド／アドレス／オプションデータを発行後にバーストリードを行います。

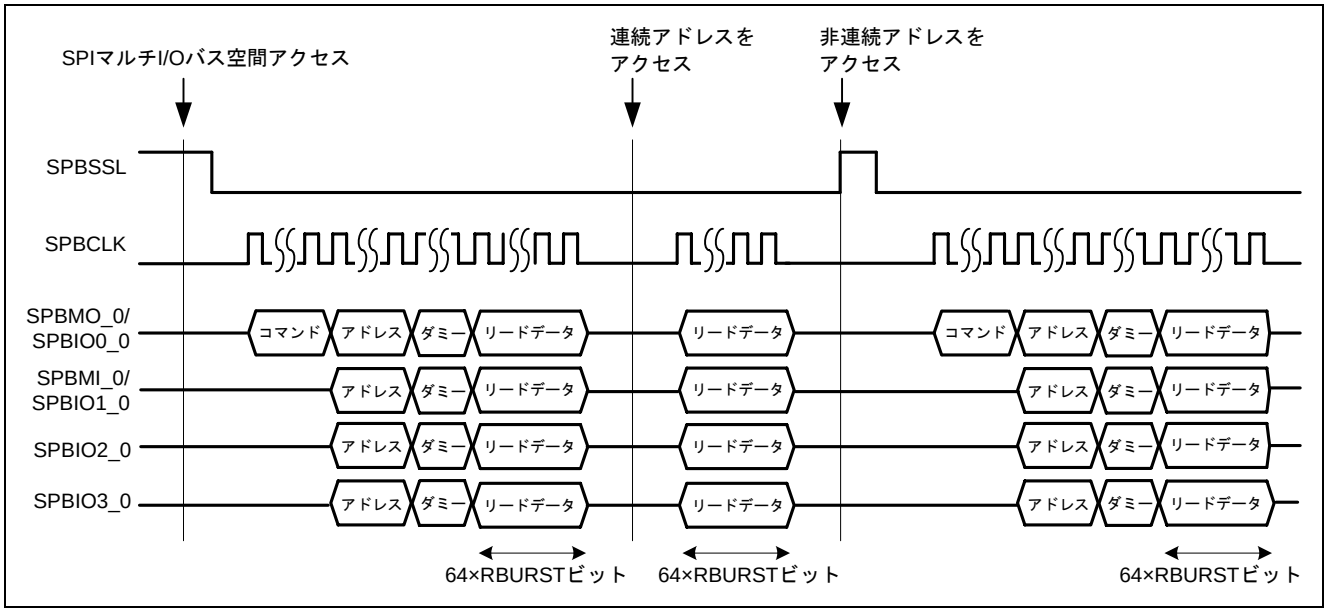


図13　バーストリード動作の SPBSSL 自動ネゲート機能

2.6.6 外部アドレス空間リードモード設定フロー

図 14に本応用例における外部アドレス空間リードモードの設定フローを示します。

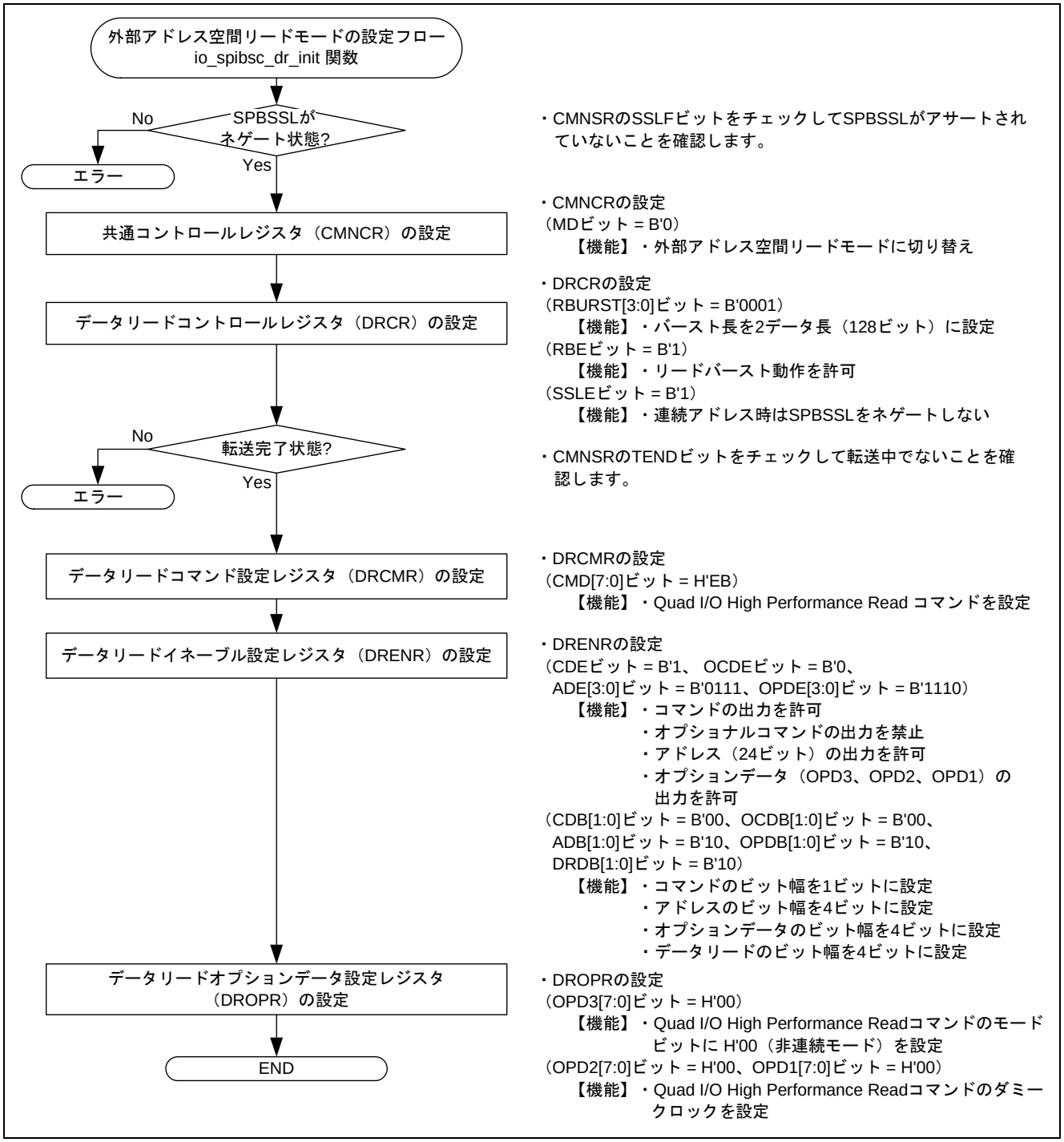


図14　外部アドレス空間リードモードの設定フロー

2.7 参考プログラムの動作概要

ここでは、参考プログラムの動作概要を説明します。参考プログラムは、まず外部アドレス空間リードモードが有効な状態で起動して SPI マルチ I/O バス空間上に配置したメイン関数を実行します。次に SPI 動作モードを使用してシリアルフラッシュメモリをリードライトしますが、SPI マルチ I/O 空間上では SPI 動作モードに切り換えられないため、大容量内蔵 RAM 上に配置した関数で実行します。最後に、再び外部アドレス空間リードモードを有効にしてメイン関数に戻ります。

