

SH7266/SH7267 グループ

R01AN0647JJ0100

Rev. 1.00

2011.09.09

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の高速リード例

要旨

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) は、マルチ I/O に対応したシリアルフラッシュメモリをアクセスすることができ、リード処理を最大 72MHz の高速なクロックで実行できます。本アプリケーションノートは、RQSPI のリード処理を高速化する手法について説明します。

動作確認デバイス

SH7267

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.....	7
4. 参考ドキュメント.....	31

1. はじめに

1.1 仕様

本アプリケーションノートは、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) で高速リードを行う応用例の説明と参考プログラムで構成しています。なお、RQSPI の基本的な使用方法はアプリケーションノート「SH7266/SH7267 グループ 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」で説明しています。そのため、本アプリケーションノートでは高速リードを行う場合の変更点のみを説明します。

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 -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。併せて参照してください。

- SH7266/SH7267 グループ 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
- 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
- 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 (マルチ I/O 対応版)

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、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) にシリアルフラッシュメモリを接続して、72MHz の QSPCLK でデータを高速にリードします。(Bφ=72MHz 時) *

36MHz の QSPCLK でデータをリードライトする手順は、アプリケーションノート「SH7266/SH7267 グループ 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」で説明しています。ここでは 72MHz の QSPCLK でリードするために、セットアップ期間を拡張する方法と参考プログラムの変更箇所について説明します。

【注】ライト時は QSCLK を 72MHz に設定できません。

2.1 セットアップ期間の拡張

アクセス速度は QSPCLK の周波数に比例します。SH7267 の RQSPI は、受信動作に限り QSPCLK を 72MHz に設定できるため、本応用例ではデータリード時の QSPCLK を 72MHz に設定します。そしてデータ入力セットアップ時間 (t_{su}) を満たすためにデータラッチタイミングを 1/2 サイクル遅らせています。

図 1 に本応用例のリード処理タイミングを示します。

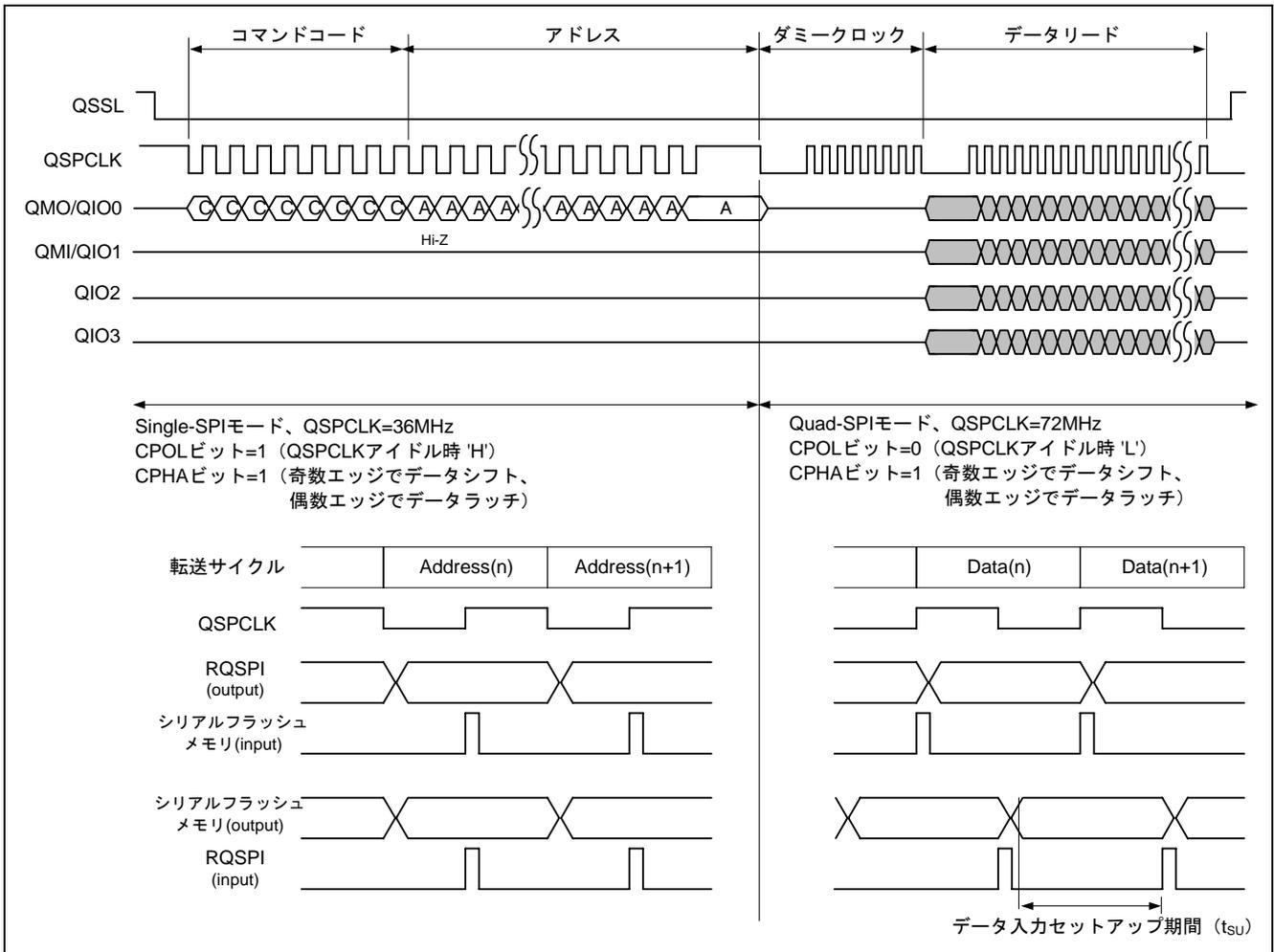


図1 本応用例のリード処理タイミング

2.2 インタフェースタイミング

図2にセットアップ期間拡張時のインタフェースタイミングを示します。表1および表2にシリアルフラッシュメモリおよびSH7267のタイミング条件を示します。これらの条件を満たすように設定を行ってください。

