

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、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。

---

# SH7263 グループ

## サンプリングレートコンバータ (SRC) 設定例

---

### 要旨

この資料は、SH7263 のサンプリングレートコンバータ (SRC) 設定例を掲載しています。

### 動作確認デバイス

SH7263

### 目次

1. はじめに .....	2
2. 応用例の説明 .....	3
3. 参考プログラム例 .....	22
4. 参考ドキュメント .....	48

## 1. はじめに

### 1.1 仕様

- サンプリングレートコンバータ (SRC) を用いて、PCMデータのサンプリングレートを 22.05 kHz から 44.1 kHz に変換し、シリアルサウンドインタフェース (SSI) を用いて、PCM データを送信します。
- SRC へのデータ入出力および SSI へのデータ入力に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

- サンプリングレートコンバータ (SRC)
- ダイレクトメモリアクセスコントローラ (DMAC)
- シリアルサウンドインタフェース (SSI)

### 1.3 適用条件

- マイコン: SH7263
- 動作周波数:
 

内部クロック	200 MHz
バスクロック	66.67 MHz
周辺クロック	33.33 MHz
- C コンパイラ: ルネサス テクノロジ製  
SuperH RISC engine ファミリー C/C++ コンパイラパッケージ Ver.9.01 Release01
- コンパイルオプション:
 

```
-cpu = sh2afpu -fpu = single -include = "${WORKSPDIR}\inc"
-object = "${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

なし

## 2. 応用例の説明

本応用例では、サンプリングレートコンバータ (SRC) を用いて、PCM データのサンプリングレートを 22.05 kHz から 44.1 kHz に変換します。

### 2.1 使用機能の動作概要

サンプリングレートコンバータ (SRC) は、WMA/MP3/AAC などの各種デコーダで生成されたデータのサンプリングレートを変換するモジュールです。

表 1 に本応用例で使用するサンプリングレートコンバータ (SRC) のモジュール概要を示します。また、図 1 にサンプリングレートコンバータ (SRC) の概念図を示します。サンプリングレートコンバータ (SRC) についての詳細は、「SH7263 グループ ハードウェアマニュアル (RJJ09B0284) サンプリングレートコンバータ (SRC)」の章を参照ください。

表 1 本応用例で使用するモジュール概要

項目	概要
データ形式	16 ビット (ステレオ/モノラル)
サンプリングレート	入力: 8 kHz, 11.025 kHz, 12 kHz, 16 kHz, 22.05 kHz, 24 kHz, 32 kHz, 44.1 kHz, 48 kHz 出力: 44.1 kHz, 48 kHz
処理性能	1 サンプルの出力間隔は最大約 8 $\mu$ s ( $P\phi = 33$ MHz)
SNR	80 dB 以上
割り込み要因	3 種類: 入力データ FIFO エンプティ, 出力データ FIFO フル, 出力データ FIFO オーバライト
DMA 転送要因	2 種類: 入力データ FIFO エンプティ, 出力データ FIFO フル

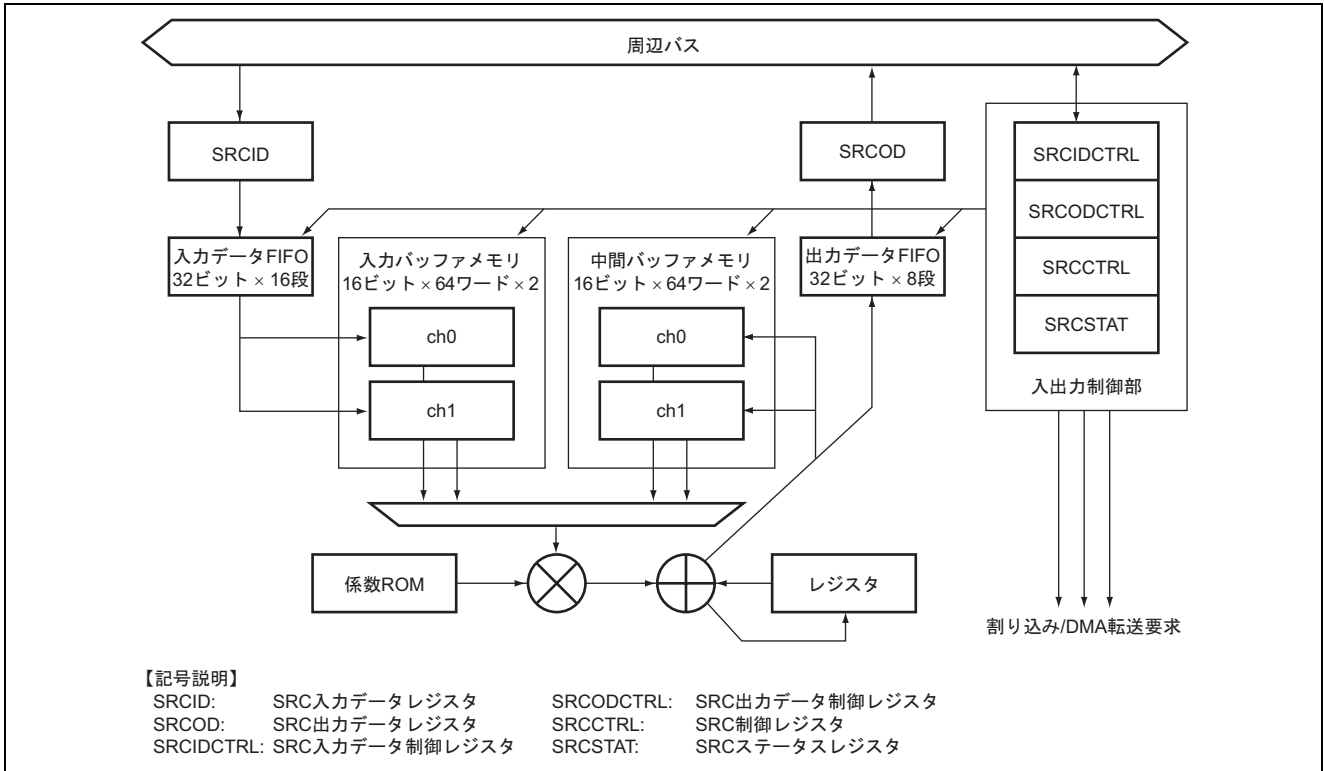


図1 サンプリングレートコンバータ (SRC) の概念図

2.2 使用機能の設定手順

図2にサンプリングレートコンバータ (SRC) の初期設定フロー例を示します。なお、各レジスタ設定の詳細は、「SH7263 グループ ハードウェアマニュアル (RJJ09B0284)」を参照ください。

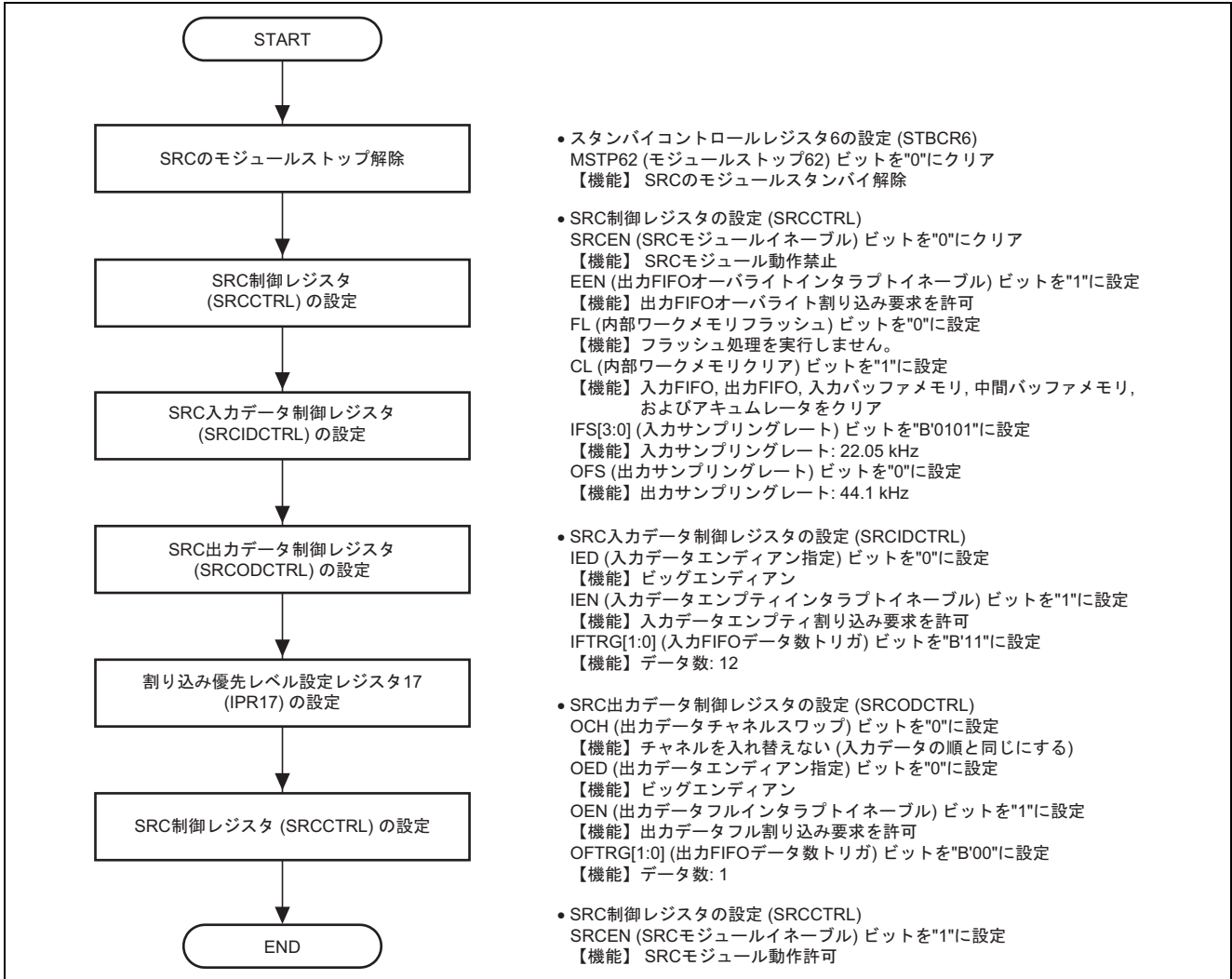


図2 サンプリングレートコンバータの初期設定フロー例

### 2.3 参考プログラムの動作

参考プログラムでは、PCM データのサンプリングレートを 22.05 kHz から 44.1 kHz に変換します。変換された PCM データは DMAC を用いて SSI のトランスミットデータレジスタ (SSITDR) に転送されます。図 3 に PCM データのデータフローを、図 4 に参考プログラムのシーケンスフローを示します。

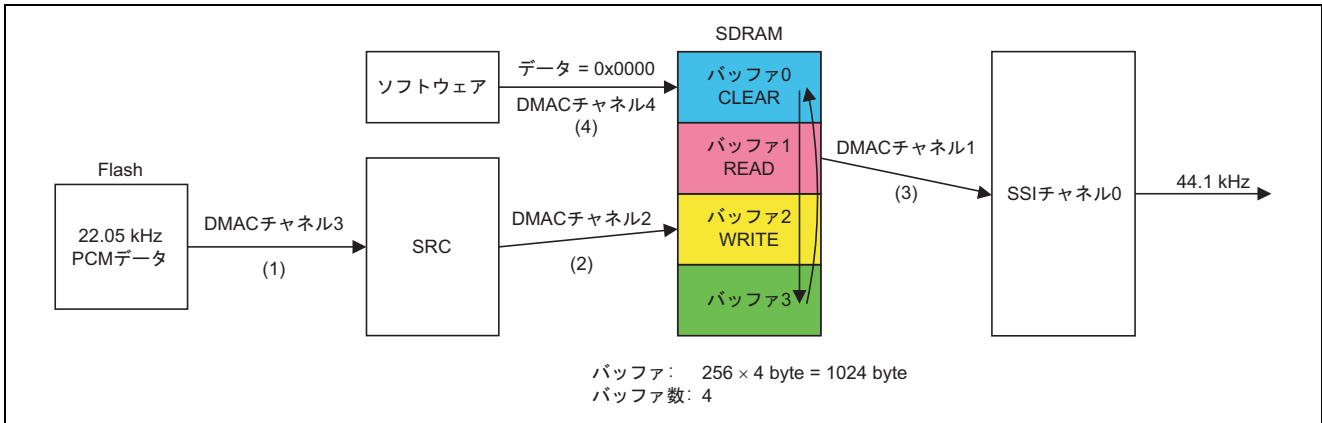


図 3 PCM データのデータフロー

#### 動作説明

- (1) SRC の IDE 割り込み要求により DMAC チャンネル 3 を起動します。DMA 転送により SRC の SRCID レジスタにデータを書き込みます。
- (2) SRC の ODF 割り込み要求により DMAC チャンネル 2 を起動します。DMA 転送により SRC の SRCOD レジスタにデータを書き込みます。  
バッファに空きがある場合は、そのバッファに対して PCM データを転送します。  
バッファに空きがない場合は、DMAC チャンネル 2 を転送禁止に設定しており、バッファが空いた時点で DMAC チャンネル 2 を転送許可に再度設定しています。
- (3) SSI チャンネル 0 の割り込み要求により DMAC チャンネル 1 を起動します。DMA 転送により SSI の SSITDR レジスタにデータを書き込みます。  
バッファに格納された PCM データを SSI チャンネル 0 に転送します。サンプリングレート変換された PCM データすべての転送が終わると SSI をミュートに設定します。
- (4) SSI チャンネル 0 にバッファのデータ転送が終わると、動作の確認用にバッファをクリアします。