2.7.1 メイン関数フロー

図 15に参考プログラムのメイン関数フローを示します。

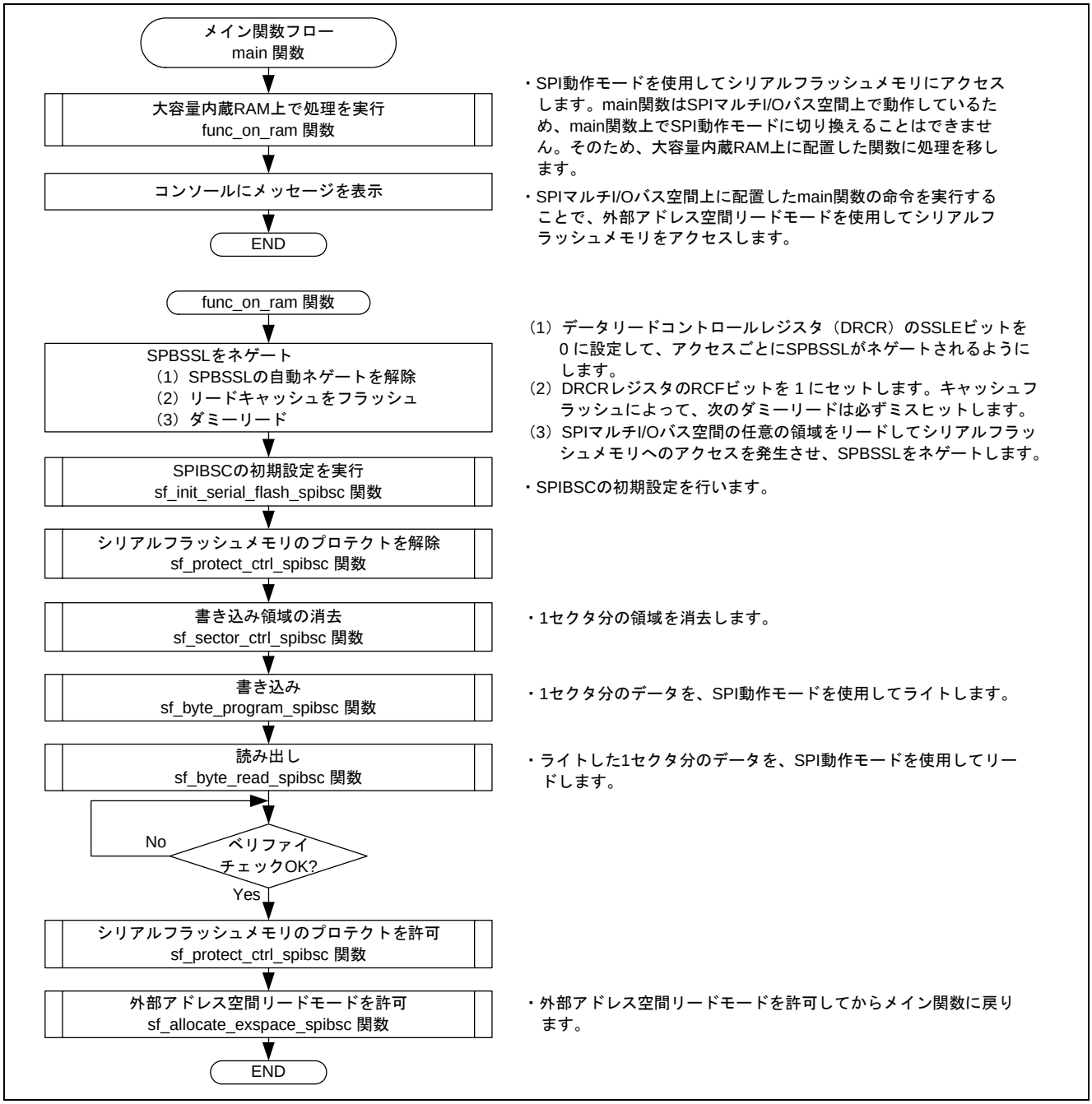


図15　参考プログラムのメイン関数フロー

3. 参考プログラムリスト

3.1 参考プログラムについての補足

ブートモード 0 およびブートモード 1（CS0 空間に接続したメモリからのブート）使用時は、端子設定を SPIBSC 機能に設定することはできません。そのため、参考プログラムはブートモード 3（シリアルフラッシュブート）で起動します。

シリアルフラッシュブートを使用する際のブート手順や、プログラムをシリアルフラッシュメモリに書き込む方法については、アプリケーションノート「SH7268/SH7269 グループ SPI マルチ I/O バスコントローラを使用したシリアルフラッシュメモリからのブート例」を参照してください。

3.2 サンプルプログラムリスト"main.c"（1）

```
1   /***********************************************************************
2   *   DISCLAIMER
3   *
4   *   This software is supplied by Renesas Electronics Corporation and is only
5   *   intended for use with Renesas products. No other uses are authorized.
6   *
7   *   This software is owned by Renesas Electronics Corporation and is protected under
8   *   all applicable laws, including copyright laws.
9   *
10  *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11  *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12  *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13  *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14  *   DISCLAIMED.
15  *
16  *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17  *   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18  *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19  *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20  *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21  *
22  *   Renesas reserves the right, without notice, to make changes to this
23  *   software and to discontinue the availability of this software.
24  *   By using this software, you agree to the additional terms and
25  *   conditions found by accessing the following link:
26  *   http://www.renesas.com/disclaimer
27  ************************************************************************
28  *   Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29  *************************** Technical reference data *************************
30  *   System Name : SH7268/SH7269 Sample Program
31  *   File Name   : main.c
32  *   Abstract    : Sample Program Main
33  *   Version     : 1.00.00
34  *   Device      : SH7268/SH7269
35  *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36  *               : C/C++ compiler package for the SuperH RISC engine family
37  *               :                          (Ver.9.03Release02).
38  *   OS          : None
39  *   H/W Platform: R0K57269(CPU board)
40  *   Description :
41  ************************************************************************
42  *   History     : Jul.06,2011 Ver.1.00.00
43  ************************************************************************/
44  #include <stdio.h>
45  #include <string.h>
46  #include <machine.h>
47  #include "serial_flash.h"
48  #include "iodefine.h"
49
```

3.3 サンプルプログラムリスト"main.c"（2）

```
50    /* ==== prototype declaration ==== */
51    void main(void);
52    void func_on_ram(void);
53
54    /***********************************************************************
55     * ID         :
56     * Outline    : main
57     * Include    :
58     * Declaration : void main(void);
59     * Description :
60     * Argument    : void
61     * Return Value: void
62     * Note       : None
63     ***********************************************************************/
64    void main(void)
65    {
66        func_on_ram();
67
68        puts("¥nSH7269 SPIBSC Sample Program. Ver.1.00.00");
69        puts("Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.");
70        puts("¥n");
71
72        while(1){
73            /* loop */
74        }
75    }
76
77
78    #pragma section SPIBSC
79    /***********************************************************************
80     * ID         :
81     * Outline    : SPI operating mode
82     * Include    :
83     * Declaration : void exe_spibsc_spi(void) ;
84     * Description :
85     * Argument    : void
86     * Return Value: void
87     * Note       : None
88     ***********************************************************************/
89    void func_on_ram(void)
90    {
91        volatile short dummy;
92        int w_size  = SF_PAGE_SIZE;
93        int w_sctno = (SF_NUM_OF_SECTOR - 1);
94        int w_addr, bsz, i;
95        static char r_data[ SF_PAGE_SIZE ];
96        static char w_data[ SF_PAGE_SIZE ];
97
```

3.4 サンプルプログラムリスト"main.c"（3）

```
98        /* ==== Use SPI operating mode ==== */
99
100       /* Initialize data */
101       for(i=0; i<w_size; i++){
102          r_data[i] = 'R';
103          w_data[i] = 'W';
104       }
105       bsz = 1;
106       w_addr = (w_sctno * SF_SECTOR_SIZE * bsz);
107
108       /* Negate SPBSSL */
109       SPIBSC.DRCR.BIT.SSLE = 0;       /* No keep SSL */
110       SPIBSC.DRCR.BIT.RCF = 1;        /* Chach flush */
111       dummy = *(short *)0x18000000;   /* Dummy read */
112
113       /* Initializes the SPIBSC */
114       sf_init_serial_flash_spibsc();
115
116       /* Disables the software protection in serial flash memory */
117       sf_protect_ctrl_spibsc(SF_REQ_UNPROTECT);
118
119       /* Erase */
120       sf_sector_erase_spibsc(w_sctno);
121
122       /* Write */
123       sf_byte_program_spibsc(w_addr, w_data, w_size );
124
125       /* Read */
126       sf_byte_read_spibsc(w_addr,r_data, w_size);
127
128       /* Verifies data */
129       for(i=0; i<w_size; i++){
130          if( r_data[i] != w_data[i] ){
131             while(1){
132                /* error */
133             }
134          }
135       }
136       /* Enables the software protection in serial flash memory */
137       sf_protect_ctrl_spibsc(SF_REQ_PROTECT);
138
139
140       /* ==== Enable external address space read mode ==== */
141       sf_allocate_exspace_spibsc();
142
143    }
144
145    /* End of File */
```

3.5 サンプルプログラムリスト"qserial_flash_spibsc.c"（1）

```
1     /***********************************************************************
2     *   DISCLAIMER
3     *
4     *   This software is supplied by Renesas Electronics Corporation and is only
5     *   intended for use with Renesas products. No other uses are authorized.
6     *
7     *   This software is owned by Renesas Electronics Corporation and is protected under
8     *   all applicable laws, including copyright laws.
9     *
10    *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11    *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12    *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13    *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14    *   DISCLAIMED.
15    *
16    *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17    *   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18    *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19    *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20    *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21    *
22    *   Renesas reserves the right, without notice, to make changes to this
23    *   software and to discontinue the availability of this software.
24    *   By using this software, you agree to the additional terms and
25    *   conditions found by accessing the following link:
26    *   http://www.renesas.com/disclaimer
27    ************************************************************************
28    *   Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29    *************************** Technical reference data *********************
30    *   System Name : SH7268/SH7269 Firm Update Sample Program
31    *   File Name   : qserial_flash_spibsc.c
32    *   Abstract    :
33    *   Version     : 1.00.00
34    *   Device      : SH7268/SH7269
35    *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36    *               : C/C++ compiler package for the SuperH RISC engine family
37    *               :                          (Ver.9.03Release02).
38    *   OS          : None
39    *   H/W Platform: R0K57269(CPU board)
40    *   Description :
41    ************************************************************************
42    *   History     : Jul.06,2011 Ver.1.00.00
43    ************************************************************************/
44    #include "io_spibsc.h"
45    #include "serial_flash.h"
46    #include "qserial_flash_spibsc.h"
47
```

