

SH7262/SH7264 グループ

R01AN0547JJ0101

Rev. 1.01

シリアルサウンドインタフェース マスタランシーバ設定例

2011.02.23

要旨

本アプリケーションノートは、SH7262/SH7264 のシリアルサウンドインタフェース (SSI) を使用したマスタランシーバ設定例について説明します。

動作確認デバイス

SH7262/SH7264

以下、総称して「SH7264」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	15
4. 参考ドキュメント	28

1. はじめに

1.1 仕様

シリアルサウンドインタフェース (SSI) をマスタランシーバモードに設定し、PCM データの全二重通信を行います。

SSI とのデータ転送にはダイレクトメモリアクセスコントローラ (DMAC) を使用します。

1.2 使用機能

- シリアルサウンドインタフェース (SSI)
- ダイレクトメモリアクセスコントローラ (DMAC)
- 汎用入出力ポート
- 割り込みコントローラ

1.3 適用条件

マイコン	SH7262/SH7264
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 Release00
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)

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。

- SH7262/SH7264 グループ 初期設定例
- SH7262/SH7264 グループ シリアルサウンドインタフェース スレーブレシーバ設定例
- SH7262/SH7264 グループ シリアルサウンドインタフェース マスタランスマッタ設定例

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

本応用例ではシリアルサウンドインタフェース (SSI) をサンプリング周期 44.1kHz に設定し、マスタランシーバとして動作します。

2.1 SSI の動作概要

SSI の特長は以下の通りです。

- チャンネル数：4 チャンネル
- 動作モード：非圧縮モード
非圧縮モードはチャンネルに分割されるシリアルオーディオストリームをサポートします。
- トランスミッタまたはレシーバいずれとしても動作可能
- チャンネル 0 は全二重通信が可能
- シリアルバスフォーマットを使用可能
- データバッファとシフトレジスタ間は非同期転送
- シリアルバスインタフェースで使用するクロックの分周比が選択可能
- DMAC または割り込みで、データ送受信を制御可能
- オーバサンプルクロックを以下の端子から選択可能
AUDIO_CLK
AUDIO_X1、AUDIO_X2
- 送信部、受信部に 8 段 FIFO バッファ内蔵