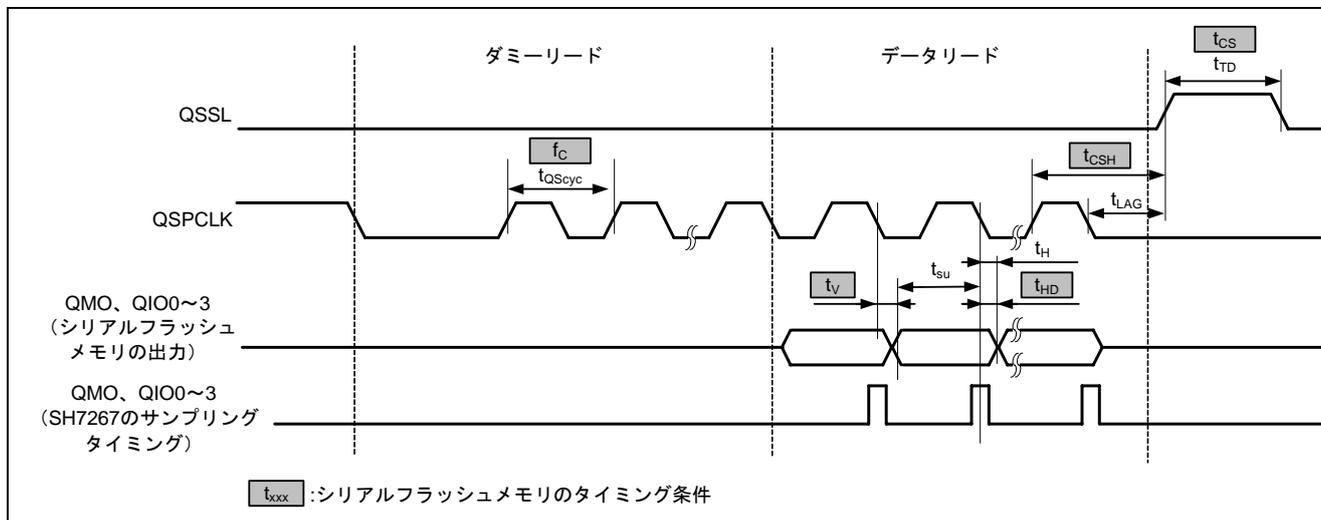


図2 セットアップ期間拡張時のインタフェースタイミング

表1 セットアップ期間拡張時の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_{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。 $t_{TD}(=次アクセス遅延) \times t_{QScyc} \times t_{cyc} \geq t_{CS}(\min)$	SPND レジスタ SPCMD レジスタ
f_{SCK}	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。 $f_c(\max) \geq 1/(t_{QScyc} \times t_{cyc})$	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。 $t_{LAG}(=QSSL \text{ ネゲート遅延}) \times t_{QScyc} \times t_{cyc} \geq t_{CSH}(\min)$	SSLND レジスタ SPCMD レジスタ

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

表2 セットアップ期間拡張時の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リング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。 $t_{QScyc} \times t_{cyc} - t_v(\max) \geq t_{SU}(\min)$	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。 $t_{HO}(\min) \geq t_H(\min)$	

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

2.3 参考プログラムの変更点

ここでは本応用例の参考プログラムにおける、アプリケーションノート「SH7266/SH7267 グループ 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」の参考プログラムからの変更点について説明します。

図 3に本応用例の参考プログラムで使用するシーケンス制御を示します。

Quad Output Read コマンド (H'6B) のリード処理を 72MHz のクロックで実行するためにコマンドレジスタ 1 (SPCMD1) およびコマンドレジスタ 2 (SPCMD2) の設定値を変更します。変更後の設定値を表 3 に示します。またデータレジスタ (SPDR) の読み出しは CPU ではなく DMAC を使用した 16 バイトバースト転送で行います。DMAC の設定値を表 4 に示します。