3.6 サンプルプログラムリスト"qserial_flash_spibsc.c"（2）

```
48    #pragma section SPIBSC
49
50    /* ---- serial flash command[S25FL032P(Spansion)] ---- */
51    #define SFLASHCMD_CHIP_ERASE  0xc7
52    #define SFLASHCMD_SECTOR_ERASE0xd8
53    #define SFLASHCMD_BYTE_PROGRAM0x02
54    #define SFLASHCMD_BYTE_READ       0x0B       /* fast read   */
55    #define SFLASHCMD_DUAL_READ       0x3B
56    #define SFLASHCMD_QUAD_READ       0x6B
57    #define SFLASHCMD_DUAL_IO_READ0xBB
58    #define SFLASHCMD_QUAD_IO_READ0xEB
59    #define SFLASHCMD_WRITE_ENABLE0x06
60    #define SFLASHCMD_READ_STATUS 0x05
61    #define SFLASHCMD_READ_CONFIG 0x35
62    #define SFLASHCMD_WRITE_STATUS0x01
63    #define SFLASHCMD_QUAD_PROGRAM0x32
64    /* ---- serial flash register definitions ---- */
65    #define CFREG_QUAD_BIT        0x02       /* Quad mode bit(Configuration Register) */
66    #define CFREG_FREEZE_BIT      0x01       /* freeze bit(Configuration Register) */
67    #define STREG_BPROTECT_BIT    0x1c       /* protect bit(Status Register) */
68
69    /* ==== Prototype Declaration ==== */
70    static void read_status(unsigned char* status1,unsigned char* status2);
71    static void read_config(unsigned char* config1,unsigned char* config2);
72    static void busy_wait(void);
73    static void write_status(unsigned char status, unsigned char config);
74    static void sf_set_mode(enum sf_req req);
75    static void write_enable(void);
76    #if (SPI_QUAD != 0)
77    static void sf_byte_read_spibsc_quad(unsigned long addr, unsigned char *buf, int unit);
78    #endif
79
80    /* ==== Global variable ==== */
81    ST_SPIBSC_SM SpibscSm;
82
83
84
85          （省略）
86
```

3.7 サンプルプログラムリスト"qserial_flash_spibsc.c"（3）

```
126
127  /***********************************************************************
128   * ID          :
129   * Outline     : External address space read mode
130   * Include     :
131   * Declaration  : void sf_allocate_exspace_spibsc(void);
132   * Description  : Set to the external address space read mode
133   * Argument     : void
134   * Return Value : void
135   * Note        : None
136   ***********************************************************************/
137  void sf_allocate_exspace_spibsc (void)
138  {
139  #if (SFLASH_DUAL == 0)
140     sf_bsz_set_spibsc(1);                  /* s-flash x 1  */
141  #else
142     sf_bsz_set_spibsc(2);                  /* s-flash x 2  */
143  #endif
144
145  #if (SPI_QUAD == 0)
146     io_spibsc_dr_init(SFLASHCMD_BYTE_READ);    /* Single-SPI   */
147  #else
148     io_spibsc_dr_init(SFLASHCMD_QUAD_IO_READ);  /* Quad-SPI    */
149  #endif
150  }
151
152  /***********************************************************************
153   * ID          :
154   * Outline     : Initialize the serial flash memory
155   * Include     :
156   * Declaration  : void sf_init_serial_flash_spibsc(void);
157   * Description  : Initialize to access to the serial flash memory
158   *              : Initialize the SPI multi bus I/O bus controller(SPIBSC)
159   *              : to set the serial flash memory to Quad mode
160   * Argument    : void
161   * Return Value : void
162   * Note        : None
163   ***********************************************************************/
164  void sf_init_serial_flash_spibsc(void)
165  {
166     /* ==== SPIBSC の初期化 ==== */
167  #if (SFLASH_DUAL == 0)
168     io_spibsc_common_init(SPIBSC_CMNCR_BSZ_SINGLE); /* s-flash x 1  */
169  #else
170     io_spibsc_common_init(SPIBSC_CMNCR_BSZ_DUAL);   /* s-flash x 2  */
171  #endif
172
173  #if (SPI_QUAD == 0)
174     sf_set_mode( SF_REQ_SERIALMODE );
175  #else
176     /* ==== setting serial-flash quad mode ==== */
177     sf_set_mode( SF_REQ_QUADMODE );
178  #endif
179  }
```

3.8 サンプルプログラムリスト"qserial_flash_spibsc.c"（4）

```
180
181   /*****************************************************************************
182    * ID          :
183    * Outline     : Protection
184    * Include     :
185    * Declaration  : void sf_protect_ctrl_spibsc(enum sf_req req);
186    * Description  : Serial flash memory protect setting/clearing the setting
187    *              : Specify the setting by the argument, reg. The initial value of
188    *              : protection/clearance differ to the specification of the serial
189    *              : flash memory.
190    * Argument    : enum sf_req req ; I : SF_REQ_UNPROTECT -> clear all sector protection
191    *              :                     SF_REQ_PROTECT   -> protect all sectors
192    * Return Value : void
193    * Note        : None
194    *****************************************************************************/void
195   sf_protect_ctrl_spibsc(enum sf_req req)
196   {
197       unsigned char st_reg1, st_reg2;
198       unsigned char cf_reg1, cf_reg2;
199
200       read_status(&st_reg1,&st_reg2);
201       read_config(&cf_reg1,&cf_reg2);
202
203       /* ==== Set value of Serial Flash(0) ==== */
204
205       /* ---- clear freeze bit in configuration register  ---- */
206       write_status( st_reg1 , (unsigned char)(cf_reg1 & (~CFREG_FREEZE_BIT)) );
207
208       if( req == SF_REQ_UNPROTECT ){
209           st_reg1 &= ~STREG_BPROTECT_BIT;    /* un-protect in all area   */
210       }
211       else{
212           st_reg1 |= STREG_BPROTECT_BIT;     /* protect in all area     */
213       }
214
215       /* ---- clear or set protect bit in status register ---- */
216       /* ---- with freeze bit in configuration register   ---- */
217       write_status( st_reg1 , (unsigned char)(cf_reg1 | CFREG_FREEZE_BIT) );
218   }
```

（省略）

3.9 サンプルプログラムリスト"qserial_flash_spibsc.c"（5）

```
303
304  /*****************************************************************************
305   * ID          :
306   * Outline     : Sector erase
307   * Include     :
308   * Declaration  : void sf_sector_erase_spibsc(int sector_no);
309   * Description  : Erase the specified sector in the serial flash memory
310   *              : A write enable command should be issued before erasing or programming.
311   *              : After erasing or programming, check the serial flash memory status
312   *              : with the busy state is cleared.
313   * Argument    : int sector_no ; I : sector number
314   * Return Value : void
315   * Note         : None
316   *****************************************************************************/
317  void sf_sector_erase_spibsc(int sector_no)
318  {
319      unsigned long addr = sector_no * SF_SECTOR_SIZE;
320
321  #if (SFLASH_DUAL == 1)
322      int bsz;
323
324      /*  set BE in both of serial-flash */
325      bsz = sf_bsz_get_spibsc();
326      sf_bsz_set_spibsc(2);                       /* s-flash x 2  */
327  #endif
328
329      /* sector erase in Single-SPI   */
330
331      write_enable();                             /* WREN Command */
332
333      SpibscSm.cdb = SPIBSC_1BIT;                /* Commmand bit-width = Single */
334      SpibscSm.adb = SPIBSC_1BIT;                /* Address bit-width = Single */
335
336      SpibscSm.cde = SPIBSC_OUTPUT_ENABLE;       /* Command Enable */
337      SpibscSm.ocde = SPIBSC_OUTPUT_DISABLE;     /* Optional-Command Disable */
338      SpibscSm.ade = SPIBSC_OUTPUT_ADDR_24;      /* Enable(Adr[23:0]) */
339      SpibscSm.opde = SPIBSC_OUTPUT_DISABLE;     /* Option-Data Disable */
340      SpibscSm.spide = SPIBSC_OUTPUT_DISABLE;    /* Disable */
341
342      SpibscSm.sslkp = SPIBSC_SPISSL_NEGATE;     /* Negate after transfer */
343      SpibscSm.spire = SPIBSC_SPIDATA_DISABLE;   /* Data Access (Read Disable) */
344      SpibscSm.spiwe = SPIBSC_SPIDATA_DISABLE;   /* Data Access (Write Disable) */
345
346      SpibscSm.cmd = SFLASHCMD_SECTOR_ERASE;     /* SE:Sector Erase */
347
348      SpibscSm.addr = addr;      /* dont care in dual mode   */
349                                 /* because address is calcurated with sector_no */
350
```