図 1 に SSI のブロック図を示します。

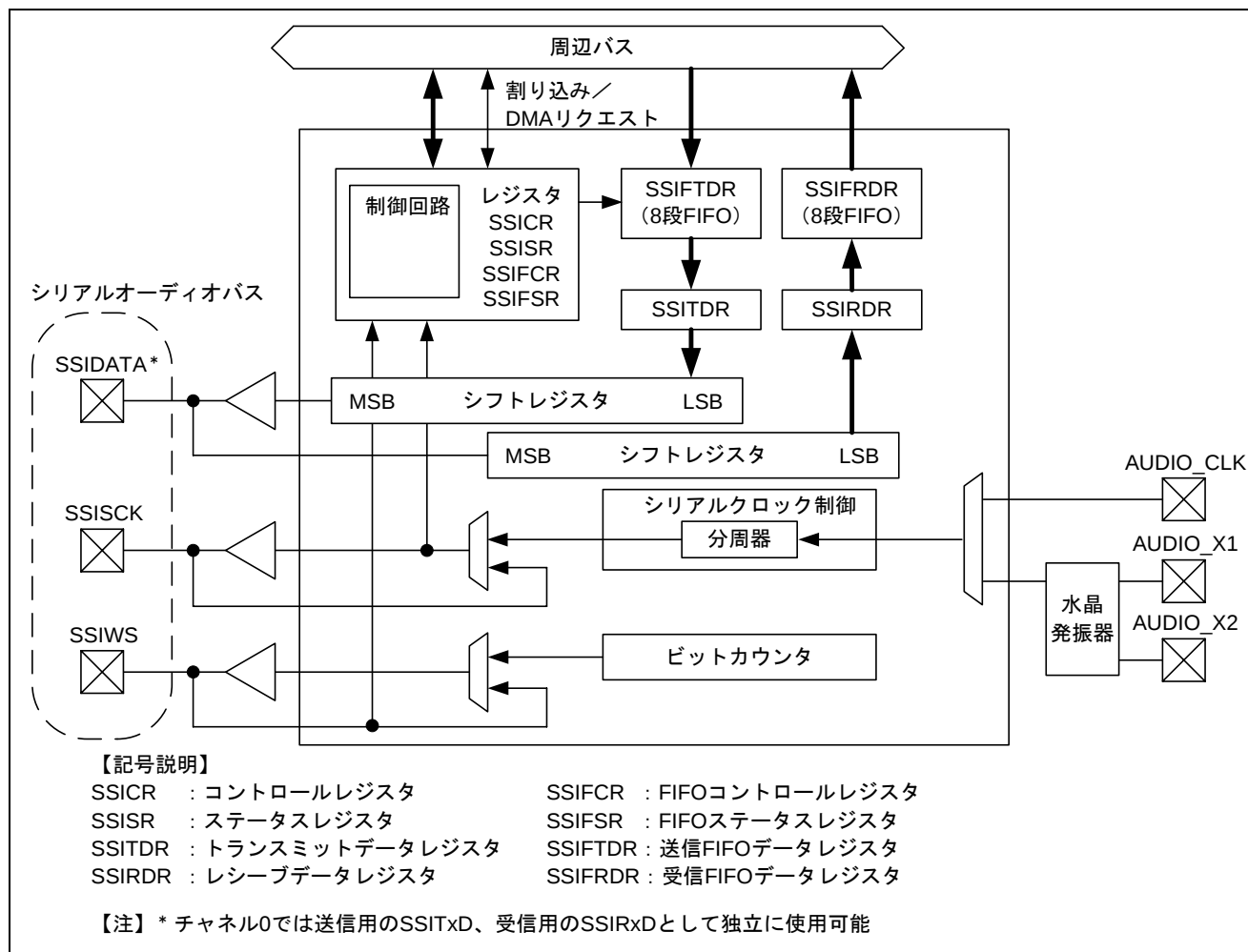


図1 SSIのブロック図

2.2 使用機能の設定手順

図2にSSI設定フロー例を、図3にDMAC設定フロー例を示します。

なお、各レジスタの詳細は「SH7262 グループ、SH7264 グループ ハードウェアマニュアル」を参照してください。

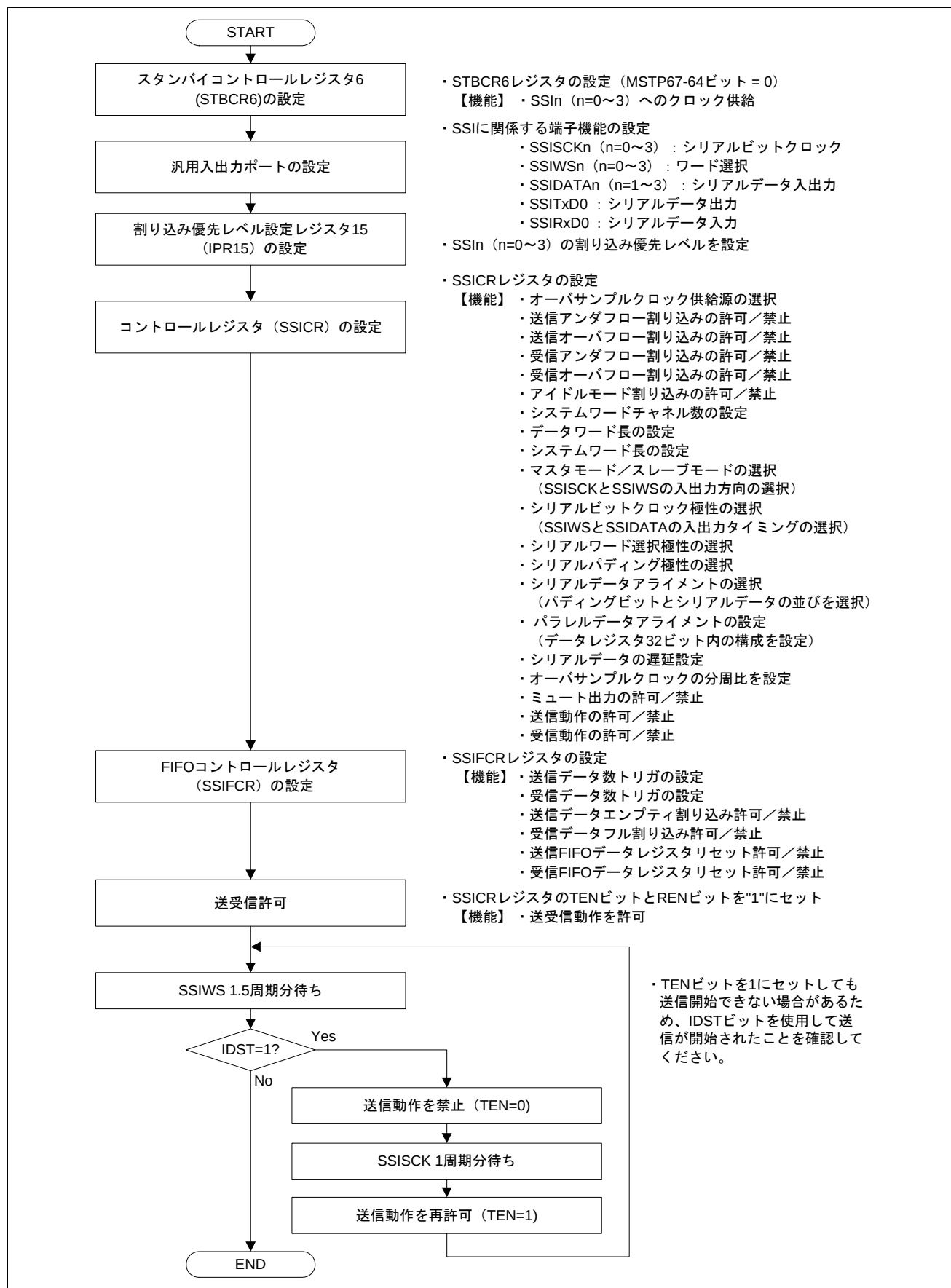


図2 SSI 設定フロー例

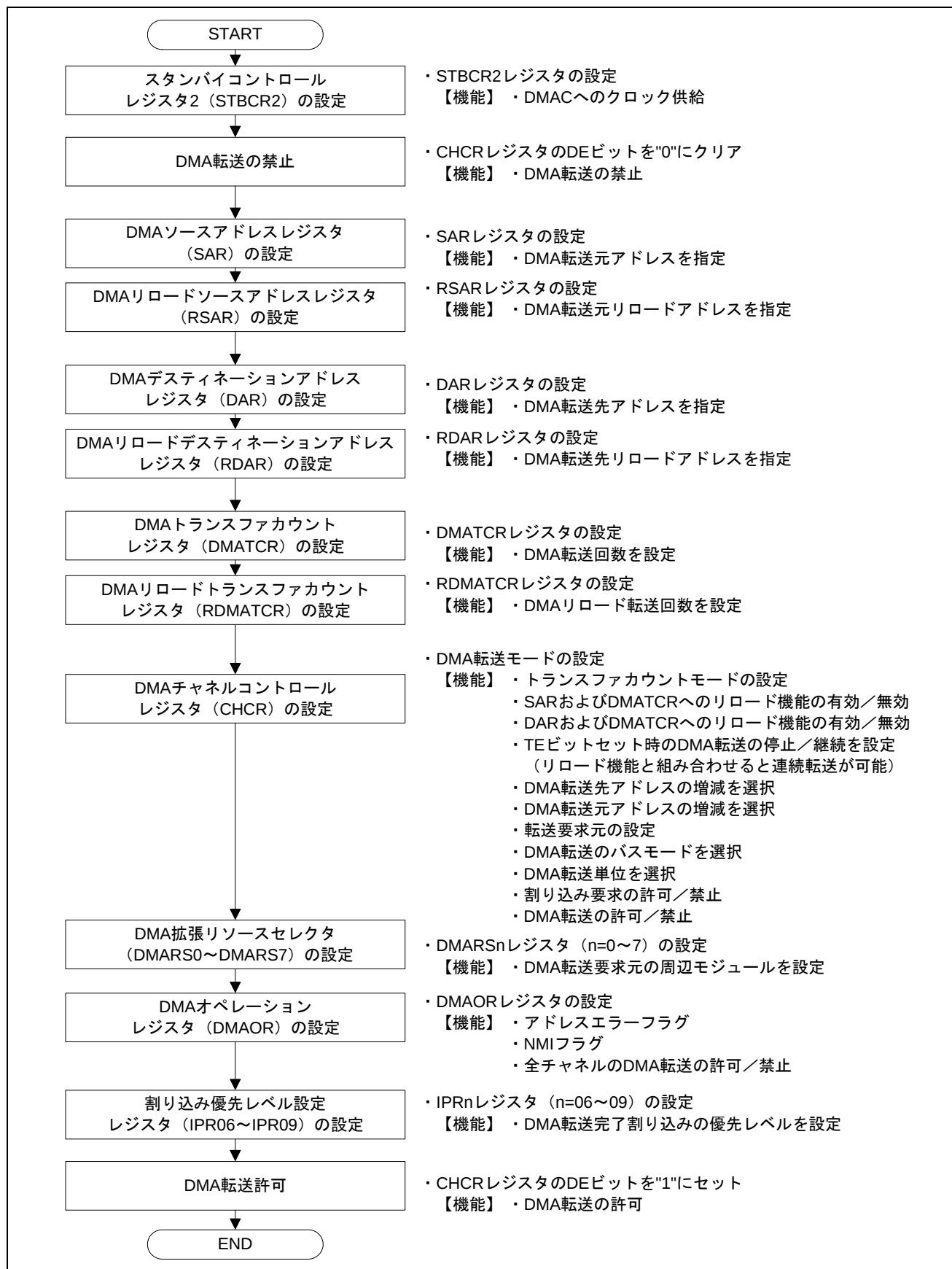


図3 DMAC 設定フロー例

2.3 参考プログラムの動作

参考プログラムでは、SSITxD0 端子と SSIRxD0 端子を直結して送信したデータを受信します。

SSI チャンネル 0 の送信データエンベティ割り込みによる DMA 転送要求を使用して DMAC チャンネル 1 を起動します。DMAC は、外部メモリから SSI チャンネル 0 の送信 FIFO データレジスタ (SSIFTDR) へデータを転送します。SSIFTDR レジスタに転送されたデータは、トランスミットデータレジスタ (SSITDR) を経由して SSITxD0 端子から出力されます。送信するデータは、10 サンプル (40 バイト) 分の PCM データです。繰り返し 4 回転送し、転送完了後に SSI をミュート出力に変更します。

SSIRxD0 端子から受信したデータは SSI チャンネル 0 のレシーブデータレジスタ (SSIRDR) を経由して、受信 FIFO データレジスタ (SSIFRDR) に転送されます。SSIFRDR レジスタにデータが転送されると、SSI は受信データフル割り込みによる DMA 転送要求を使用して DMAC チャンネル 2 を起動します。DMAC チャンネル 2 の転送先には 10 サンプル (40 バイト) 分の受信バッファを設定します。受信バッファは 2 面構成で交互に切り替えながら連続受信します。受信バッファのデータは、DMA 転送完了割り込みで読み出します。

参考プログラムの SSI 設定を以下に示します。

- 使用チャンネル：チャンネル 0
- 動作モード：マスタランシーバ
- データ送信制御方法：DMAC (チャンネル 1)
- データ受信制御方法：DMAC (チャンネル 2)
- オーバサンプルクロック：AUDIO_X1 入力 (11.2896MHz)
- シリアルオーバーサンプルクロック周波数：オーバーサンプルクロック周波数/4 (2.822MHz)
- データワード長：16 ビット
- システムワード長：32 ビット
- パディングビット："L"レベル
- SSIWS 信号と SSIDATA 信号間の遅延なし
- SSIWS 信号と SSIDATA 信号は SSISCK 信号の立ち下がりで変化
- サンプリング周波数：44.1kHz [354ns (2.8224MHz) × 32 ビット × 2]
- 第 1 チャンネルのデータワード 1 (L チャンネル) に "H'FFFF" を、第 2 チャンネルのデータワード 2 (R チャンネル) に "H'0000" を出力します。

