

SH7266/SH7267 グループ

R01AN0167JJ0100

Rev. 1.00

2010.10.29

ルネサスクワッドシリアルペリフェラルインタフェース

シリアルフラッシュメモリ接続例

要旨

本アプリケーションノートは、SH7266/SH7267 のルネサスクワッドシリアルペリフェラルインタフェース (RQSPI) を使用した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

SH7266/SH7267

以下、総称して「SH7267」として説明します。

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.....	18
4. 参考ドキュメント.....	37

1. はじめに

1.1 仕様

- シリアルペリフェラルインタフェース (SPI) マルチ I/O バスに対応したシリアルフラッシュメモリを SH7267 と接続します。
- ルネサスシリアルペリフェラルインタフェース (RSPI) を使用してブートします。
- ブート後、RQSPI の Quad-SPI 動作を用いて、シリアルフラッシュメモリをリードライトします。

1.2 使用機能

- ルネサスクワッドシリアルペリフェラルインタフェース (RQSPI)
- 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	SH7266/SH7267
動作周波数	内部クロック : 144 MHz バスクロック : 72 MHz 周辺クロック : 36 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 -chginclpath -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。

- SH7266/SH7267 グループ 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

本応用例では、SH7267（マスタ）と SPI 互換のシリアルフラッシュメモリ（スレーブ）を接続して、ルネサスクワッドシリアルペリフェラルインタフェース（RQSPI）を用いてリードライトします。この章では、端子接続例と参考プログラムフローを説明します。

2.1 RQSPI の動作概要

SH7267 の RQSPI は、Single/Dual/Quad-SPI 動作でシリアルフラッシュメモリとの通信が可能です。転送データ長は 8 ビット～128G ビットに設定できます。ビットレートは $B\phi$ の 1～4080 分周に設定できます。ただし送信時は 1 分周に設定することができません。また 4 つのコマンドレジスタ（SPCMDn [n=0～3]）を用いて異なる転送方式をシーケンシャルにループ実行できます。

図 1～図 3 に Single/Dual/Quad-SPI モードの転送フォーマットを示します。Single-SPI モードは QMO 端子を出力、QMI 端子を入力とした全二重通信です。Dual-SPI モードは QIO0 端子と QIO1 端子を入出力端子とした 2 ビット幅の半二重通信です。Quad-SPI モードは QIO2 端子と QIO3 端子を加えた 4 ビット幅の半二重通信です。なお QMO 端子は QIO0 端子と、QMI 端子は QIO1 端子とのマルチプレクス端子です。

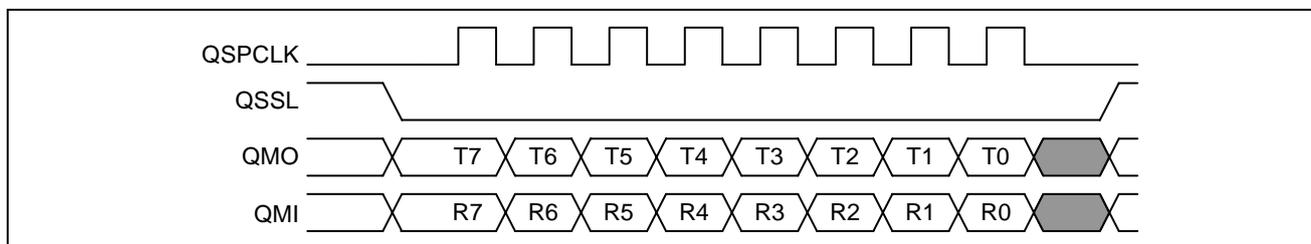


図1 Single-SPI モード転送フォーマット

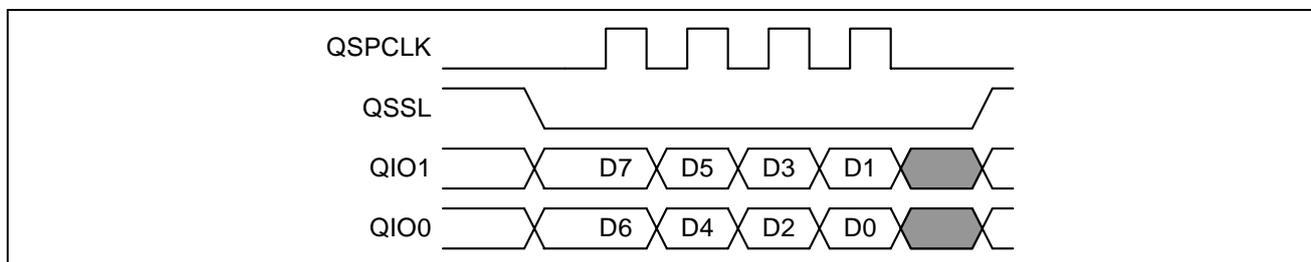


図2 Dual-SPI モード転送フォーマット

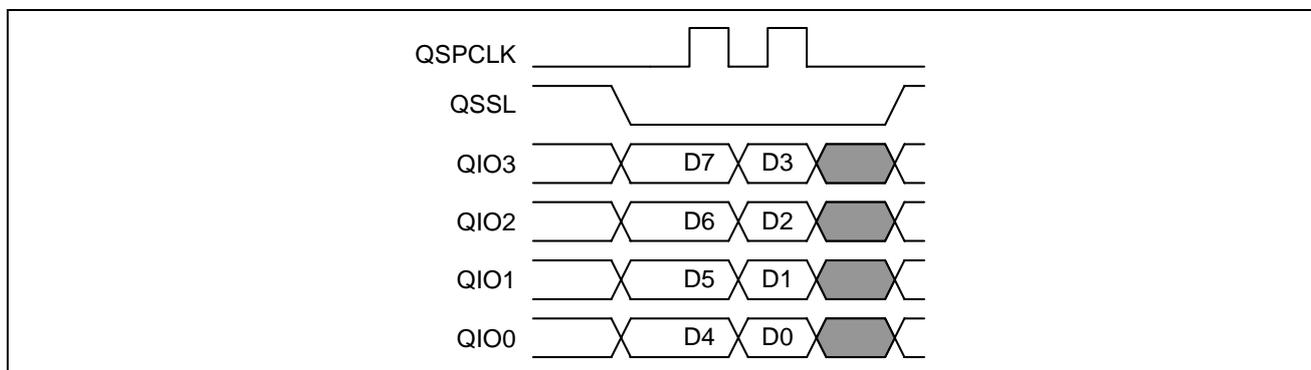


図3 Quad-SPI モード転送フォーマット

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 4にシリアルフラッシュメモリ接続回路例を示します。SH7267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。

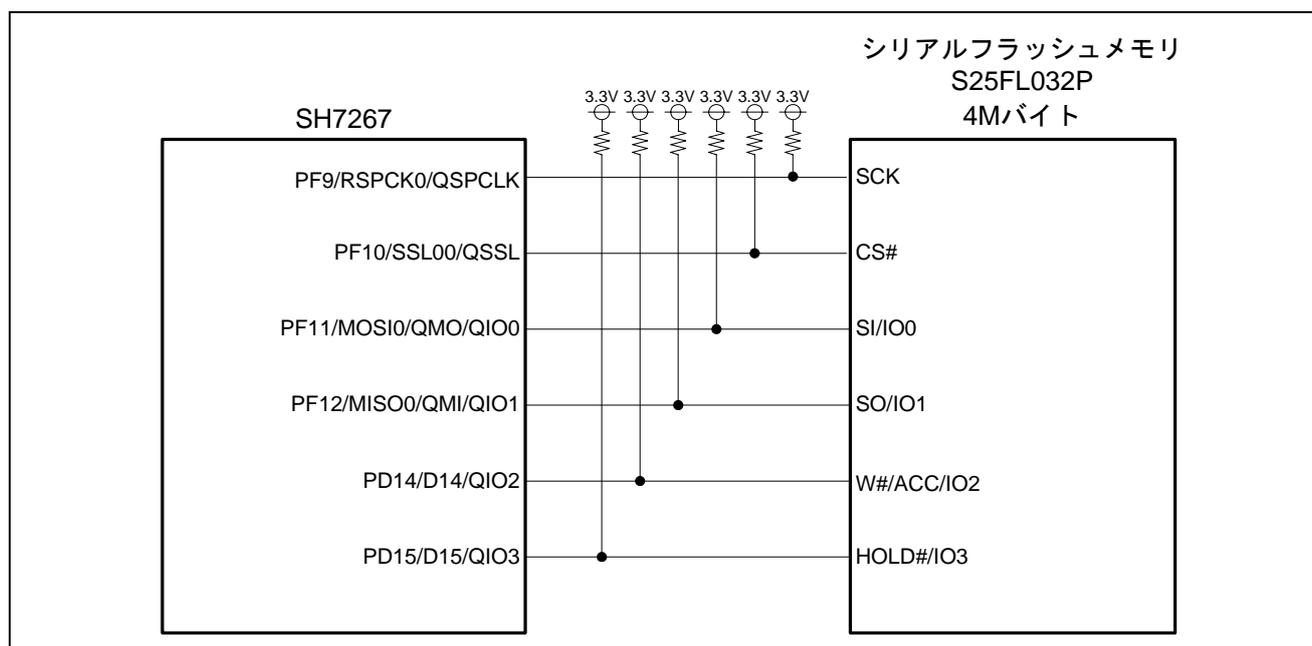


図4 シリアルフラッシュメモリ接続回路例

【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	SH7267 ポートコントロールレジスタ		SH7267 マルチプレクス端子名
		レジスタ名	MD ビット設定値	
RQSPI	QSPCLK	PFCR2	PF9MD[2:0] = B'110	PF9/A23/SSISCK3/RSPCK0/TIOC3A/FRB/ QSPCLK
	QSSL	PFCR2	PF10MD[2:0] = B'110	PF10/A24/SSIWS3/SSL00/TIOC3B/FCE#/QSSL
	QMO/QIO0	PFCR2	PF11MD[2:0] = B'110	PF11/A25/SSIDATA3/MOSIO/TIOC3C/SPDIF_IN/ QMO/QIO0
	QMI/QIO1	PFCR3	PF12MD[2:0] = B'110	PF12/BS/MISO0/TIOC3D/SPDIF_OUT/QMI/QIO1
	QIO2	PDCR3	PD14MD[1:0] = B'11	PD14/D14/NAF6/PWM2G/QIO2
	QIO3	PDCR3	PD15MD[1:0] = B'11	PD15/D15/NAF7/PWM2H/QIO3

【注】 SH7267 のマルチプレクス端子について

QSPCLK、QSSL、QMO/QIO0、QMI/QIO1、QIO2、QIO3 端子はマルチプレクス端子であり、初期状態は汎用入出力ポートになっています。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RQSPI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（CS0空間に接続したメモリからのブート）使用時は、QIO2、QIO3端子をRQSPI機能に設定できません。ブートモード1または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

SH7267 とシリアルフラッシュメモリ間のインタフェースタイミング例を示します。スレーブとなるシリアルフラッシュメモリのタイミング条件に合わせて RQSPI 設定およびクロック周波数設定を行います。