3.10 サンプルプログラムリスト"qserial_flash_spibsc.c"（6）

```
351      io_spibsc_transfer(&SpibscSm);
352
353      busy_wait();
354
355   #if (SFLASH_DUAL == 1)
356      sf_bsz_set_spibsc(bsz);
357   #endif
358   }
359
360   /*********************************************************************************
361    * ID          :
362    * Outline     : Data program
363    * Include     :
364    * Declaration  : void sf_byte_program_spibsc(unsigned long addr, unsigned char *buf, int size);
365    * Description  : Program the assigned program in the serial flash memory
366    *              : Erase the specified sector in the serial flash memory
367    *              : A write enable command should be issued before erasing or programming.
368    *              : After erasing or programming, check the serial flash memory status
369    *              : with the busy state is cleared.
370    *              : The maximum write data size is limited by the device.
371    * Argument    : unsigned long addr ; I : address in the serial flash memory to write to
372    *              : unsigned char *buf ; I : address of the buffer to store write data
373    *              : int size          ; I : number of byte to write
374    * Return Value : void
375    * Note        : None
376    *********************************************************************************/
377   void sf_byte_program_spibsc(unsigned long addr, unsigned char *buf, int size)
378   {
379      int unit;
380
381      write_enable();                              /* WREN Command */
382
383      /* ---- Command,Address ---- */
384      SpibscSm.cdb = SPIBSC_1BIT;                 /* Commmand bit-width = Single */
385      SpibscSm.adb = SPIBSC_1BIT;                 /* Address bit-width = Single */
386
387      SpibscSm.cde = SPIBSC_OUTPUT_ENABLE;        /* Command Enable */
388      SpibscSm.ocde = SPIBSC_OUTPUT_DISABLE;      /* Optional-Command Disable */
389      SpibscSm.ade = SPIBSC_OUTPUT_ADDR_24;       /* Enable Adr[23:0] */
390      SpibscSm.opde = SPIBSC_OUTPUT_DISABLE;      /* Option-Data Disable */
391      SpibscSm.spide = SPIBSC_OUTPUT_DISABLE;     /* Disable */
392
393      SpibscSm.sslkp = SPIBSC_SPISSL_KEEP;        /* Keep after transfer */
394      SpibscSm.spire = SPIBSC_SPIDATA_DISABLE;    /* Data Access (Read Disable) */
395      SpibscSm.spiwe = SPIBSC_SPIDATA_DISABLE;    /* Data Access (Write Disable) */
396
397   #if (SPI_QUAD == 0)
398      SpibscSm.cmd = SFLASHCMD_BYTE_PROGRAM;      /* PP: Page Program */
399   #else
400      SpibscSm.cmd = SFLASHCMD_QUAD_PROGRAM;      /* QPP: Quad Page Program */
401   #endif
```

3.11 サンプルプログラムリスト"qserial_flash_spibsc.c"（7）

```
402
403      if(io_spibsc_bsz_get() == SPIBSC_CMNCR_BSZ_DUAL){
404          SpibscSm.addr = (unsigned long)(addr >> 1);
405      }
406      else{
407          SpibscSm.addr = addr;
408      }
409
410      io_spibsc_transfer(&SpibscSm);              /* Command,Address */
411
412      /* ---- Data ---- */
413  #if (SPI_QUAD == 0)
414      SpibscSm.spidb = SPIBSC_1BIT;               /* Single */
415  #else
416      SpibscSm.spidb = SPIBSC_4BIT;               /* Quad */
417  #endif
418
419      SpibscSm.cde = SPIBSC_OUTPUT_DISABLE;      /* Command Disable */
420      SpibscSm.ocde = SPIBSC_OUTPUT_DISABLE;     /* Optional-Command Disable */
421      SpibscSm.ade = SPIBSC_OUTPUT_DISABLE;      /* Disable Adr */
422      SpibscSm.opde = SPIBSC_OUTPUT_DISABLE;     /* Option-Data Disable */
423
424      SpibscSm.spire = SPIBSC_SPIDATA_DISABLE;    /* Data Access (Read Disable) */
425      SpibscSm.spiwe = SPIBSC_SPIDATA_ENABLE;     /* Data Access (Write Ensable) */
426
427      if(io_spibsc_bsz_get() == SPIBSC_CMNCR_BSZ_DUAL){
428          if((size % 8) == 0){
429              SpibscSm.spide = SPIBSC_OUTPUT_SPID_32; /* Enable(64bit) */
430              unit = 8;
431          }
432          else if((size % 4) == 0){
433              SpibscSm.spide = SPIBSC_OUTPUT_SPID_16; /* Enable(32bit) */
434              unit = 4;
435          }
436          else if((size % 2) == 0){
437              SpibscSm.spide = SPIBSC_OUTPUT_SPID_8;  /* Enable(16bit) */
438              unit = 2;
439          }
440          else{
441              return;
442          }
443      }
```

3.12 サンプルプログラムリスト"qserial_flash_spibsc.c"（8）

```
444      else{
445         if((size % 4) == 0){
446            SpibscSm.spide = SPIBSC_OUTPUT_SPID_32; /* Enable(32bit) */
447            unit = 4;
448         }
449         else if((size % 2) == 0){
450            SpibscSm.spide = SPIBSC_OUTPUT_SPID_16; /* Enable(16bit) */
451            unit = 2;
452         }
453         else{
454            SpibscSm.spide = SPIBSC_OUTPUT_SPID_8;  /* Enable(8bit) */
455            unit = 1;
456         }
457      }
458
459      while(size > 0){
460         SpibscSm.smwdr[0] = (unsigned long)(((unsigned long)*buf++ << 24) & 0xff000000ul);
461                                          /* Data[63:56] or Data[31:24] */
462         if(unit >= 2){
463            SpibscSm.smwdr[0] |= (unsigned long)(((unsigned long)*buf++ << 16) & 0x00ff0000ul);
464                                          /* Data[55:48] or Data[23:16] */
465         }
466         if(unit >= 4){
467            SpibscSm.smwdr[0] |= (unsigned long)(
468                              (((unsigned long)*buf++ << 8 ) & 0x0000ff00ul) |
469                              (((unsigned long)*buf++     ) & 0x000000ffful));
470                                          /* Data[47:40] or Data[15:0] */
471         }
472         if(unit >= 8){
473            SpibscSm.smwdr[1] = (unsigned long)(
474                              (((unsigned long)*buf++ << 24) & 0xff000000ul) |
475                              (((unsigned long)*buf++ << 16) & 0x00ff0000ul) |
476                              (((unsigned long)*buf++ << 8 ) & 0x0000ff00ul) |
477                              (((unsigned long)*buf++     ) & 0x000000ffful));
478                                          /*Data[31: 0] or nothing */
479         }
480
481         size -= unit;
482         if(size <= 0){
483            SpibscSm.sslkp = SPIBSC_SPISSL_NEGATE;
484         }
485         io_spibsc_transfer(&SpibscSm);        /* Data */
486      }
487
488      busy_wait();
489
490   }
491
```

3.13 サンプルプログラムリスト"qserial_flash_spibsc.c"（9）

```
492   /***********************************************************************
493    * ID         :
494    * Outline     : Data read
495    * Include     :
496    * Declaration  : void sf_byte_read_spibsc(unsigned long addr, unsigned char *buf, int size);
497    * Description  : Read the serial memory by the specified number of byte
498    * Argument    : unsigned long addr ; I : address of the serial flash memory to read
499    *            : unsigned char *buf ; I : address of the buffer to store the read data
500    *            : int size         ; I : number of byte to read
501    * Return Value : void
502    * Note        : None
503    ***********************************************************************/
504   #if (SPI_QUAD == 0)


         （省略）



605
606   #else
607
608   void sf_byte_read_spibsc(unsigned long addr, unsigned char *buf, int size)
609   {
610       int unit;
611
612       if(io_spibsc_bsz_get() == SPIBSC_CMNCR_BSZ_DUAL){
613           if((size % 8) == 0){
614               unit = 8;
615           }
616           else if((size % 4) == 0){
617               unit = 4;
618           }
619           else if((size % 2) == 0){
620               unit = 2;
621           }
622           else{
623               return;
624           }
625       }
626       else{
627           if((size % 4) == 0){
628               unit = 4;
629           }
630           else if((size % 2) == 0){
631               unit = 2;
632           }
633           else{
634               unit = 1;
635           }
636       }
637
```

3.14 サンプルプログラムリスト"qserial_flash_spibsc.c"（10）

```
638       while(size > 0){
639          sf_byte_read_spibsc_quad(addr, buf, unit);
640
641          /* increment address and buf */
642          addr += unit;
643          buf  += unit;
644
645          size -= unit;
646       }
647    }
648
649    static void sf_byte_read_spibsc_quad(unsigned long addr, unsigned char *buf, int unit)
650    {
651       /* ---- Command,Address,Dummy ---- */
652       SpibscSm.cdb = SPIBSC_1BIT;                /* Commmand bit-width = Single */
653       SpibscSm.adb = SPIBSC_1BIT;                /* Address bit-width = Single */
654
655       SpibscSm.cde = SPIBSC_OUTPUT_ENABLE;       /* Command Enable */
656       SpibscSm.ocde = SPIBSC_OUTPUT_DISABLE;     /* Optional-Command Disable */
657       SpibscSm.ade = SPIBSC_OUTPUT_ADDR_24;      /* Enable Adr[23:0] */
658       SpibscSm.opde = SPIBSC_OUTPUT_OPD_3;       /* Option-Data OPD3 */
659       SpibscSm.spide = SPIBSC_OUTPUT_DISABLE;    /* Disable */
660
661       SpibscSm.sslkp = SPIBSC_SPISSL_KEEP;       /* Keep after transfer */
662       SpibscSm.spire = SPIBSC_SPIDATA_DISABLE;   /* Data Access (Read Disable) */
663       SpibscSm.spiwe = SPIBSC_SPIDATA_DISABLE;   /* Data Access (Write Disable) */
664
665       SpibscSm.cmd = SFLASHCMD_QUAD_READ;        /* QOR: Quad Output Read */
666
667       if(io_spibsc_bsz_get() == SPIBSC_CMNCR_BSZ_DUAL){
668          SpibscSm.addr = (unsigned long)(addr >> 1);
669       }
670       else{
671          SpibscSm.addr = addr;
672       }
673
674       io_spibsc_transfer(&SpibscSm);            /* Command,Address */
675
676
677       /* ---- Data ---- */
678       SpibscSm.spidb = SPIBSC_4BIT;              /* Quad */
679
680       SpibscSm.cde = SPIBSC_OUTPUT_DISABLE;      /* Command Disable */
681       SpibscSm.ocde = SPIBSC_OUTPUT_DISABLE;     /* Optional-Command Disable */
682       SpibscSm.ade = SPIBSC_OUTPUT_DISABLE;      /* Disable Adr */
683       SpibscSm.opde = SPIBSC_OUTPUT_DISABLE;     /* Option-Data Disable */
684
685       SpibscSm.spire = SPIBSC_SPIDATA_ENABLE;    /* Data Access (Read Enable) */
686       SpibscSm.spiwe = SPIBSC_SPIDATA_DISABLE;   /* Data Access (Write Disable) */
687
```