図 4に参考プログラムの出力波形を、図 5にブロック図を示します。

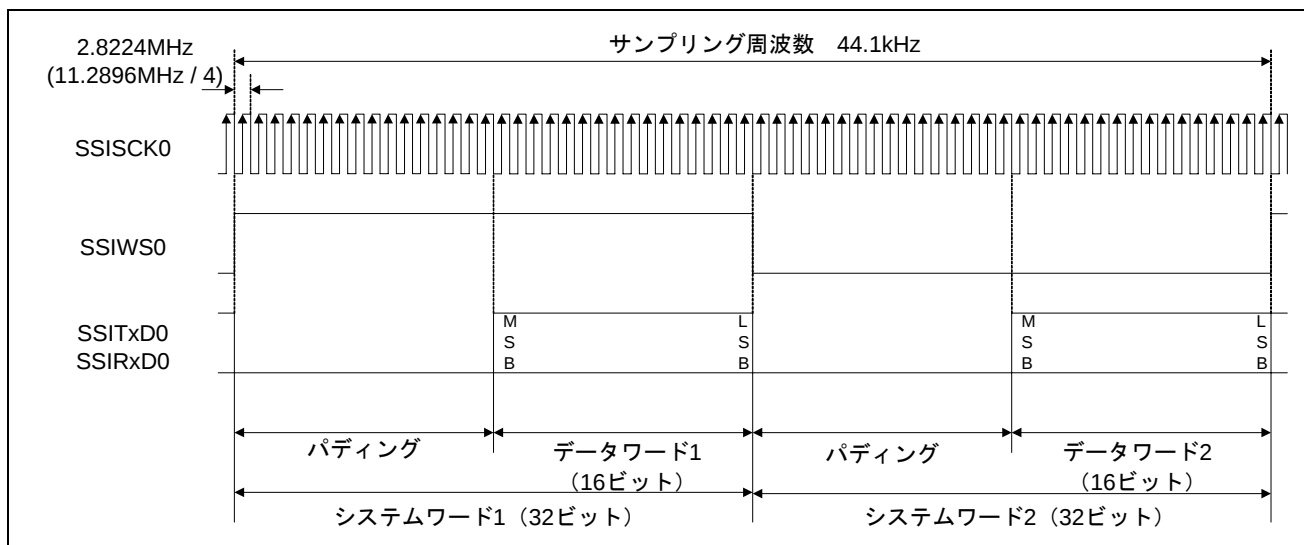


図4 参考プログラムの出力波形

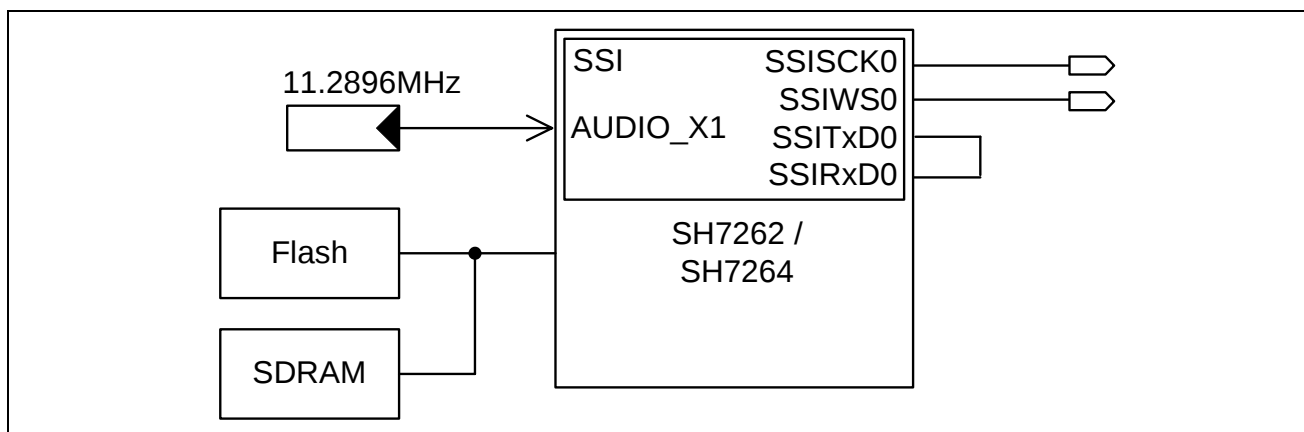


図5 ブロック図

2.4 参考プログラムの処理手順

表 1に参考プログラムの SSI レジスタ設定を、表 2と表 3に参考プログラムの DMAC レジスタ設定を示します。また、図 6～図 8に参考プログラムの処理フローを示します。

表1 参考プログラムの SSI レジスタ設定

レジスタ名	アドレス	設定値	機能
コントロール レジスタ 0 (SSICR_0)	H'FFFF 0000	H'3C0B D520	• CKS ビット= B'0 (オーバーサンプルクロックは AUDIO_X1 入力) • TUIEN ビット= B'1 (送信アンダフロー割り込み許可) • TOIEN ビット= B'1 (送信オーバフロー割り込み許可) • RUIEN ビット= B'1 (受信アンダフロー割り込み許可) • ROIEN ビット= B'1 (受信オーバフロー割り込み許可) • IIEN ビット= B'0 (アイドルモード割り込み禁止) • CHNL[1:0]ビット= B'00 (各システムワードは 1 チャンネルで構成) • DWL[2:0]ビット= B'001 (データワード長: 16 ビット) • SWL[2:0]ビット= B'011 (システムワード長: 32 ビット) • SCKD ビット= B'1 (シリアルビットクロック出力、マスタモード) • SWSD ビット= B'1 (シリアルワード選択出力、マスタモード) • SCKP ビット= B'0 (SSIWS と SSIDATA は SSISCK の立ち下がリエッジで変化、立ち上がりエッジでサンプリング) • SWSP ビット= B'1 (SSIWS は、第 1 チャンネルではハイレベル、第 2 チャンネルはローレベル) • SPDP ビット= B'0 (パディングビットはローレベル) • SDTA ビット= B'1 (パディングビット、シリアルデータの順に送受信) • PDTA ビット= B'0 (パラレルデータの下位側を先行して送受信) • DEL ビット= B'1 (SSIWS と SSIDATA 間の遅延なし) • CKDV ビット[3:0]= B'0010 (オーバーサンプルクロックは 4 分周) • MUEN ビット= B'0 (非ミュート状態) • TEN ビット= B'0 (送信動作を禁止) • REN ビット= B'0 (受信動作を禁止)
		H'3C0B D523	• TEN ビット= B'1 (送信動作を許可) • REN ビット= B'1 (受信動作を許可)
FIFO コントロール レジスタ 0 (SSIFCR_0)	H'FFFF 0010	H'0000 000C	• TTRG[1:0]ビット= B'00 (送信データ数トリガ: 7) • RTRG[1:0]ビット= B'00 (受信データ数トリガ: 1) • TIE ビット= B'1 (送信データエンプティ割り込み許可) • RIE ビット= B'1 (受信データフル割り込み許可) • TFRST ビット= B'0 (送信 FIFO リセット動作禁止) • RFRST ビット= B'0 (受信 FIFO リセット動作禁止)