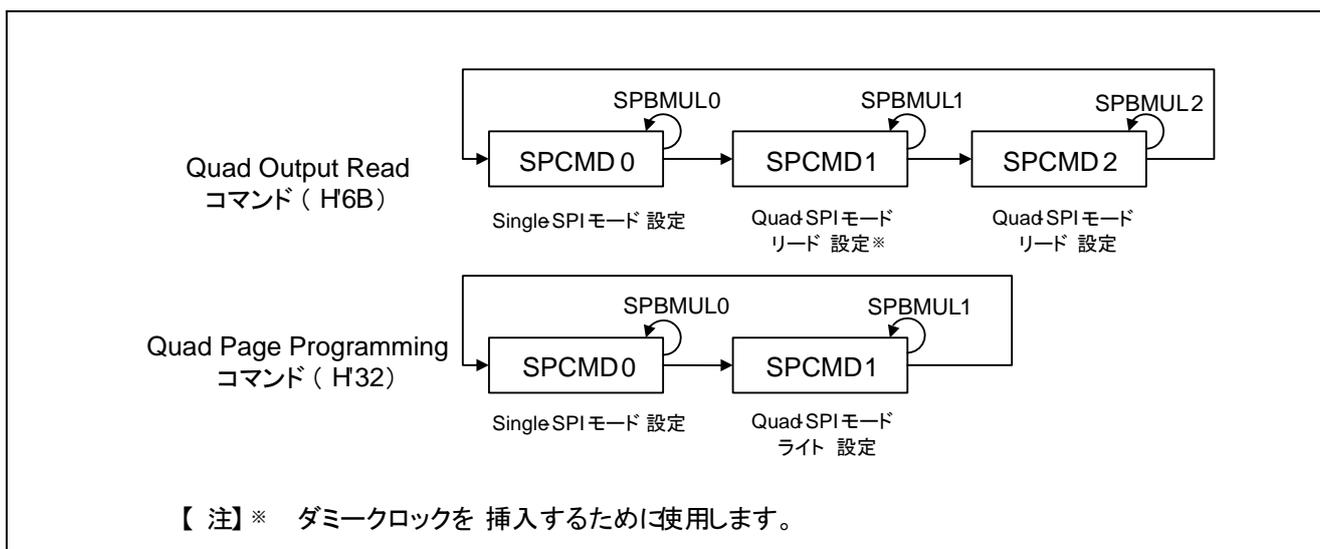


図3 参考プログラムのシーケンス制御

表3 高速リード時の SPCMDn レジスタ設定値

レジスタ名	設定値	説明
SPCMD0	H'E287	<ul style="list-style-type: none"> SCKDEN="B'1": クロック遅延期間は SPCKD 設定値 SLNDEN="B'1": QSSL ネゲート遅延期間は SSLND 設定値 SPNDEN="B'1": 次アクセス遅延期間は SPND 設定値 LSBF="B'0": MSB ファースト SPB[3:0]="B'0010": 転送データ長 32 ビット SSLKP="B'1": 転送終了後も QSSL 信号を保持 SPIMOD[1:0]="B'00": Single-SPI SPRW="B'0": ライト動作 BRDV[1:0]="B'01": ベースのビットレートの 2 分周 (36MHz) CPOL="B'1": アイドル時に QSPCLK は 1 CPHA="B'1": 奇数エッジでデータシフト、偶数エッジでデータラッチ
SPCMD1	H'E2D1	<ul style="list-style-type: none"> SPIMOD[1:0]="B'10": Quad-SPI SPRW="B'1": リード動作 BRDV[1:0]="B'00": ベースのビットレート (72MHz) CPOL="B'0": アイドル時に QSPCLK は 0 (その他のビット設定値は SPCMD0 レジスタと同様)
SPCMD2	H'E2D1	(SPCMD1 レジスタと同じ設定値)
SPSCR	H'02	<ul style="list-style-type: none"> SPSC[1:0]="B'10": 参照する SPCMD 番号 (0→1→2→0→・・・)

表4 DMAC のレジスタ設定値

レジスタ名	設定値	説明
CHCR4	H'02004818	<ul style="list-style-type: none">• TC=B'0 : 1 回の転送要求で 1 回転送• SAF=B'1 : SAR レジスタから 4 バイトを 4 回転送• DM[1:0]=B'01 : デスティネーションアドレスは増加• SM[1:0]=B'00 : ソースアドレスは固定• RS[3:0]=B'1000 : 転送要求元は DMA 拡張リソースセクタ• TB=B'0 : サイクルスチールモード• TS[1:0]=B'11 : 転送単位は 16 バイト (ロングワード×4) 単位• IE=B'0 : 割り込み要求を禁止• DE=B'1 : DMA 転送を許可
SAR4	H'FFFFFFE004	RQSPI の SPDR レジスタ
DAR4	-	リード用バッファのアドレス
DMARS2	H'A2	<ul style="list-style-type: none">• 転送要求元は RQSPI の受信
DMAOR	H'0001	<ul style="list-style-type: none">• 全チャンネルの DMA 転送を許可

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 (マルチ I/O 対応版)」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 : SH7266/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の高速リード例
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(2011) 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の高速リード例
33 *  Version     : 1.02.00
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 *              : 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 *              : Apr.14,2011 Ver.1.02.00 ダミーリードデータを一時格納するバッファを追加
45 *  "FILE COMMENT END" *****/
46 #include <stdio.h>
47 #include <machine.h>
48 #include "iodefine.h"
49 #include "qserial_flash.h"
50 #include "rqspi.h"
```

3.6 サンプルプログラムリスト"qserial_flash.c" (2)

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

    (省略)

286  /*"FUNC COMMENT"*****
287  * ID          :
288  * Outline     : データリード
289  *-----
290  * Include     :
291  *-----
292  * Declaration : void sf_byte_read(unsigned long addr, unsigned char *buf, int size);
293  *-----
294  * 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。
295  *-----
296  * 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
297  *              : unsigned char *buf ; I : リードデータを格納するバッファのアドレス
298  *              : int size          ; I : リードするバイト数
299  *-----
300  * Return Value : void
301  *-----
302  * Note        : None
303  *"FUNC COMMENT END"*****/
304  void sf_byte_read(unsigned long addr, unsigned char *buf, int size)
305  {
306      unsigned char cmd[4];
307      unsigned long dummy[2];
308
309      cmd[0] = SFLASHCMD_QUAD_READ;
310      cmd[1] = (unsigned char)((addr >> 16) & 0xff);
311      cmd[2] = (unsigned char)((addr >> 8) & 0xff);
312      cmd[3] = (unsigned char)( addr      & 0xff);
313
314      io_rqspi_set_cmd( 0, SPI_SINGLE, cmd, NULL, sizeof(cmd));
315      io_rqspi_set_cmd( 1, SPI_QUAD_RD, NULL, dummy, 4); /* (2clk/byte)×4 = 8clk */
316      io_rqspi_set_cmd( 2, SPI_QUAD_RD, NULL, buf, size);
317      io_rqspi_transfer(2);
318  }

    (以下、省略)

481  /* 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の高速リード例
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(2011) 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の高速リード例
33 *  Version     : 1.01.00
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 *              : Apr.14,2011 Ver.1.01.00 Dual/Quad リード時のクロックを 72MHz に変更
44 *              : SPDR の読み出し前のデータ数確認処理を追加
45 *              : 16byte 単位のリード処理に DMAC を使用
46 *  "FILE COMMENT END"*****/
47 #include <stdio.h>
48 #include <machine.h>
49 #include "iodefine.h"
50 #include "rqspi.h"
51
```