3.15 サンプルプログラムリスト"qserial_flash_spibsc.c"（11）

```
688       if(io_spibsc_bsz_get() == SPIBSC_CMNCR_BSZ_DUAL){
689          if( unit == 8 ){
690             SpibscSm.spide = SPIBSC_OUTPUT_SPID_32; /* Enable(64bit) */
691          }
692          else if( unit == 4 ){
693             SpibscSm.spide = SPIBSC_OUTPUT_SPID_16; /* Enable(32bit) */
694          }
695          else if( unit == 2 ){
696             SpibscSm.spide = SPIBSC_OUTPUT_SPID_8;  /* Enable(16bit) */
697          }
698          else{
699             return;
700          }
701       }
702       else{
703          if( unit == 4 ){
704             SpibscSm.spide = SPIBSC_OUTPUT_SPID_32; /* Enable(32bit) */
705          }
706          else if( unit == 2 ){
707             SpibscSm.spide = SPIBSC_OUTPUT_SPID_16; /* Enable(16bit) */
708          }
709          else{
710             SpibscSm.spide = SPIBSC_OUTPUT_SPID_8;  /* Enable(8bit) */
711          }
712       }
713
714       SpibscSm.sslkp = SPIBSC_SPISSL_NEGATE;
715       io_spibsc_transfer(&SpibscSm);                /* Data 入力 */
716
717       *buf++ = (unsigned char)((SpibscSm.smrdr[0] >> 24) & 0x000000fful);
718                                                  /* Data[63:56],Data[31:24] */
719       if(unit >= 2){
720          *buf++ = (unsigned char)((SpibscSm.smrdr[0] >> 16) & 0x000000fful);
721                                                  /* Data[55:48],Data[23:16] */
722       }
723       if(unit >= 4){
724          *buf++ = (unsigned char)((SpibscSm.smrdr[0] >> 8 ) & 0x000000fful);
725          *buf++ = (unsigned char)((SpibscSm.smrdr[0]     ) & 0x000000fful);
726                                                  /* Data[47:40],Data[15:0] */
727       }
728       if(unit >= 8){
729          *buf++ = (unsigned char)((SpibscSm.smrdr[1] >> 24) & 0x000000fful);
730          *buf++ = (unsigned char)((SpibscSm.smrdr[1] >> 16) & 0x000000fful);
731          *buf++ = (unsigned char)((SpibscSm.smrdr[1] >> 8 ) & 0x000000fful);
732          *buf++ = (unsigned char)((SpibscSm.smrdr[1]     ) & 0x000000fful);
733                                                  /*Data[31:0] */
734       }
735    }
736    #endif
737
```

3.16 サンプルプログラムリスト"qserial_flash_spibsc.c"（12）

```
          （省略）


/*****************************************************************************
 * ID         :
 * Outline     : Enable writing
 * Include     :
 * Declaration  : static void write_enable(void);
 * Description  : Issuing the write enable command to permit to erase/program
 *            : in the serial flash memory
 * Argument    : void
 * Return Value : void
 * Note        : None
 *****************************************************************************/
static void write_enable(void)
{
    SpibscSm.cdb = SPIBSC_1BIT;              /* Single */

    SpibscSm.cde = SPIBSC_OUTPUT_ENABLE;       /* Command Enable */
    SpibscSm.ocde = SPIBSC_OUTPUT_DISABLE;     /* Optional-Command Disable */
    SpibscSm.ade = SPIBSC_OUTPUT_DISABLE;      /* Address Disable */
    SpibscSm.opde = SPIBSC_OUTPUT_DISABLE;     /* Option-Data Disable */
    SpibscSm.spide = SPIBSC_OUTPUT_DISABLE;    /* Disable */

    SpibscSm.sslkp = SPIBSC_SPISSL_NEGATE;     /* Negate after transfer */
    SpibscSm.spire = SPIBSC_SPIDATA_DISABLE;   /* Data Access (Read Disable) */
    SpibscSm.spiwe = SPIBSC_SPIDATA_DISABLE;   /* Data Access (Write Disable) */

    SpibscSm.cmd = SFLASHCMD_WRITE_ENABLE;     /* WREN:Write Enable */

    io_spibsc_transfer(&SpibscSm);
}


          （以下、省略）
```

3.17 サンプルプログラムリスト"qserial_flash_spibsc.h"（1）

```
1     /***********************************************************************
2     *   DISCLAIMER

      （省略）

27    ***********************************************************************
28    *   Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29    *************************** Technical reference data *********************
30    *   System Name : SH7268/SH7269 Firm Update Sample Program
31    *   File Name   : qserial_flash_spibsc.h
32    *   Abstract    :
33    *   Version     : 1.00.00
34    *   Device      : SH7268/SH7269
35    *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36    *               : C/C++ compiler package for the SuperH RISC engine family
37    *               :                          (Ver.9.03Release02).
38    *   OS          : None
39    *   H/W Platform: R0K57269(CPU board)
40    *   Description :
41    ***********************************************************************
42    *   History     : Jul.06,2011 Ver.1.00.00
43    ***********************************************************************/
44    #ifndef _QSERIAL_FLASH_SPIBSC_H_
45    #define _QSERIAL_FLASH_SPIBSC_H_
46
47    /* ==== Function prototype declaration ==== */
48    int sf_bsz_get_spibsc(void);
49    void sf_bsz_set_spibsc(int bsz);
50    void sf_allocate_exspace_spibsc (void);
51    void sf_init_serial_flash_spibsc(void);
52    void sf_protect_ctrl_spibsc(enum sf_req req);
53    void sf_chip_erase_spibsc(void);
54    void sf_sector_erase_spibsc(int sector_no);
55    void sf_byte_program_spibsc(unsigned long addr, unsigned char *buf, int size);
56    void sf_byte_read_spibsc(unsigned long addr, unsigned char *buf, int size);
57
58    #endif /* _QSERIAL_FLASH_SPIBSC_H_ */
59    /* End of File */
60
```

3.18　サンプルプログラムリスト"io_spibsc.c"（1）

```
1     /***********************************************************************
2     *   DISCLAIMER
3     *
4     *   This software is supplied by Renesas Electronics Corporation and is only
5     *   intended for use with Renesas products. No other uses are authorized.
6     *
7     *   This software is owned by Renesas Electronics Corporation and is protected under
8     *   all applicable laws, including copyright laws.
9     *
10    *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11    *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12    *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13    *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14    *   DISCLAIMED.
15    *
16    *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17    *   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18    *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19    *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20    *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21    *
22    *   Renesas reserves the right, without notice, to make changes to this
23    *   software and to discontinue the availability of this software.
24    *   By using this software, you agree to the additional terms and
25    *   conditions found by accessing the following link:
26    *   http://www.renesas.com/disclaimer
27    ************************************************************************
28    *   Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29    *************************** Technical reference data *************************
30    *   System Name : SH7268/SH7269 Firm Update Sample Program
31    *   File Name   : io_spibsc.c
32    *   Abstract    : loader program for spibsc
33    *   Version     : 1.00.00
34    *   Device      : SH7268/SH7269
35    *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36    *               : C/C++ compiler package for the SuperH RISC engine family
37    *               :                          (Ver.9.03Release02).
38    *   OS          : None
39    *   H/W Platform: R0K57269(CPU board)
40    *   Description :
41    ************************************************************************
42    *   History     : Jul.06,2011 Ver.1.00.00
43    ************************************************************************/
44    #include "iodefine.h"
45    #include "io_spibsc.h"
46    #include "machine.h"
47
```