表2 参考プログラムの DMAC チャンネル 1 のレジスタ設定

レジスタ名	アドレス	設定値	機能
DMA チャンネルコントロールレジスタ_1 (CHCR_1)	H'FFFE 101C	H'0000 0000	DE ビット= B'0 (DMA 転送を禁止)
		H'2010 1814	- TC ビット= B'0 (1 回転送) - RLDSAR ビット= B'1 (SAR のリロード機能有効) - RLDDAR ビット= B'0 (DAR のリロード機能無効) - DAF ビット、SAF ビット= B'00 (未使用) - DO ビット= B'0 (未使用) - TL ビット= B'0 (未使用) - TEMASK ビット= B'1 (TE ビットがセットされても DMA 転送を継続) - HE ビット、HIE ビット= B'00 (未使用) - AM ビット、AL ビット= B'00 (未使用) - DM[1:0] ビット= B'00 (デスティネーションアドレス固定) - SM[1:0] ビット= B'01 (ソースアドレス増加) - RS[3:0] ビット= B'1000 (DMA 拡張リソース選択) - DL ビット、DS ビット= B'00 (未使用) - TB ビット= B'0 (サイクルスチールモード) - TS[1:0] ビット= B'10 (ロングワード転送) - IE ビット= B'1 (割り込み要求許可) - DE ビット= B'0 (DMA 転送を禁止)
		H'2010 1815	DE ビット= B'1 (DMA 転送許可)
DMA ソースアドレスレジスタ_1 (SAR_1)	H'FFFE 1010	内蔵 RAM	転送元開始アドレスに内蔵 RAM 領域を設定
DMA リロードソースアドレスレジスタ_1 (RSAR_1)	H'FFFE 1110	内蔵 RAM	リロード用転送元開始アドレスに内蔵 RAM 領域を設定
DMA デスティネーションアドレスレジスタ_1 (DAR_1)	H'FFFE 1014	H'FFFF 0018	転送先開始アドレスに SSIFTDR レジスタ 0 を設定
DMA トランスファカウンタレジスタ_1 (DMATCR_1)	H'FFFE 1018	H'0000 000A	転送回数 10 回
DMA リロードトランスファカウンタレジスタ_1 (RDMATCR_1)	H'FFFE 1118	H'0000 000A	転送回数 10 回
DMA オペレーションレジスタ (DMAOR)	H'FFFE 1200	H'0001	- CMS[1:0] ビット= B'00 (通常モード) - PR[1:0] ビット= B'00 (チャンネル優先順位: 固定モード 1) - AE ビット= B'0 (アドレスエラーフラグクリア) - NMIF ビット= B'0 (NMI 割り込みクリア) - DME ビット= B'1 (全チャンネルの DMA 転送許可)
DMA 拡張リソースセクタ 0 (DMARS0)	H'FFFE 1300	H'2100	DMA チャンネル 1 の転送要求元に SSI チャンネル 0 の送信機能を設定

表3 参考プログラムの DMAC チャンネル 2 のレジスタ設定

レジスタ名	アドレス	設定値	機能
DMA チャンネルコントロールレジスタ_2 (CHCR_2)	H'FFFE 102C	H'0000 0000	DE ビット= B'0 (DMA 転送を禁止)
		H'1010 4814	- TC ビット= B'0 (1 回転送) - RLDSAR ビット= B'0 (SAR のリロード機能無効) - RLDDAR ビット= B'1 (DAR のリロード機能有効) - DAF ビット、SAF ビット= B'00 (未使用) - DO ビット= B'0 (未使用) - TL ビット= B'0 (未使用) - TEMASK ビット= B'1 (TE ビットがセットされても DMA 転送を継続) - HE ビット、HIE ビット= B'00 (未使用) - AM ビット、AL ビット= B'00 (未使用) - DM[1:0] ビット= B'01 (デスティネーションアドレス増加) - SM[1:0] ビット= B'00 (ソースアドレス固定) - RS[3:0] ビット= B'1000 (DMA 拡張リソース選択) - DL ビット、DS ビット= B'00 (未使用) - TB ビット= B'0 (サイクルスチールモード) - TS[1:0] ビット= B'10 (ロングワード転送) - IE ビット= B'1 (割り込み要求許可) - DE ビット= B'0 (DMA 転送禁止)
		H'1010 4815	DE ビット= B'1 (DMA 転送許可)
DMA ソースアドレスレジスタ_2 (SAR_2)	H'FFFE 1020	H'FFFF 001C	転送元開始アドレスに SSIFRDR レジスタ 0 を設定
DMA デスティネーションアドレスレジスタ_2 (DAR_2)	H'FFFE 1024	内蔵 RAM	転送先開始アドレスに内蔵 RAM 領域を設定
DMA リロードデスティネーションアドレスレジスタ_2 (RDAR_2)	H'FFFE 1124	内蔵 RAM	リロード用転送先開始アドレスに内蔵 RAM 領域を設定
DMA トランスファカウンタレジスタ_2 (DMATCR_2)	H'FFFE 1028	H'0000 000A	転送回数 10 回
DMA リロードトランスファカウンタレジスタ_2 (RDMATCR_2)	H'FFFE 1128	H'0000 000A	転送回数 10 回
DMA オペレーションレジスタ (DMAOR)	H'FFFE 1200	H'0001	- CMS[1:0] ビット= B'00 (通常モード) - PR[1:0] ビット= B'00 (チャンネル優先順位: 固定モード 1) - AE ビット= B'0 (アドレスエラーフラグクリア) - NMIF ビット= B'0 (NMI 割り込みクリア) - DME ビット= B'1 (全チャンネルの DMA 転送許可)
DMA 拡張リソースセクタ 1 (DMARS1)	H'FFFE 1304	H'0022	DMA チャンネル 2 の転送要求元に SSI チャンネル 0 の受信機能を設定

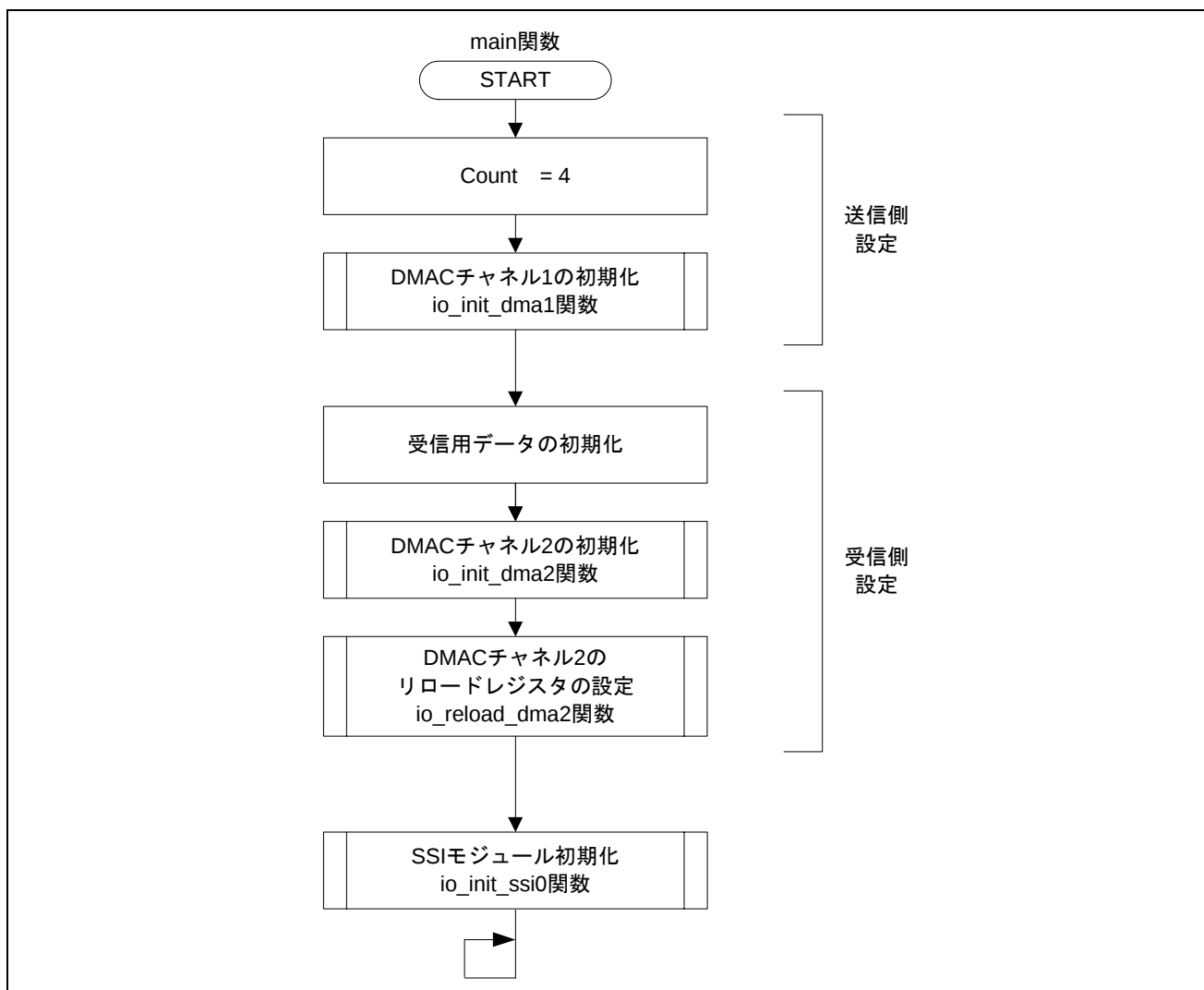


図6 参考プログラムの処理フロー (1)