図 5 にデータ転送タイミング例を示します。本応用例で使用するシリアルフラッシュメモリは、クロックの立ち上がりでデータサンプルを行い、立ち下がりでデータ変化する仕様のため、コマンドレジスタ (SPCMD) の CPOL ビットと CPHA ビットにはともに 1 を設定します。本設定により、アイドル時の RSPCK は 1 に設定され、RSPI のデータ変化タイミングを奇数エッジ (ここでは立ち下がりエッジ) に設定することができます。表 3 と表 4 に示すタイミング条件を満たすように RQSPI を設定してください。

本応用例ではビットレートを 36Mbps ($B\phi=72\text{MHz}$ 時) に設定します。SH7267 は 1 分周に設定した状態での送信を禁止するため、Single-SPI 動作時は 72Mbps に設定できません。ただし、Dual または Quad-SPI 動作の場合は、半二重動作となるためリードだけであれば 72Mbps に設定できます。

なお、図 5 の転送方式でセットアップ時間が不足する場合はデータ変化からデータサンプルまでの期間を 1 サイクル分に拡張する転送方式を利用してください。詳細については、アプリケーションノート「SH7262/SH7264 グループ ルネサスシリアルペリフェラルインタフェース シリアルフラッシュメモリの高速リードライト例」を参照してください。

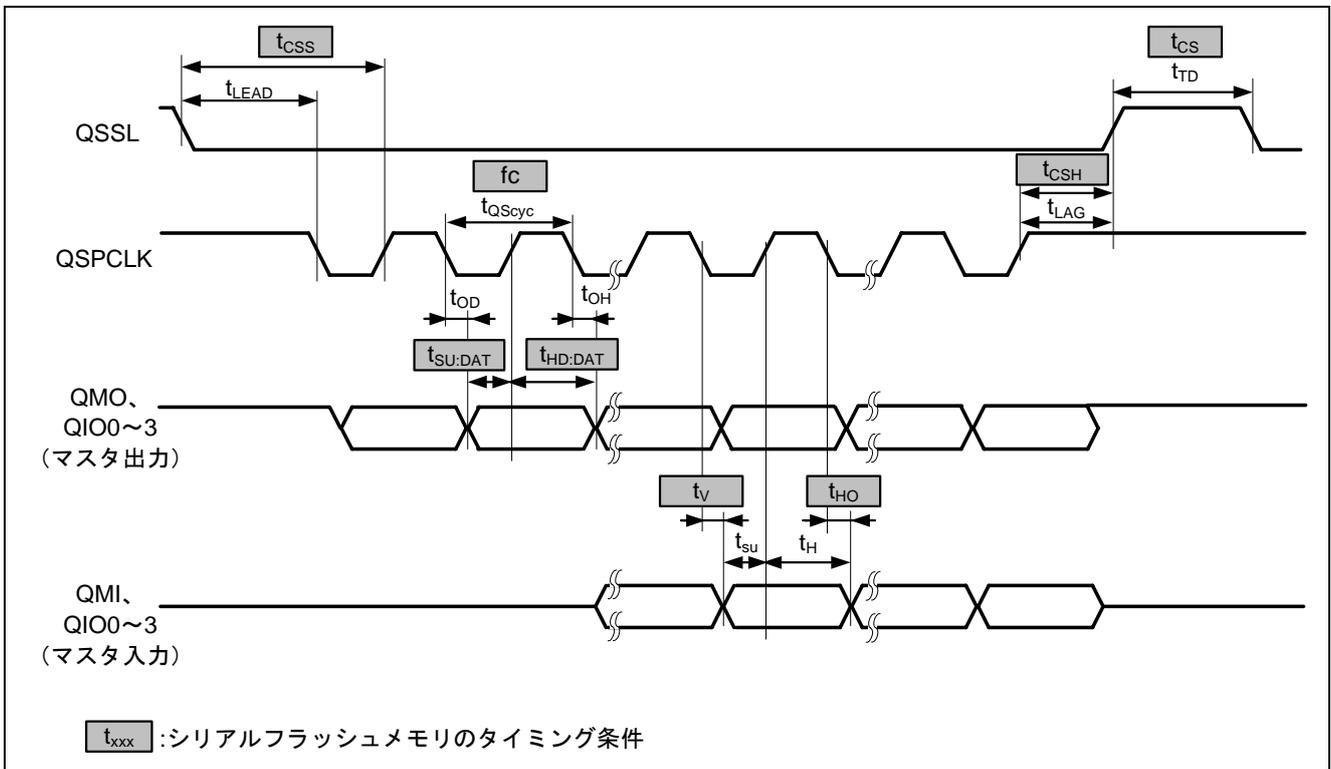


図5 データ転送タイミング例 (CPOL=1, CPHA=1 の場合)

表3 データ転送における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	関連レジスタ
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 QSPCLK の立ち上がりでスレーブがデータサンプルするまでに必要な時間です。 以下の式を満たす設定を行います。 $(t_{LEAD}(=QSPCLK \text{ 遅延}) + 1/2) \times t_{QScyc} \times t_{cyc} \geq t_{CSS} \text{ (min)}$	SPCKD レジスタ SPCMD レジスタ SPBR レジスタ
t_{CS}	チップセレクト'H'時間	QSSL のネゲート期間として必要な時間です。 以下の式を満たす設定を行います。 $t_{TD}(=\text{次アクセス遅延}) \times t_{QScyc} \times t_{cyc} \geq t_{CS} \text{ (min)}$	SPND レジスタ SPCMD レジスタ
f_c	シリアルクロック周波数	スレーブが対応可能な最大動作周波数です。 以下の式を満たす設定を行います。 $f_c(\text{max}) \geq 1 / (t_{QScyc} \times t_{cyc})$	SPBR レジスタ SPCMD レジスタ
t_{CSH}	チップセレクト'L'ホールド時間	最後の QSPCLK の立ち上がりから QSSL のネゲートまでに必要なホールド時間です。 以下の式を満たす設定を行います。 $t_{LAG}(=QSSL \text{ ネゲート遅延}) \times t_{QScyc} \times t_{cyc} \geq t_{CSH} \text{ (min)}$	SSLND レジスタ SPCMD レジスタ
$t_{SU:DAT}$	データ入力セットアップ時間	マスタのデータ出力からデータサンプルまでに必要な時間です。 以下の式を満たす設定を行います。 $1/2 \times t_{QScyc} \times t_{cyc} - t_{OD}(\text{max}) \geq t_{SU:DAT} \text{ (min)}$	SPBR レジスタ SPCMD レジスタ
$t_{HD:DAT}$	データ入力ホールド時間	マスタがデータ出力を保持しなければならない時間です。 以下の式を満たす設定を行います。 $t_{OH}(\text{min}) + 1/2 \times t_{QScyc} \times t_{cyc} \geq t_{HD:DAT} \text{ (min)}$	SPBR レジスタ SPCMD レジスタ

【注】 tはバスクロック（Bφ）の1サイクル時間を示します。

表4 データ転送における SH7267 のタイミング条件

シンボル	項目	説明	関連レジスタ
t_{SU}	データ入力セットアップ時間	スレーブのデータ出力からデータサンプルまでに必要な時間です。 以下の式を満たす設定を行います。 $1/2 \times t_{QScyc} \times t_{cyc} - t_V(\text{max}) \geq t_{SU}(\text{min})$	SPBR レジスタ SPCMD レジスタ
t_H	データ入力ホールド時間	スレーブがデータ出力を保持しなければならない時間です。 以下の式を満たす設定を行います。 $t_{HO}(\text{min}) + 1/2 \times t_{QScyc} \times t_{cyc} \geq t_H(\text{min})$	SPBR レジスタ SPCMD レジスタ

【注】 tはバスクロック（Bφ）の1サイクル時間を示します。

2.4 参考プログラムの動作

2.4.1 RQSPI の初期設定例

図 6に本参考プログラムにおけるRQSPI初期設定フローを示します。なお、転送方式に応じたコマンドレジスタの設定は転送開始前に設定します。

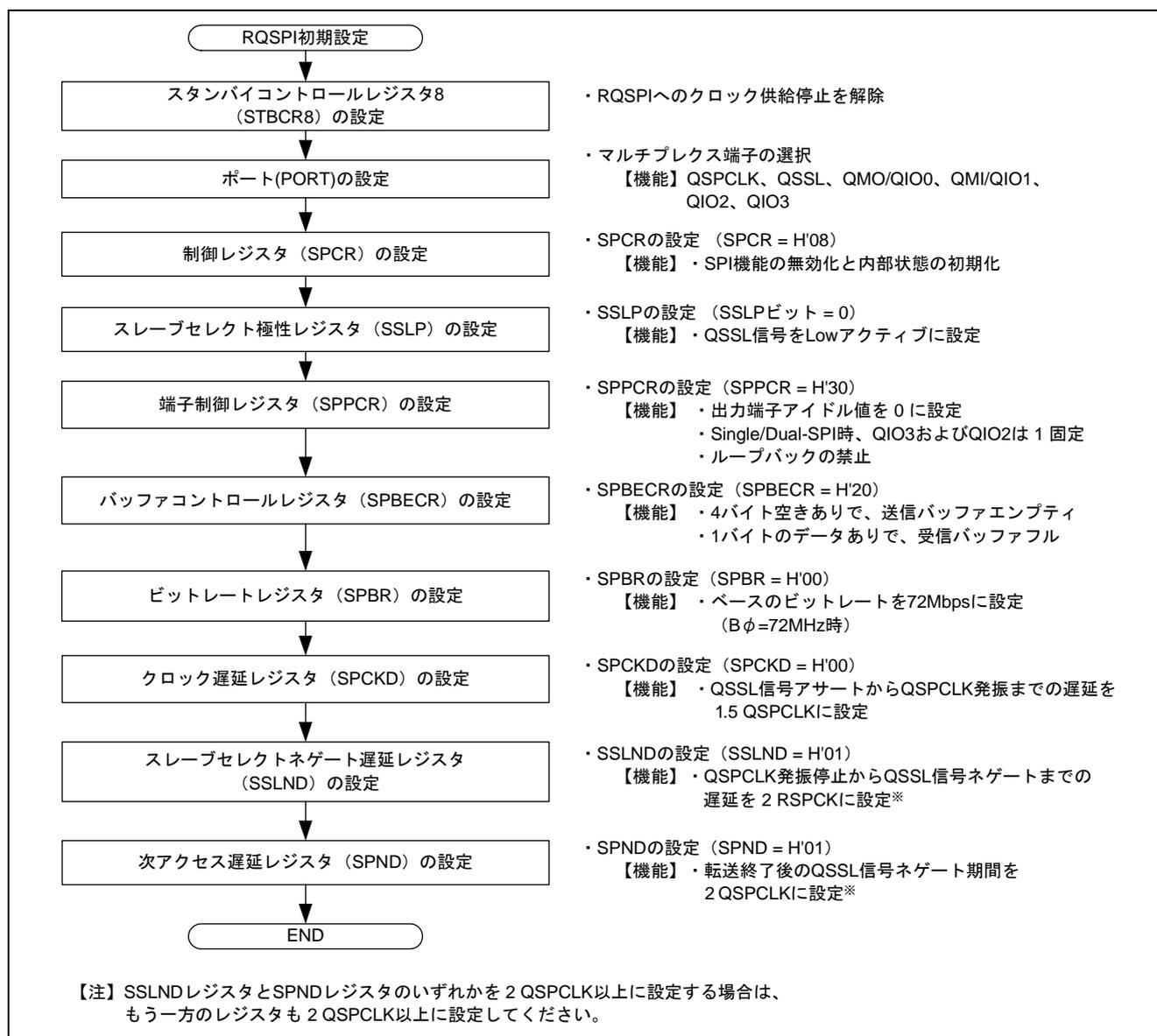


図6 参考プログラムの RQSPI 初期設定フロー (1)