3.19 サンプルプログラムリスト"io_spibsc.c"（2）

```
48    #pragma section SPIBSC
49
50    /* ==== define values ==== */
51
52    /* ==== Prototype Declaration ==== */
53
54    /* ==== Global variable ==== */
55
56    /*******************************************************************************
57     * ID          :
58     * Outline     :
59     * Include     : io_spibsc.h
60     * Declaration  : int io_spibsc_bsz_set(unsigned long bsz);
61     * Description  :
62     * Argument    : unsigned long bsz : BSZ bit
63     * Return Value :
64     * Note        : None
65     *******************************************************************************/
66    int io_spibsc_bsz_set(unsigned long bsz)
67    {
68      if(SPIBSC.CMNSR.BIT.SSLF != SPIBSC_SSL_NEGATE){
69        return -1;
70      }
71      if(SPIBSC.CMNCR.BIT.BSZ != bsz){
72        if(bsz == SPIBSC_CMNCR_BSZ_DUAL){
73            /* s-flash x 2 (4bit x 2)  */
74            PORT.PBCR3.BIT.PB14MD = 6; /* PB14:SPBIO3_1 */
75            PORT.PBCR3.BIT.PB13MD = 6; /* PB13:SPBIO2_1 */
76            PORT.PFCR0.BIT.PF3MD = 6;  /* PF3:SPBMI_1/SPBIO1_1 */
77            PORT.PFCR0.BIT.PF2MD = 6;  /* PF2:SPBMO_1/SPBIO0_1 */
78        }
79        SPIBSC.CMNCR.BIT.BSZ = bsz;
80        SPIBSC.DRCR.BIT.RCF = SPIBSC_DRCR_RCF_EXE;    /* flush read-cache    */
81      }
82      return 0;
83    }
84
```

3.20 サンプルプログラムリスト"io_spibsc.c"（3）

```
85    /***********************************************************************
86     * ID          :
87     * Outline     :
88     * Include     : io_spibsc.h
89     * Declaration  : unsigned long io_spibsc_bsz_get(void);
90     * Description  :
91     * Argument    : void
92     * Return Value : BSZ bit
93     * Note        : None
94     ***********************************************************************/
95    unsigned long io_spibsc_bsz_get(void)
96    {
97      return (unsigned long)SPIBSC.CMNCR.BIT.BSZ;
98    }
99
100   /***********************************************************************
101    * ID          :
102    * Outline     :
103    * Include     : io_spibsc.h
104    * Declaration  : int io_spibsc_common_init(unsigned long bsz);
105    * Description  :
106    * Argument    : unsigned long bsz : BSZ bit
107    * Return Value :
108    * Note        : None
109    ***********************************************************************/
110   int io_spibsc_common_init(unsigned long bsz)
111   {
112     CPG.STBCR7.BIT.MSTP75 = 0;
113
114     PORT.PBCR5.BIT.PB20MD = 6;  /* PB20:SPBMI_0/SPBIO1_0  */
115     PORT.PBCR4.BIT.PB19MD = 6;  /* PB19:SPBMO_0/SPBIO0_0  */
116     PORT.PBCR4.BIT.PB18MD = 6;  /* PB18:SPBSSL  */
117     PORT.PBCR4.BIT.PB17MD = 6;  /* PB17:SPBCLK */
118     PORT.PBCR4.BIT.PB16MD = 6;  /* PB16:SPBIO3_0 */
119     PORT.PBCR3.BIT.PB15MD = 6;  /* PB15:SPBIO2_0 */
120
121     if(bsz == SPIBSC_CMNCR_BSZ_DUAL){
122       /* s-flash x 2 (4bit x 2)  */
123       PORT.PBCR3.BIT.PB14MD = 6; /* PB14:SPBIO3_1 */
124       PORT.PBCR3.BIT.PB13MD = 6; /* PB13:SPBIO2_1 */
125       PORT.PFCR0.BIT.PF3MD = 6;  /* PF3:SPBMI_1/SPBIO1_1 */
126       PORT.PFCR0.BIT.PF2MD = 6;  /* PF2:SPBMO_1/SPBIO0_1 */
127     }
128
129     if(SPIBSC.CMNSR.BIT.SSLF != SPIBSC_SSL_NEGATE){
130       return -1;
131     }
132
```

3.21 サンプルプログラムリスト"io_spibsc.c"（4）

```
133     SPIBSC.CMNCR.BIT.MOIIO3 = SPIBSC_OUTPUT_HiZ;
134     SPIBSC.CMNCR.BIT.MOIIO2 = SPIBSC_OUTPUT_HiZ;
135     SPIBSC.CMNCR.BIT.MOIIO1 = SPIBSC_OUTPUT_HiZ;
136     SPIBSC.CMNCR.BIT.MOIIO0 = SPIBSC_OUTPUT_HiZ;
137
138     SPIBSC.CMNCR.BIT.IO3FV = SPIBSC_OUTPUT_HiZ;
139     SPIBSC.CMNCR.BIT.IO2FV = SPIBSC_OUTPUT_HiZ;
140     SPIBSC.CMNCR.BIT.IO0FV = SPIBSC_OUTPUT_HiZ;
141
142     /* S-flash mode 0    */
143     SPIBSC.CMNCR.BIT.CPHAT = SPIBSC_CMNCR_CPHAT_EVEN;
144                                             /* even edge : write */
145     SPIBSC.CMNCR.BIT.CPHAR = SPIBSC_CMNCR_CPHAR_EVEN;
146                                             /* even edge : read */
147     SPIBSC.CMNCR.BIT.SSLP = SPIBSC_CMNCR_SSLP_LOW;
148                                             /* SPBSSL：low active */
149     SPIBSC.CMNCR.BIT.CPOL = SPIBSC_CMNCR_CPOL_LOW;
150                                             /* SPBCLK：low at negate */
151
152     io_spibsc_bsz_set(bsz);
153
154     SPIBSC.SSLDR.BIT.SPNDL = SPIBSC_DELAY_1SPBCLK;
155                                             /* next access delay */
156     SPIBSC.SSLDR.BIT.SLNDL = SPIBSC_DELAY_1SPBCLK;
157                                             /* SPBSSL negate delay */
158     SPIBSC.SSLDR.BIT.SCKDL = SPIBSC_DELAY_1SPBCLK;
159                                             /* clock delay */
160
161     /* ---- Bit rate 66.67Mbps ---- */
162     SPIBSC.SPBCR.BIT.SPBR = 1;                  /* divide 2 base bit rate B clock(133.33MHz) */
163     SPIBSC.SPBCR.BIT.BRDV = 0;
164
165     return 0;
166   }
167
```

3.22 サンプルプログラムリスト"io_spibsc.c"（5）

```
168   /***********************************************************************
169    * ID          :
170    * Outline     :
171    * Include     : io_spibsc.h
172    * Declaration  : int io_spibsc_dr_init(unsigned long cmd);
173    * Description  :
174    * Argument    : void
175    * Return Value :
176    * Note        : None
177    ***********************************************************************/
178   int io_spibsc_dr_init(unsigned long cmd)
179   {
180     if(SPIBSC.CMNSR.BIT.SSLF != SPIBSC_SSL_NEGATE){
181       return -1;
182     }
183
184     SPIBSC.CMNCR.BIT.MD = SPIBSC_CMNCR_MD_EXTRD;   /* SPI I/O mode*/
185
186     SPIBSC.DRCR.BIT.RBURST = SPIBSC_BURST_2;
187     SPIBSC.DRCR.BIT.RBE = SPIBSC_BURST_ENABLE;
188     SPIBSC.DRCR.BIT.SSLE = SPIBSC_SPISSL_KEEP;     /* Keep SSL after read */
189                                 /* if not continuous address it negeted   */
190
191     if(SPIBSC.CMNSR.BIT.TEND != SPIBSC_TRANS_END){
192       return -1;
193     }
194
195     /* ---- Command ---- */
196     SPIBSC.DRCMR.BIT.CMD = cmd;                /* Command */
197     SPIBSC.DRENR.BIT.CDB = SPIBSC_1BIT;        /* Single */
198     SPIBSC.DRENR.BIT.CDE = SPIBSC_OUTPUT_ENABLE;
199                                             /* Enable */
200     /* ---- Option Command ---- */
201     SPIBSC.DRCMR.BIT.OCMD = 0x00;
202     SPIBSC.DRENR.BIT.OCDB = SPIBSC_1BIT;       /* Single */
203     SPIBSC.DRENR.BIT.OCDE = SPIBSC_OUTPUT_DISABLE;
204                                             /* Disable */
205
206     /* ---- Address ---- */
207     if(cmd == 0xBB){
208                                             /* Dual I/O High Performance */
209       SPIBSC.DRENR.BIT.ADB = SPIBSC_2BIT;      /* Dual */
210       SPIBSC.DRENR.BIT.ADE = SPIBSC_OUTPUT_ADDR_24;
211                                             /* S-flash x 1 Enable(ADDR[23:0]) */
212                                             /* S-flash x 2 Enable(ADDR[24:1]) */
213     }
```