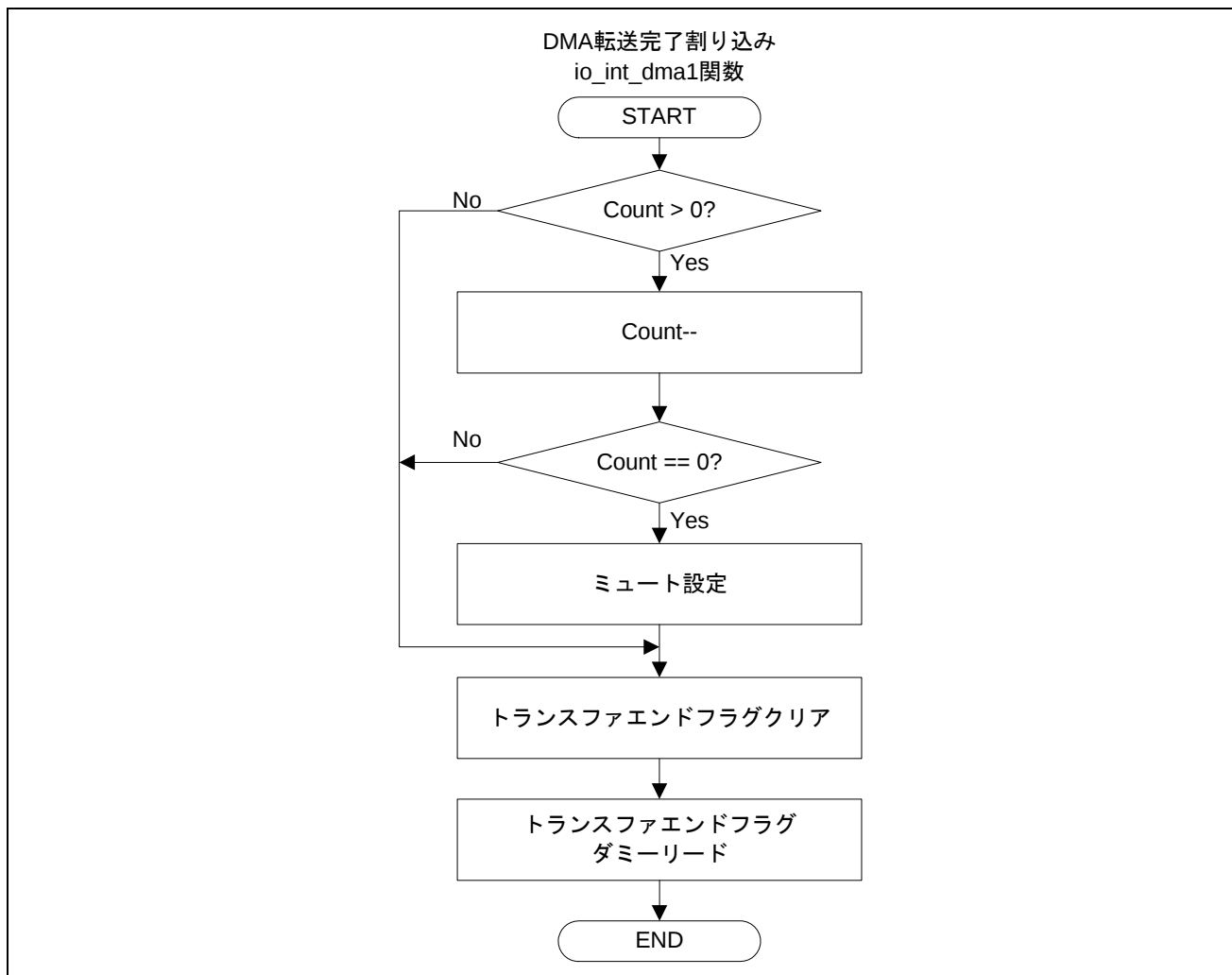


図7 参考プログラムの処理フロー (2)

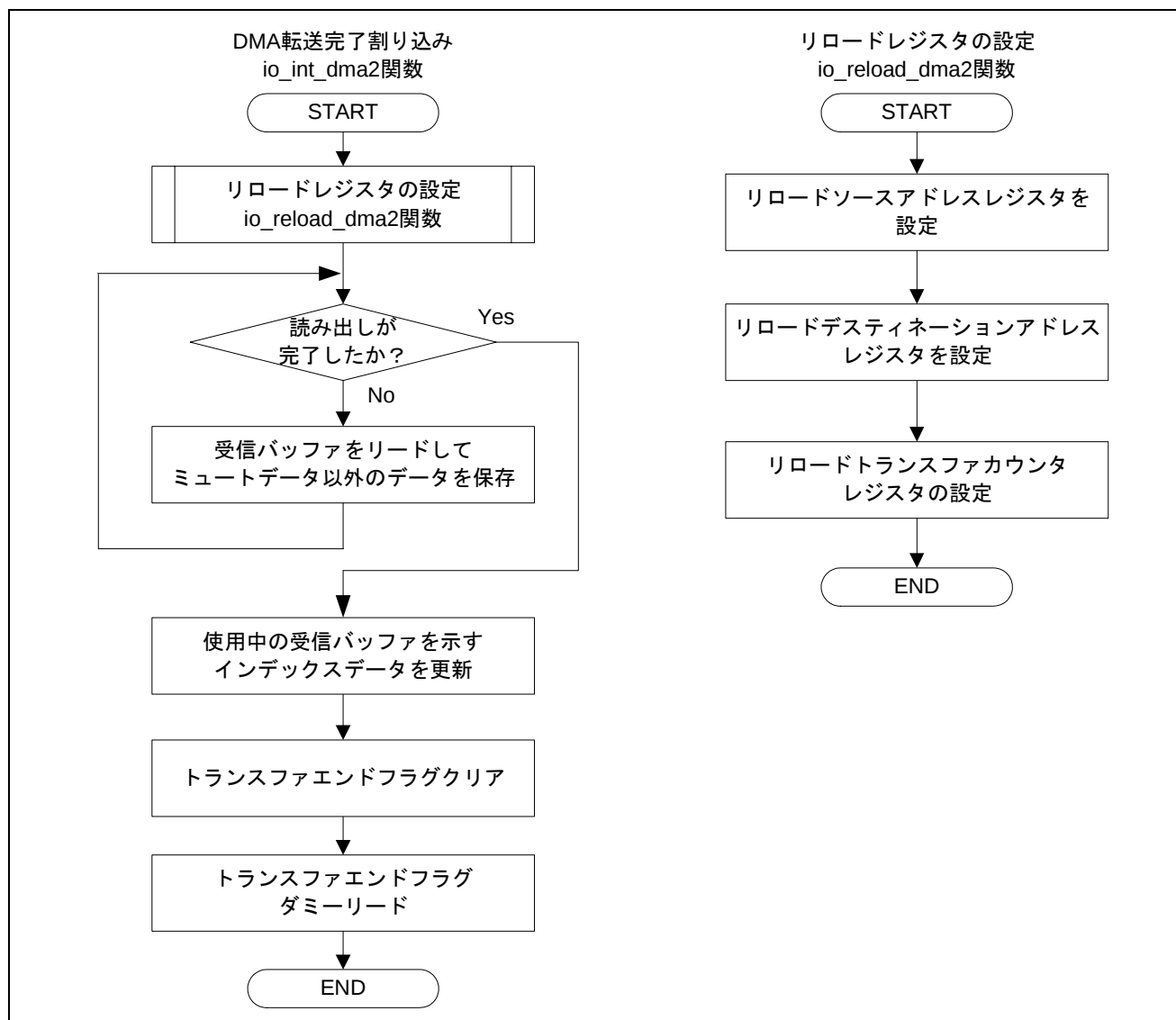


図8 参考プログラムの処理フロー (3)