2.4.2 シーケンス制御

シリアルフラッシュメモリのアクセスは全てコマンド形式で行います。以下に、主なコマンドおよびコマンドシーケンス例、参考プログラムでのシーケンス制御を示します。

なお、本応用例では Spansion 製 S25FL032P のコマンドを参考にしています。コマンドの詳細は使用するシリアルフラッシュメモリのデータシートを参照してください。

A. 主なコマンド

表 5にS25FL032Pの主なコマンドを示します。

表5 S25FL032P の主なコマンド

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
Read Data bytes at Fast Speed	H'0B	3	1	1 以上 ^{※1}	データのリード
Quad Output Read	H'6B	3	1	1 以上 ^{※1}	データのリード (Quad-SPI)
Write Enable	H'06	0	0	0	プログラム/イレース コマンドの許可
Write Disable	H'04	0	0	0	プログラム/イレース コマンドの禁止
64KB Sector Erase	H'D8	3	0	0	セクタ (64KB) 単位のイレース
Bulk Erase	H'C7	0	0	0	全領域のイレース
Page Programming	H'02	3	0	1 以上 ^{※2}	データのライト
Quad Page Programming	H'32	3	0	1 以上 ^{※2}	データのライト (Quad-SPI)
Read Status Register	H'05	0	0	1 以上	ステータスリード
Read Configuration Register	H'35	0	0	1 以上	コンフィグレーション レジスタリード
Write (Status & Configuration) Register	H'01	0	0	1 or 2	ステータス&コンフィグレーション レジスタライト

【注】 ^{※1} 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。)

^{※2} 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。)

B. コマンドシーケンス例

図 7にQuad Output Read (H'6B) コマンドのシーケンス例を示します。

S25FL032P の Quad Output Read コマンドは次の手順で実行します。まず QSSL 信号をアサートし、コマンドコードとアドレス (3 バイト) を Single-SPI モードで転送します。次にダミークロックを挿入します。ダミークロックは QMO/QIO0 端子を入力方向に切り換えるために Quad-SPI モード (リード設定) で実行します。その後、リードデータを Quad-SPI モード (リード設定) で読み出します。

このような複雑な転送は、コマンドレジスタ n (SPCMDn) と転送データ長倍数設定レジスタ (SPBMULn) を組み合わせたシーケンス制御を使用して実現します。

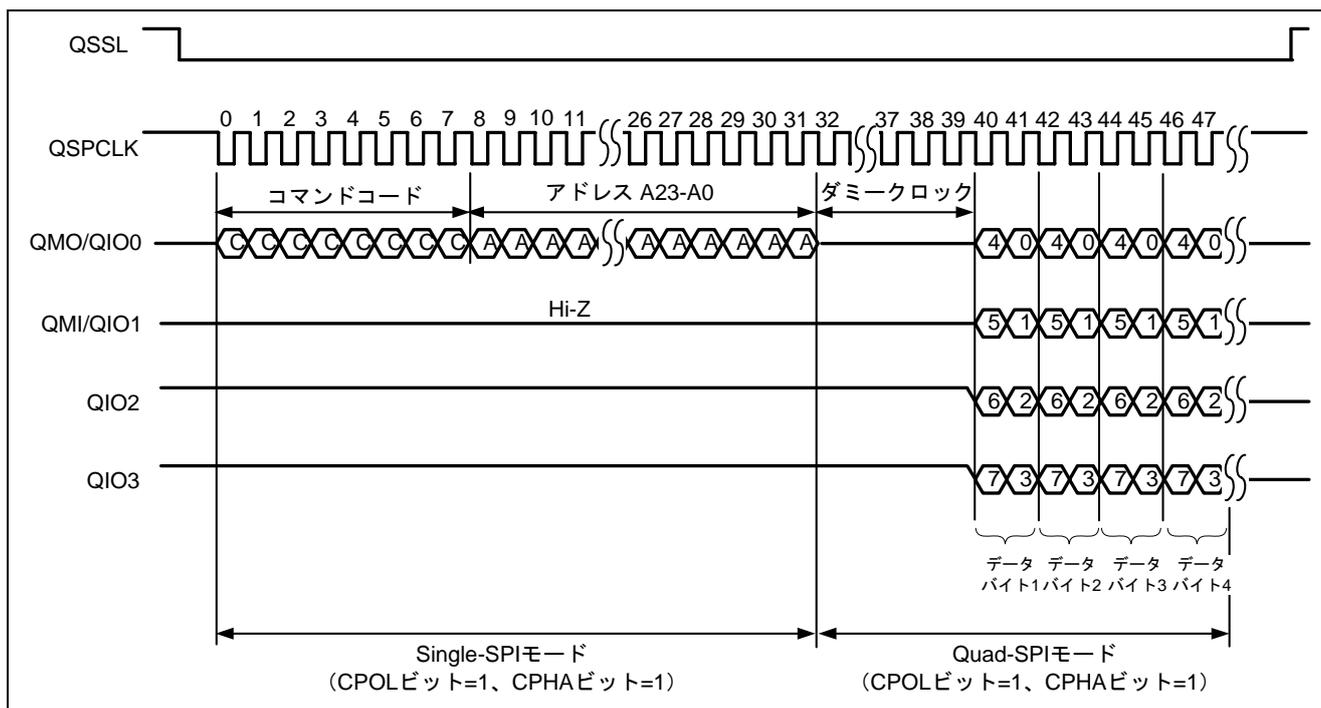


図7 リードコマンドシーケンス

C. 参考プログラムでのシーケンス制御例

図8に参考プログラムで使用するシーケンス制御例を示します。

Single-SPIモードでマスタ送信するコマンドコード、アドレスはSPCMD0レジスタを使用して転送します。Quad Output Read コマンド (H'6B) や Quad Page Programming (H'32) コマンドなど、Quad-SPIモードへの切り換えが必要な場合は、SPCMD1レジスタおよびSPCMD2レジスタを使用します。SPCMDnレジスタごとの転送回数はSPBMULnレジスタに設定します。

なお、Dual/Quad-SPIモードのリード設定は、受信バッファに空きがあると即座にクロックを発振してマスタ受信処理を開始します。そのため、SPBFCRレジスタの受信バッファリセットを実行すると、タイミングによっては受信したデータを破棄してしまう可能性があります。注意してください。

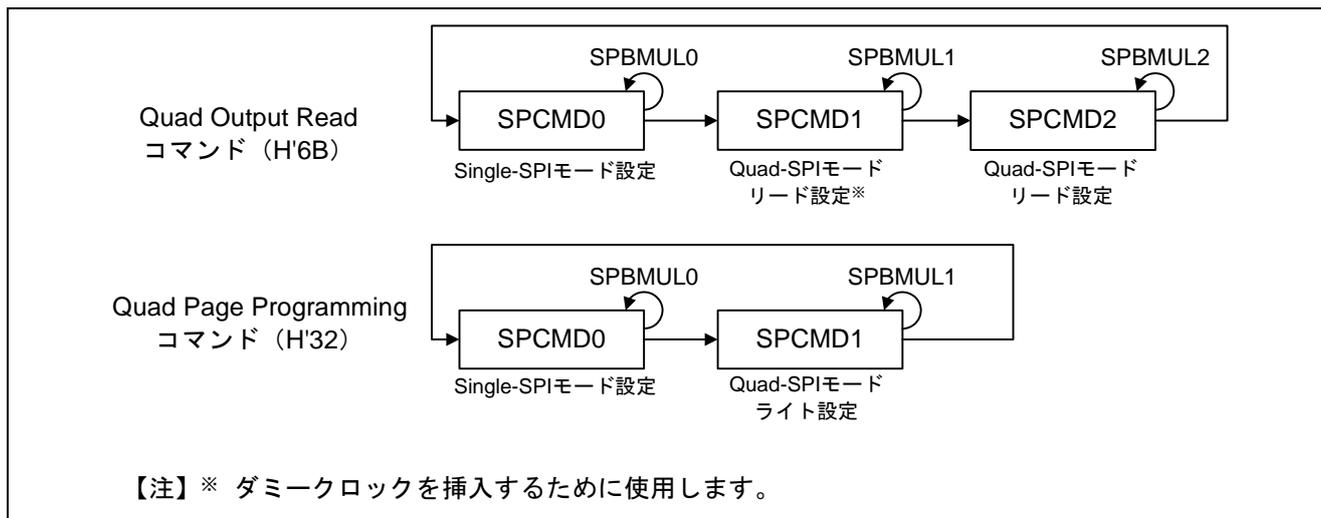


図8 参考プログラムのシーケンス制御

図9に参考プログラムでシーケンス制御情報を定義している構造体を示します。この構造体でSPCMDnレジスタごとの転送設定を管理します。

rqspi_cmd_set[4]構造体は、io_rqspi_transfer関数でRQSPI転送を行う際に使用します。io_rqspi_transfer関数を実行する前にio_rqspi_set_cmd関数を使用してrqspi_cmd_set[4]構造体に値を設定してください。

```

/* ---- RQSPIのシーケンス制御情報を定義する構造体 ---- */
typedef struct{
    unsigned short spcmd;          /* SPCMDnレジスタへの設定値 */
    unsigned long  spbmul;        /* SPBMULnレジスタへの設定値 */
    void          *wr_ptr;        /* 送信データの格納先アドレス */
    void          *rd_ptr;        /* 受信データの格納先アドレス */
}RQSPI_CMD_ST;

RQSPI_CMD_ST rqspi_cmd_set[4];

```

図9 参考プログラムのシーケンス制御情報を定義する構造体

図 10にspcmdメンバの設定に使用するマクロ定義を示します。

SPI_SINGLE/SPI_QUAD_WR/SPI_QUAD_RD マクロのいずれかを io_rqspi_set_cmd 関数の引数に指定してください。io_rqspi_set_cmd 関数が指定モードを反映した値を rqspi_cmd_set[4]構造体に設定します。

■SPCMDレジスタ設定のベースとなるマクロ定義

マクロ名 : SPCMD_DEFAULT_SET

設定値 : 0xE087

/* bit 15	: クロック遅延	: SPCKD(1.5 QSPCLK)	*/
/* bit 14	: QSSLネゲート遅延	: SSLND(2 QSPCLK)	*/
/* bit 13	: 次アクセス遅延	: SPND(2 QSPCLK)	*/
/* bit 12	: フォーマット	: MSBファースト	*/
/* bit 11-8	: 転送データ長	: 8ビット	*/
/* bit 7	: QSSL信号保持	: 転送終了後も保持	*/
/* bit 6-5	: SPI動作モード	: Single-SPI	*/
/* bit 4	: リードライト	: ライト(無効)	*/
/* bit 3-2	: ビットレート	: SPBRを2分周(36Mbps)	*/
/* bit 1	: CPOL(QSPCLK極性)	: アイドル時="1"	*/
/* bit 0	: CPHA(QSPCLK位相)	: 奇数エッジでデータシフト	*/
/*		: 偶数エッジでデータラッチ	*/

■Single-SPIモードを使用する場合のマクロ定義

マクロ名 : (SPCMD_DEFAULT_SET | SPI_SINGLE)