3.9 サンプルプログラムリスト"rqspi.c" (2)

```

52  /* ==== マクロ定義 ==== */
53  #define SPCR_SPI_ENABLE 0x48          /* ビット 3 は常に"1"をライトすること */
54  #define SPCR_SPI_DISABLE 0x08        /* ビット 3 は常に"1"をライトすること */
55  #define SPCMD_SPIRW_BIT 0x0070      /* SPI 動作モードとリードライトの対象ビット */
56  #define SPCMD_SPB_BIT 0x0f00        /* 転送データ長(アクセスサイズ)の対象ビット */
57  #define SPCMD_SPB_8BITS 0x0000
58  #define SPCMD_SPB_16BITS 0x0100
59  #define SPCMD_SPB_32BITS 0x0200
60  #define SPCMD_DEFAULT_SET 0xe087
61          /* bit 15 : クロック遅延      : SPCKD( 1.5 QSPCLK) */
62          /* bit 14 : QSSL ネゲート遅延 : SSLND( 2 QSPCLK) */
63          /* bit 13 : 次アクセス遅延   : SPND( 2 QSPCLK) */
64          /* bit 12 : フォーマット      : MSB ファースト */
65          /* bit 11-8: 転送データ長     : 8 ビット */
66          /* bit 7 : QSSL 信号保持      : 転送終了後も保持 */
67          /* bit 6-5 : SPI 動作モード   : Single-SPI */
68          /* bit 4 : リードライト      : ライト(無効) */
69          /* bit 3-2 : ビットレート     : SPBR を 2 分周(36Mbps) */
70          /* bit 1 : CPOL(QSPCLK 極性) : アイドル時="1" */
71          /* bit 0 : CPHA(QSPCLK 位相) : 奇数エッジでデータシフト*/
72          /*                               : 偶数エッジでデータラッチ*/
73  #define SPCMD_FAST_SET 0xe081
74          /* bit 15 : クロック遅延      : SPCKD( 1.5 QSPCLK) */
75          /* bit 14 : QSSL ネゲート遅延 : SSLND( 2 QSPCLK) */
76          /* bit 13 : 次アクセス遅延   : SPND( 2 QSPCLK) */
77          /* bit 12 : フォーマット      : MSB ファースト */
78          /* bit 11-8: 転送データ長     : 8 ビット */
79          /* bit 7 : QSSL 信号保持      : 転送終了後も保持 */
80          /* bit 6-5 : SPI 動作モード   : Single-SPI */
81          /* bit 4 : リードライト      : ライト(無効) */
82          /* bit 3-2 : ビットレート     : SPBR を 1 分周(72Mbps) */
83          /* bit 1 : CPOL(QSPCLK 極性) : アイドル時="0" */
84          /* bit 0 : CPHA(QSPCLK 位相) : 奇数エッジでデータシフト*/
85          /*                               : 偶数エッジでデータラッチ*/
86
87  /* ---- RQSPI のシーケンス制御情報を定義する構造体 ---- */
88  typedef struct{
89      unsigned short spcmd;          /* SPCMDn レジスタへの設定値 */
90      unsigned long spbmul;         /* SPBMULn レジスタへの設定値 */
91      void *wr_ptr;                 /* 送信データの格納先アドレス */
92      void *rd_ptr;                 /* 受信データの格納先アドレス */
93  }RQSPI_CMD_ST;
94
95  /* ==== 関数プロトタイプ宣言 ==== */
96  extern void io_dmac_read_SPDR_reg_16burst( void *rdp, int cnt );
97
98  static int io_rqspi_update_SPCMD_reg( int seq );
99  static void io_rqspi_write_SPDR_reg( void *wrp, int cnt, unsigned short bitsz );
100 static void io_rqspi_read_SPDR_reg( void *rdp, int cnt , unsigned short bitsz );
101 static void io_rqspi_rdw_r_SPDR_reg( void *wrp, void *rdp, int cnt, unsigned short bitsz );
102

```

3.10 サンプルプログラムリスト"rqspi.c" (3)

```
103  /* ==== 変数定義 ==== */
104  RQSPI_CMD_ST rqspi_cmd_set[4];
105
106  /*"FUNC COMMENT"*****
107  * ID      :
108  * Outline : RQSPI の初期化
109  *-----
110  * Include : iodefine.h
111  *-----
112  * Declaration : static void io_init_rqspi(void);
113  *-----
114  * 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。
115  *              : 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
116  *              : 転送設定を行います。
117  *-----
118  * Argument  : void
119  *-----
120  * Return Value: void
121  *-----
122  * Note      : None
123  *"FUNC COMMENT END"*****/
124  void io_rqspi_initialize(void)
125  {
126  /* ==== クロック供給 ==== */
127  CPG.STBCR8.BIT.MSTP82 = 0;
128
129  /* ==== Port ==== */
130  PORT.PFCR3.BIT.PF12MD = 0x06u; /* QMI/QIO1 */
131  PORT.PFCR2.BIT.PF11MD = 0x06u; /* QMO/QIO0 */
132  PORT.PFCR2.BIT.PF10MD = 0x06u; /* QSSL */
133  PORT.PFCR2.BIT.PF9MD  = 0x06u; /* QSPCLK */
134  PORT.PDCR3.BIT.PD15MD = 0x03u; /* QIO3 */
135  PORT.PDCR3.BIT.PD14MD = 0x03u; /* QIO2 */
136
137  /* ==== SPI 機能の無効化と内部状態の初期化 ==== */
138  RQSPI.SPCR.BYTE = SPCR_SPI_DISABLE;
139
140  /* ==== スレーブセレクト極性レジスタ (SSLP) ==== */
141  RQSPI.SSLP.BIT.SSLP = 0; /* QSSL 信号は Low アクティブ */
142
143  /* ==== 端子制御レジスタ (SPPCR) ==== */
144  RQSPI.SPPCR.BYTE = 0x26; /* 出力端子アイドル値は 0 */
145  /* Single/Dual-SPI 時、QIO3 は 1 固定 */
146  /* Single/Dual-SPI 時、QIO2 は 1 固定 */
147  /* 通常動作モード(非ループバック) */
148  /* ==== バッファコントロールレジスタ (SPBFCR) ==== */
149  RQSPI.SPBFCR.BYTE = 0xC0; /* バッファリセット */
150  RQSPI.SPBFCR.BYTE = 0x25; /* バッファリセット解除 */
151  /* 送信バッファデータ数トリガ 4 バイト空き */
152  /* 受信バッファデータ数トリガ 16 バイト受信 */
153  /* (受信トリガ数は必要に応じて転送前に再設定) */
154
```