3. 参考プログラムリスト

3.1 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) 2009(2011) Renesas Electronics Corporation. All rights reserved.
29 *   "FILE COMMENT"***** Technical reference data *****
30 *   System Name : SH7264 Sample Program
31 *   File Name   : main.c
32 *   Abstract    : SSI マスタランシーバ設定例
33 *   Version     : 1.01.00
34 *   Device      : SH7262/SH7264
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 Release00).
38 *   OS          : None
39 *   H/W Platform: M3A-HS64G50(CPU board)
40 *   Description :
41 *****/
42 *   History     : Feb.24,2009 Ver.1.00.00
43 *               : Feb.23,2011 Ver.1.01.00 送信の開始確認を追加
44 *   "FILE COMMENT END"*****
45 #include <string.h>
46 #include "iodefine.h" /* iodefine.h は、HEW で自動生成されるファイルです。 */
```

3.2 サンプルプログラムリスト"main.c" (2)

```
47
48  /* ==== マクロ宣言 ==== */
49  #define SSI_DATASIZE 40u
50  #define SSI_MUTEDATA 0x00000000ul
51
52  /* ==== プロトタイプ宣言 ==== */
53  void main(void);
54  void io_init_ssi0(void);
55  void io_init_dma1(void *src, void *dst, size_t size);
56  void io_init_dma2(void *src, void *dst, size_t size);
57  void io_reload_dma2(void *src, void *dst, size_t size);
58
59  /* ==== 変数宣言 ==== */
60
61  /* ---- 送信側データ ---- */
62  unsigned long Data[SSI_DATASIZE/sizeof(unsigned long)] = {
63      0x0000FFFFul, 0x0000FFFFul,
64      0x0000FFFFul, 0x0000FFFFul,
65      0x0000FFFFul, 0x0000FFFFul,
66      0x0000FFFFul, 0x0000FFFFul,
67      0x0000FFFFul, 0x0000FFFFul};
68  unsigned int Count;
69
70  /* ---- 受信側データ ---- */
71  unsigned long Buff[2][SSI_DATASIZE/sizeof(unsigned long)]; /* 受信バッファ */
72  unsigned int BuffIdx; /* 受信中のバッファを示すインデックス */
73  unsigned long RcvData[100]; /* ミュートを除く受信データを保存する領域 */
74  unsigned int RcvCnt; /* 保存した受信データ数 */
75
```

3.3 サンプルプログラムリスト"main.c" (3)

```

76  /*"FUNC COMMENT"*****
77  * ID      :
78  * Outline  : main
79  *-----
80  * Include  : #include "iodefine.h"
81  *-----
82  * Declaration : void main(void);
83  *-----
84  * Description : SSI モジュール初期化後、データ送信を行います。
85  *-----
86  * Argument   : void
87  *-----
88  * Return Value: void
89  *"FUNC COMMENT END"*****/
90  void main(void)
91  {
92      /* ==== 送信側設定 ==== */
93      Count = 4u;                                /* DMA 転送回数 */
94      /* ---- DMAC の初期化/転送許可設定 ---- */
95      io_init_dma1(    Data,                        /* ソースアドレス */
96                     (void *)&SSI0.SSIFTDR,        /* デスティネーションアドレス */
97                     SSI_DATASIZE);                /* バイト数 */
98
99
100     /* ==== 受信側設定 ==== */
101     RcvCnt = 0u;                                /* 受信データ数 */
102     BuffIdx = 0u;                                /* 受信バッファのインデックス */
103     /* ---- DMAC の初期化/転送許可設定 ---- */
104     io_init_dma2(    (void *)&SSI0.SSIFRDR,        /* ソースアドレス */
105                     Buff[BuffIdx],                /* デスティネーションアドレス */
106                     SSI_DATASIZE);                /* バイト数 */
107     io_reload_dma2(  (void *)&SSI0.SSIFRDR,
108                     Buff[BuffIdx^1u],
109                     SSI_DATASIZE);
110
111     /* ==== SSI0 初期化 ==== */
112     io_init_ssi0();
113
114     while(1){
115         /* Program end */
116     }
117 }

```

3.4 サンプルプログラムリスト"main.c" (4)

```

118  /*"FUNC COMMENT"*****
119  * ID      :
120  * Outline : SSI モジュール初期化
121  *-----
122  * Include : #include "iodefine.h"
123  *-----
124  * Declaration : void io_init_ssi0(void);
125  *-----
126  * Description : マスタランシーバモードでデータ転送します。
127  *              : サンプリング周期は 44.1kHz です。
128  *-----
129  * Argument  : void
130  *-----
131  * Return Value: void
132  *"FUNC COMMENT END"*****
133  void io_init_ssi0(void)
134  {
135      volatile int w;
136
137      /* ---- SSI クロック供給 ---- */
138      CPG.STBCR6.BIT.MSTP67 = 0u;          /* SSI0 */
139
140      /* ----SSSI 端子機能選択 ---- */
141      PORT.PGCR2.BIT.PG8MD = 2u;          /* SSISCK0 */
142      PORT.PGCR2.BIT.PG9MD = 2u;          /* SSIWS0 */
143      PORT.PGCR2.BIT.PG10MD = 2u;         /* SSIRXD0 */
144      PORT.PGCR2.BIT.PG11MD = 2u;         /* SSITXD0 */
145
146      /* ---- SSI 割り込みレベル設定 ---- */
147      INTC.IPR15.BIT._SSI0 = 1u;          /* SSI0 */
148
149      /* ---- コントロールレジスタ(SSICR) ---- */
150      SSIF0.SSICR.LONG = 0x3C0BD520uL;
151      /*
152          bit31      : reserve 0
153          bit30      : CKS : 0----- AUDIO_X1 入力
154          bit29      : TUIEN : 1----- 送信アンダフロー割り込み許可
155          bit28      : TOIEN : 1----- 送信オーバフロー割り込み許可
156          bit27      : RUIEN : 1----- 受信アンダフロー割り込み許可
157          bit26      : ROIEN : 1----- 受信オーバフロー割り込み許可
158          bit25      : IIEN : 0----- アイドルモード割り込み禁止
159          bit24      : reserve 0
160          bit23-22   : CHNL : B'00----- 各システムワードは 1 チャンネル
161          bit21-19   : DWL : B'001----- データワード長は 16 ビット
162          bit18-16   : SWL : B'011----- システムワード長は 32 ビット
163          bit15      : SCKD : 1----- シリアルビットクロック出力、マスタモード
164          bit14      : SWSD : 1----- シリアルワード WS 出力、マスタモード
165          bit13      : SCKP : 0----- SSISCK の立ち下がりで変化、立ち上がりでサンプリング
166          bit12      : SWSP : 1----- 第一チャンネルは"H"、第二チャンネルは"L"
167          bit11      : SPDP : 0----- パディングビットは"L レベル"
168          bit10      : SDTA : 1----- パディングビット、シリアルデータの順に送受信

```

3.5 サンプルプログラムリスト"main.c" (5)

```

169         bit9      : PDTA : 0----- パラレルデータの下位側を先行して送受信
170         bit8      : DEL : 1----- SSIWS と SSIDATA 間の遅延なし
171         bit7-4    : CKDV : B'0010----- サンプルクロック分周比 AUDIO φ / 4 (44.1kHz)
172         bit3      : MUEN : 0----- ミュート状態ではない
173         bit2      : reserve 0
174         bit1      : TEN : 0----- 送信動作 禁止
175         bit0      : REN : 0----- 受信動作 禁止
176     */
177     /* ---- FIFO コントロールレジスタ設定 (SSIFCR) ---- */
178     SSIF0.SSIFCR.LONG = 0x00000000Cul;
179     /*
180         bit31-8    : reserve 0
181         bit7-6     : TTRG : B'00----- 送信データ数トリガ 7
182         bit5-4     : RTRG : B'00----- 受信データ数トリガ 1
183         bit3       : TIE : 1----- 送信データエンプティ割り込み要求を許可
184         bit2       : RIE : 1----- 受信データフル割り込み要求を許可
185         bit1       : TFRST : 0----- 送信 FIFO データレジスタリセット禁止
186         bit0       : RFRST : 0----- 受信 FIFO データレジスタリセット禁止
187     */
188     /* ---- 受信許可 ---- */
189     SSIF0.SSICR.BIT.REN = 1u;
190     /* ---- 送信許可 ---- */
191     SSIF0.SSICR.BIT.TEN = 1u;
192
193     /* ---- 送信が開始されたことを確認 ---- */
194     while(1){
195         /* SSIWS 1.5 周期分待ち */
196         for( w = 16000 ; w > 0 ; w--){ /* 1.1 ms = (1/44.1 kHz) * 32 bit * 1.5 cyc */
197             /* wait */
198         }
199         /* シリアルバスが動作していれば正常終了 */
200         if( SSIF0.SSISR.BIT.IDST == 0 ){
201             break;
202         }
203         /* 送信禁止 */
204         SSIF0.SSICR.BIT.TEN = 0u;
205
206         /* SSISCK 1 周期分待ち */
207         for( w = 400 ; w > 0 ; w--){ /* 23 us = 1/44.1 kHz */
208             /* wait */
209         }
210         /* 送信許可 */
211         SSIF0.SSICR.BIT.TEN = 1u;
212     }
213 }

```

3.6 サンプルプログラムリスト"main.c" (6)