バッファに空きがなく DMAC チャンネル 2 の転送を禁止にしている場合、出力 FIFO の読み出しを行えないため出力 FIFO がフルになる場合があります。出力 FIFO がフルのときに次のデータの変換処理が終了すると SRC ステータスレジスタ (SRCSTAT) の OVF ビットが 1 にセットされます。OVF ビットがクリアされるまで、SRC の変換処理は停止します。変換処理が停止している間、DMA チャンネル 3 に対して転送要求は発生しません。OVF ビットクリアは、DMAC チャンネル 2 を再度転送許可に設定するときに行っています。



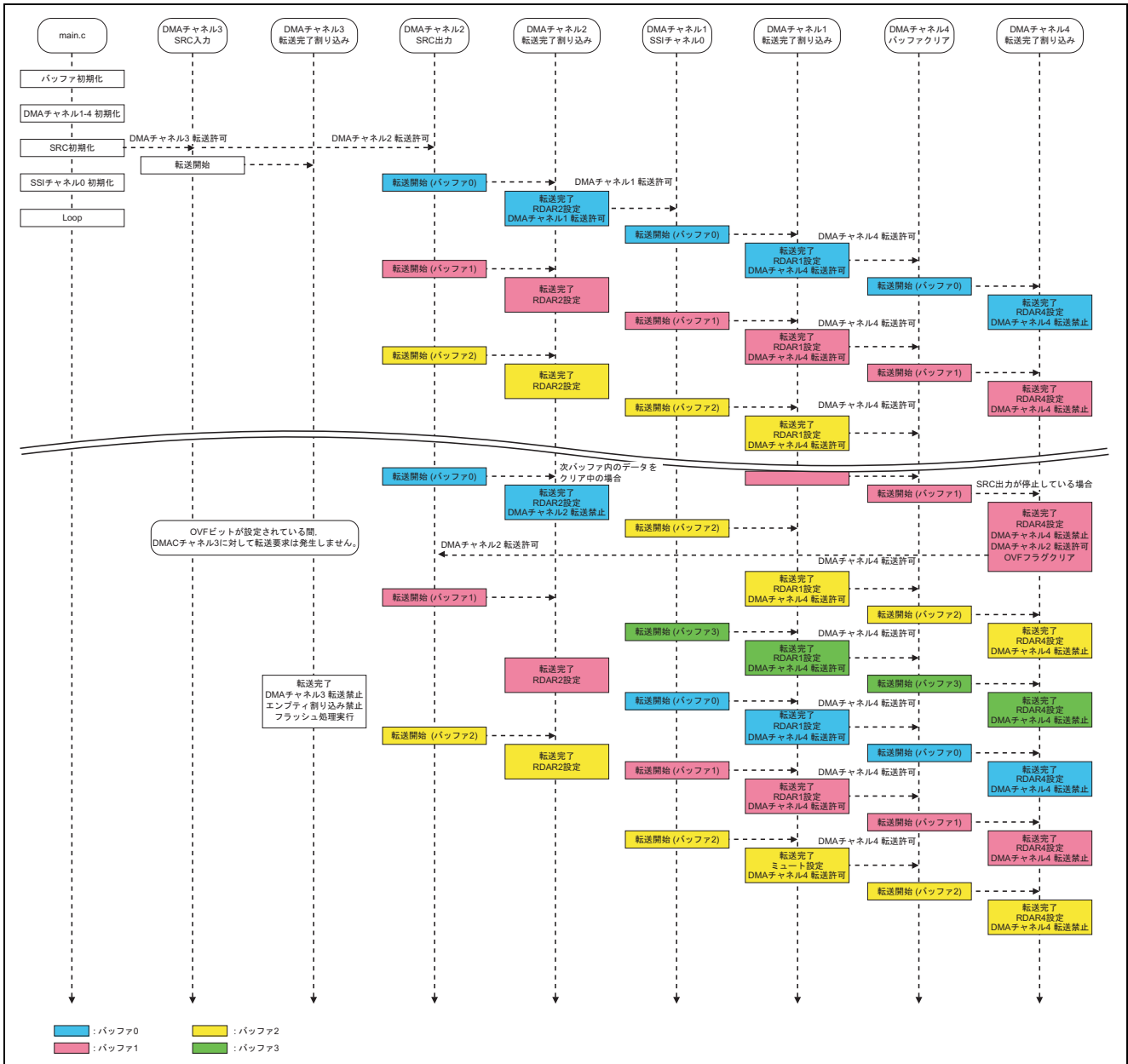


図4 参考プログラムのシーケンスフロー

図 5 ~ 図 10 に参考プログラムのフローチャートを示します。

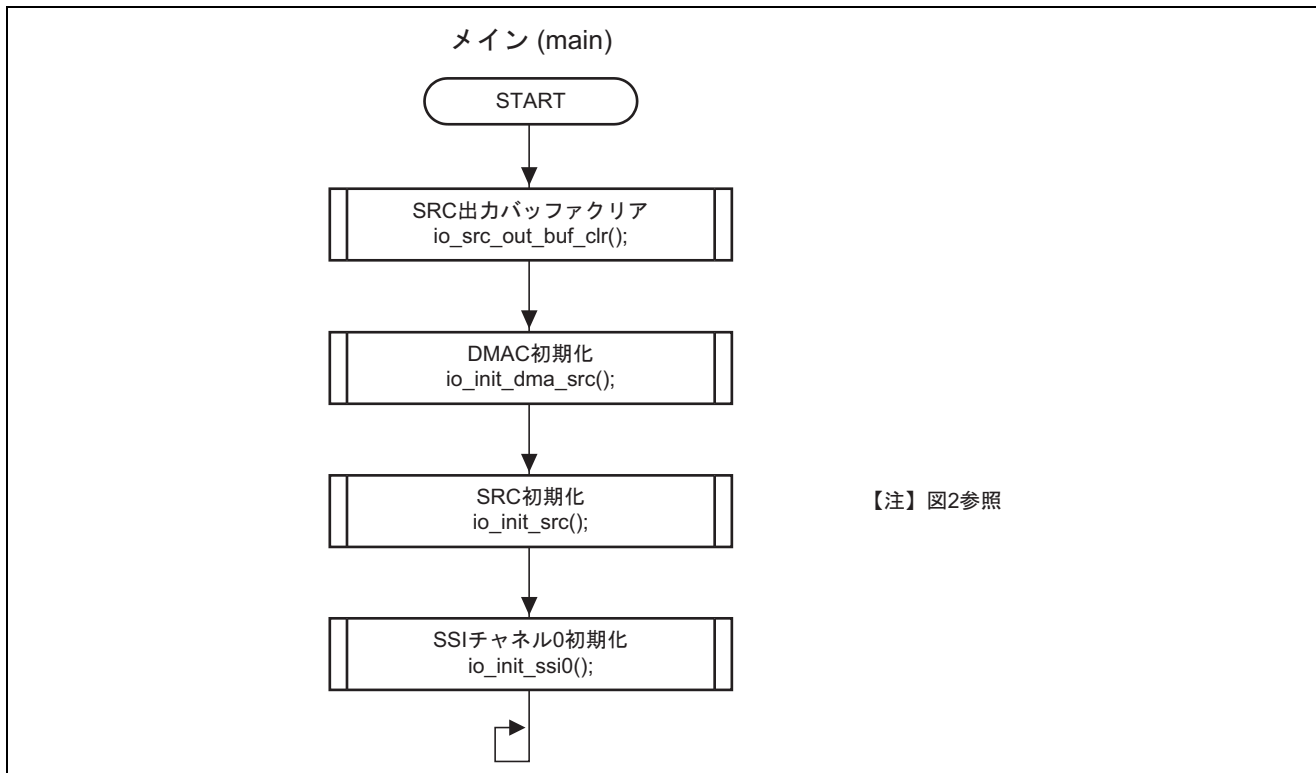


図 5 参考プログラムのフローチャート (1)

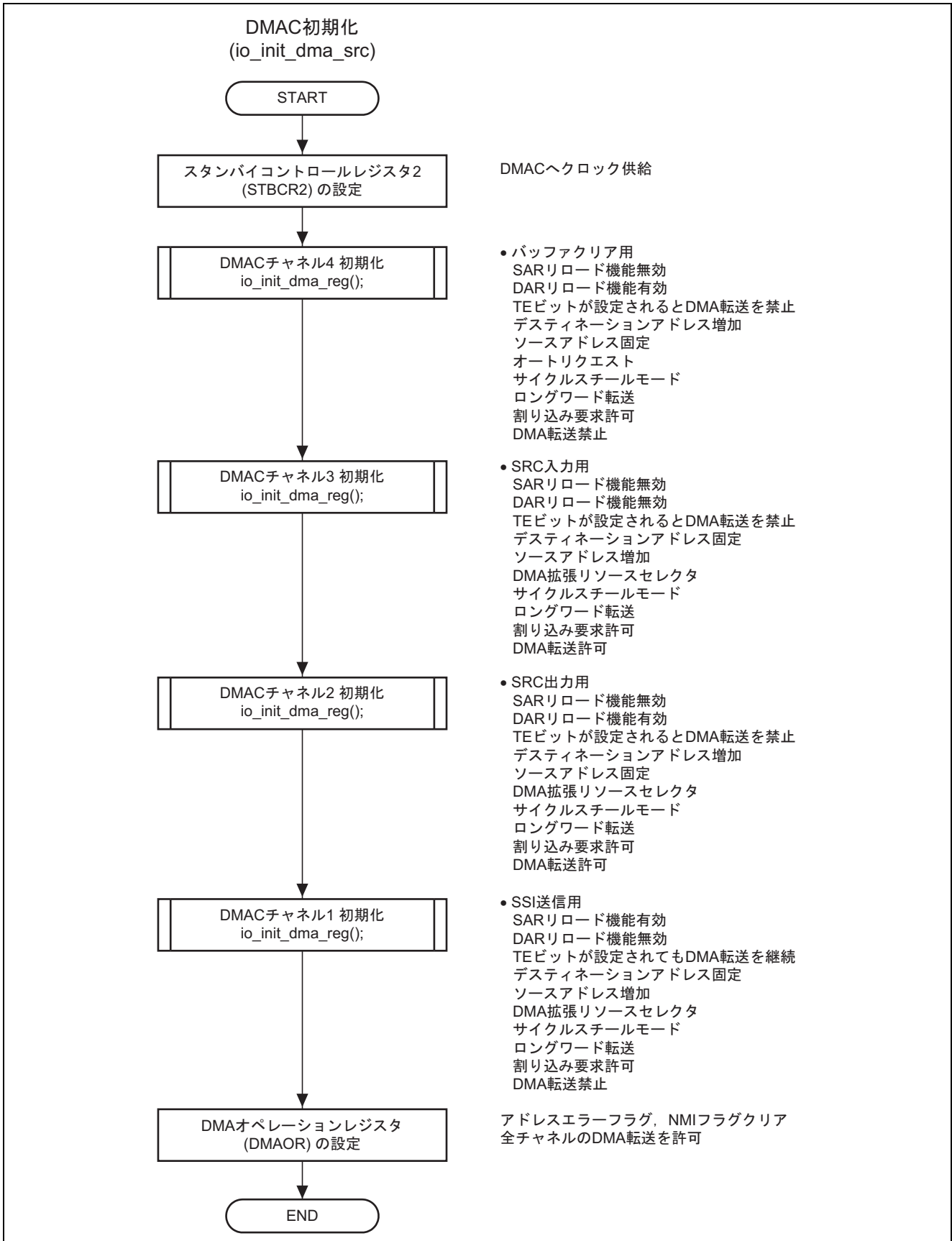


図 6 参考プログラムのフローチャート (2)

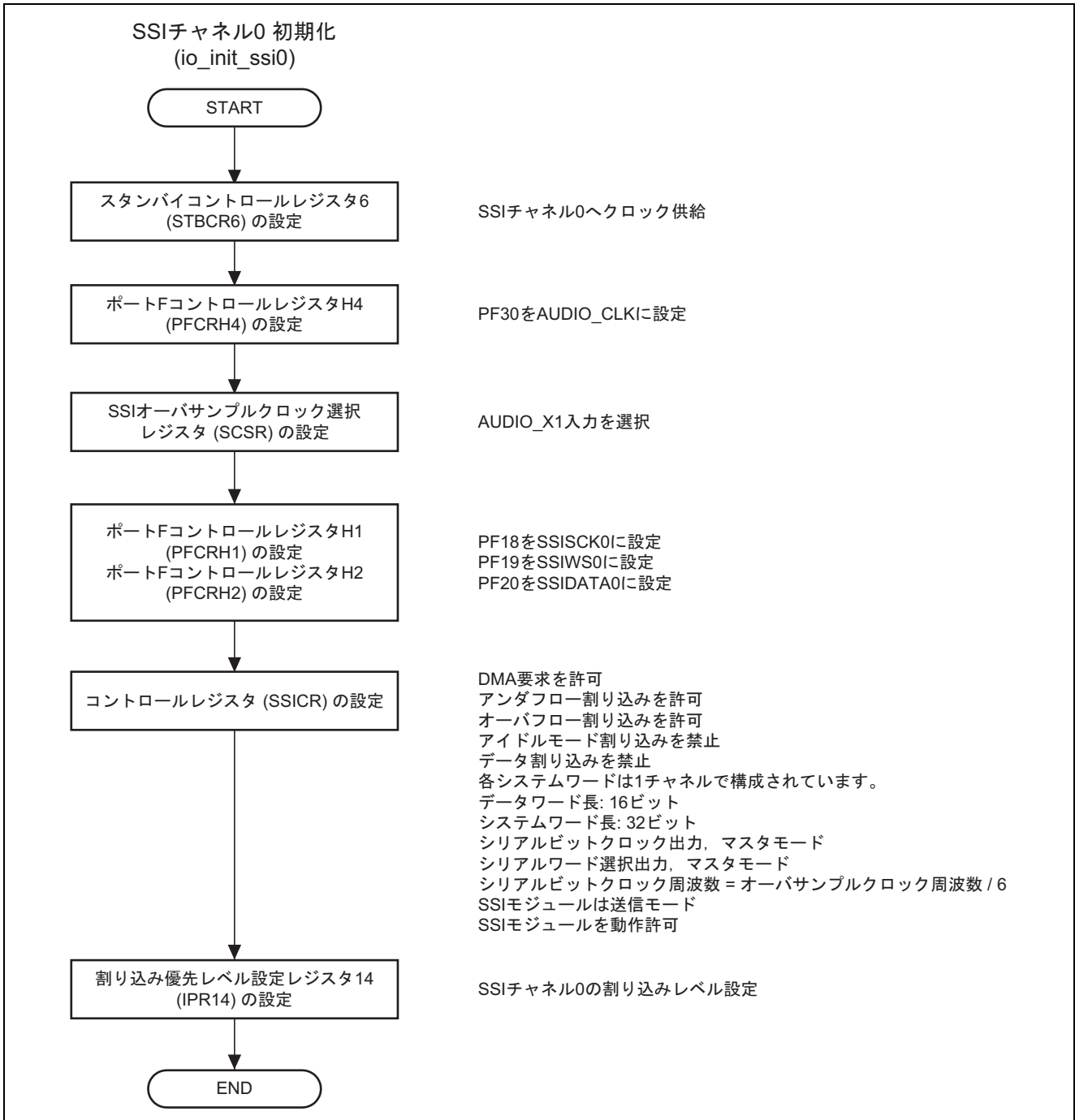


図7 参考プログラムのフローチャート (3)

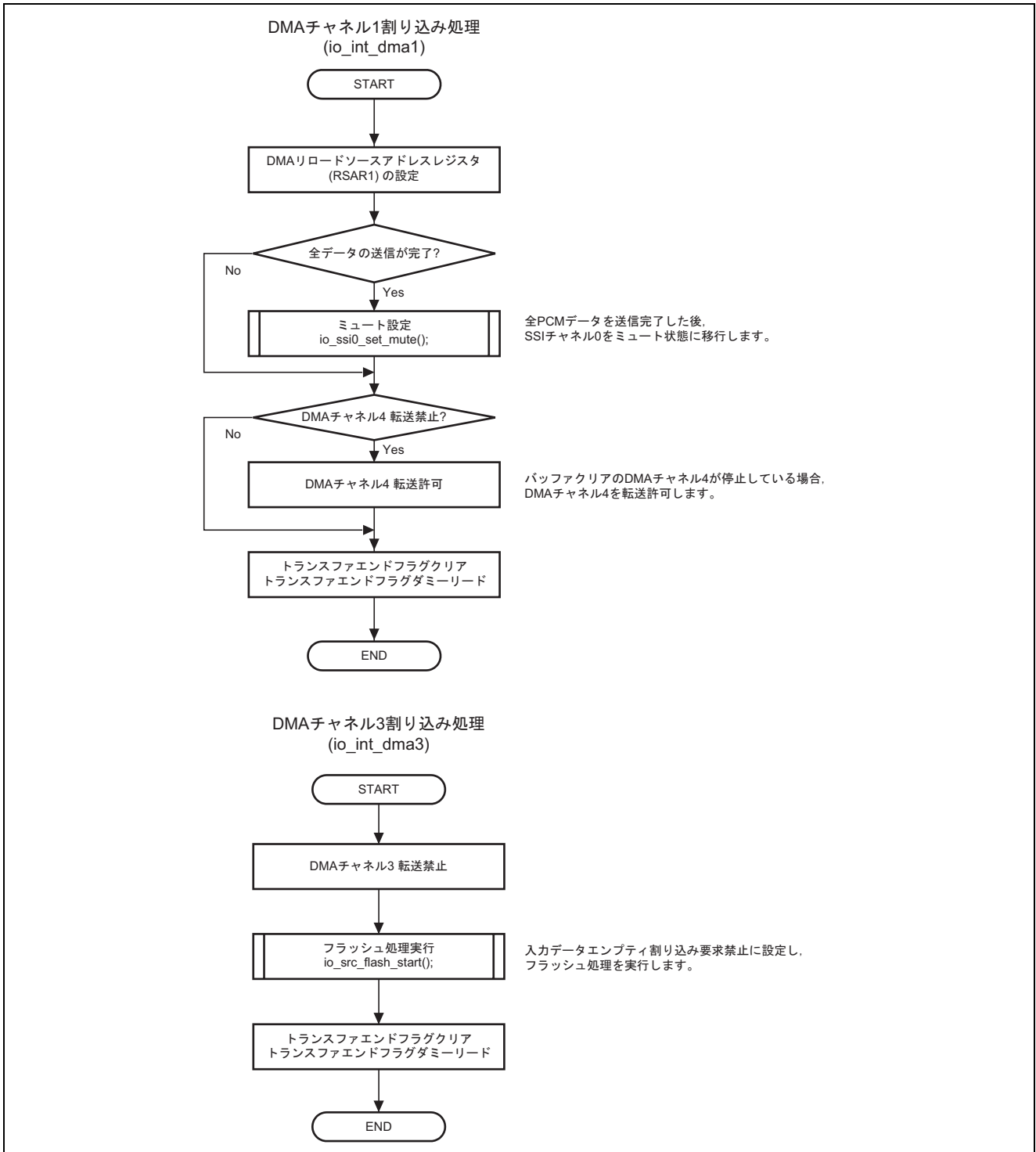


図8 参考プログラムのフローチャート (4)

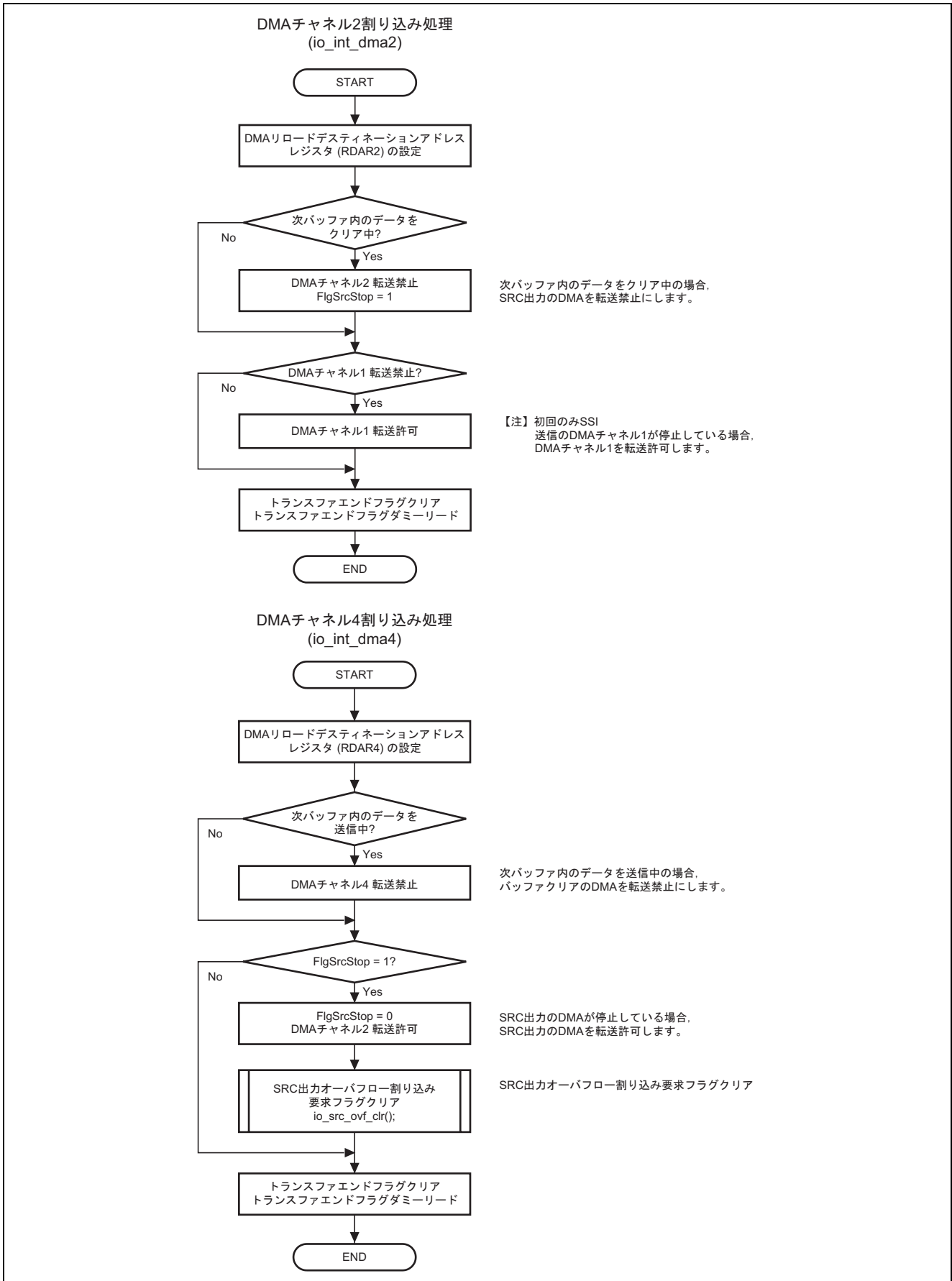


図9 参考プログラムのフローチャート (5)

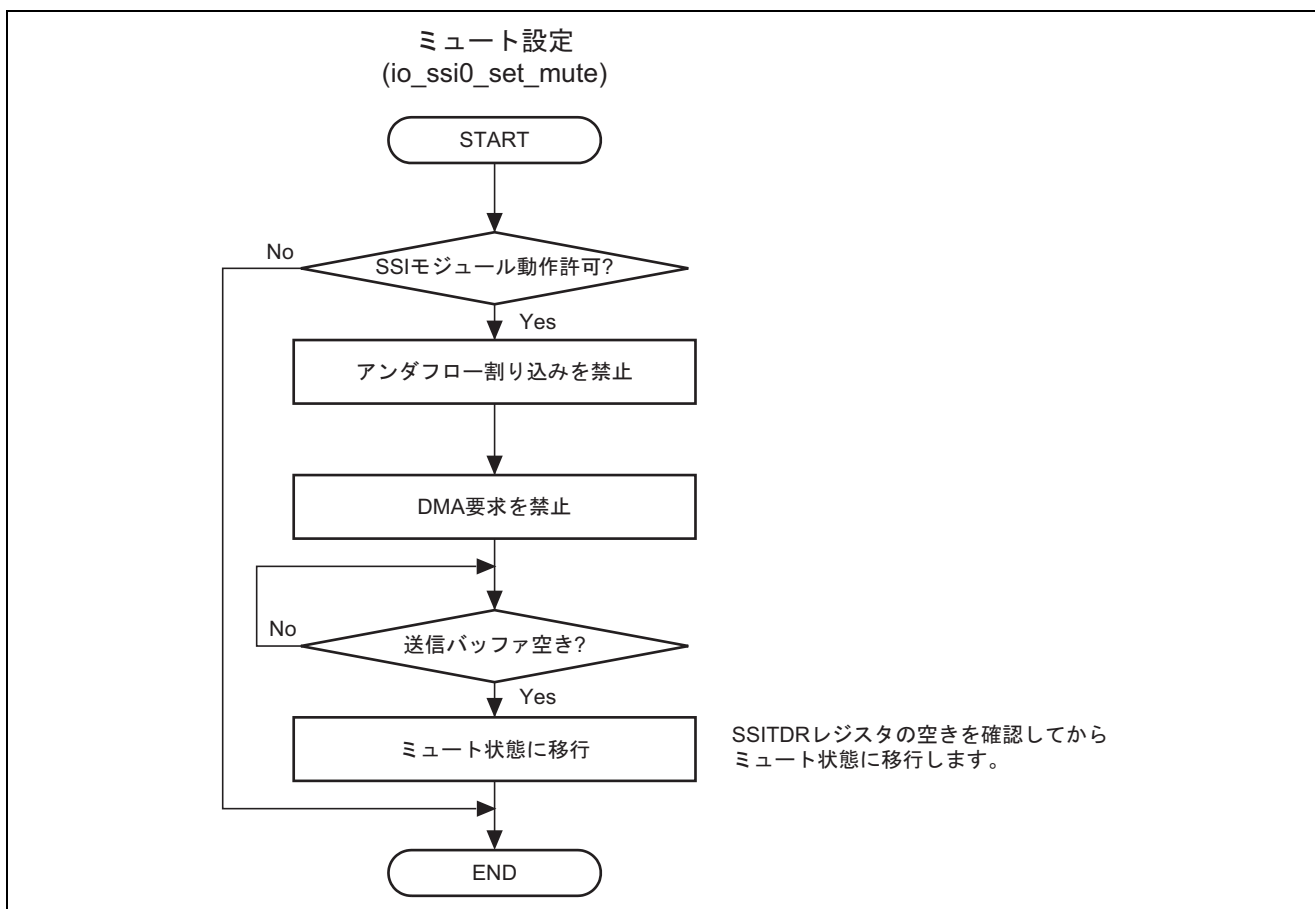


図 10 参考プログラムのフローチャート (6)

## 2.4 参考プログラムの処理手順

表 2~表 7 に参考プログラムでのレジスタ設定を示します。

表 2 参考プログラムでの SRC レジスタ設定

レジスタ名	アドレス	設定値	機能
SRC 入力データ制御レジスタ (SRCIDCTRL)	H'FFFF 4008	H'0103	IED = "0": ビッグエンディアン IEN = "1": 入力データエンプティ割り込み要求を許可 IFTRG[1:0] = "B'11": データ数 12
SRC 出力データ制御レジスタ (SRCODCTRL)	H'FFFF 400A	H'0100	OCH = "0": チャネルを入れ替えない OED = "0": ビッグエンディアン OEN = "1": 出力データフル割り込み要求を許可 OFTRG[1:0] = "B'00": データ数 1
SRC 制御レジスタ (SRCCTRL)	H'FFFF 400C	H'0550	SRCEN = "0": SRC モジュール動作禁止 EEN = "1": 出力 FIFO オーバライト割り込み要求を許可 FL = "0": フラッシュ処理を実行しません。 CL = "1": 入力 FIFO, 出力 FIFO, 入力バッファメモリ, 中間バッファメモリ, およびアキュムレータをクリア IFS[3:0] = "B'0101": 22.05 kHz OFS = "0": 44.1 kHz
		H'1450	SRCEN = "1": SRC モジュール動作許可



表 3 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 C000	H'1C0B D553	DMEN = "1": DMA 要求を許可 UIEN = "1": アンダフロー割り込みを許可 OIEN = "1": オーバフロー割り込みを許可 ILEN = "0": アイドルモード割り込みを禁止 CHNL = "B'00": 各システムワードは 1 チャンネルで構成 DWL = "B'001": データワード長 16 ビット SWL = "B'011": システムワード長 32 ビット SCKD = "1": シリアルビットクロック出力, マスタモード SWSD = "1": シリアルワード選択出力, マスタモード SCKP = "0": シリアルビットクロック極性 SSISCK の立ち下がりがエッジで変化 SWSP = "1": シリアル WS 極性 第 1 チャンネルは "H", 第 2 チャンネルは "L" SPDP = "0": パディングビットは "L" SDTA = "1": パディングビット, シリアルデータの順に送受信 PDTA = "0": データワード長 16 ビット時, 設定は無視されます。ビット 15-0 には第 1 のデータワード, ビット 31-16 には第 2 のデータワードが格納されています。 DEL = "1": SSIWS と SSIDATA 間の遅延なし CKDV = "B'101": オーバサンプルクロック / 6 MUEN = "0": ミュート状態ではない TRMD = "1": SSI モジュールは送信モード EN = "1": SSI モジュール動作を許可

表 4 参考プログラムの DMAC レジスタ設定 (SSI)

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 (CHCR1)	H'FFFE 101C	H'0000 0000	DE = "0": DMA 転送禁止
		H'2010 1814	TC = "0": 1 回転送 RLDSAR = "1": SAR リロード機能有効 RLDDAR = "0": DAR リロード機能無効 TEMASK = "1": *1 TE ビットが設定されても DMA 転送を継続 DM = "B'00": デスティネーションアドレス固定 SM = "B'01": ソースアドレス増加 RS = "B'1000": DMA 拡張リソースセクタ TB = "0": サイクルスチールモード TS = "B'10": ロングワード転送 IE = "1": 割り込み要求許可
		H'2010 1815	DE = "1": DMA 転送許可
DMA ソースアドレス レジスタ_1 (SAR1)	H'FFFE 1010	H'2C00 0000	転送元の開始アドレス: SDRAM 領域に設定
DMA リロードソース アドレスレジスタ_1 (RSAR1)	H'FFFE 1110	H'2C00 0400	転送元の開始アドレス: SDRAM 領域に設定
DMA デスティネーション アドレスレジスタ_1 (DAR1)	H'FFFE 1014	H'FFFF C008	転送先の開始アドレス: SSI トランスミットデータレジスタ (SSITDR_0)