設定値 : 0xE087 | 0x0000

■Quad-SPIモードを使用してライト動作を行う場合のマクロ定義

マクロ名 : (SPCMD_DEFAULT_SET | SPI_QUAD_WR)

設定値 : 0xE087 | 0x0040

■Quad-SPIモードを使用してリード動作を行う場合のマクロ定義

マクロ名 : (SPCMD_DEFAULT_SET | SPI_QUAD_RD)

設定値 : 0xE087 | 0x0050

【注】 転送終了後にQSSL信号をネゲートするため、最後に指定したSPCMDnレジスタだけは上記の設定値にSSLKPビットを"0"に設定した値を書き込んでいます。

図10 参考プログラムの SPCMDn レジスタ設定用マクロ定義

2.4.3 メイン関数フロー

図 11に参考プログラムのメイン関数フローを示します。参考プログラムは、シリアルフラッシュメモリのセクタ 0 を除く全領域にライトした後、リードした値と等しいかをチェックします。

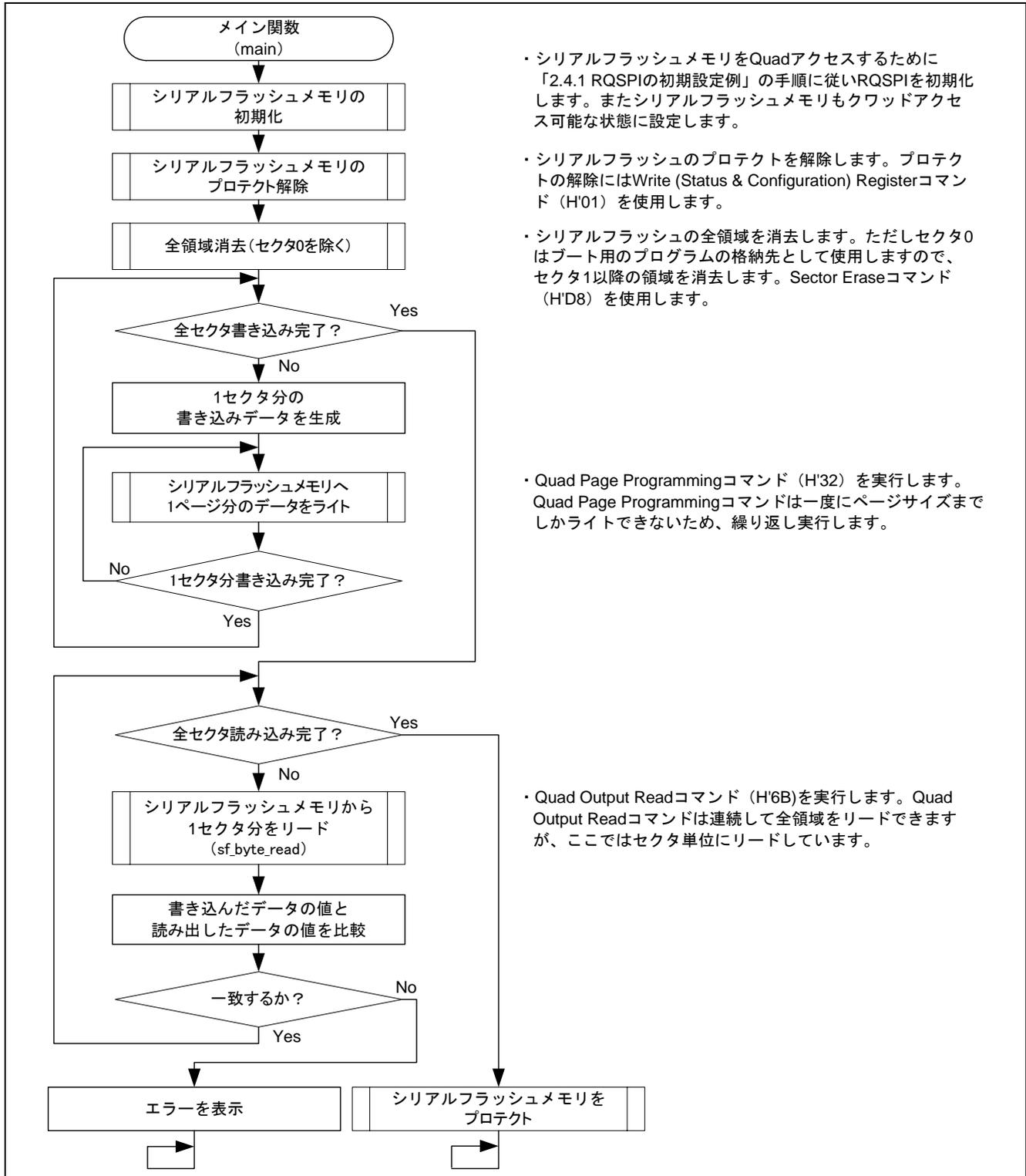


図11 参考プログラムのメイン関数フロー

2.4.4 シリアルフラッシュメモリのコマンド実行フロー

図 12に参考プログラムのコマンド実行フローを示します。シリアルフラッシュメモリのコマンドは複数ありますが、例としてQuad_Output_readコマンド (H'6B) の処理フローを示します。

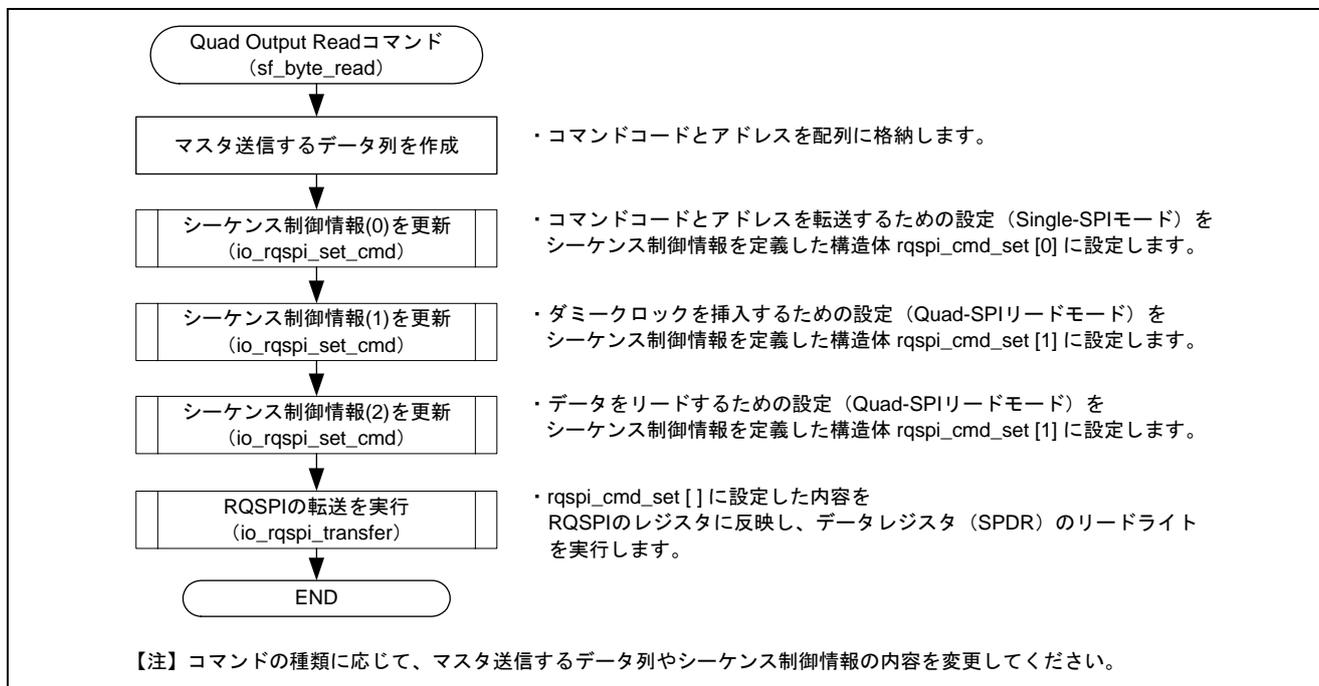


図12 参考プログラムのコマンド実行フロー (Quad Output read の場合)

2.4.5 RQSPI の転送処理フロー

図 13に参考プログラムのシーケンス制御情報の更新処理フローを示します。RQSPIの転送を行う前に、必要なコマンドレジスタの数だけ、この処理を実行してください。

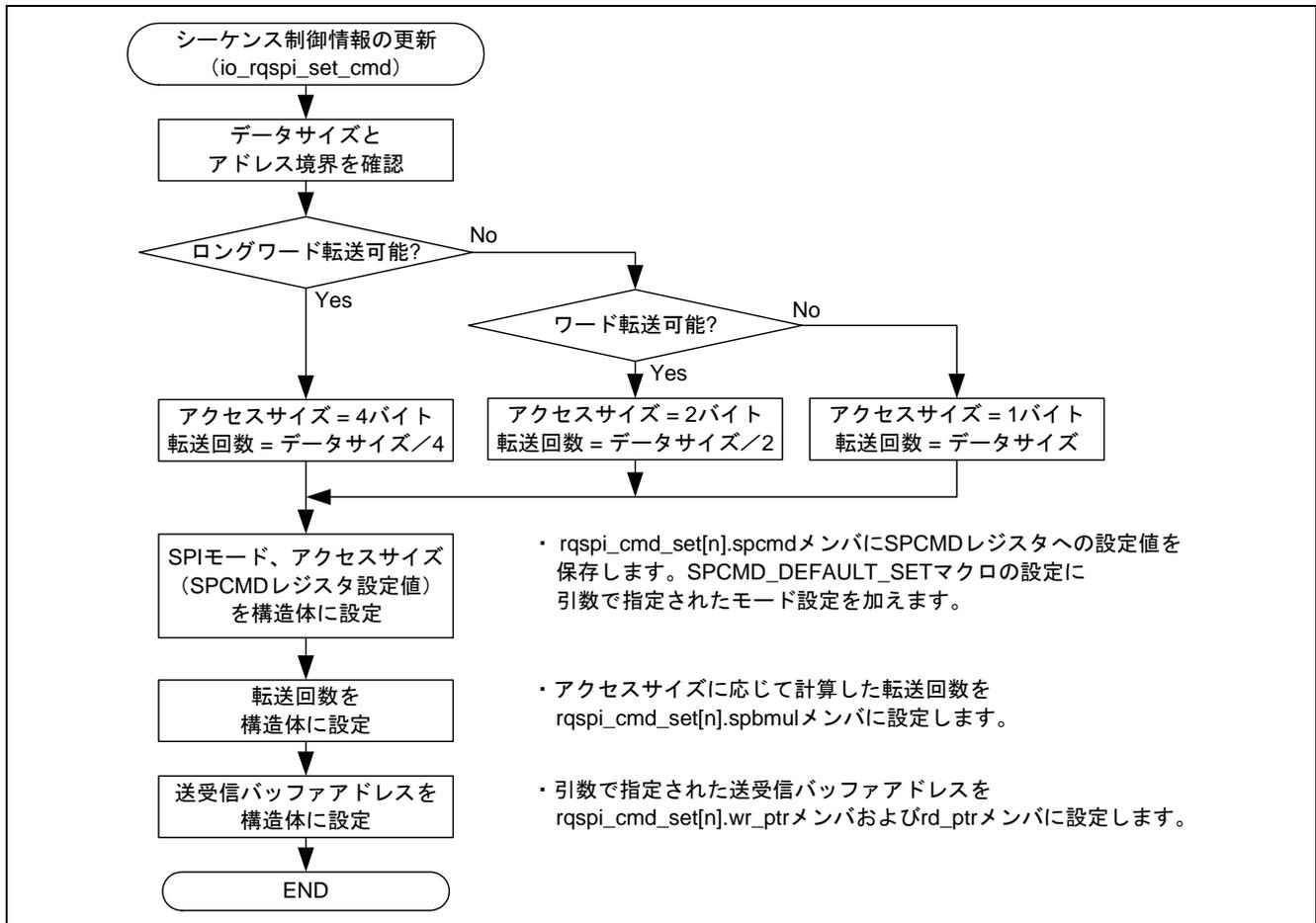


図13 参考プログラムの RQSPI 転送処理フロー (1)