```
214  /*"FUNC COMMENT"*****
215  * ID      :
216  * Outline  : SSI 割り込み
217  *-----
218  * Include  : #include "iodefine.h"
219  *-----
220  * Declaration : void io_int_ssi0(void);
221  *-----
222  * Description : SSI 割り込み処理を行います。
223  *-----
224  * Argument   : void
225  *-----
226  * Return Value: void
227  *"FUNC COMMENT END"*****/
228 void io_int_ssi0(void)
229 {
230     /* 送信アンダフローエラー */
231     if(SSIF0.SSISR.BIT.TUIRQ == 1u){
232         SSIF0.SSISR.BIT.TUIRQ = 0u;
233         while(1){
234             /* dead loop */
235         }
236     }
237     /* 送信オーバフローエラー */
238     if(SSIF0.SSISR.BIT.TOIRQ == 1u){
239         SSIF0.SSISR.BIT.TOIRQ = 0u;
240         while(1){
241             /* dead loop */
242         }
243     }
244     /* 受信アンダフローエラー */
245     if(SSIF0.SSISR.BIT.RUIRQ == 1u){
246         SSIF0.SSISR.BIT.RUIRQ = 0u;
247         while(1){
248             /* dead loop */
249         }
250     }
251     /* 受信オーバフローエラー */
252     if(SSIF0.SSISR.BIT.ROIRQ == 1u){
253         SSIF0.SSISR.BIT.ROIRQ = 0u;
254         while(1){
255             /* dead loop */
256         }
257     }
258 }
259
```

3.7 サンプルプログラムリスト"main.c" (7)

```

260  /*"FUNC COMMENT"*****
261  * ID      :
262  * Outline  : DMA 転送初期設定
263  * -----
264  * Include  : #include "iodefine.h"
265  * -----
266  * Declaration : void io_init_dma1(void *src, void *dst, size_t size);
267  * -----
268  * Description : ソースアドレス src からデスティネーションアドレス dst へ
269  *              : size バイト分 DMAC による転送を行います。
270  *              : DMA 転送完了後も繰り返し同じデータを転送します。
271  *              : DMA 転送の完了割り込みは許可します。
272  *              : 転送サイズはロングワードとし、転送要求元は SSI1 を設定します。
273  *              : なお、転送サイズとソース/デスティネーションアドレスの
274  *              : アライメントが一致しない場合の動作は保証しません。
275  * -----
276  * Argument  : void *src   : ソースアドレス
277  *              : void *dst   : デスティネーションアドレス
278  *              : size_t size : 転送サイズ (バイト)
279  * -----
280  * Return Value: void
281  *"FUNC COMMENT END"*****/
282  void io_init_dma1(void *src, void *dst, size_t size)
283  {
284      /* ---- スタンバイコントロールレジスタ 2 の設定 ---- */
285      CPG.STBCR2.BIT.MSTP8 = 0u;          /* DMAC のモジュールストップ解除*/
286
287      /* ---- DMA 転送禁止 ---- */
288      DMAC.CHCR1.BIT.DE = 0u;
289
290      /* ---- DMA ソースアドレスレジスタの設定 ---- */
291      /* ---- DMA リロードソースアドレスレジスタの設定 ---- */
292      DMAC.SAR1.LONG = (unsigned long)src;
293      DMAC.RSAR1.LONG= (unsigned long)src;
294
295      /* ---- DMA デスティネーションアドレスレジスタの設定 ---- */
296      DMAC.DAR1.LONG = (unsigned long)dst;
297
298      /* ---- DMA トランスファカウントレジスタの設定 ---- */
299      /* ---- DMA リロードトランスファカウントレジスタの設定 ---- */
300      DMAC.DMATCR1.LONG = size >> 2u;
301      DMAC.RDMATCR1.LONG= size >> 2u;
302

```

3.8 サンプルプログラムリスト"main.c" (8)

```

303  /* ---- DMA チャンネルコントロールレジスタ設定 ---- */
304  DMAC.CHCR1.LONG = 0x20101814u1;
305  /*
306      bit31    : TC : 0----- 1 回転送
307      bit30    : reserve 0
308      bit29    : RLDSAR : 1----- SAR リロード機能有効
309      bit28    : RLDDAR : 0----- DAR リロード機能無効
310      bit27    : reserve 0
311      bit26    : DAF : 0----- 未使用
312      bit25    : SAF : 0----- 未使用
313      bit24    : reserve 0
314      bit23    : D0 : 0----- 未使用
315      bit22    : TL : 0----- 未使用
316      bit21    : reserve 0
317      bit20    : TEMASK : 1----- TE ビットセット後も DMA 転送継続
318      bit19    : HE : 0----- 未使用
319      bit18    : HIE : 0----- 未使用
320      bit17    : AM : 0----- 未使用
321      bit16    : AL : 0----- 未使用
322      bit15-14 : DM[1:0] : B'00----- 転送先アドレス固定
323      bit13-12 : SM[1:0] : B'01----- 転送元アドレス増加
324      bit11-8  : RS[3:0] : B'1000----- DMA 拡張リソースセクタ
325      bit7     : DL : 0----- 未使用
326      bit6     : DS : 0----- 未使用
327      bit5     : TB : 0----- サイクルスチールモード
328      bit4-3   : TS : B'10----- ロングワード転送
329      bit2     : IE : 1----- 割り込み許可
330      bit1     : TE : 0----- トランスファエンドフラグ
331      bit0     : DE : 0----- DMA 転送禁止
332  */
333  /* ----DMA 拡張リソースセクタ 0 の設定---- */
334  DMAC.DMARS0.BIT.CH1MID = 0x08u; /* MID = SSI0 */
335  DMAC.DMARS0.BIT.CH1RID = 0x01u; /* RID = TxD */
336
337  /* ----DMA オペレーションレジスタの設定---- */
338  DMAC.DMAOR.WORD &= 0xFFFF9u; /* AE, NMIF ビットクリア */
339  DMAC.DMAOR.BIT.DME = 1u; /* 全チャンネルの DMA 転送を許可 */
340
341  /* ---- 割り込み優先レベル設定レジスタの設定 ---- */
342  INTC.IPR06.BIT._DMAC1 = 1u;
343
344  /* ---- DMA 転送許可 ---- */
345  DMAC.CHCR1.BIT.DE = 1u1;
346  }

```

3.9 サンプルプログラムリスト"main.c" (9)

```

347  /*"FUNC COMMENT"*****
348  * ID      :
349  * Outline  : DMA 転送初期設定
350  * -----
351  * Include  : #include "iodefine.h"
352  * -----
353  * Declaration : void io_init_dma2(void *src, void *dst, size_t size);
354  * -----
355  * Description : ソースアドレス src からデスティネーションアドレス dst へ
356  *              : size バイト分 DMAC による転送を行います。
357  *              : DMA 転送完了後もデータ転送を継続しますがリロードレジスタの設定は
358  *              : 別途行う必要があります。
359  *              : DMA 転送の完了割り込みは許可します。
360  *              : 転送サイズはロングワードとし、転送要求元は SSI2 を設定します。
361  *              : なお、転送サイズとソース/デスティネーションアドレスの
362  *              : アライメントが一致しない場合の動作は保証しません。
363  * -----
364  * Argument  : void *src   : ソースアドレス
365  *              : void *dst   : デスティネーションアドレス
366  *              : size_t size : 転送サイズ (バイト)
367  * -----
368  * Return Value: void
369  *"FUNC COMMENT END"*****
370 void io_init_dma2(void *src, void *dst, size_t size)
371 {
372     /* ---- スタンバイコントロールレジスタ 2 の設定 ---- */
373     CPG.STBCR2.BIT.MSTP8 = 0u;          /* DMAC のモジュールストップ解除*/
374
375     /* ---- DMA 転送禁止 ---- */
376     DMAC.CHCR2.BIT.DE = 0u;
377
378     /* ---- DMA ソースアドレスレジスタの設定 ---- */
379     DMAC.SAR2.LONG = (unsigned long)src;
380
381     /* ---- DMA デスティネーションアドレスレジスタの設定 ---- */
382     DMAC.DAR2.LONG = (unsigned long)dst;
383
384     /* ---- DMA トランスファカウンタレジスタの設定 ---- */
385     DMAC.DMATCR2.LONG = size >> 2u;
386

```

3.10 サンプルプログラムリスト"main.c" (10)