DMA リロード デスティネーション アドレスレジスタ_1 (RDAR1)	H'FFFE 1114	H'FFFF C008	転送先の開始アドレス: SSI トランスミットデータレジスタ (SSITDR_0)
DMA トランスファ カウントレジスタ_1 (DMATCR1)	H'FFFE 1018	H'0000 0100	転送回数: 256 回
DMA リロード トランスファカウント レジスタ_1 (RDMATCR1)	H'FFFE 1118	H'0000 0100	転送回数: 256 回
DMA 拡張リソース セクタ 0 (DMARSO)	H'FFFE 1300	H'0023	SSI チャンネル 0 を選択

【注】 1. SSI は一定のタイミングで PCM データを出力する必要があります。

TEMASK = "0"とした場合は DMA 転送完了時に DMA が停止状態となります。そのため DMA 転送完了割り込み処理がメイン処理の割り込み禁止期間などで遅延した場合に SSI がアンダフローする場合があります。

この問題を回避するため、TEMASK = "1"として DMA 転送完了時も DMA 転送を継続することを推奨します。

表 5 参考プログラムの DMAC レジスタ設定 (SRC 入力)

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_3 (CHCR3)	H'FFFE 103C	H'0000 0000	DE = "0": DMA 転送禁止
		H'0000 1814	TC = "0": 1 回転送 RLDSAR = "0": SAR リロード機能無効 RLDDAR = "0": DAR リロード機能無効 TEMASK = "0": TE ビットが設定されると DMA 転送を禁止 DM = "B'00": デスティネーションアドレス 固定 SM = "B'01": ソースアドレス増加 RS = "B'1000": DMA 拡張リソースセクタ TB = "0": サイクルスチールモード TS = "B'10": ロングワード転送 IE = "1": 割り込み要求許可
		H'0000 1815	DE = "1": DMA 転送許可
DMA ソースアドレス レジスタ_3 (SAR3)	H'FFFE 1030	H'0000 1900	転送元の開始アドレス: Flash 領域に設定
DMA リロードソース アドレスレジスタ_3 (RSAR3)	H'FFFE 1130	H'0000 1900	転送元の開始アドレス: Flash 領域に設定
DMA デスティネーション アドレスレジスタ_3 (DAR3)	H'FFFE 1034	H'FFFF 4000	転送先の開始アドレス: SRC 入力データレジスタ (SRCID)
DMA リロード デスティネーション アドレスレジスタ_3 (RDAR3)	H'FFFE 1134	H'FFFF 4000	転送先の開始アドレス: SRC 入力データレジスタ (SRCID)
DMA トランスファ カウントレジスタ_3 (DMATCR3)	H'FFFE 1038	H'0000 4EDF	転送回数: 20191 回
DMA リロード トランスファカウント レジスタ_3 (RDMATCR3)	H'FFFE 1138	H'0000 4EDF	転送回数: 20191 回
DMA 拡張リソース セクタ 1 (DMARS1)	H'FFFE 1304	H'4100	SRC 入力データエンプティを選択

表 6 参考プログラムの DMAC レジスタ設定 (SRC 出力)

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 (CHCR2)	H'FFFE 102C	H'0000 0000	DE = "0": DMA 転送禁止
		H'1000 4814	TC = "0": 1 回転送 RLDSAR = "0": SAR リロード機能無効 RLDDAR = "1": DAR リロード機能有効 TEMASK = "0": TE ビットが設定されると DMA 転送を禁止 DM = "B'01": デスティネーションアドレス 増加 SM = "B'00": ソースアドレス固定 RS = "B'1000": DMA 拡張リソースセクタ TB = "0": サイクルスチールモード TS = "B'10": ロングワード転送 IE = "1": 割り込み要求許可
		H'1000 4815	DE = "1": DMA 転送許可
DMA ソースアドレス レジスタ_2 (SAR2)	H'FFFE 1020	H'FFFF 4004	転送元の開始アドレス: SRC 出力データレジスタ (SRCOD)
DMA リロードソース アドレスレジスタ_2 (RSAR2)	H'FFFE 1120	H'FFFF 4004	転送元の開始アドレス: SRC 出力データレジスタ (SRCOD)
DMA デスティネーション アドレスレジスタ_2 (DAR2)	H'FFFE 1024	H'2C00 0000	転送先の開始アドレス: SDRAM 領域に設定
DMA リロード デスティネーション アドレスレジスタ_2 (RDAR2)	H'FFFE 1124	H'2C00 0000	転送先の開始アドレス: SDRAM 領域に設定
DMA トランスファ カウントレジスタ_2 (DMATCR2)	H'FFFE 1028	H'0000 0100	転送回数: 256 回
DMA リロード トランスファカウント レジスタ_2 (RDMATCR2)	H'FFFE 1128	H'0000 0100	転送回数: 256 回
DMA 拡張リソース セクタ 1 (DMARS1)	H'FFFE 1304	H'0042	SRC 出力データフルを選択

表7 参考プログラムの DMAC レジスタ設定 (バッファクリア)

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_4 (CHCR4)	H'FFFE 104C	H'0000 0000	DE = "0": DMA 転送禁止