図 14と 図 15に参考プログラムにおけるRQSPIの転送処理フローを示します。

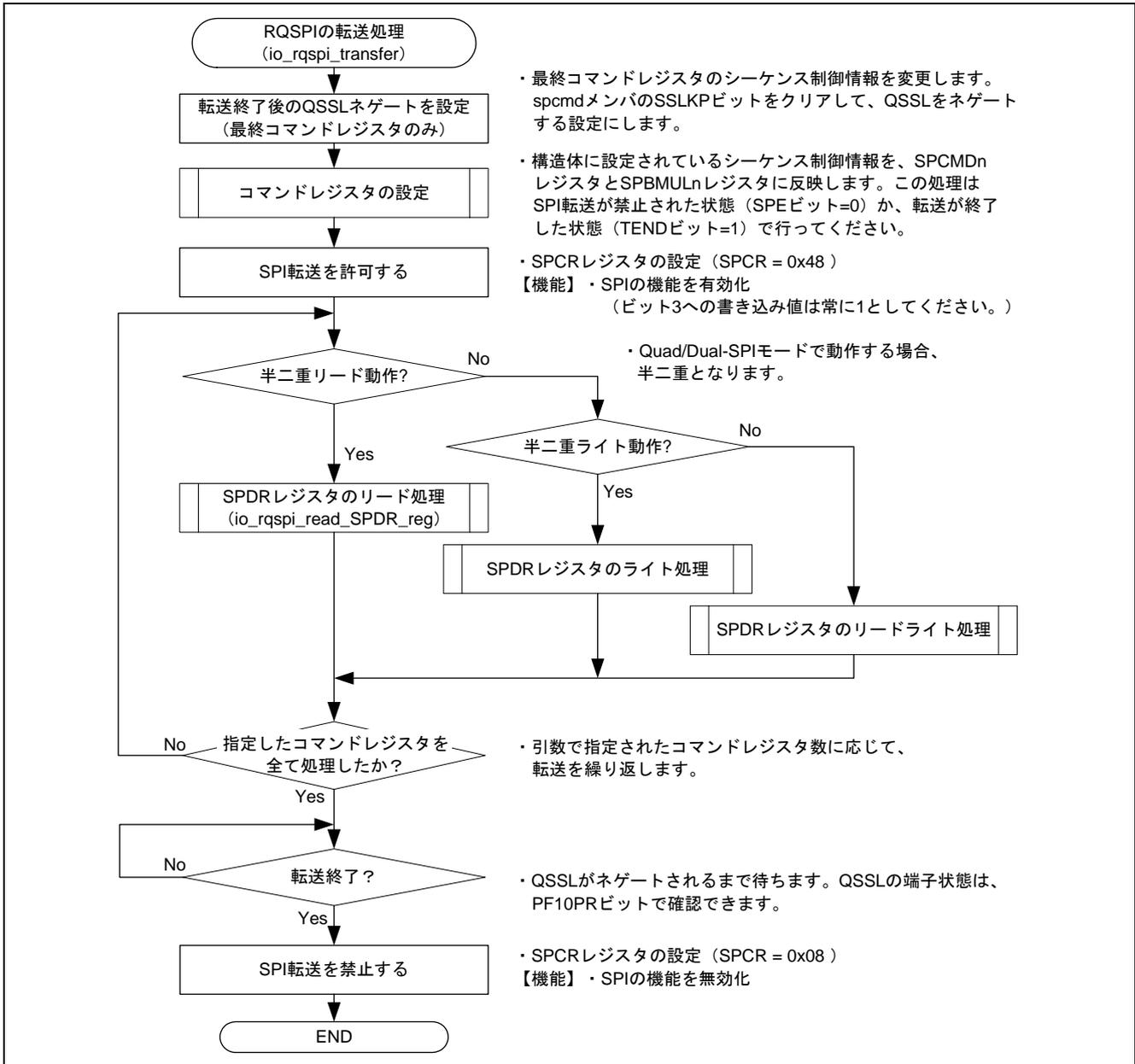


図14 参考プログラムのRQSPI転送処理フロー（1）

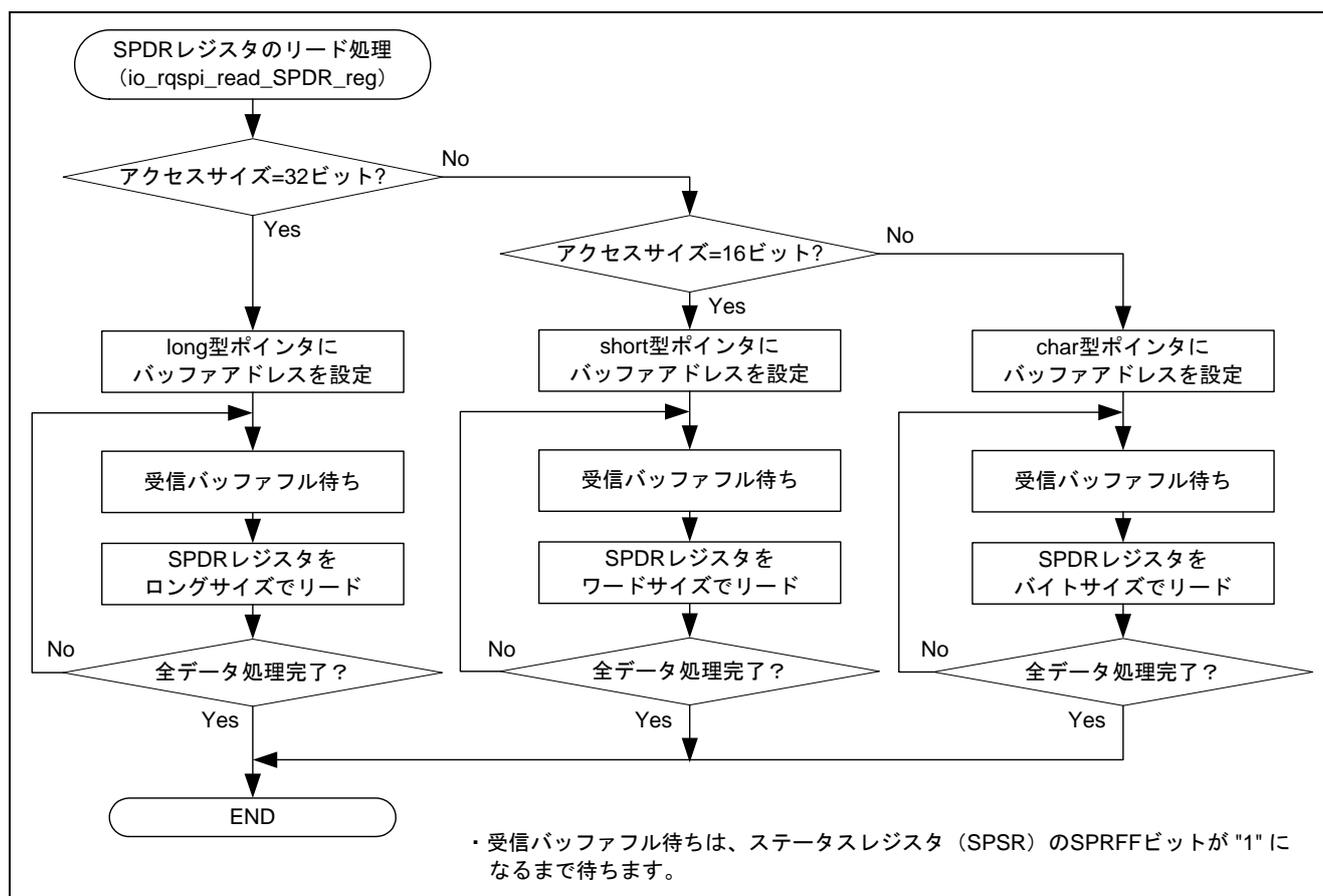


図15 参考プログラムの RQSPI 転送処理フロー (2)

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 (CS0 空間に接続したメモリからのブート) 使用時は、QIO2、QIO3 端子を RQSPI 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1 または 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「SH7266/SH7267 グループ 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) 2010 Renesas Electronics Corporation. All Rights Reserved.
29 *  "FILE COMMENT"***** Technical reference data *****
30 *  System Name : SH7267 Sample Program
31 *  File Name   : main.c
32 *  Abstract    : RQSPI クワッドシリアルフラッシュメモリ接続例
33 *  Version     : 1.00.00
34 *  Device      : SH7266/SH7267
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                      (Ver.9.03 Release02).
38 *  OS          : None
39 *  H/W Platform: R0K57267(CPU board), Spansion[S25FL032P](serial flash)
40 *  Description :
41 *****/
42 *  History     : Aug.20,2010 Ver.1.00.00
43 *  "FILE COMMENT END"*****/
44 #include <stdio.h>
45 #include "qserial_flash.h"
46
```

3.3 サンプルプログラムリスト"main.c" (2)

```
47  /* ==== マクロ定義 ==== */
48  #define TOP_ADDRESS    0                /* シリアルフラッシュメモリの先頭アドレス */
49
50  /* ==== 関数プロトタイプ宣言 ==== */
51  void main(void);
52
53  /* ==== 変数定義 ==== */
54  #pragma section DEBUG_BUFFER
55  static unsigned char data[SF_SECTOR_SIZE];
56  static unsigned char rbuf[SF_SECTOR_SIZE];
57  #pragma section
58
59  /*"FUNC COMMENT"*****
60  * ID          :
61  * Outline     : シリアルフラッシュメモリアクセス メイン処理
62  *-----
63  * Include     :
64  *-----
65  * Declaration : void main(void);
66  *-----
67  * Description : シリアルフラッシュメモリへのイレース、プログラム、リード処理を
68  *               : 行います。RSPi チャンネル 0 を初期化後、セクタ 0 を除く全領域を
69  *               : イレースした後、データを書き込みます。結果は読み出して確認します。
70  *-----
71  * Argument    : void
72  *-----
73  * Return Value : void
74  *-----
75  * Note        : None
76  *"FUNC COMMENT END"*****/
77  void main(void)
78  {
79      int i, j;
80      static unsigned long addr;
81
82      /* ==== シリアルフラッシュメモリの初期化 ==== */
83      sf_init_serial_flash();
84
85      /* ==== シリアルフラッシュメモリのプロテクト解除 ==== */
86      sf_protect_ctrl( SF_REQ_UNPROTECT );
87
88      /* ==== セクタイレース (セクタ 0 以外の全領域) ==== */
89      for(i = 1; i < SF_NUM_OF_SECTOR; i++){
90          sf_sector_erase( i );
91      }
```

3.4 サンプルプログラムリスト"main.c" (3)