3.11 サンプルプログラムリスト"rqspi.c" (4)

```
155  /* ==== ビットレートレジスタ (SPBR) ==== */
156  RQSPI.SPBR.BYTE = 0;          /* ベースビットレート 72Mbps (Bφ=72MHz 時) */
157
158  /* ==== クロック遅延レジスタ (SPCKD) ==== */
159  RQSPI.SPCKD.BYTE = 0x00;     /* SSLセットアップ時間= 1.5 QSPCLK */
160
161  /* ==== スレーブセレクトネゲート遅延レジスタ (SSLND) ==== */
162  RQSPI.SSLND.BYTE = 0x01;     /* SSL ホールド時間= 2 QSPCLK */
163
164  /* ==== 次アクセス遅延レジスタ (SPND) ==== */
165  RQSPI.SPND.BYTE = 0x01;     /* 連続転送遅延時間= 2 QSPCLK */
166
167  /* ==== コマンドレジスタ n (SPCMDn) ==== */
168  RQSPI.SPCMD0.WORD = SPCMD_DEFAULT_SET; /* (転送前に再設定) */
169  RQSPI.SPCMD1.WORD = SPCMD_DEFAULT_SET; /* (転送前に再設定) */
170  RQSPI.SPCMD2.WORD = SPCMD_DEFAULT_SET; /* (転送前に再設定) */
171  RQSPI.SPCMD3.WORD = SPCMD_DEFAULT_SET; /* (転送前に再設定) */
172  }
```

3.12 サンプルプログラムリスト"rqspi.c" (5)

```
173  /*"FUNC COMMENT"*****
174  * ID      :
175  * Outline  : シーケンス制御情報の更新
176  *-----
177  * Include  :
178  *-----
179  * Declaration : void io_rqspi_set_cmd( int idx, unsigned short mode, void *wrp,
180  *      :                               void *rdp, unsigned long sz)
181  *-----
182  * 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。
183  *-----
184  * Argument   : int      idx ; I : 対象のコマンドレジスタ番号 (0~3)
185  *             : unsigned short mode ; I : SPCMD に指定する SPI モードおよび RW 設定
186  *             : void      *wrp ; I : 送信データ格納先アドレス
187  *             : void      *rdp ; O : 受信データ格納先アドレス
188  *             : unsigned long sz ; I : 送受信データ数 (バイト)
189  *-----
190  * Return Value : void
191  *-----
192  * Note       :
193  *"FUNC COMMENT END"*****/
194  void io_rqspi_set_cmd( int idx, unsigned short mode, void *wrp, void *rdp, unsigned long sz)
195  {
196     RQSPI_CMD_ST *cmd = rqspi_cmd_set;
197     unsigned short bitsz = SPCMD_SPB_8BITS;
198     unsigned long trncnt = sz;
199
200     /* ---- 最適なデータサイズを設定 ---- */
201     if( ((sz&0x3)==0) && (((int)wrp&0x3)==0) && (((int)rdp&0x3)==0) ){
202         bitsz = SPCMD_SPB_32BITS;
203         trncnt = sz >> 2;
204     }
205     else if( ((sz&0x1)==0) && (((int)wrp&0x1)==0) && (((int)rdp&0x1)==0) ){
206         bitsz = SPCMD_SPB_16BITS;
207         trncnt = sz >> 1;
208     }
209
210     /* ---- 半二重(リード)動作の場合は高速クロックを使用 --- */
211     if( mode==SPI_QUAD_RD || mode==SPI_DUAL_RD ){
212         cmd[idx].spcmd = (mode | SPCMD_FAST_SET | bitsz);
213     }
214     else{
215         cmd[idx].spcmd = (mode | SPCMD_DEFAULT_SET | bitsz);
216     }
217     cmd[idx].spbmul = trncnt;
218     cmd[idx].wr_ptr = wrp;
219     cmd[idx].rd_ptr = rdp;
220 }
```

3.13 サンプルプログラムリスト"rqspi.c" (6)

```
221  /*"FUNC COMMENT"*****
222  * ID      :
223  * Outline  : RQSPI の転送処理
224  *-----
225  * Include   : iodef.h
226  *-----
227  * Declaration : int io_rqspi_transfer( int seq );
228  *-----
229  * Description : RQSPI の転送処理を行います。
230  *              : 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
231  *              : 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
232  *              : 送受信を行います。
233  *-----
234  * 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)
235  *-----
236  * Return Value : 0 : 正常終了
237  *-----
238  * Note       : None
239  *"FUNC COMMENT END"*****/
240  int io_rqspi_transfer( int seq )
241  {
242      int i, cnt;
243      unsigned short spirw, bitsz;
244
245      /* ==== コマンドレジスタの設定 ==== */
246      rqspi_cmd_set[seq].spcmd &= ~(0x0080);      /* QSSL のネゲート設定を追加 */
247      io_rqspi_update_SPCMD_reg(seq);
248
249      /* ==== SPI 転送許可 ==== */
250      RQSPI.SPCR.BYTE = SPCR_SPI_ENABLE;
251
252      /* ==== データレジスタのリードライト処理 ==== */
253      for( i=0; i<=seq; i++){
254          bitsz = rqspi_cmd_set[i].spcmd & SPCMD_SPB_BIT;
255          spirw = rqspi_cmd_set[i].spcmd & SPCMD_SPIRW_BIT;
256          cnt   = rqspi_cmd_set[i].spbmul;
257
258          /* ---- 半二重(ライト)動作の場合 --- */
259          if( spirw==SPI_QUAD_WR || spirw==SPI_DUAL_WR ){
260              io_rqspi_write_SPDR_reg( rqspi_cmd_set[i].wr_ptr, cnt, bitsz);
261          }
262          /* ---- 半二重(リード)動作の場合 --- */
263          else if( spirw==SPI_QUAD_RD || spirw==SPI_DUAL_RD ){
264              if( (bitsz == SPCMD_SPB_32BITS) && ((cnt & 0x3) == 0) ){
265                  /* 16 バイト単位のリード処理は DMA 転送を使用する */
266                  io_dmac_read_SPDR_reg_16burst( rqspi_cmd_set[i].rd_ptr, (cnt/4) );
267              }
268              else{
269                  io_rqspi_read_SPDR_reg( rqspi_cmd_set[i].rd_ptr, cnt, bitsz);
270              }
271          }
272      }
```