		H'1000 4414	TC = "0": 1 回転送 【注】内蔵周辺モジュールリクエストの場合のみ有効 RLDSAR = "0": SAR リロード機能無効 RLDDAR = "1": DAR リロード機能有効 TEMASK = "0": TE ビットが設定されると DMA 転送を禁止 DM = "B'01": デスティネーションアドレス増加 SM = "B'00": ソースアドレス固定 RS = "B'0100": オートリクエスト TB = "0": サイクルスチールモード TS = "B'10": ロングワード転送 IE = "1": 割り込み要求許可
		H'1000 4415	DE = "1": DMA 転送許可
DMA ソースアドレス レジスタ_4 (SAR4)	H'FFFE 1040	H'FFF8 07E0	転送元の開始アドレス: 内蔵 RAM 領域に設定
DMA リロードソース アドレスレジスタ_4 (RSAR4)	H'FFFE 1140	H'FFF8 07E0	転送元の開始アドレス: 内蔵 RAM 領域に設定
DMA デスティネーション アドレスレジスタ_4 (DAR4)	H'FFFE 1044	H'2C00 0000	転送先の開始アドレス: SDRAM 領域に設定
DMA リロード デスティネーション アドレスレジスタ_4 (RDAR4)	H'FFFE 1144	H'2C00 0000	転送先の開始アドレス: SDRAM 領域に設定
DMA トランスファ カウントレジスタ_4 (DMATCR4)	H'FFFE 1048	H'0000 0100	転送回数: 256 回
DMA リロードトランスファ カウントレジスタ_4 (RDMATCR4)	H'FFFE 1148	H'0000 0100	転送回数: 256 回
DMA 拡張リソース セクタ 2 (DMARS2)	H'FFFE 1308	H'0000	オートリクエストでは使用しません。

表 8 に参考プログラムで使用する変数を，表 9 に参考プログラムで使用しているマクロ定義を示します。

表 8 参考プログラムで使用する変数

変数名	機能	領域	使用モジュール名
unsigned long rom_ding22_05_bin[ ]	サンプリングレート 22.05 kHz PCM データ	Flash	void io_init_dma3(void *src, void *dst, size_t size, unsigned int mode)
unsigned long size_ding22_05_bin	サンプリングレート 22.05 kHz PCM データのサイズ	Flash	void io_init_dma3(void *src, void *dst, size_t size, unsigned int mode)

表 9 参考プログラムで使用しているマクロ定義

マクロ定義	設定値	機能
IN_SAMPLING_8_KHZ	H'0000	SRC 入力サンプリングレート 8 kHz
IN_SAMPLING_11_025_KHZ	H'0001	SRC 入力サンプリングレート 11.025 kHz
IN_SAMPLING_12_KHZ	H'0002	SRC 入力サンプリングレート 12 kHz
IN_SAMPLING_16_KHZ	H'0003	SRC 入力サンプリングレート 16 kHz
IN_SAMPLING_22_05_KHZ	H'0004	SRC 入力サンプリングレート 22.05 kHz
IN_SAMPLING_24_KHZ	H'0005	SRC 入力サンプリングレート 24 kHz
IN_SAMPLING_32_KHZ	H'0006	SRC 入力サンプリングレート 32 kHz
IN_SAMPLING_44_1_KHZ	H'0008	SRC 入力サンプリングレート 44.1 kHz
IN_SAMPLING_48_KHZ	H'0009	SRC 入力サンプリングレート 48 kHz
OUT_SAMPLING_44_1_KHZ	H'0000	SRC 出力サンプリングレート 44.1 kHz
OUT_SAMPLING_48_KHZ	H'0001	SRC 出力サンプリングレート 48 kHz
SRC_OUT_BUF_SIZE	256	SRC 出力側バッファサイズ
SRC_OUT_BUF_NUM	4	SRC 出力側バッファ数
BUF_CLR	0	バッファクリア動作
BUF_SRC	1	SRC 出力動作
BUF_SSI	2	SSI 送信動作
DMA_RLDSAR_OFF	H'0000 0000	DMA SAR リロード機能 無効
DMA_RLDSAR_ON	H'2000 0000	DMA SAR リロード機能 有効
DMA_RLDDAR_OFF	H'0000 0000	DMA DAR リロード機能 無効
DMA_RLDDAR_ON	H'1000 0000	DMA DAR リロード機能 有効
DMA_TE_STOP	H'0000 0000	DMA TE ビットがセットされると DMA 転送停止
DMA_TE_CONT	H'0010 0000	DMA TE ビットがセットされても DMA 転送継続
DMA_DAR_FIX	H'0000 0000	DMA デスティネーションアドレス 固定
DMA_DAR_INC	H'0000 4000	DMA デスティネーションアドレス 増加
DMA_DAR_DEC	H'0000 8000	DMA デスティネーションアドレス 減少
DMA_SAR_FIX	H'0000 0000	DMA ソースアドレス 固定
DMA_SAR_INC	H'0000 1000	DMA ソースアドレス 増加
DMA_SAR_DEC	H'0000 2000	DMA ソースアドレス 減少
DMA_AUTO_REQ	H'0000 0400	DMA リソースセクタ: オートリクエスト
DMA_ENH_RES	H'0000 0800	DMA リソースセクタ: 拡張リソースセクタ
DMA_SIZE_BYTE	H'0000 0000	DMA バイト転送
DMA_SIZE_WORD	H'0000 0008	DMA ワード転送
DMA_SIZE_LONG	H'0000 0010	DMA ロングワード転送
DMA_SIZE_LONGx4	H'0000 0018	DMA 16 バイト転送
DMA_INT_DISABLE	H'0000 0000	DMA DMA 転送完了割り込み未使用
DMA_INT_ENABLE	H'0000 0004	DMA DMA 転送完了割り込み使用
DMA_DISABLE	H'0000 0000	DMA DMA 転送を禁止
DMA_ENABLE	H'0000 0001	DMA DMA 転送を許可

### 3. 参考プログラム例