```
92  /* ==== データライト (セクタ 0 以外の全領域) ==== */
93  addr = TOP_ADDRESS + SF_SECTOR_SIZE;      /* セクタ 1 */
94  for(i = 1; i < SF_NUM_OF_SECTOR; i++){    /* セクタ 1~ */
95
96      /* ---- データ初期化 (1セクタ分) ---- */
97      for(j = 0; j < SF_SECTOR_SIZE; j++){
98          data[j] = (i + j) % 100;
99      }
100     /* ---- セクタサイズをライト ---- */
101     for(j = 0; j < ( SF_SECTOR_SIZE / SF_PAGE_SIZE ); j++){
102
103         /* ---- ページサイズをライト ---- */
104         sf_byte_program( addr, data+(j*SF_PAGE_SIZE), SF_PAGE_SIZE );
105         addr += SF_PAGE_SIZE;              /* 書き込み先アドレス更新 */
106     }
107 }
108 /* ==== データリード (セクタ 0 以外の全領域) ==== */
109 addr = TOP_ADDRESS + SF_SECTOR_SIZE;      /* セクタ 1 */
110 for(i = 1; i < SF_NUM_OF_SECTOR; i++){    /* セクタ 1~ */
111
112     /* ---- セクタサイズをリード ---- */
113     sf_byte_read( addr, rbuf, SF_SECTOR_SIZE );
114     addr += SF_SECTOR_SIZE;                /* 読み込み先アドレス更新 */
115
116     /* ---- バリファイチェック ---- */
117     for(j = 0; j < SF_SECTOR_SIZE; j++){
118         data[j] = (i + j) % 100;          /* 書き込んだデータを再生 */
119         if( data[j] != rbuf[j] ){
120             puts("Error: verify error\n");
121             fflush(stdout);
122             while(1){
123                 /* error */
124             }
125         }
126     }
127 }
128 /* ==== シリアルフラッシュメモリのプロテクト ==== */
129 sf_protect_ctrl( SF_REQ_PROTECT );
130
131 while(1){
132     /* loop */
133 }
134 }
135
136 /* End of File */
```

3.5 サンプルプログラムリスト"qserial_flash.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) 2010 Renesas Electronics Corporation. All Rights Reserved.
29 *  "FILE COMMENT" ***** Technical reference data *****
30 *  System Name : SH7267 Sample Program
31 *  File Name   : qserial_flash.c
32 *  Abstract    : RQSPI クワッドシリアルフラッシュメモリ接続例
33 *  Version     : 1.00.01
34 *  Device      : SH7266/SH7267
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              : (Ver.9.03 Release02).
38 *  OS          : None
39 *  H/W Platform: R0K57267(CPU board), Spansion[S25FL032P](serial flash)
40 *  Description :
41 *****/
42 *  History     : Aug.20,2010 Ver.1.00.00
43 *              : Oct.13,2010 Ver.1.01.00 クワッドリード時のシーケンス制御方法を変更
44 *  "FILE COMMENT END" *****/
45 #include <stdio.h>
46 #include <machine.h>
47 #include "iodefine.h"
48 #include "qserial_flash.h"
49 #include "rqspi.h"
50
```

3.6 サンプルプログラムリスト"qserial_flash.c" (2)

```
51  /* ==== マクロ定義 ==== */
52  /* ---- シリアルフラッシュメモリのコマンド[S25FL032P(Spansion)] ---- */
53  #define SFLASHCMD_CHIP_ERASE      0xc7
54  #define SFLASHCMD_SECTOR_ERASE    0xd8
55  #define SFLASHCMD_BYTE_PROGRAM    0x02
56  #define SFLASHCMD_BYTE_READ       0x0B
57  #define SFLASHCMD_QUAD_PROGRAM    0x32
58  #define SFLASHCMD_QUAD_READ       0x6B
59  #define SFLASHCMD_WRITE_ENABLE    0x06
60  #define SFLASHCMD_WRITE_DISABLE   0x04

    (省略)

285  /*"FUNC COMMENT"*****
286  * ID          :
287  * Outline     : データリード
288  *-----
289  * Include     :
290  *-----
291  * Declaration : void sf_byte_read(unsigned long addr, unsigned char *buf, int size);
292  *-----
293  * Description : シリアルフラッシュメモリを指定バイト数だけリードします。
294  *-----
295  * Argument    : unsigned long addr ; I : リードするシリアルフラッシュメモリのアドレス
296  *              : unsigned char *buf ; I : リードデータを格納するバッファのアドレス
297  *              : int size          ; I : リードするバイト数
298  *-----
299  * Return Value : void
300  *-----
301  * Note        : None
302  *"FUNC COMMENT END"*****/
303  void sf_byte_read(unsigned long addr, unsigned char *buf, int size)
304  {
305      unsigned char cmd[4];
306
307      cmd[0] = SFLASHCMD_QUAD_READ;
308      cmd[1] = (unsigned char)((addr >> 16) & 0xff);
309      cmd[2] = (unsigned char)((addr >> 8) & 0xff);
310      cmd[3] = (unsigned char)( addr      & 0xff);
311
312      io_rqspi_set_cmd( 0, SPI_SINGLE, cmd, NULL, sizeof(cmd));
313      io_rqspi_set_cmd( 1, SPI_QUAD_RD, NULL, NULL, 4); /* (2clk/byte)×4 = 8clk */
314      io_rqspi_set_cmd( 2, SPI_QUAD_RD, NULL, buf, size);
315      io_rqspi_transfer(2);
316  }

    (以下、省略)

478  /* End of File */
```

3.7 サンプルプログラムリスト"qserial_flash.h" (1)

```
1  /*****
2  *   DISCLAIMER
3
4  (省略)
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27  ****
28  *   Copyright (C) 2010 Renesas Electronics Corporation. All Rights Reserved.
29  *   "FILE COMMENT" ***** Technical reference data *****
30  *   System Name : SH7267 Sample Program
31  *   File Name   : qserial_flash.h
32  *   Abstract    : QSPI クワッドシリアルフラッシュメモリ接続例
33  *   Version     : 1.00.00
34  *   Device      : SH7266/SH7267
35  *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36  *                : C/C++ compiler package for the SuperH RISC engine family
37  *                :                               (Ver.9.03 Release02).
38  *   OS          : None
39  *   H/W Platform: R0K57267(CPU board), Spansion[S25FL032P](serial flash)
40  *   Description :
41  ****
42  *   History     : Aug.20,2010 Ver.1.00.00
43  *   "FILE COMMENT END" *****
44  #ifndef _QSERIAL_FLASH_H_
45  #define _QSERIAL_FLASH_H_
46
47  /* ==== マクロ定義 ==== */
48  #define SF_PAGE_SIZE      256          /* ページサイズ */
49  #define SF_SECTOR_SIZE    0x10000     /* セクタサイズ = 64KB */
50  #define SF_NUM_OF_SECTOR  64          /* セクタ数 64 */
51
52  enum sf_req{
53      SF_REQ_PROTECT = 0,              /* プロテクト要求 */
54      SF_REQ_UNPROTECT,                /* プロテクト解除要求 */
55      SF_REQ_SERIALMODE,               /* Serial/Dual モード要求 */
56      SF_REQ_QUADMODE,                 /* Quad モード要求 */
57  };
58
59  /* ==== 関数プロトタイプ宣言 ==== */
60  void sf_init_serial_flash(void);
61  void sf_protect_ctrl(enum sf_req req);
62  void sf_set_mode(enum sf_req req);
63  void sf_chip_erase(void);
64  void sf_sector_erase(int sector_no);
65  void sf_byte_program(unsigned long addr, unsigned char *buf, int size);
66  void sf_byte_read(unsigned long addr, unsigned char *buf, int size);
67
68  #endif /* _QSERIAL_FLASH_H_ */
69  /* End of File */
70
```

3.8 サンプルプログラムリスト"rqspi.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) 2010 Renesas Electronics Corporation. All Rights Reserved.
29 *  "FILE COMMENT"***** Technical reference data *****
30 *  System Name : SH7267 Sample Program
31 *  File Name   : rqspi.c
32 *  Abstract    : RQSPI クワッドシリアルフラッシュメモリ接続例
33 *  Version     : 1.00.00
34 *  Device      : SH7266/SH7267
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              : (Ver.9.03 Release02).
38 *  OS          : None
39 *  H/W Platform: R0K57267(CPU board), Spansion[S25FL032P](serial flash)
40 *  Description :
41 *****/
42 *  History     : Aug.20,2010 Ver.1.00.00
43 *  "FILE COMMENT END"*****/
44 #include <stdio.h>
45 #include <machine.h>
46 #include "iodefine.h"
47 #include "rqspi.h"
48
```

3.9 サンプルプログラムリスト"rqspi.c" (2)

```
49  /* ==== マクロ定義 ==== */
50  #define SPCR_SPI_ENABLE 0x48          /* ビット 3 は常に"1"をライトすること */
51  #define SPCR_SPI_DISABLE 0x08        /* ビット 3 は常に"1"をライトすること */
52  #define SPCMD_SPIRW_BIT 0x0070       /* SPI 動作モードとリードライトの対象ビット */
53  #define SPCMD_SPB_BIT 0x0f00        /* 転送データ長(アクセスサイズ)の対象ビット */
54  #define SPCMD_SPB_8BITS 0x0000
55  #define SPCMD_SPB_16BITS 0x0100
56  #define SPCMD_SPB_32BITS 0x0200
57  #define SPCMD_DEFAULT_SET 0xe087
58          /* bit 15 : クロック遅延      : SPCKD( 1.5 QSPCLK) */
59          /* bit 14 : QSSL ネゲート遅延 : SSLND( 2 QSPCLK) */
60          /* bit 13 : 次アクセス遅延   : SPND( 2 QSPCLK) */
61          /* bit 12 : フォーマット      : MSB ファースト */
62          /* bit 11-8: 転送データ長     : 8 ビット */
63          /* bit 7  : QSSL 信号保持     : 転送終了後も保持 */
64          /* bit 6-5 : SPI 動作モード   : Single-SPI */
65          /* bit 4  : リードライト     : ライト(無効) */
66          /* bit 3-2 : ビットレート     : SPBR を 2 分周(36Mbps) */
67          /* bit 1  : CPOL(QSPCLK 極性) : アイドル時="1" */
68          /* bit 0  : CPHA(QSPCLK 位相) : 奇数エッジでデータシフト*/
69          /*                               : 偶数エッジでデータラッチ */
70
71  /* ---- RQSPI のシーケンス制御情報を定義する構造体 ---- */
72  typedef struct{
73      unsigned short spcmd;           /* SPCMDn レジスタへの設定値 */
74      unsigned long  spbmul;         /* SPBMULn レジスタへの設定値 */
75      void *wr_ptr;                  /* 送信データの格納先アドレス */
76      void *rd_ptr;                  /* 受信データの格納先アドレス */
77  }RQSPI_CMD_ST;
78
79  /* ==== 関数プロトタイプ宣言 ==== */
80  static int io_rqspi_update_SPCMD_reg( int seq );
81  static void io_rqspi_write_SPDR_reg( void *wrp, int cnt, unsigned short bitsz );
82  static void io_rqspi_read_SPDR_reg( void *rdp, int cnt , unsigned short bitsz );
83  static void io_rqspi_rdw_r_SPDR_reg( void *wrp, void *rdp, int cnt, unsigned short bitsz );
84
85  /* ==== 変数定義 ==== */
86  RQSPI_CMD_ST rqspi_cmd_set[4];
87
```

3.10 サンプルプログラムリスト"rqspi.c" (3)