```

387  /* ---- DMA チャンネルコントロールレジスタ設定 ---- */
388  DMAC.CHCR2.LONG = 0x10104814u1;
389  /*
390      bit31    : TC : 0----- 1 回転送
391      bit30    : reserve 0
392      bit29    : RLDSAR : 0----- SAR リロード機能無効
393      bit28    : RLDDAR : 1----- DAR リロード機能有効
394      bit27    : reserve 0
395      bit26    : DAF : 0----- 未使用
396      bit25    : SAF : 0----- 未使用
397      bit24    : reserve 0
398      bit23    : D0 : 0----- 未使用
399      bit22    : TL : 0----- 未使用
400      bit21    : reserve 0
401      bit20    : TEMASK : 1----- TE ビットセット後も DMA 転送継続
402      bit19    : HE : 0----- 未使用
403      bit18    : HIE : 0----- 未使用
404      bit17    : AM : 0----- 未使用
405      bit16    : AL : 0----- 未使用
406      bit15-14 : DM[1:0] : B'01----- 転送先アドレス増加
407      bit13-12 : SM[1:0] : B'00----- 転送元アドレス固定
408      bit11-8  : RS[3:0] : B'1000----- DMA 拡張リソースセクタ
409      bit7     : DL : 0----- 未使用
410      bit6     : DS : 0----- 未使用
411      bit5     : TB : 0----- サイクルスチールモード
412      bit4-3   : TS : B'10----- ロングワード転送
413      bit2     : IE : 1----- 割り込み許可
414      bit1     : TE : 0----- トランスファエンドフラグ
415      bit0     : DE : 0----- DMA 転送禁止
416  */
417  /* ----DMA 拡張リソースセクタ 0 の設定---- */
418  DMAC.DMARS1.BIT.CH2MID = 0x08u; /* MID = SSI0 */
419  DMAC.DMARS1.BIT.CH2RID = 0x02u; /* RID = RxD */
420
421  /* ----DMA オペレーションレジスタの設定---- */
422  DMAC.DMAOR.WORD &= 0xFFF9u; /* AE, NMIF ビットクリア */
423  DMAC.DMAOR.BIT.DME = 1u; /* 全チャンネルの DMA 転送を許可 */
424
425  /* ---- 割り込み優先レベル設定レジスタの設定 ---- */
426  INTC.IPR06.BIT._DMAC2 = 1u;
427  /* ---- DMA の実行 ---- */
428
429  DMAC.CHCR2.BIT.DE = 1u1; /* DMA 転送許可 */
430 }
431

```

3.11 サンプルプログラムリスト"main.c" (11)

```
432  /*"FUNC COMMENT"*****
433  * ID      :
434  * Outline : DMA 転送リロード設定
435  * -----
436  * Include : #include "iodefine.h"
437  * -----
438  * Declaration : void io_reload_dma2(void *src, void *dst, size_t size);
439  * -----
440  * Description : リロードソースアドレスレジスタおよびリロードデスティネーション
441  *              : アドレスレジスタ、リロードトランスファカウンタレジスタに値を設定
442  *              : します。転送単位はロングワードとします。
443  *              : 転送サイズとソース/デスティネーションアドレスの
444  *              : アライメントが一致しない場合の動作は保証しません。
445  * -----
446  * Argument : void *src   : ソースアドレス
447  *           : void *dst   : デスティネーションアドレス
448  *           : size_t size : 転送サイズ (バイト)
449  * -----
450  * Return Value: void
451  * "FUNC COMMENT END"*****/
452  void io_reload_dma2(void *src, void *dst, size_t size)
453  {
454      /* ---- DMA リロードソースアドレスレジスタの設定 ---- */
455      DMAC.RSAR2.LONG= (unsigned long)src;
456
457      /* ---- DMA リロードデスティネーションアドレスレジスタの設定 ---- */
458      DMAC.RDAR2.LONG= (unsigned long)dst;
459
460      /* ---- DMA リロードトランスファカウンタレジスタの設定 ---- */
461      DMAC.RDMATCR2.LONG= size >> 2u;
462
463  }
464
```

3.12 サンプルプログラムリスト"main.c" (12)

```
465  /*"FUNC COMMENT"*****
466  * ID      :
467  * Outline  : DMA 転送完了割り込み
468  * -----
469  * Include  : #include "iodefine.h"
470  * -----
471  * Declaration : void io_int_dma1(void);
472  * -----
473  * Description : DMA 転送が指定数完了すると、SSI をミュートに移行させます。
474  * -----
475  * Argument   : void
476  * -----
477  * Return Value: void
478  *"FUNC COMMENT END"*****/
479 void io_int_dma1(void)
480 {
481     volatile unsigned long dummy;
482
483     if( Count > 0 ){
484         /* ---- 転送回数カウンタ ---- */
485         Count--;
486
487         if( Count == 0 ){
488             /* ---- 指定回数転送したらミュートに移行する (DMA 転送は停止しない) ---- */
489             SSIF0.SSICR.BIT.MUEN = 1u;      /* ミュート開始 */
490             /* FIFO のデータはミュートに置き換えられます */
491         }
492     }
493     /* ---- トランスファエンドフラグクリア ---- */
494     DMAC.CHCR1.BIT.TE = 0u;
495     dummy = DMAC.CHCR1.BIT.TE;      /* ダミーリード */
496 }
```

3.13 サンプルプログラムリスト"main.c" (13)

```
497  /*"FUNC COMMENT"*****
498  * ID      :
499  * Outline  : DMA 転送完了割り込み
500  *-----
501  * Include  : #include "iodefine.h"
502  *-----
503  * Declaration : void io_int_dma2(void);
504  *-----
505  * Description : 受信バッファからミュートを除く受信データを読み出します。
506  *-----
507  * Argument   : void
508  *-----
509  * Return Value: void
510  *"FUNC COMMENT END"*****/
511 void io_int_dma2(void)
512 {
513     volatile unsigned long dummy;
514     unsigned long rdata;
515     int i;
516
517     /* ---- リロードレジスタ更新 ---- */
518     io_reload_dma2((void *)&SSI2.SSIFRDR, Buff[BuffIdx], SSI_DATASIZE);
519
520     /* ---- 受信データの読み出し ---- */
521     for(i=0; i<SSI_DATASIZE/sizeof(unsigned long); i++){
522         rdata = Buff[BuffIdx][i];
523
524         if(SSI_MUTEDATA != rdata){
525             RcvData[RcvCnt++] = rdata;
526             if( RcvCnt >= (sizeof(RcvData)/sizeof(unsigned long)) ){
527                 RcvCnt = 0u;
528             }
529         }
530     }
531     /* ---- 受信バッファのインデックスを更新 ---- */
532     BuffIdx ^= 1u;
533
534     /* ---- トランスファエンドフラグクリア ---- */
535     DMAC.CHCR2.BIT.TE = 0u;
536     dummy = DMAC.CHCR2.BIT.TE;    /* ダミーリード */
537 }
538
539 /* 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、SH2A-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
SH7262 グループ、SH7264 グループ ハードウェアマニュアル Rev.2.00
(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/>

お問い合わせ先

<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	2009.03.10	—	初版発行
1.01	2011.02.23	5, 19	テクニカルアップデート(TN-SH7-A799A/J)に対応

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

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

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

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

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

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

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

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

5. 製品間の相違について

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続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がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

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

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

(03)5201-5307

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