```

1  /*"FILE COMMENT"*****
2  *
3  *   System Name : SH7263 Sample Program
4  *   File Name   : main.c
5  *   Contents    : SRC サンプルプログラム
6  *   Version     : 1.00.00
7  *   Model       : R0K572630D001BR
8  *   CPU         : SH7263
9  *   Compiler    : SHC9.1.1.0
10 *   note        :
11 *
12 *               <注意事項>
13 *               本サンプルプログラムはすべて参考資料であり、
14 *               その動作を保証するものではありません。
15 *               本サンプルプログラムはお客様のソフトウェア開発時の
16 *               技術参考資料としてご利用ください。
17 *
18 *   The information described here may contain technical inaccuracies or
19 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
20 *   assume no responsibility for any damage, liability, or other loss rising
21 *   from these inaccuracies or errors.
22 *
23 *   Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
24 *   AND Renesas Solutions Corp. All Rights Reserved
25 *
26 *   history      : 2008.05.09 ver.1.00.00
27 *"FILE COMMENT END"*****/
28 #include <string.h>
29 #include "iodefine.h" /* iodefine.h は、HEW で自動生成されるファイルです。 */
30 #include "src.h"
31
32 /* ==== プロトタイプ宣言 ==== */
33 void main(void);
34
    
```

図 11 サンプルプログラムリスト"main.c" (1)



```

35  /*"FUNC COMMENT"*****
36  * Outline      : サンプルプログラムメイン
37  *-----
38  * Include      : #include <string.h>
39  *-----
40  * Declaration : void main(void);
41  *-----
42  * Function     : サンプリングレート変更後の PCM データ格納バッファをクリア
43  *              : した後, DMA ch1~4, SRC, SSI ch0 を設定します。
44  *              : SRC の設定は入力サンプリングレートを 22.05kHz ,
45  *              : 出力サンプリングレートを 44.1kHz に設定しています。
46  *-----
47  * Argument    : void
48  *-----
49  * Return Value: void
50  *-----
51  * Notice      :
52  *"FUNC COMMENT END"*****/
53  void main(void)
54  {
55      /* ---- SRC バッファ初期化 ---- */
56      src_out_buf_clr();                /* SRC 出力側バッファクリア */
57
58      /* ---- DMA ch1,2,3,4 初期化 ---- */
59      io_init_dma_src();
60          /* ch1 : SSI 送信 */
61          /* ch2 : SRC 出力 */
62          /* ch3 : SRC 入力 */
63          /* ch4 : バッファクリア */
64      /* ---- SRC 初期化 ---- */
65      io_init_src(    IN_SAMPLING_22_05_KHZ,    /* 入力サンプリングレート */
66                   OUT_SAMPLING_44_1_KHZ);    /* 出力サンプリングレート */
67
68      /* ---- SSI ch0 初期化 ---- */
69      io_init_ssi0();
70
71      while(1){
72          /* Program end */
73      }
74  }
75
76  /* End of File */
    
```

図 12 サンプルプログラムリスト"main.c" (2)

```

1  /*"FILE COMMENT"*****
2  *
3  *      System Name : SH7263 Sample Program
4  *      File Name   : dmac.c
5  *      Contents    : DMAC
6  *      Version     : 1.00.00
7  *      Model       : R0K572630D001BR
8  *      CPU         : SH7263
9  *      Compiler    : SHC9.1.1.0
10 *      note        :
11 *
12 *      <注意事項>
13 *      本サンプルプログラムはすべて参考資料であり、
14 *      その動作を保証するものではありません。
15 *      本サンプルプログラムはお客様のソフトウェア開発時の
16 *      技術参考資料としてご利用ください。
17 *
18 *      The information described here may contain technical inaccuracies or
19 *      typographical errors. Renesas Technology Corporation and Renesas Solutions
20 *      assume no responsibility for any damage, liability, or other loss rising
21 *      from these inaccuracies or errors.
22 *
23 *      Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
24 *      AND Renesas Solutions Corp. All Rights Reserved
25 *
26 *      history      : 2008.05.09 ver.1.00.00
27 *"FILE COMMENT END"*****/
28 #include <string.h>
29 #include "iodefine.h"      /* iodefine.h は、HEW で自動生成されるファイルです。 */
30 #include "src.h"
31 #include "dmac.h"
32
33 extern unsigned long size_ding22_05_bin;
34 extern unsigned long rom_ding22_05_bin[];
35
36 const ST_DMA DMA[8]={
37     { /* ch0 */
38         &DMAC.SAR0.LONG,      /* DMA ソースアドレスレジスタ (SAR) */
39         &DMAC.DAR0.LONG,      /* DMA ディスティネーションアドレスレジスタ (DAR) */
40         &DMAC.DMATCR0.LONG,   /* DMA トランスファカウントレジスタ (DMATCR) */
41         &DMAC.CHCR0.LONG,     /* DMA チャネルコントロールレジスタ (CHCR) */
42         &DMAC.RSAR0.LONG,     /* DMA リロードソースアドレスレジスタ (RSAR) */
43         &DMAC.RDAR0.LONG,     /* DMA リロードディスティネーションアドレスレジスタ (RDAR) */
44         &DMAC.RDMATCR0.LONG,  /* DMA リロードトランスファカウントレジスタ (RDMATCR) */
45     },

```

図 13 サンプルプログラムリスト"dmac.c" (1)