3.14 サンプルプログラムリスト"rqspi.c" (7)

```
272     /* ---- 全二重動作の場合 --- */
273     else{
274         io_rqspi_rdwr_SPDR_reg( rqspi_cmd_set[i].wr_ptr, rqspi_cmd_set[i].rd_ptr, cnt, bitsz);
275     }
276 }
277 /* ==== 転送終了待ち (QSSLがネゲートされるまで) ==== */
278 while( PORT.PFPR0.BIT.PF10PR == 0 ){
279     /* wait */
280 }
281 /* ==== SPI 転送終了 ==== */
282 RQSPI.SPCR.BYTE = SPCR_SPI_DISABLE;
283 }
284 /*"FUNC COMMENT"*****
285 * ID          :
286 * Outline     : コマンドレジスタの更新
287 *-----
288 * Include     : iodef.h
289 *-----
290 * Declaration : static int io_rqspi_update_SPCMD_reg( int seq );
291 *-----
292 * 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
293 *              : 反映します。
294 *-----
295 * 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)
296 *-----
297 * Return Value : 0 : 正常終了
298 *-----
299 * Note        : None
300 *"FUNC COMMENT END"*****
301 static int io_rqspi_update_SPCMD_reg( int seq )
302 {
303     RQSPI_CMD_ST *cmd = rqspi_cmd_set;
304
305     /* ==== シーケンス制御レジスタの設定 ==== */
306     RQSPI.SPSCR.BYTE = seq;
307
308     /* ==== コマンドレジスタの設定 ==== */
309     /* ---- CMD3 ---- */
310     RQSPI.SPCMD3.WORD = cmd[3].spcmd;
311     RQSPI.SPB MUL3.LONG = cmd[3].spbmul;
312     /* ---- CMD2 ---- */
313     RQSPI.SPCMD2.WORD = cmd[2].spcmd;
314     RQSPI.SPB MUL2.LONG = cmd[2].spbmul;
315     /* ---- CMD1 ---- */
316     RQSPI.SPCMD1.WORD = cmd[1].spcmd;
317     RQSPI.SPB MUL1.LONG = cmd[1].spbmul;
318     /* ---- CMD0 ---- */
319     RQSPI.SPCMD0.WORD = cmd[0].spcmd;
320     RQSPI.SPB MUL0.LONG = cmd[0].spbmul;
321
322     return 0;
323 }
```

3.15 サンプルプログラムリスト"rqspi.c" (8)

```
324  /*"FUNC COMMENT"*****
325  * ID      :
326  * Outline  : データレジスタへのライト処理 (半二重動作)
327  *-----
328  * Include  : iodef.h
329  *-----
330  * Declaration : static void io_rqspi_write_SPDR_reg( void *wrp, int cnt,
331  *          :          unsigned short bitsz );
332  *-----
333  * 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
334  *          : 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。
335  *-----
336  * Argument  : void          *wrp   ; I : ライトデータへのポインタ
337  *          : int           cnt    ; I : 書き込み回数
338  *          : unsigned short bitsz ; I : データレジスタへのアクセスサイズ
339  *-----
340  * Return Value : void
341  *-----
342  * Note      : None
343  *"FUNC COMMENT END"*****/
344  static void io_rqspi_write_SPDR_reg( void *wrp, int cnt, unsigned short bitsz )
345  {
346  if( bitsz == SPCMD_SPB_32BITS ){
347  unsigned long *wrp_l = (unsigned long *)wrp;
348  while( cnt-- ){
349  while(RQSPI.SPSR.BIT.SPTEF == 0){
350  /* 送信バッファエンプティ待ち */
351  }
352  RQSPI.SPDR.LONG = *wrp_l++;
353  }
354  }
355  else if( bitsz == SPCMD_SPB_16BITS ){
356  unsigned short *wrp_w = (unsigned short *)wrp;
357  while( cnt-- ){
358  while(RQSPI.SPSR.BIT.SPTEF == 0){
359  /* 送信バッファエンプティ待ち */
360  }
361  RQSPI.SPDR.WORD = *wrp_w++;
362  }
363  }
364  else{
365  unsigned char *wrp_c = (unsigned char *)wrp;
366  while( cnt-- ){
367  while(RQSPI.SPSR.BIT.SPTEF == 0){
368  /* 送信バッファエンプティ待ち */
369  }
370  RQSPI.SPDR.BYTE = *wrp_c++;
371  }
372  }
373  }
```

3.16 サンプルプログラムリスト"rqspi.c" (9)

```

374  /*"FUNC COMMENT"*****
375  * ID      :
376  * Outline  : データレジスタのリード処理 (半二重動作)
377  *-----
378  * Include  : iodef.h
379  *-----
380  * Declaration : static void io_rqspi_read_SPDR_reg( void *rdp, int cnt,
381  *          :          unsigned short bitsz );
382  *-----
383  * 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、
384  *          : 引数 rdp で指定されたバッファ領域へ格納します。
385  *          : 引数 cnt の回数だけ繰り返します。
386  *-----
387  * Argument  : void          *rdp ; I : リードデータを格納するバッファアドレス
388  *          : int           cnt  ; I : 読み出し回数
389  *          : unsigned short bitsz ; I : データレジスタへのアクセスサイズ
390  *-----
391  * Return Value : void
392  *-----
393  * Note      : None
394  *"FUNC COMMENT END"*****/
395  static void io_rqspi_read_SPDR_reg( void *rdp, int cnt , unsigned short bitsz )
396  {
397  if( bitsz == SPCMD_SPB_32BITS ){
398  unsigned long *rdp_l = (unsigned long *)rdp;
399  while( cnt-- ){
400  while( RQSPI.SPBDCR.BIT.RXBC < 4 ){
401  /* データ受信待ち */
402  }
403  *rdp_l++ = RQSPI.SPDR.LONG;
404  }
405  }
406  else if( bitsz == SPCMD_SPB_16BITS ){
407  unsigned short *rdp_w = (unsigned short *)rdp;
408  while( cnt-- ){
409  while( RQSPI.SPBDCR.BIT.RXBC < 2 ){
410  /* データ受信待ち */
411  }
412  *rdp_w++ = RQSPI.SPDR.WORD;
413  }
414  }
415  else{
416  unsigned char *rdp_c = (unsigned char *)rdp;
417  while( cnt-- ){
418  while( RQSPI.SPBDCR.BIT.RXBC < 1 ){
419  /* データ受信待ち */
420  }
421  *rdp_c++ = RQSPI.SPDR.BYTE;
422  }
423  }
424  }