3.23 サンプルプログラムリスト"io_spibsc.c"（6）

```
214     else if(cmd == 0xEB){
215                                             /* Quad I/O High Performance */
216       SPIBSC.DRENR.BIT.ADB = SPIBSC_4BIT;   /* Quad */
217       SPIBSC.DRENR.BIT.ADE = SPIBSC_OUTPUT_ADDR_24;
218                                             /* S-flash x 1 Enable(ADDR[23:0]) */
219                                             /* S-flash x 2 Enable(ADDR[24:1]) */
220     }
221     else{
222       SPIBSC.DRENR.BIT.ADB = SPIBSC_1BIT;   /* Single */
223       SPIBSC.DRENR.BIT.ADE = SPIBSC_OUTPUT_ADDR_24;
224                                             /* S-flash x 1 Enable(ADDR[23:0]) */
225                                             /* S-flash x 2 Enable(ADDR[24:1]) */
226     }
227
228     /* ---- Option Data ---- */
229     if(cmd == 0xBB){
230                                             /* Dual I/O High Performance */
231       SPIBSC.DROPR.BIT.OPD3 = 0x00;         /* Option Data(Mode bit) */
232       SPIBSC.DROPR.BIT.OPD2 = 0x00;         /* Option Data */
233       SPIBSC.DROPR.BIT.OPD1 = 0x00;         /* Option Data */
234       SPIBSC.DROPR.BIT.OPD0 = 0x00;         /* Option Data */
235       SPIBSC.DRENR.BIT.OPDB = SPIBSC_2BIT;  /* Dual */
236       SPIBSC.DRENR.BIT.OPDE = SPIBSC_OUTPUT_OPD_3;
237                                             /* Enable(OPD3) */
238     }
239     else if(cmd == 0xEB){
240                                             /* Quad I/O High Performance */
241       SPIBSC.DROPR.BIT.OPD3 = 0x00;         /* Option Data(Mode bit) */
242       SPIBSC.DROPR.BIT.OPD2 = 0x00;         /* Option Data(Dummy) */
243       SPIBSC.DROPR.BIT.OPD1 = 0x00;         /* Option Data(Dummy) */
244       SPIBSC.DROPR.BIT.OPD0 = 0x00;         /* Option Data */
245       SPIBSC.DRENR.BIT.OPDB = SPIBSC_4BIT;  /* Quad */
246       SPIBSC.DRENR.BIT.OPDE = SPIBSC_OUTPUT_OPD_321;
247                                             /* Enable(OPD3,OPD2,OPD1) */
248     }
249     else{
250       SPIBSC.DROPR.BIT.OPD3 = 0x00;         /* Option Data(Dummy) */
251       SPIBSC.DROPR.BIT.OPD2 = 0x00;         /* Option Data */
252       SPIBSC.DROPR.BIT.OPD1 = 0x00;         /* Option Data */
253       SPIBSC.DROPR.BIT.OPD0 = 0x00;         /* Option Data */
254       SPIBSC.DRENR.BIT.OPDB = SPIBSC_1BIT;  /* Single */
255       SPIBSC.DRENR.BIT.OPDE = SPIBSC_OUTPUT_OPD_3;
256                                             /* Enable(OPD3) */
257     }
```

3.24 サンプルプログラムリスト"io_spibsc.c"（7）

```
258     /* ---- Data ---- */
259     if(cmd == 0x6B){
260       SPIBSC.DRENR.BIT.DRDB = SPIBSC_4BIT;      /* Quad */
261     }
262     else if(cmd == 0x3B){
263       SPIBSC.DRENR.BIT.DRDB = SPIBSC_2BIT;      /* Dual */
264     }
265     else if(cmd == 0x0B){
266       SPIBSC.DRENR.BIT.DRDB = SPIBSC_1BIT;      /* Single */
267     }
268     else if(cmd == 0xBB){
269       SPIBSC.DRENR.BIT.DRDB = SPIBSC_2BIT;      /* Dual I/O High Performance */
270     }
271     else if(cmd == 0xEB){
272       SPIBSC.DRENR.BIT.DRDB = SPIBSC_4BIT;      /* Quad I/O High Performance */
273     }
274     else{
275       return -1;
276     }
277     return 0;
278   }
279
280   /***********************************************************************
281    * ID           :
282    * Outline      :
283    * Include      : io_spibsc.h
284    * Declaration  : int io_spibsc_transfer(ST_SPIBSC_SM *SpibscSm);
285    * Description  :
286    * Argument     : void
287    * Return Value :
288    * Note         : None
289    ***********************************************************************/
290   int io_spibsc_transfer(ST_SPIBSC_SM *SpibscSm)
291   {
292     int i;
293     volatile unsigned long dummy;
294
295     if(SPIBSC.CMNCR.BIT.MD != SPIBSC_CMNCR_MD_SPI){
296       if(SPIBSC.CMNSR.BIT.SSLF != SPIBSC_SSL_NEGATE){
297           return -1;
298       }
299       SPIBSC.CMNCR.BIT.MD = SPIBSC_CMNCR_MD_SPI;
300                                                 /* SPI Mode */
301     }
302
303     if(SPIBSC.CMNSR.BIT.TEND != SPIBSC_TRANS_END){
304       return -1;
305     }
306
```

3.25 サンプルプログラムリスト"io_spibsc.c"（8）

```
307       /* ---- Command ---- */
308       SPIBSC.SMENR.BIT.CDE = SpibscSm->cde;      /* Enable/Disable */
309       if(SpibscSm->cde != SPIBSC_OUTPUT_DISABLE){
310          SPIBSC.SMCMR.BIT.CMD = SpibscSm->cmd;   /* Command */
311          SPIBSC.SMENR.BIT.CDB = SpibscSm->cdb;   /* Single/Dual/Quad */
312       }
313
314       /* ---- Option Command ---- */
315       SPIBSC.SMENR.BIT.OCDE = SpibscSm->ocde;    /* Enable/Disable */
316       if(SpibscSm->ocde != SPIBSC_OUTPUT_DISABLE){
317          SPIBSC.SMCMR.BIT.OCMD = SpibscSm->ocmd; /* Option Command */
318          SPIBSC.SMENR.BIT.OCDB = SpibscSm->ocdb; /* Single/Dual/Quad */
319       }
320
321       /* ---- Address ---- */
322       SPIBSC.SMENR.BIT.ADE = SpibscSm->ade;         /* Enable/Disable */
323       if(SpibscSm->ade != SPIBSC_OUTPUT_DISABLE){
324          SPIBSC.SMADR.BIT.ADR = SpibscSm->addr;     /* Address */
325          SPIBSC.SMENR.BIT.ADB = SpibscSm->adb;      /* Single/Dual/Quad */
326       }
327
328       /* ---- Option Data ---- */
329       SPIBSC.SMENR.BIT.OPDE = SpibscSm->opde;       /* Enable/Disable */
330       if(SpibscSm->opde != SPIBSC_OUTPUT_DISABLE){
331          SPIBSC.SMOPR.BIT.OPD3 = SpibscSm->opd[0];   /* Option Data */
332          SPIBSC.SMOPR.BIT.OPD2 = SpibscSm->opd[1];   /* Option Data */
333          SPIBSC.SMOPR.BIT.OPD1 = SpibscSm->opd[2];   /* Option Data */
334          SPIBSC.SMOPR.BIT.OPD0 = SpibscSm->opd[3];   /* Option Data */
335          SPIBSC.SMENR.BIT.OPDB = SpibscSm->opdb;    /* Single/Dual/Quad */
336       }
337
338       /* ---- Data ---- */
339       SPIBSC.SMENR.BIT.SPIDE = SpibscSm->spide;      /* Enable/Disable */
340       if(SpibscSm->spide != SPIBSC_OUTPUT_DISABLE){
341          SPIBSC.SMWDR0.LONG = SpibscSm->smwdr[0];
342          SPIBSC.SMWDR1.LONG = SpibscSm->smwdr[1];    /* シリフラ 2 個接続時のみ有効 */
343          SPIBSC.SMENR.BIT.SPIDB = SpibscSm->spidb;   /* Single/Dual/Quad */
344       }
345
346       SPIBSC.SMCR.BIT.SSLKP = SpibscSm->sslkp;
347
348       if((SpibscSm->spidb != SPIBSC_1BIT) && (SpibscSm->spide != SPIBSC_OUTPUT_DISABLE)){
349          if((SpibscSm->spire  == SPIBSC_SPIDATA_ENABLE) &&
                (SpibscSm->spiwe  == SPIBSC_SPIDATA_ENABLE)){
350           /* not set in same time */
351           return -1;
352        }
353      }
```

3.26　サンプルプログラムリスト"io_spibsc.c"（9）

```
354     SPIBSC.SMCR.BIT.SPIRE = SpibscSm->spire;
355     SPIBSC.SMCR.BIT.SPIWE = SpibscSm->spiwe;
356
357     SPIBSC.SMCR.BIT.SPIE = SPIBSC_SPI_ENABLE;      /* execute after setting SPNDL bit */
358
359     /* wait for transfer-start */
360     dummy = SPIBSC.CMNSR.LONG;
361     dummy = SPIBSC.CMNSR.LONG;
362     dummy = SPIBSC.CMNSR.LONG;
363     dummy = SPIBSC.CMNSR.LONG;
364
365     while(SPIBSC.CMNSR.BIT.TEND != SPIBSC_TRANS_END){
366       /* wait for transfer-end */
367     }
368     SpibscSm->smrdr[0] = SPIBSC.SMRDR0.LONG;
369     SpibscSm->smrdr[1] = SPIBSC.SMRDR1.LONG;       /* valid in two serial-flash  */
370
371     return 0;
372   }
373
374   /* End of File */
375
```