```

46  { /* ch1 */
47      &DMAC.SAR1.LONG,
48      &DMAC.DAR1.LONG,
49      &DMAC.DMATCR1.LONG,
50      &DMAC.CHCR1.LONG,
51      &DMAC.RSAR1.LONG,
52      &DMAC.RDAR1.LONG,
53      &DMAC.RDMATCR1.LONG,
54  },
55  { /* ch2 */
56      &DMAC.SAR2.LONG,
57      &DMAC.DAR2.LONG,
58      &DMAC.DMATCR2.LONG,
59      &DMAC.CHCR2.LONG,
60      &DMAC.RSAR2.LONG,
61      &DMAC.RDAR2.LONG,
62      &DMAC.RDMATCR2.LONG,
63  },
64  { /* ch3 */
65      &DMAC.SAR3.LONG,
66      &DMAC.DAR3.LONG,
67      &DMAC.DMATCR3.LONG,
68      &DMAC.CHCR3.LONG,
69      &DMAC.RSAR3.LONG,
70      &DMAC.RDAR3.LONG,
71      &DMAC.RDMATCR3.LONG,
72  },
73  { /* ch4 */
74      &DMAC.SAR4.LONG,
75      &DMAC.DAR4.LONG,
76      &DMAC.DMATCR4.LONG,
77      &DMAC.CHCR4.LONG,
78      &DMAC.RSAR4.LONG,
79      &DMAC.RDAR4.LONG,
80      &DMAC.RDMATCR4.LONG,
81  },
82  { /* ch5 */
83      &DMAC.SAR5.LONG,
84      &DMAC.DAR5.LONG,
85      &DMAC.DMATCR5.LONG,
86      &DMAC.CHCR5.LONG,
87      &DMAC.RSAR5.LONG,
88      &DMAC.RDAR5.LONG,
89      &DMAC.RDMATCR5.LONG,
90  },
91  { /* ch6 */
92      &DMAC.SAR6.LONG,
93      &DMAC.DAR6.LONG,
94      &DMAC.DMATCR6.LONG,
95      &DMAC.CHCR6.LONG,
96      &DMAC.RSAR6.LONG,
97      &DMAC.RDAR6.LONG,
98      &DMAC.RDMATCR6.LONG,
99  },
100 { /* ch7 */
101     &DMAC.SAR7.LONG,
102     &DMAC.DAR7.LONG,
103     &DMAC.DMATCR7.LONG,
104     &DMAC.CHCR7.LONG,
105     &DMAC.RSAR7.LONG,
106     &DMAC.RDAR7.LONG,
107     &DMAC.RDMATCR7.LONG,
108 } };
109

```

図 14 サンプルプログラムリスト"dmac.c" (2)

```

110  const ST_DMA_REG DmaReg[8] = {
111      { /* ch0 */
112          0ul,          /* DMA チャンネルコントロールレジスタ (CHCR) */
113          0u,          /* DMA 拡張リソースセクタ (DMARS) */
114          0u           /* 割り込み優先レベル設定レジスタ (IPR) */
115      },
116      { /* ch1 */
117          DMA_RLDSAR_ON | DMA_RLDDAR_OFF | DMA_TE_CONT | DMA_DAR_FIX | DMA_SAR_INC | DMA_ENH_RES
118          | DMA_SIZE_LONG | DMA_INT_ENABLE | DMA_DISABLE,
119          DMARS_SSI_0,
120          0x0fu
121      },
122      { /* ch2 */
123          DMA_RLDSAR_OFF | DMA_RLDDAR_ON | DMA_TE_STOP | DMA_DAR_INC | DMA_SAR_FIX | DMA_ENH_RES
124          | DMA_SIZE_LONG | DMA_INT_ENABLE | DMA_ENABLE,
125          DMARS_SRC_FULLL,
126          0x0au
127      },
128      { /* ch3 */
129          DMA_RLDSAR_OFF | DMA_RLDDAR_OFF | DMA_TE_STOP | DMA_DAR_FIX | DMA_SAR_INC | DMA_ENH_RES
130          | DMA_SIZE_LONG | DMA_INT_ENABLE | DMA_ENABLE,
131          DMARS_SRC_EMP,
132          0x01u
133      },
134      { /* ch4 */
135          DMA_RLDSAR_OFF | DMA_RLDDAR_ON | DMA_TE_STOP | DMA_DAR_INC | DMA_SAR_FIX | DMA_AUTO_REQ
136          | DMA_SIZE_LONG | DMA_INT_ENABLE | DMA_DISABLE,
137          0u,
138          0x0au
139      },
140      { /* ch5 */
141          0ul,
142          0u,
143          0u
144      },
145      { /* ch6 */
146          0ul,
147          0u,
148          0u
149      },
150      { /* ch7 */
151          0ul,
152          0u,
153          0u
154      }
155  };
156  /* ==== プロトタイプ宣言 ==== */
157  void io_init_dma_reg(unsigned int ch, void *src, void *r_src, void *dst, void *r_dst, size_t size);
    
```

図 15 サンプルプログラムリスト"dmac.c" (3)

```

158  /*"FUNC COMMENT"*****
159  * Outline      : DMA 初期化
160  *-----
161  * Include      : #include <iodef.h>
162  *-----
163  * Declaration  : void io_init_dma_src(void);
164  *-----
165  * Function     : チャンネル 1~4 を初期化します。
166  *-----
167  * Argument    : void
168  *-----
169  * Return Value: void
170  *-----
171  * Notice      :
172  *"FUNC COMMENT END"*****/
173  void io_init_dma_src(void)
174  {
175      static unsigned int src_clr = 0x00000000ul;
176
177      /* ====スタンバイコントロールレジスタ 2(STBCR2)の設定==== */
178      CPG.STBCR2.BIT.MSTP8 = 0u;          /* DMAC のモジュールストップ解除*/
179
180      /* ==== DMAC の初期化/転送許可設定 ==== */
181      /*---- バッファクリア ---- */
182      io_init_dma_reg( 4,                /* チャンネル */
183                      &src_clr,         /* ソースアドレス */
184                      &src_clr,         /* リロードソースアドレス */
185                      get_buf_adr_proc(BUF_CLR), /* デスティネーションアドレス */
186                      get_buf_adr_proc(BUF_CLR), /* リロードデスティネーションアドレス */
187                      src_out_buf_size()); /* バイト数 */
188      /* ---- SRC 入力 ---- */
189      io_init_dma_reg( 3,                /* チャンネル */
190                      rom_ding22_05_bin, /* ソースアドレス */
191                      rom_ding22_05_bin, /* リロードソースアドレス */
192                      (void *)&SRC.SRCID, /* デスティネーションアドレス */
193                      (void *)&SRC.SRCID, /* リロードデスティネーションアドレス */
194                      size_ding22_05_bin); /* バイト数 */
195      /* ---- SRC 出力 ---- */
196      io_init_dma_reg( 2,                /* チャンネル */
197                      (void *)&SRC.SRCOD, /* ソースアドレス */
198                      (void *)&SRC.SRCOD, /* リロードソースアドレス */
199                      get_buf_adr_proc(BUF_SRC), /* デスティネーションアドレス */
200                      get_buf_adr_proc(BUF_SRC), /* リロードデスティネーションアドレス */
201                      src_out_buf_size()); /* バイト数 */
202      /* ---- SSI 送信 ---- */
203      io_init_dma_reg( 1,                /* チャンネル */
204                      get_buf_adr_proc(BUF_SSI), /* ソースアドレス */
205                      get_buf_adr_proc(BUF_SSI), /* リロードソースアドレス */
206                      (void *)&SSI0.SSITDR, /* デスティネーションアドレス */
207                      (void *)&SSI0.SSITDR, /* リロードデスティネーションアドレス */
208                      src_out_buf_size()); /* バイト数 */
209
210      /* ----DMA オペレーションレジスタの設定---- */
211      DMAC.DMAOR.WORD &= 0xffff9u;      /* AE,NMIF ビットクリア */
212
213      if(DMAC.DMAOR.BIT.DME == 0u){      /* 全チャンネル DMA 転送を許可 */
214          DMAC.DMAOR.BIT.DME = 1u;
215      }
216  }
    
```

図 16 サンプルプログラムリスト"dmac.c" (4)

```

217 /*"FUNC COMMENT"*****
218 * Outline      : DMA 初期設定
219 *-----
220 * Include      : #include <iodef.h>
221 *              : #include "dmac.h"
222 *-----
223 * Declaration  : void io_init_dma_reg(unsigned int ch,void *src, void *r_src,
224 *              :              void *dst, void *r_dst, size_t size);
225 *-----
226 * Function     : ソースアドレス src からディスティネーションアドレス dst へ
227 *              : size バイト分 DMAC による転送を行います。
228 *-----
229 * Argument     : unsigned int : チャンネル
230 *              : void *src    : ソースアドレス
231 *              : void *r_src  : リロードソースアドレス
232 *              : void *dst    : ディスティネーションアドレス
233 *              : void *r_dst  : リロードディスティネーションアドレス
234 *              : size_t size  : 転送サイズ(バイト)
235 *-----
236 * Return Value: void
237 *-----
238 * Notice       : 転送サイズとソース/ディスティネーションアドレスの
239 *              : アライメントが一致しない場合の動作は保証しません。
240 *              : 割り込み使用時は、割り込みルーチンを登録する必要があります
241 *"FUNC COMMENT END"*****/
242 void io_init_dma_reg(unsigned int ch,void *src, void *r_src,void *dst, void *r_dst, size_t size)
243 {
244     unsigned long ts = DmaReg[ch].chcr & 0x00000018ul;
245     unsigned long ie = DmaReg[ch].chcr & 0x00000004ul;
246     unsigned long de = DmaReg[ch].chcr & 0x00000001ul;
247     unsigned long rs = DmaReg[ch].chcr & 0x00000f00ul;
248
249     /* ---- DMA チャンネルコントロールレジスタの設定 ---- */
250     *DMA[ch].CHCR = 0ul; /* DMA 禁止 */
251     /* ---- DMA ソースアドレスレジスタの設定 ---- */
252     *DMA[ch].SAR = (unsigned long)src;
253     /* ---- DMA リロードソースアドレスレジスタの設定 ---- */
254     *DMA[ch].RSAR = (unsigned long)r_src;
255     /* ---- DMA デスティネーションアドレスレジスタの設定 ---- */
256     *DMA[ch].DAR = (unsigned long)dst;
257     /* ---- DMA リロードディスティネーションアドレスレジスタの設定 ---- */
258     *DMA[ch].RDAR = (unsigned long)r_dst;
259     /* ---- DMA トランスファカウントレジスタの設定 ---- */
260     /* ---- DMA リロードトランスファカウントレジスタの設定 ---- */
261     switch(ts){
262     case DMA_SIZE_BYTE:
263         *DMA[ch].DMATCR = size; /* 転送回数(1/1)を設定 */
264         *DMA[ch].RDMATCR = size;
265         break;
266     case DMA_SIZE_WORD:
267         *DMA[ch].DMATCR = size >> 1u; /* 転送回数(1/2)を設定 */
268         *DMA[ch].RDMATCR = size >> 1u;
269         break;
270     case DMA_SIZE_LONG:
271         *DMA[ch].DMATCR = size >> 2u; /* 転送回数(1/4)を設定 */
272         *DMA[ch].RDMATCR = size >> 2u;
273         break;
274     case DMA_SIZE_LONGx4:
275         *DMA[ch].DMATCR = size >> 4u; /* 転送回数(1/16)を設定 */
276         *DMA[ch].RDMATCR = size >> 4u;
277         break;
278     default:
279         break;
280     }

```

図 17 サンプルプログラムリスト"dmac.c" (5)