```

3.17 サンプルプログラムリスト"rqspi.c" (10)

```
425 /*"FUNC COMMENT"*****
426 * ID      :
427 * Outline : データレジスタのリードライト処理 (全二重動作)
428 *-----
429 * Include : iodef.h
430 *-----
431 * Declaration : static void io_rqspi_rdwr_SPDR_reg( void *wrp, void *rdp,
432 *           :                               int cnt, unsigned short bitsz );
433 *-----
434 * 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、
435 *           : 引数 rdp で指定されたバッファ領域へ格納します。
436 *           : 引数 cnt の回数だけ繰り返します。
437 *-----
438 * Argument  : void      *wrp ; I : ライトデータへのポインタ
439 *           : void      *rdp ; I : リードデータを格納するバッファアドレス
440 *           : int       cnt  ; I : 書き込みおよび読み出し回数
441 *           : unsigned short bitsz ; I : データレジスタへのアクセスサイズ
442 *-----
443 * Return Value : void
444 *-----
445 * Note       : None
446 *"FUNC COMMENT END"*****/
447 static void io_rqspi_rdwr_SPDR_reg( void *wrp, void *rdp, int cnt, unsigned short bitsz )
448 {
449     unsigned long tmp = 0;
450
451     /* ==== アクセスサイズ 32 ビットの場合 ==== */
452     if( bitsz == SPCMD_SPB_32BITS ){
453         unsigned long *wrp_l = (unsigned long *)wrp;
454         unsigned long *rdp_l = (unsigned long *)rdp;
455         while( cnt-- ){
456             /* ---- ライト処理 ---- */
457             while(RQSPI.SPSR.BIT.SPTEF == 0){
458                 /* 送信バッファエンブティ待ち */
459             }
460             if( wrp != NULL){
461                 RQSPI.SPDR.LONG = *wrp_l++;
462             }
463             else{
464                 RQSPI.SPDR.LONG = tmp; /* ダミー送信 */
465             }
466             /* ---- リード処理 ---- */
467             while( RQSPI.SPBDCR.BIT.RXBC < 4 ){
468                 /* データ受信待ち */
469             }
470             if(rdp != NULL){
471                 *rdp_l++ = RQSPI.SPDR.LONG;
472             }
473             else{
474                 tmp = RQSPI.SPDR.LONG; /* ダミー受信 */
475             }
476         }
477     }
```

3.18 サンプルプログラムリスト"rqspi.c" (11)

```
478     /* ==== アクセスサイズ 16 ビットの場合 ==== */
479     else if( bitsz == SPCMD_SPB_16BITS ){
480         unsigned short *wrp_w = (unsigned short *)wrp;
481         unsigned short *rdp_w = (unsigned short *)rdp;
482         while( cnt-- ){
483             /* ---- ライト処理 ---- */
484             while(RQSPI.SPSR.BIT.SPTEF == 0){
485                 /* 送信バッファエンプティ待ち */
486             }
487             if( wrp != NULL){
488                 RQSPI.SPDR.WORD = *wrp_w++;
489             }
490             else{
491                 RQSPI.SPDR.WORD = (unsigned short)tmp; /* ダミー送信 */
492             }
493             /* ---- リード処理 ---- */
494             while( RQSPI.SPBDCR.BIT.RXBC < 2 ){
495                 /* データ受信待ち */
496             }
497             if(rdp != NULL){
498                 *rdp_w++ = RQSPI.SPDR.WORD;
499             }
500             else{
501                 tmp = RQSPI.SPDR.WORD; /* ダミー受信 */
502             }
503         }
504     }
```

3.19 サンプルプログラムリスト"rqspi.c" (12)

```
505     /* ==== アクセスサイズ 8 ビットの場合 ==== */
506     else{
507         unsigned char *wrp_c = (unsigned char *)wrp;
508         unsigned char *rdp_c = (unsigned char *)rdp;
509         while( cnt-- ){
510             /* ---- ライト処理 ---- */
511             while(RQSPI.SPSR.BIT.SPTEF == 0){
512                 /* 送信バッファエンプティ待ち */
513             }
514             if( wrp != NULL){
515                 RQSPI.SPDR.BYTE = *wrp_c++;
516             }
517             else{
518                 RQSPI.SPDR.BYTE = (unsigned char)tmp; /* ダミー送信 */
519             }
520             /* ---- リード処理 ---- */
521             while( RQSPI.SPBDCR.BIT.RXBC < 1 ){
522                 /* データ受信待ち */
523             }
524             if(rdp != NULL){
525                 *rdp_c++ = RQSPI.SPDR.BYTE;
526             }
527             else{
528                 tmp = RQSPI.SPDR.BYTE; /* ダミー受信 */
529             }
530         }
531     }
532 }
533 /* End of File */
```

3.20 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の高速リード例
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
```

3.21 サンプルプログラムリスト"dmac.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 *"FILE COMMENT"***** Technical reference data *****/
30 *   System Name : SH7267 Sample Program
31 *   File Name   : dmac.c
32 *   Abstract    : RQSPI クワッドシリアルフラッシュメモリの高速リード例
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     : Apr.14,2011 Ver.1.00.00
43 *"FILE COMMENT END"*****/
44 #include "iodefine.h"
45
46 /* ==== プロトタイプ宣言 ==== */
47 void io_dmac_read_SPDR_reg_16burst( void *rdp, int cnt );
48 static void io_init_dmac4(void *src, void *dst, int count);
49
```