```
88  /*"FUNC COMMENT"*****
89  * ID      :
90  * Outline : RQSPI の初期化
91  *-----
92  * Include : iodefine.h
93  *-----
94  * Declaration : static void io_init_rqspi(void);
95  *-----
96  * Description : ルネサスクワッドシリアルペリフェラルインタフェースを初期化します。
97  *              : マスタモードに設定し、シリアルフラッシュメモリの仕様に合わせた
98  *              : 転送設定を行います。
99  *-----
100 * Argument  : void
101 *-----
102 * Return Value: void
103 *-----
104 * Note      : None
105 *"FUNC COMMENT END"*****/
106 void io_rqspi_initialize(void)
107 {
108     /* ==== クロック供給 ==== */
109     CPG.STBCR8.BIT.MSTP82 = 0;
110
111     /* ==== Port ==== */
112     PORT.PFCR3.BIT.PF12MD = 0x06u; /* QMI/QIO1 */
113     PORT.PFCR2.BIT.PF11MD = 0x06u; /* QMO/QIO0 */
114     PORT.PFCR2.BIT.PF10MD = 0x06u; /* QSSL */
115     PORT.PFCR2.BIT.PF9MD  = 0x06u; /* QSPCLK */
116     PORT.PDCR3.BIT.PD15MD = 0x03u; /* QIO3 */
117     PORT.PDCR3.BIT.PD14MD = 0x03u; /* QIO2 */
118
119     /* ==== SPI 機能の無効化と内部状態の初期化 ==== */
120     RQSPI.SPCR.BYTE = SPCR_SPI_DISABLE;
121
122     /* ==== スレーブセレクト極性レジスタ (SSLP) ==== */
123     RQSPI.SSLP.BIT.SSLP = 0; /* QSSL 信号は Low アクティブ */
124
125     /* ==== 端子制御レジスタ (SPPCR) ==== */
126     RQSPI.SPPCR.BYTE = 0x26; /* 出力端子アイドル値は 0 */
127                             /* Single/Dual-SPI 時、QIO3 は 1 固定 */
128                             /* Single/Dual-SPI 時、QIO2 は 1 固定 */
129                             /* 通常動作モード(非ループバック) */
130     /* ==== バッファコントロールレジスタ (SPBFCR) ==== */
131     RQSPI.SPBFCR.BYTE = 0x20; /* 4 バイト空きありで、送信バッファエンプティ */
132                             /* 1 バイトのデータありで、受信バッファフル */
133     /* ==== ビットレートレジスタ (SPBR) ==== */
134     RQSPI.SPBR.BYTE = 0; /* ベースビットレート 72Mbps (Bφ=72MHz 時) */
135
136     /* ==== クロック遅延レジスタ (SPCKD) ==== */
137     RQSPI.SPCKD.BYTE = 0x00; /* SSL セットアップ時間= 1.5 QSPCLK */
138
139     /* ==== スレーブセレクトネゲート遅延レジスタ (SSLND) ==== */
140     RQSPI.SSLND.BYTE = 0x01; /* SSL ホールド時間= 2 QSPCLK */
```

3.11 サンプルプログラムリスト"rqspi.c" (4)

```

141
142  /* ==== 次アクセス遅延レジスタ (SPND) ==== */
143  RQSPI.SPND.BYTE = 0x01;          /* 連続転送遅延時間= 2 QSPCLK */
144
145  /* ==== コマンドレジスタ n (SPCMDn) ==== */
146  RQSPI.SPCMD0.WORD = SPCMD_DEFAULT_SET;    /* (転送前に再設定) */
147  RQSPI.SPCMD1.WORD = SPCMD_DEFAULT_SET;    /* (転送前に再設定) */
148  RQSPI.SPCMD2.WORD = SPCMD_DEFAULT_SET;    /* (転送前に再設定) */
149  RQSPI.SPCMD3.WORD = SPCMD_DEFAULT_SET;    /* (転送前に再設定) */
150  }
151  /*"FUNC COMMENT"*****
152  * ID          :
153  * Outline     : シーケンス制御情報の更新
154  *-----
155  * Include     :
156  *-----
157  * Declaration : void io_rqspi_set_cmd( int idx, unsigned short mode, void *wrp,
158  *          :                          void *rdp, unsigned long sz)
159  *-----
160  * Description : シーケンス制御情報を定義する構造体 rqspi_cmd_set を更新します。
161  *-----
162  * Argument    : int          idx ; I : 対象のコマンドレジスタ番号 (0~3)
163  *              : unsigned short mode ; I : SPCMD に指定する SPI モードおよび RW 設定
164  *              : void          *wrp ; I : 送信データ格納先アドレス
165  *              : void          *rdp ; O : 受信データ格納先アドレス
166  *              : unsigned long  sz  ; I : 送受信データ数 (バイト)
167  *-----
168  * Return Value : void
169  *-----
170  * Note        :
171  *"FUNC COMMENT END"*****
172  void io_rqspi_set_cmd( int idx, unsigned short mode, void *wrp, void *rdp, unsigned long sz)
173  {
174  RQSPI_CMD_ST *cmd = rqspi_cmd_set;
175  unsigned short bitsz = SPCMD_SPB_8BITS;
176  unsigned long trncnt = sz;
177
178  /* ---- 最適なデータサイズを設定 ---- */
179  if( ((sz&0x3)==0) && (((int)wrp&0x3)==0) && (((int)rdp&0x3)==0) ){
180  bitsz = SPCMD_SPB_32BITS;
181  trncnt = sz >> 2;
182  }
183  else if( ((sz&0x1)==0) && (((int)wrp&0x1)==0) && (((int)rdp&0x1)==0) ){
184  bitsz = SPCMD_SPB_16BITS;
185  trncnt = sz >> 1;
186  }
187
188  cmd[idx].spcmd = (mode | SPCMD_DEFAULT_SET | bitsz);
189  cmd[idx].spbmul = trncnt;
190  cmd[idx].wr_ptr = wrp;
191  cmd[idx].rd_ptr = rdp;
192  }

```

3.12 サンプルプログラムリスト"rqspi.c" (5)

```
193  /*"FUNC COMMENT"*****
194  * ID      :
195  * Outline  : RQSPI の転送処理
196  *-----
197  * Include  : iodefine.h
198  *-----
199  * Declaration : int io_rqspi_transfer( int seq );
200  *-----
201  * Description : RQSPI の転送処理を行います。
202  *              : シーケンス制御情報を定義する構造体 rqspi_cmd_set を使用して
203  *              : RQSPI の転送を行います。データレジスタを必要回数分リードライトし
204  *              : 送受信を行います。
205  *-----
206  * Argument   : int seq ; I : 使用するコマンドレジスタ数-1 (SPCMD0 のみの場合:0)
207  *-----
208  * Return Value : 0 : 正常終了
209  *-----
210  * Note       : None
211  *"FUNC COMMENT END"*****/
212  int io_rqspi_transfer( int seq )
213  {
214      int i, cnt;
215      unsigned short spirw, bitsz;
216
217      /* ==== コマンドレジスタの設定 ==== */
218      rqspi_cmd_set[seq].spcmd &= ~(0x0080);      /* QSSL のネゲート設定を追加 */
219      io_rqspi_update_SPCMD_reg(seq);
220
221      /* ==== SPI 転送許可 ==== */
222      RQSPI.SPCR.BYTE = SPCR_SPI_ENABLE;
223
224      /* ==== データレジスタのリードライト処理 ==== */
225      for( i=0; i<=seq; i++){
226          bitsz = rqspi_cmd_set[i].spcmd & SPCMD_SPB_BIT;
227          spirw = rqspi_cmd_set[i].spcmd & SPCMD_SPIRW_BIT;
228          cnt = rqspi_cmd_set[i].spbmul;
229
230          /* ---- 半二重(ライト)動作の場合 --- */
231          if( spirw==SPI_QUAD_WR || spirw==SPI_DUAL_WR ){
232              io_rqspi_write_SPDR_reg( rqspi_cmd_set[i].wr_ptr, cnt, bitsz);
233          }
234          /* ---- 半二重(リード)動作の場合 --- */
235          else if( spirw==SPI_QUAD_RD || spirw==SPI_DUAL_RD ){
236              io_rqspi_read_SPDR_reg( rqspi_cmd_set[i].rd_ptr, cnt, bitsz);
237          }
238          /* ---- 全二重動作の場合 --- */
239          else{
240              io_rqspi_rdwr_SPDR_reg( rqspi_cmd_set[i].wr_ptr, rqspi_cmd_set[i].rd_ptr, cnt, bitsz);
241          }
242      }
```

3.13 サンプルプログラムリスト"rqspi.c" (6)

```
243     /* ==== 転送終了待ち (QSSLがネゲートされるまで) ==== */
244     while( PORT.PFPR0.BIT.PF10PR == 0 ){
245         /* wait */
246     }
247     /* ==== SPI 転送終了 ==== */
248     RQSPI.SPCR.BYTE = SPCR_SPI_DISABLE;
249 }
250 /*"FUNC COMMENT"*****
251 * ID          :
252 * Outline     : コマンドレジスタの更新
253 *-----
254 * Include     : iodef.h
255 *-----
256 * Declaration : static int io_rqspi_update_SPCMD_reg( int seq );
257 *-----
258 * Description : シーケンス制御レジスタおよびコマンドレジスタにシーケンス制御情報を
259 *              : 反映します。
260 *-----
261 * Argument    : int seq ; I : 使用するコマンドレジスタ数-1 (SPCMD0 のみの場合:0)
262 *-----
263 * Return Value : 0 : 正常終了
264 *-----
265 * Note        : None
266 *"FUNC COMMENT END"*****/
267 static int io_rqspi_update_SPCMD_reg( int seq )
268 {
269     RQSPI_CMD_ST *cmd = rqspi_cmd_set;
270
271     /* ==== シーケンス制御レジスタの設定 ==== */
272     RQSPI.SPSCR.BYTE = seq;
273
274     /* ==== コマンドレジスタの設定 ==== */
275     /* ---- CMD3 ---- */
276     RQSPI.SPCMD3.WORD = cmd[3].spcmd;
277     RQSPI.SPB MUL3.LONG = cmd[3].spbmul;
278     /* ---- CMD2 ---- */
279     RQSPI.SPCMD2.WORD = cmd[2].spcmd;
280     RQSPI.SPB MUL2.LONG = cmd[2].spbmul;
281     /* ---- CMD1 ---- */
282     RQSPI.SPCMD1.WORD = cmd[1].spcmd;
283     RQSPI.SPB MUL1.LONG = cmd[1].spbmul;
284     /* ---- CMD0 ---- */
285     RQSPI.SPCMD0.WORD = cmd[0].spcmd;
286     RQSPI.SPB MUL0.LONG = cmd[0].spbmul;
287
288     return 0;
289 }
```

3.14 サンプルプログラムリスト"rqspi.c" (7)