3.27　サンプルプログラムリスト"io_spibsc.h"（1）

```
1     /***********************************************************************************
2     *   DISCLAIMER

       （省略）

27    ***********************************************************************************
28    *   Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29    *************************** Technical reference data ************************
30    *   System Name : SH7268/SH7269 Firm Update Sample Program
31    *   File Name   : io_spibsc.h
32    *   Abstract    : spibsc structure
33    *   Version     : 1.00.00
34    *   Device      : SH7268/SH7269
35    *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36    *               : C/C++ compiler package for the SuperH RISC engine family
37    *               :                          (Ver.9.03Release02).
38    *   OS          : None
39    *   H/W Platform: R0K57269(CPU board)
40    *   Description :
41    ***********************************************************************************
42    *   History     : Jul.06,2011 Ver.1.00.00
43    ***********************************************************************************/
44    #ifndef _IO_SPIBSC_H_
45    #define _IO_SPIBSC_H_
46
47    /* ==== define values ==== */
48    #define SPIBSC_CMNCR_MD_EXTRD     0
49    #define SPIBSC_CMNCR_MD_SPI       1
50
51    #define SPIBSC_OUTPUT_LOW         0
52    #define SPIBSC_OUTPUT_HIGH        1
53    #define SPIBSC_OUTPUT_LAST        2
54    #define SPIBSC_OUTPUT_HiZ         3
55
56    #define SPIBSC_CMNCR_CPHAT_EVEN   0
57    #define SPIBSC_CMNCR_CPHAT_ODD
58
59    #define SPIBSC_CMNCR_CPHAR_ODD    0
60    #define SPIBSC_CMNCR_CPHAR_EVEN   1
61
62    #define SPIBSC_CMNCR_SSLP_LOW     0
63    #define SPIBSC_CMNCR_SSLP_HIGH    1
64
65    #define SPIBSC_CMNCR_CPOL_LOW     0
66    #define SPIBSC_CMNCR_CPOL_HIGH    1
67
68    #define SPIBSC_CMNCR_BSZ_SINGLE       0
69    #define SPIBSC_CMNCR_BSZ_DUAL     1
70
```

3.28　サンプルプログラムリスト"io_spibsc.h"（2）

```
71    #define SPIBSC_DELAY_1SPBCLK      0
72    #define SPIBSC_DELAY_2SPBCLK      1
73    #define SPIBSC_DELAY_3SPBCLK      2
74    #define SPIBSC_DELAY_4SPBCLK      3
75    #define SPIBSC_DELAY_5SPBCLK      4
76    #define SPIBSC_DELAY_6SPBCLK      5
77    #define SPIBSC_DELAY_7SPBCLK      6
78    #define SPIBSC_DELAY_8SPBCLK      7
79
80
81    #define SPIBSC_BURST_1            0x00
82    #define SPIBSC_BURST_2            0x01
83    #define SPIBSC_BURST_3            0x02
84    #define SPIBSC_BURST_4            0x03
85    #define SPIBSC_BURST_5            0x04
86    #define SPIBSC_BURST_6            0x05
87    #define SPIBSC_BURST_7            0x06
88    #define SPIBSC_BURST_8            0x07
89    #define SPIBSC_BURST_9            0x08
90    #define SPIBSC_BURST_10           0x09
91    #define SPIBSC_BURST_11           0x0a
92    #define SPIBSC_BURST_12           0x0b
93    #define SPIBSC_BURST_13           0x0c
94    #define SPIBSC_BURST_14           0x0d
95    #define SPIBSC_BURST_15           0x0e
96    #define SPIBSC_BURST_16           0x0f
97
98    #define SPIBSC_BURST_DISABLE      0
99    #define SPIBSC_BURST_ENABLE       1
100
101   #define SPIBSC_DRCR_RCF_EXE       1
102
103   #define SPIBSC_SSL_NEGATE         0
104   #define SPIBSC_TRANS_END          1
105
106   #define SPIBSC_1BIT               0
107   #define SPIBSC_2BIT               1
108   #define SPIBSC_4BIT               2
109
110   #define SPIBSC_OUTPUT_DISABLE     0
111   #define SPIBSC_OUTPUT_ENABLE      1
112   #define SPIBSC_OUTPUT_ADDR_24     0x07
113   #define SPIBSC_OUTPUT_ADDR_32     0x0f
114   #define SPIBSC_OUTPUT_OPD_3           0x08
115   #define SPIBSC_OUTPUT_OPD_32      0x0c
116   #define SPIBSC_OUTPUT_OPD_321     0x0e
117   #define SPIBSC_OUTPUT_OPD_3210    0x0f
118
```

3.29 サンプルプログラムリスト"io_spibsc.h"（3）

```
119   #define SPIBSC_OUTPUT_SPID_8       0x08
120   #define SPIBSC_OUTPUT_SPID_16      0x0c
121   #define SPIBSC_OUTPUT_SPID_32      0x0f
122
123   #define SPIBSC_SPISSL_NEGATE       0
124   #define SPIBSC_SPISSL_KEEP         1
125
126   #define SPIBSC_SPIDATA_DISABLE     0
127   #define SPIBSC_SPIDATA_ENABLE      1
128
129   #define SPIBSC_SPI_DISABLE         0
130   #define SPIBSC_SPI_ENABLE          1
131
132   typedef struct{
133      unsigned long cdb:2;
134      unsigned long ocdb:2;
135      unsigned long adb:2;
136      unsigned long opdb:2;
137      unsigned long spidb:2;
138      unsigned long cde:1;
139      unsigned long ocde:1;
140      unsigned long ade:4;
141      unsigned long opde:4;
142      unsigned long spide:4;
143      unsigned long sslkp:1;
144      unsigned long spire:1;
145      unsigned long spiwe:1;
146      unsigned long :5;
147
148      unsigned char cmd;
149      unsigned char ocmd;
150      unsigned long addr;
151      unsigned char opd[4];
152      unsigned long smrdr[2];
153      unsigned long smwdr[2];
154   }ST_SPIBSC_SM;
155
156   int io_spibsc_bsz_set(unsigned long bsz);
157   unsigned long io_spibsc_bsz_get(void);
158   int io_spibsc_common_init(unsigned long bsz);
159   int io_spibsc_dr_init(unsigned long cmd);
160   int io_spibsc_transfer(ST_SPIBSC_SM *SpibscSm);
161
162   #endif /* IO_SPIBSC_H */
163   /* End of File */
```

3.30 サンプルプログラムリスト"serial_flash.h"（1）

```
/***********************************************************************************
*   DISCLAIMER

（省略）

***********************************************************************************
*   Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
*************************** Technical reference data ***************************
*   System Name : SH7268/SH7269 Firm Update Sample Program
*   File Name   : serial_flash.h
*   Abstract    : Renesas Serial Peripheral Interface
*               : read/write serial flash memory in high-speed
*   Version     : 1.00.00
*   Device      : SH7268/SH7269
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                          (Ver.9.03 Release02).
*   OS          : None
*   H/W Platform: R0K57269(CPU board)
*   Description :
***********************************************************************************
*   History     : Jul.06,2011 Ver.1.00.00
***********************************************************************************/
#ifndef _SERIAL_FLASH_H_
#define _SERIAL_FLASH_H_

/* ==== Macro definition ==== */
/* S25FL032P(Spansion) */
#define SF_PAGE_SIZE      256         /* Page size of serial flash memory */
#define SF_SECTOR_SIZE    (64*1024)   /* Sector size = 64 KB   */
#define SF_NUM_OF_SECTOR  64          /* Number of sectors: 64   */

enum sf_req{
  SF_REQ_PROTECT = 0,                 /* request protect */
  SF_REQ_UNPROTECT,                   /* release protect */
  SF_REQ_SERIALMODE,                  /* request Serial/Dual mode */
  SF_REQ_QUADMODE,                    /* request Quad mode */
};
/* ==== Function prototype declaration ==== */
void sf_init_serial_flash(void);
void sf_protect_ctrl(enum sf_req req);
void sf_set_mode(enum sf_req req);
void sf_chip_erase(void);
void sf_sector_erase(int sector_no);
void sf_byte_program(unsigned long addr, unsigned char *buf, int size);
void sf_byte_read(unsigned long addr, unsigned char *buf, int size);
void sf_byte_read_long(unsigned long addr, unsigned long *buf, int size);

/* ==== Variant definition ==== */

#endif /* _SERIAL_FLASH_H_ */
/* End of File */
```

4. 参考ドキュメント

- ソフトウェアマニュアル
  SH-2A/SH-2A-FPU ソフトウェアマニュアル　Rev.3.00
  （最新版をルネサス エレクトロニクスホームページから入手してください。）
- ハードウェアマニュアル
  SH7268 グループ、SH7269 グループ ユーザーズマニュアル ハードウェア編　Rev.1.00
  （最新版をルネサス エレクトロニクスホームページから入手してください。）

ホームページとサポート窓口

- ルネサス エレクトロニクスホームページ
  http://japan.renesas.com/
- お問合せ先
  http://japan.renesas.com/inquiry

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2011.07.11	—	初版発行
1.01	2012.02.16	—	SH726B の参考プログラムを追加

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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

   【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

   CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

   【注意】電源投入時は，製品の状態は不定です。

   電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

   外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

   同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

   【注意】リザーブアドレスのアクセスを禁止します。

   アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

   【注意】リセット時は、クロックが安定した後、リセットを解除してください。

   プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

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

5. 製品間の相違について

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

   同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 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お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、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に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   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

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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その総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口 http://www.renesas.com

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

ルネサス エレクトロニクス販売株式会社　〒100-0004　千代田区大手町2-6-2（日本ビル） (03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：http://japan.renesas.com/inquiry

Colophon 1.0