```

281  /* ---- DMA チャンネルコントロールレジスタ設定 ---- */
282  *DMA[ch].CHCR = DmaReg[ch].chcr & 0xffffffff;
283  /*
284     bit31      : TC :0----- 1 回転送
285     bit30      : reserve 0
286     bit29      : RLDSAR : x----- SAR リロード機能 有効/無効
287     bit28      : RLDDAR : x----- DAR リロード機能 有効/無効
288     bit27-24   : reserve 0
289     bit23      : DO over run0 : 0----- 未使用
290     bit22      : TL TEND low active : 0----- 未使用
291     bit21      : reserve 0
292     bit20      : TEMASK : x----- DMA 転送停止/DMA 転送継続
293     bit19      : HE :0----- 未使用
294     bit18      : HIE :0----- 未使用
295     bit17      : AM :0----- 未使用
296     bit16      : AL :0----- 未使用
297     bit15-14   : DM1:x DM0:x----- デスティネーションアドレス 固定/増加/減少
298     bit13-12   : SM1:x SM0:x----- ソースアドレス 固定/増加/減少
299     bit11-8    : RS : B'xxxx----- リソースセクタ
300     bit7       : DL : DREQ level : 0 ----- 未使用
301     bit6       : DS : DREQ select :0 Low level 未使用
302     bit5       : TB : cycle :0----- サイクルスチールモード
303     bit4-3     : TS : transfer size:B'10--- ロングワード単位転送
304     bit2       : IE : interrupt enable:1--- 割り込み許可
305     bit1       : TE : transfer end-----
306     bit0       : DE : DMA enable bit:0----- DMA 転送禁止
307  */
308  /* ----DMA 拡張リソースセクタの設定---- */
309  if(rs == DMA_ENH_RES){
310     switch(ch){
311     case 0:
312         DMAC.DMARS0.BYTE.CH0 = (unsigned char)DmaReg[ch].dmars;
313         break;
314     case 1:
315         DMAC.DMARS0.BYTE.CH1 = (unsigned char)DmaReg[ch].dmars;
316         break;
317     case 2:
318         DMAC.DMARS1.BYTE.CH2 = (unsigned char)DmaReg[ch].dmars;
319         break;
320     case 3:
321         DMAC.DMARS1.BYTE.CH3 = (unsigned char)DmaReg[ch].dmars;
322         break;
323     case 4:
324         DMAC.DMARS2.BYTE.CH4 = (unsigned char)DmaReg[ch].dmars;
325         break;
326     case 5:
327         DMAC.DMARS2.BYTE.CH5 = (unsigned char)DmaReg[ch].dmars;
328         break;
329     case 6:
330         DMAC.DMARS3.BYTE.CH6 = (unsigned char)DmaReg[ch].dmars;
331         break;
332     case 7:
333         DMAC.DMARS3.BYTE.CH7 = (unsigned char)DmaReg[ch].dmars;
334         break;
335     default:
336         break;
337     }
338  }

```

図 18 サンプルプログラムリスト"dmac.c" (6)

```

339  /* ----割り込み優先レベル設定---- */
340  if(ie == DMA_INT_ENABLE){
341      switch(ch){
342          case 0:
343              INTC.IPR06.BIT._DMAC0 = (unsigned short)DmaReg[ch].ipr;
344              break;
345          case 1:
346              INTC.IPR06.BIT._DMAC1 = (unsigned short)DmaReg[ch].ipr;
347              break;
348          case 2:
349              INTC.IPR06.BIT._DMAC2 = (unsigned short)DmaReg[ch].ipr;
350              break;
351          case 3:
352              INTC.IPR06.BIT._DMAC3 = (unsigned short)DmaReg[ch].ipr;
353              break;
354          case 4:
355              INTC.IPR07.BIT._DMAC4 = (unsigned short)DmaReg[ch].ipr;
356              break;
357          case 5:
358              INTC.IPR07.BIT._DMAC5 = (unsigned short)DmaReg[ch].ipr;
359              break;
360          case 6:
361              INTC.IPR07.BIT._DMAC6 = (unsigned short)DmaReg[ch].ipr;
362              break;
363          case 7:
364              INTC.IPR07.BIT._DMAC7 = (unsigned short)DmaReg[ch].ipr;
365              break;
366          default:
367              break;
368      }
369  }
370
371  /* ---- DMA の実行 ---- */
372  *DMA[ch].CHCR |= de;          /* DMA 転送許可 */
373
374  }
    
```

図 19 サンプルプログラムリスト"dmac.c" (7)



```

375  /*"FUNC COMMENT"*****
376  * Outline      : DMA 転送終了割り込み[SSI 送信]
377  *-----
378  * Include      : #include "iodefine.h"
379  *-----
380  * Declaration  : void io_int_dmal(void);
381  *-----
382  * Function     : リロードレジスタを更新し、割り込み処理を行います。
383  *-----
384  * Argument     : void
385  *-----
386  * Return Value: void
387  *-----
388  * Notice      :
389  *"FUNC COMMENT END"*****/
390 void io_int_dmal(void)
391 {
392     volatile unsigned long dummy;
393     /* ---- DMA リロードソースアドレスレジスタの設定 ---- */
394     DMAC.RSAR1.LONG = (unsigned long)get_buf_adr_proc(BUF_SSI);
395
396     (*trans_cmpplt_hdr[BUF_SSI])(); /* 割り込み処理 */
397
398     DMAC.CHCR1.BIT.TE = 0ul;
399     dummy = DMAC.CHCR1.BIT.TE;
400 }
401 /*"FUNC COMMENT"*****
402 * Outline      : DMA 転送終了割り込み[SRC 出力]
403 *-----
404 * Include      : #include <iodefine.h>
405 *-----
406 * Declaration  : void io_int_dma2(void);
407 *-----
408 * Function     : リロードレジスタを更新し、割り込み処理を行います。
409 *-----
410 * Argument     : void
411 *-----
412 * Return Value: void
413 *-----
414 * Notice      :
415 *"FUNC COMMENT END"*****/
416 void io_int_dma2(void)
417 {
418     volatile unsigned long dummy;
419     /* ---- 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 ---- */
420     DMAC.RDAR2.LONG = (unsigned long)get_buf_adr_proc(BUF_SRC);
421
422     (*trans_cmpplt_hdr[BUF_SRC])(); /* 割り込み処理 */
423
424     DMAC.CHCR2.BIT.TE = 0ul;
425     dummy = DMAC.CHCR2.BIT.TE;
426 }
    
```

図 20 サンプルプログラムリスト"dmac.c" (8)

```

427  /*"FUNC COMMENT"*****
428  * Outline      : DMA 転送終了割り込み[SRC 入力]
429  *-----
430  * Include      : #include <iodef.h>
431  *-----
432  * Declaration  : void io_int_dma3(void);
433  *-----
434  * Function     : DMA 転送を禁止し, SRC フラッシュ処理を実行します。
435  *-----
436  * Argument     : void
437  *-----
438  * Return Value: void
439  *-----
440  * Notice      :
441  *"FUNC COMMENT END"*****/
442  void io_int_dma3(void)
443  {
444      volatile unsigned long dummy;
445
446      DMAC.CHCR3.BIT.DE = 0ul;      /* DMA ch3(割り込み要因:SRC 入力)を停止 */
447
448      io_src_flash_start();        /* フラッシュ処理実行 */
449
450      DMAC.CHCR3.BIT.TE = 0ul;
451      dummy = DMAC.CHCR3.BIT.TE;
452  }
453  /*"FUNC COMMENT"*****
454  * Outline      : DMA 転送終了割り込み[バッファクリア]
455  *-----
456  * Include      : #include <iodef.h>
457  *-----
458  * Declaration  : void io_int_dma4(void);
459  *-----
460  * Function     : リロードレジスタを更新し, 割り込み処理を行います。
461  *-----
462  * Argument     : void
463  *-----
464  * Return Value: void
465  *-----
466  * Notice      :
467  *"FUNC COMMENT END"*****/
468  void io_int_dma4(void)
469  {
470      volatile unsigned long dummy;
471      /* ---- 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 ---- */
472      DMAC.RDAR4.LONG = (unsigned long)get_buf_adr_proc(BUF_CLR);
473
474      (*trans_cmpplt_hdr[BUF_CLR])(); /* 割り込み処理 */
475
476      DMAC.CHCR4.BIT.TE = 0ul;
477      dummy = DMAC.CHCR4.BIT.TE;
478  }
479
480  /* End of File */
    
```

図 21 サンプルプログラムリスト"dmac.c" (9)

```

1  /*"FILE COMMENT"*****
2  *
3  *      System Name : SH7263 Sample Program
4  *      File Name   : main.c
5  *      Contents    : SRC サンプルプログラム
6  *      Version     : 1.00.00
7  *      Model       : R0K572630D001BR
8  *      CPU         : SH7263
9  *      Compiler    : SHC9.1.1.0
10 *      note        :
11 *
12 *          <注意事項>
13 *          本サンプルプログラムはすべて参考資料であり、
14 *          その動作を保証するものではありません。
15 *          本サンプルプログラムはお客様のソフトウェア開発時の
16 *          技術参考資料としてご利用ください。
17 *
18 *      The information described here may contain technical inaccuracies or
19 *      typographical errors. Renesas Technology Corporation and Renesas Solutions
20 *      assume no responsibility for any damage, liability, or other loss rising
21 *      from these inaccuracies or errors.
22 *
23 *      Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
24 *      AND Renesas Solutions Corp. All Rights Reserved
25 *
26 *      history      : 2008.05.12 ver.1.00.00
27 *"FILE COMMENT END"*****/
28 #include <string.h>
29 #include "iodefine.h"      /* iodefine.h は、HEW で自動生成されるファイルです。 */
30 #include "src.h"
31
32 /* ==== プロトタイプ宣言 ==== */
33 void io_init_src(unsigned int in_sampling, unsigned int out_sampling);
34

```

図 22 サンプルプログラムリスト"src.c" (1)

```

35  /*"FUNC COMMENT"*****
36  * Outline      : SRC レジスタ初期化
37  *-----
38  * Include      : #include <iodef.h>
39  *-----
40  * Declaration  : void io_init_src(unsigned int in_sampling,
41  *                :                unsigned int out_sampling);
42  *-----
43  * Function     : SRC レジスタ初期設定後, SRC モジュール動作を許可します。
44  *-----
45  * Argument     : unsigned int in_sampling  : 入力サンプリングレート
46  *                : unsigned int out_sampling : 出力サンプリングレート
47  *-----
48  * Return Value: void
49  *-----
50  * Notice       :
51  *"FUNC COMMENT END"*****/
52  void io_init_src(unsigned int in_sampling,unsigned int out_sampling)
53  {
54      /* ==== SRC クロック供給 ==== */
55      CPG.STBCR6.BIT.MSTP62 = 0u;
56
57      /* ==== SRC 制御レジスタ ==== */
58      SRC.SRCCTRL.WORD = 0x0500u | (in_sampling << 4u) | out_sampling;
59      /*
60          bit15-13 : reserve : 0
61          bit12   : SRCEN : 0 ----- SRC モジュール動作禁止
62          bit11   : reserve : 0
63          bit10   : EEN : 1 ----- 出力 FIFO オーバーライト割り込みを許可
64          bit9    : FL : 0 -----
65          bit8    : CL : 1 ----- 内部ワークメモリクリア
66          bit7-4  : IFS : B'0110 ----- 入力サンプリングレート 24.0kHz
67          bit3-1  : reserve : 0
68          bit0    : OFS : 0 ----- 出力サンプリングレート 44.1kHz
69      */
70      /* ==== SRC 入力データ制御レジスタ ==== */
71      SRC.SRCIDCTRL.WORD = 0x0103u;
72      /*
73          bit15-10 : reserve : 0
74          bit9     : IED : 0 ----- 入力データ:ビッグエンディアン
75          bit8     : IEN : 1 ----- 入力データエンプティ割り込み許可
76          bit7-2  : reserve : 0
77          bit1-0  : IFTRG : B'11 ----- 入力 FIFO データ数トリガ:12
78      */
79      /* ==== SRC 出力データ制御レジスタ ==== */
80      SRC.SRCODCTRL.WORD = 0x0100u;
81      /*
82          bit15-11 : reserve : 0
83          bit10    : OCH : 0 ----- チャンネルを入れ替えない
84          bit9     : OED : 0 ----- 出力データ:ビッグエンディアン
85          bit8     : OEN : 1 ----- 出力 FIFO データフル割り込み許可
86          bit7-2  : reserve : 0
87          bit1-0  : OFTRG : B'00 ----- 出力 FIFO データ数トリガ:1
88      */
89
90      /* ==== Setting of interrupt priority level ==== */
91      INTC.IPR17.BIT._SRC = 0x01u;
92
93      SRC.SRCCTRL.BIT.SRCEN = 1u; /* SRC モジュール動作許可 */
94  }

```

図 23 サンプルプログラムリスト"src.c" (2)