```
290  /*"FUNC COMMENT"*****
291  * ID      :
292  * Outline  : データレジスタへのライト処理 (半二重動作)
293  *-----
294  * Include  : iodef.h
295  *-----
296  * Declaration : static void io_rqspi_write_SPDR_reg( void *wrp, int cnt,
297  *              :                               unsigned short bitsz );
298  *-----
299  * Description : 引数 wrp に格納されたライトデータを、引数 bitsz で指定されたアクセス
300  *              : サイズでデータレジスタに書き込みます。引数 cnt の回数だけ繰り返します。
301  *-----
302  * Argument  : void      *wrp ; I : ライトデータへのポインタ
303  *              : int      cnt ; I : 書き込み回数
304  *              : unsigned short bitsz ; I : データレジスタへのアクセスサイズ
305  *-----
306  * Return Value : void
307  *-----
308  * Note      : None
309  *"FUNC COMMENT END"*****/
310  static void io_rqspi_write_SPDR_reg( void *wrp, int cnt, unsigned short bitsz )
311  {
312  if( bitsz == SPCMD_SPB_32BITS ){
313  unsigned long *wrp_l = (unsigned long *)wrp;
314  while( cnt-- ){
315  while(RQSPI.SPSR.BIT.SPTEF == 0){
316  /* 送信バッファエンプティ待ち */
317  }
318  RQSPI.SPDR.LONG = *wrp_l++;
319  }
320  }
321  else if( bitsz == SPCMD_SPB_16BITS ){
322  unsigned short *wrp_w = (unsigned short *)wrp;
323  while( cnt-- ){
324  while(RQSPI.SPSR.BIT.SPTEF == 0){
325  /* 送信バッファエンプティ待ち */
326  }
327  RQSPI.SPDR.WORD = *wrp_w++;
328  }
329  }
330  else{
331  unsigned char *wrp_c = (unsigned char *)wrp;
332  while( cnt-- ){
333  while(RQSPI.SPSR.BIT.SPTEF == 0){
334  /* 送信バッファエンプティ待ち */
335  }
336  RQSPI.SPDR.BYTE = *wrp_c++;
337  }
338  }
339  }
```

3.15 サンプルプログラムリスト"rqspi.c" (8)

```
340  /*"FUNC COMMENT"*****
341  * ID      :
342  * Outline : データレジスタのリード処理 (半二重動作)
343  *-----
344  * Include : iodef.h
345  *-----
346  * Declaration : static void io_rqspi_read_SPDR_reg( void *rdp, int cnt,
347  *          :          unsigned short bitsz );
348  *-----
349  * Description : 引数 bitsz で指定されたアクセスサイズでデータレジスタを読み出し、
350  *          : 引数 rdp で指定されたバッファ領域へ格納します。
351  *          : 引数 cnt の回数だけ繰り返します。
352  *-----
353  * Argument  : void      *rdp ; I : リードデータを格納するバッファアドレス
354  *          : int       cnt ; I : 読み出し回数
355  *          : unsigned short bitsz ; I : データレジスタへのアクセスサイズ
356  *-----
357  * Return Value : void
358  *-----
359  * Note       : None
360  *"FUNC COMMENT END"*****/
361  static void io_rqspi_read_SPDR_reg( void *rdp, int cnt , unsigned short bitsz )
362  {
363      if( bitsz == SPCMD_SPB_32BITS ){
364          unsigned long *rdp_l = (unsigned long *)rdp;
365          while( cnt-- ){
366              while( RQSPI.SPSR.BIT.SPRFF == 0 ){
367                  /* 受信バッファフル待ち */
368              }
369              *rdp_l++ = RQSPI.SPDR.LONG;
370          }
371      }
372      else if( bitsz == SPCMD_SPB_16BITS ){
373          unsigned short *rdp_w = (unsigned short *)rdp;
374          while( cnt-- ){
375              while( RQSPI.SPSR.BIT.SPRFF == 0 ){
376                  /* 受信バッファフル待ち */
377              }
378              *rdp_w++ = RQSPI.SPDR.WORD;
379          }
380      }
381      else{
382          unsigned char *rdp_c = (unsigned char *)rdp;
383          while( cnt-- ){
384              while( RQSPI.SPSR.BIT.SPRFF == 0 ){
385                  /* 受信バッファフル待ち */
386              }
387              *rdp_c++ = RQSPI.SPDR.BYTE;
388          }
389      }
390  }
```

3.16 サンプルプログラムリスト"rqspi.c" (9)

```

391  /*"FUNC COMMENT"*****
392  * ID      :
393  * Outline  : データレジスタのリードライト処理 (全二重動作)
394  *-----
395  * Include  : iodefine.h
396  *-----
397  * Declaration : static void io_rqspi_rdwr_SPDR_reg( void *wrp, void *rdp,
398  *          :                               int cnt, unsigned short bitsz );
399  *-----
400  * Description : 引数 bitsz で指定されたアクセスサイズでデータレジスタを読み出し、
401  *          : 引数 rdp で指定されたバッファ領域へ格納します。
402  *          : 引数 cnt の回数だけ繰り返します。
403  *-----
404  * Argument  : void      *wrp   ; I : ライトデータへのポインタ
405  *          : void      *rdp   ; I : リードデータを格納するバッファアドレス
406  *          : int       cnt    ; I : 書き込みおよび読み出し回数
407  *          : unsigned short bitsz ; I : データレジスタへのアクセスサイズ
408  *-----
409  * Return Value : void
410  *-----
411  * Note      : None
412  *"FUNC COMMENT END"*****/
413  static void io_rqspi_rdwr_SPDR_reg( void *wrp, void *rdp, int cnt, unsigned short bitsz )
414  {
415      unsigned long tmp = 0;
416
417      /* ==== アクセスサイズ 32 ビットの場合 ==== */
418      if( bitsz == SPCMD_SPB_32BITS ){
419          unsigned long *wrp_l = (unsigned long *)wrp;
420          unsigned long *rdp_l = (unsigned long *)rdp;
421          while( cnt-- ){
422              /* ---- ライト処理 ---- */
423              while(RQSPI.SPSR.BIT.SPTEF == 0){
424                  /* 送信バッファエンブティ待ち */
425              }
426              if( wrp != NULL){
427                  RQSPI.SPDR.LONG = *wrp_l++;
428              }
429              else{
430                  RQSPI.SPDR.LONG = tmp; /* ダミー送信 */
431              }
432              /* ---- リード処理 ---- */
433              while( RQSPI.SPSR.BIT.SPRFF == 0 ){
434                  /* 受信バッファフル待ち */
435              }
436              if(rdp != NULL){
437                  *rdp_l++ = RQSPI.SPDR.LONG;
438              }
439              else{
440                  tmp = RQSPI.SPDR.LONG; /* ダミー受信 */
441              }
442          }
443      }

```

3.17 サンプルプログラムリスト"rqspi.c" (10)

```
444  /* ==== アクセスサイズ 16 ビットの場合 ==== */
445  else if( bitsz == SPCMD_SPB_16BITS ){
446      unsigned short *wrp_w = (unsigned short *)wrp;
447      unsigned short *rdp_w = (unsigned short *)rdp;
448      while( cnt-- ){
449          /* ---- ライト処理 ---- */
450          while(RQSPI.SPSR.BIT.SPTEF == 0){
451              /* 送信バッファエンプティ待ち */
452          }
453          if( wrp != NULL){
454              RQSPI.SPDR.WORD = *wrp_w++;
455          }
456          else{
457              RQSPI.SPDR.WORD = (unsigned short)tmp; /* ダミー送信 */
458          }
459          /* ---- リード処理 ---- */
460          while( RQSPI.SPSR.BIT.SPRFF == 0 ){
461              /* 受信バッファフル待ち */
462          }
463          if(rdp != NULL){
464              *rdp_w++ = RQSPI.SPDR.WORD;
465          }
466          else{
467              tmp = RQSPI.SPDR.WORD; /* ダミー受信 */
468          }
469      }
470  }
```

3.18 サンプルプログラムリスト"rqspi.c" (11)

```
471  /* ==== アクセスサイズ 8 ビットの場合 ==== */
472  else{
473      unsigned char *wrp_c = (unsigned char *)wrp;
474      unsigned char *rdp_c = (unsigned char *)rdp;
475      while( cnt-- ){
476          /* ---- ライト処理 ---- */
477          while(RQSPI.SPSR.BIT.SPTEF == 0){
478              /* 送信バッファエンプティ待ち */
479          }
480          if( wrp != NULL){
481              RQSPI.SPDR.BYTE = *wrp_c++;
482          }
483          else{
484              RQSPI.SPDR.BYTE = (unsigned char)tmp; /* ダミー送信 */
485          }
486          /* ---- リード処理 ---- */
487          while( RQSPI.SPSR.BIT.SPRFF == 0 ){
488              /* 受信バッファフル待ち */
489          }
490          if(rdp != NULL){
491              *rdp_c++ = RQSPI.SPDR.BYTE;
492          }
493          else{
494              tmp = RQSPI.SPDR.BYTE; /* ダミー受信 */
495          }
496      }
497  }
498 }
499 /* End of File */
```

3.19 サンプルプログラムリスト"rqspi.h" (1)

```
1  /*****
2  *   DISCLAIMER
3
4  (省略)
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27  ****
28  *   Copyright (C) 2010 Renesas Electronics Corporation. All Rights Reserved.
29  *   "FILE COMMENT"***** Technical reference data *****
30  *   System Name : SH7267 Sample Program
31  *   File Name   : rqspi.h
32  *   Abstract    : RQSPI クワッドシリアルフラッシュメモリ接続例
33  *   Version     : 1.00.00
34  *   Device      : SH7266/SH7267
35  *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36  *                : C/C++ compiler package for the SuperH RISC engine family
37  *                :                               (Ver.9.03 Release02).
38  *   OS          : None
39  *   H/W Platform: R0K57267(CPU board), Spansion[S25FL032P](serial flash)
40  *   Description :
41  ****
42  *   History     : Aug.20,2010 Ver.1.00.00
43  *   "FILE COMMENT END"*****/
44  #ifndef _RQSPI_H_
45  #define _RQSPI_H_
46
47  /* ==== マクロ定義 ==== */
48  #define SPI_SINGLE      0x0000 /* io_rqapi_set_cmd 関数の mode に指定 (Single-SPI) */
49  #define SPI_DUAL_WR     0x0020 /* io_rqapi_set_cmd 関数の mode に指定 (Dual-SPI+ライト) */
50  #define SPI_DUAL_RD     0x0030 /* io_rqapi_set_cmd 関数の mode に指定 (Dual-SPI+リード) */
51  #define SPI_QUAD_WR     0x0040 /* io_rqapi_set_cmd 関数の mode に指定 (Quad-SPI+ライト) */
52  #define SPI_QUAD_RD     0x0050 /* io_rqapi_set_cmd 関数の mode に指定 (Quad-SPI+リード) */
53
54  /* ==== 関数プロトタイプ宣言 ==== */
55  void io_rqspi_initialize(void);
56  void io_rqspi_set_cmd( int idx, unsigned short mode, void *wrp, void *rdp, unsigned long sz);
57  int io_rqspi_transfer( int seq );
58
59  #endif /* _RQSPI_H_ */
60  /* End of File */
61
```

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
SH7266 グループ、SH7267 グループ ハードウェアマニュアル Rev.1.00
(最新版をルネサス エレクトロニクスホームページから入手してください。)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.10.29	—	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っていると、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て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きを行ってください。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、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とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

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



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・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>