3.22 サンプルプログラムリスト"dmac.c" (2)

```
50  /*"FUNC COMMENT"*****
51  * ID      :
52  * Outline : DMAC setting
53  *-----
54  * Include : iodef.h
55  *-----
56  * Declaration : void io_dmac_read_SPDR_reg_16burst( void *rdp, int cnt );
57  *-----
58  * Description : ダイレクトメモリアクセスコントローラ(DMAC)チャンネル4の初期設定を行います。
59  *              : 内蔵周辺モジュールリクエスト(SPRI)を使用してRQSPIから受信データを
60  *              : 転送します。RQSPIをDual/Quad-SPI動作(受信)に設定した後に本モジュール
61  *              : を実行してください。
62  *              : 転送単位は16バイト(4×4)です。
63  *-----
64  * Argument  : void      *rdp ; I : リードデータを格納するバッファアドレス
65  *            : int       cnt ; I : 読み出し回数
66  *            : unsigned short bitsz ; I : データレジスタへのアクセスサイズ
67  *-----
68  * Return Value : void
69  *-----
70  * Note      :
71  *"FUNC COMMENT END"*****
72 void io_dmac_read_SPDR_reg_16burst( void *rdp, int cnt )
73 {
74     unsigned long *src = (unsigned long *)&RQSPI.SPDR.LONG;
75
76     /* ==== 受信データ数トリガを設定 ==== */
77     RQSPI.SPBFCCR.BIT.RXTRG = 5; /* 16バイト以上 */
78     while( RQSPI.SPBDCCR.BIT.RXBC < 16 ){
79         /* 変更前のトリガ数でSPRFFがセットされている可能性があるため確認 */
80     }
81     /* ==== DMA転送を設定 ==== */
82     io_init_dmac4( src, rdp, cnt );
83
84     /* ==== RQSPIの受信割り込み要求を許可 ==== */
85     RQSPI.SPCR.BIT.SPRIE = 1;
86
87     /* ==== 転送完了待ち ==== */
88     while(DMAC.CHCR4.BIT.TE == 0) {
89         /* wait */
90     }
91     RQSPI.SPCR.BIT.SPRIE = 0; /* 受信割り込み要求を禁止 */
92     DMAC.CHCR4.BIT.DE = 0; /* 転送禁止 */
93     DMAC.CHCR4.BIT.TE = 0; /* TEフラグのクリア */
94 }
95
```

3.23 サンプルプログラムリスト"dmac.c" (3)

```

96  /*"FUNC COMMENT"*****
97  * ID      :
98  * Outline : DMAC setting
99  *-----
100 * Include  : iodef.h
101 *-----
102 * Declaration : static void io_init_dmac4(void *src, void *dst, int count, unsigned short bitsz);
103 *-----
104 * Description : ダイレクトメモリアクセスコントローラ(DMAC)チャンネル4の初期設定を行います。
105 *              : 内蔵周辺モジュールリクエスト(SPRI)を使用して、RQSPIから受信データを
106 *              : 転送します。リロード機能は使用しません。
107 *-----
108 * Argument  : void *src ; I : 転送元アドレス
109 *              : void *dst ; O : 転送先アドレス
110 *              : int count ; I : 転送回数
111 *-----
112 * Return Value : void
113 *-----
114 * Note      :
115 * "FUNC COMMENT END"*****
116 static void io_init_dmac4(void *src, void *dst, int count)
117 {
118     /* ==== スタンバイコントロールレジスタ2の設定 ==== */
119     CPG.STBCR2.BIT.MSTP8 = 0;          /* DMACのモジュールスタンバイ解除 */
120
121     /* ====DMA_ch4 転送禁止 ==== */
122     DMAC.CHCR4.BIT.DE = 0x0;          /* DMA 禁止 */
123
124     /* ====DMA ソースアドレスレジスタ_4 (SAR_4) の設定 ==== */
125     DMAC.SAR4.LONG = (unsigned long)src;
126
127     /* ====DMA デスティネーションアドレスレジスタ_4 (DAR_4) の設定 ==== */
128     DMAC.DAR4.LONG = (unsigned long)dst;
129
130     /* ====DMA トランスファカウントレジスタ_4 (DMATCR_4) の設定 ==== */
131     DMAC.DMATCR4.LONG = count;
132
133     /* ====DMA チャンネルコントロールレジスタ_4 (CHCR_4) の設定 ==== */
134     DMAC.CHCR4.LONG = 0x02004818;
135     /*
136         bit31 : TC DMATCR 転送 : 0----- 1回転送
137         bit30 : reserve 0
138         bit29 : RLDSAR ON : 0----- リロード機能無効 (RSAR)
139         bit28 : RLDDAR ON : 0----- リロード機能無効 (RDAR)
140         bit27 : reserve 0
141         bit26 : DAF : 0----- 未使用
142         bit25 : SAF : 1----- 4バイト×4の16バイト転送
143         bit24 : reserve 0
144         bit23 : DO over run0 : 0----- 未使用
145         bit22 : TL TEND low active : 0---- 未使用
146     */

```

3.24 サンプルプログラムリスト"dmac.c" (4)

```
147          bit21      : reserve 0
148          bit20      : TEMASK :0----- 未使用
149          bit19      : HE :0----- 未使用
150          bit18      : HIE :0----- 未使用
151          bit17      : AM :0----- 未使用
152          bit16      : AL :0----- 未使用
153          bit15-14   : DM1:0 DM0:1----- デスティネーションアドレス増加
154          bit13-12   : SM1:0 SM0:0----- ソースアドレス固定
155          bit11-8    : RS : B'1000----- 転送要求元は 拡張リソースセクタ
156          bit7       : DL : DREQ level : 0 ----- 未使用
157          bit6       : DS : DREQ select :0 Low level 未使用
158          bit5       : TB :cycle :0----- サイクルスチールモード
159          bit4-3     : TS : transfer size :B'11--- 16 バイト転送
160          bit2       : IE : interrupt enable :0--- 割り込み禁止
161          bit1       : TE : transfer end :0----- TE フラグのクリア
162                                     (1 リード後の 0 クリアのみ有効)
163          bit0       : DE : DMA enable bit :0----- DMA 転送禁止
164          */
165          /* ====DMA 拡張リソースセクタ 2 (DMARS2) の設定 ==== */
166          DMAC.DMARS2.BYTE.CH4 = 0xA2;          /* RQSPI 受信 */
167
168          /* ----DMA オペレーションレジスタ (DMAOR) の設定---- */
169          DMAC.DMAOR.WORD |= 0x0007;          /* DME ビットをセット。アドレスエラーフラグ、*/
170                                     /* NMI フラグのクリア防止のため、AE ビット、*/
171                                     /* NMIF フラグには 1 をライト*/
172          /* =====DMA_ch4 転送許可===== */
173          DMAC.CHCR4.BIT.DE = 0x1;
174      }
175      /* 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.4.00
(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 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/contact/>

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	2011.09.09	—	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くと、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>