```

95  /*"FUNC COMMENT"*****
96  * Outline      : 出力 FIFO オーバフロー割り込み処理
97  *-----
98  * Include      :
99  *-----
100 * Declaration : void io_int_src_out_ovr(void);
101 *-----
102 * Function     :
103 *-----
104 * Argument     : void
105 *-----
106 * Return Value: void
107 *-----
108 * Notice       :
109 *"FUNC COMMENT END"*****/
110 void io_int_src_out_ovr(void)
111 {
112 }
113 /*"FUNC COMMENT"*****
114 * Outline      : 出力 FIFO フル割り込み処理
115 *-----
116 * Include      : #include <iodef.h>
117 *-----
118 * Declaration : void io_int_src_out_full(void);
119 *-----
120 * Function     : 出力 FIFO フル割り込み要求ビットをクリアします。
121 *-----
122 * Argument     : void
123 *-----
124 * Return Value: void
125 *-----
126 * Notice       :
127 *"FUNC COMMENT END"*****/
128 void io_int_src_out_full(void)
129 {
130     volatile unsigned long dummy;
131
132     SRC.SRCSTAT.BIT.OINT = 0u;
133     dummy = SRC.SRCSTAT.BIT.OINT;
134 }
135 /*"FUNC COMMENT"*****
136 * Outline      : 入力 FIFO エンプティ割り込み処理
137 *-----
138 * Include      : #include <iodef.h>
139 *-----
140 * Declaration : void io_int_src_in_emp(void);
141 *-----
142 * Function     : 入力 FIFO エンプティ割り込み要求ビットをクリアします。
143 *-----
144 * Argument     : void
145 *-----
146 * Return Value: void
147 *-----
148 * Notice       :
149 *"FUNC COMMENT END"*****/
150 void io_int_src_in_emp(void)
151 {
152     volatile unsigned long dummy;
153
154     SRC.SRCSTAT.BIT.IINT = 0u;
155     dummy = SRC.SRCSTAT.BIT.IINT;
156 }

```

図 24 サンプルプログラムリスト"src.c" (3)

```

157  /*"FUNC COMMENT"*****
158  * Outline      : SRC フラッシュ実行
159  *-----
160  * Include      : #include <iodef.h>
161  *-----
162  * Declaration  : void io_src_flash_start(void);
163  *-----
164  * Function     : 入力データエンブティ割り込み要求禁止後に,
165  *               : フラッシュ処理を実行します。
166  *-----
167  * Argument     : void
168  *-----
169  * Return Value: void
170  *-----
171  * Notice      :
172  *"FUNC COMMENT END"*****/
173  void io_src_flash_start(void)
174  {
175      SRC.SRCIDCTRL.BIT.IEN = 0u;    /* 入力データエンブティ割り込み禁止 */
176      SRC.SRCCTRL.BIT.FL = 1u;     /* フラッシュ処理実行 */
177  }
178  /*"FUNC COMMENT"*****
179  * Outline      : 出力 FIFO オーバフロー割り込み要求ビットのクリア
180  *-----
181  * Include      : #include <iodef.h>
182  *-----
183  * Declaration  : void io_src_ovf_clr(void);
184  *-----
185  * Function     : 出力 FIFO オーバフロー割り込み要求ビットをクリアします。
186  *-----
187  * Argument     : void
188  *-----
189  * Return Value: void
190  *-----
191  * Notice      :
192  *"FUNC COMMENT END"*****/
193  void io_src_ovf_clr(void)
194  {
195      volatile unsigned long dummy;
196
197      SRC.SRCSTAT.BIT.OVF = 0u;
198      dummy = SRC.SRCSTAT.BIT.OVF;
199  }
200
201  /* End of File */
    
```

図 25 サンプルプログラムリスト"src.c" (4)

```

1  /*"FILE COMMENT"*****
2  *
3  *   System Name : SH7263 Sample Program
4  *   File Name   : ssi.c
5  *   Contents    : SSI データ転送
6  *   Version     : 1.00.00
7  *   Model       : R0K572630D001BR
8  *   CPU         : SH7263
9  *   Compiler    : SHC9.1.1.0
10 *   note        : SSI0 を使用したデータ転送のサンプルです。
11 *
12 *
13 *           <注意事項>
14 *           本サンプルプログラムはすべて参考資料であり、
15 *           その動作を保証するものではありません。
16 *           本サンプルプログラムはお客様のソフトウェア開発時の
17 *           技術参考資料としてご利用ください。
18 *
19 *   The information described here may contain technical inaccuracies or
20 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
21 *   assume no responsibility for any damage, liability, or other loss rising
22 *   from these inaccuracies or errors.
23 *
24 *   Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
25 *   AND Renesas Solutions Corp. All Rights Reserved
26 *
27 *   history      : 2008.05.09 ver.1.00.00
28 *"FILE COMMENT END"*****/
29 #include <string.h>
30 #include "iodefine.h"      /* iodefine.h は、HEW で自動生成されるファイルです。 */
31 #include "src.h"
32
33 /* ==== プロトタイプ宣言 ==== */
34 void io_init_ssi0(void);
35 void io_ssi0_set_mute(void);
36 void io_int_ssi0(void);
37

```

図 26 サンプルプログラムリスト"ssi.c" (1)

```

38  /*"FUNC COMMENT"*****
39  * Outline      : SSI モジュール初期化
40  *-----
41  * Include      : #include <iodef.h>
42  *-----
43  * Declaration  : void io_init_ssi0(void);
44  *-----
45  * Function     : マスタトランスミッタモードでデータ転送します。
46  *               : サンプリング周期は 44.1kHz です。
47  *-----
48  * Argument     : void
49  *-----
50  * Return Value: void
51  *-----
52  * Notice      :
53  *"FUNC COMMENT END"*****/
54  void io_init_ssi0(void)
55  {
56  /* ==== SSI モジュール クロック供給 ==== */
57  CPG.STBCR6.BIT.MSTP67 = 0u; /* SSI0 */
58
59  /* ==== オーバサンプルクロック供給源選択 ==== */
60  PORT.PFCRH4.BIT.PF30MD = 1u; /* select AUDIO_CLK */
61  PORT.SCSR.BIT.SSI0CKS = 0u ; /* AUDIO_X1 入力 16.9344MHz */
62
63  /* ----SSI module pin enabled ---- */
64  PORT.PFCRH1.BIT.PF18MD = 1u; /* SSISCK0 */
65  PORT.PFCRH1.BIT.PF19MD = 1u; /* SSIWS0 */
66  PORT.PFCRH2.BIT.PF20MD = 1u; /* SSIDATA0 */
67
68  /* ==== コントロールレジスタ(SSICR) ==== */
69  SSI0.SSICR.LONG = 0x1c0bd553ul;
70  /*
71      bit31-29 : reserve 0
72      bit28    : DMEN : 1----- DMA 要求を許可
73      bit27    : UIEN : 1----- アンダフロー割り込みを許可
74      bit26    : OIEN : 1----- オーバフロー割り込みを許可
75      bit25    : IIEN : 0----- アイドルモード割り込みを禁止
76      bit24    : DIEN : 0----- データ割り込みを禁止
77      bit23-22 : CHNL : 0----- 各システムワードは 1 チャネルで構成されています
78      bit21-19 : DWL  : B'001----- データワード長 16 ビット
79      bit18-16 : SWL  : B'011----- システムワード長 32 ビット
80      bit15    : SCKD : 1----- シリアルビットクロック出力, マスタモード
81      bit14    : SWSD : 1----- シリアルワード選択出力, マスタモード
82      bit13    : SCKP : 0----- SSIWS と SSIDATA は SSISCK の立ち上がりエッジで変化
83      bit12    : SWSP : 1----- SSIWS は第一チャネルは"H"レベル, 第二チャネルは"L"レベル
84      bit11    : SPDP : 0----- パディングビットは"L"レベル
85      bit10    : SDTA : 1----- パディングビット, シリアルデータの順に送受信
86      bit9     : PDTA : 0----- 使用していません
87      bit8     : DEL  : 1----- SSIWS と SSIDATA 間の遅延なし
88      bit7     : reserve 0
89      bit6-4   : CKDV : B'101----- オーバサンプルクロック周波数(16.9344MHz) / 6[44.1kHz]
90      bit3     : MUEN : 0----- SSI モジュールはミュート状態ではない
91      bit2     : reserve 0
92      bit1    : TRMD : 1----- SSI モジュールは送信モード
93      bit0    : EN   : 1----- SSI モジュール動作を許可
94  */
95  INTC.IPR14.BIT._SSI0 = 0x01u; /* 割り込み優先レベル設定 */
96  }

```

図 27 サンプルプログラムリスト"ssi.c" (2)



```

97  /*"FUNC COMMENT"*****
98  * Outline      : SSI 割り込み
99  *-----
100 * Include      : #include <iodef.h>
101 *-----
102 * Declaration  : void io_int_ssi0(void);
103 *-----
104 * Function     : アンダフローエラーなどの割り込み要求をクリアします。
105 *-----
106 * Argument    : void
107 *-----
108 * Return Value: void
109 *-----
110 * Notice      :
111 *"FUNC COMMENT END"*****/
112 void io_int_ssi0(void)
113 {
114     /* アンダフローエラー */
115     if(SSI0.SSISR.BIT.UIRQ == 1ul){
116         SSI0.SSISR.BIT.UIRQ = 0ul;
117         while(1){
118             /* dead loop */
119         }
120     }
121     /* オーバフローエラー */
122     if(SSI0.SSISR.BIT.OIRQ == 1ul){
123         SSI0.SSISR.BIT.OIRQ = 0ul;
124         while(1){
125             /* dead loop */
126         }
127     }
128     /* アイドル */
129     if(SSI0.SSISR.BIT.IIRQ == 1ul){
130         SSI0.SSISR.BIT.IIRQ = 0ul;
131     }
132 }
    
```

図 28 サンプルプログラムリスト"ssi.c" (3)

```

133  /*"FUNC COMMENT"*****
134  * Outline      : SSI ミュート設定
135  *-----
136  * Include      : #include <iodefine.h>
137  *-----
138  * Declaration  : void io_ssi0_set_mute(void);
139  *-----
140  * Function     : ミュート状態に移行します。
141  *-----
142  * Argument     : void
143  *-----
144  * Return Value: void
145  *-----
146  * Notice      :
147  *"FUNC COMMENT END"*****/
148  void io_ssi0_set_mute(void)
149  {
150      if(SSI0.SSICR.BIT.EN == 1ul){
151          /* ---- disable SSI interrupt ---- */
152          SSI0.SSICR.BIT.UIEN = 0ul;
153
154          /* ---- disable dreq ---- */
155          SSI0.SSICR.BIT.DMEN =0ul;
156
157          while(SSI0.SSISR.BIT.DIRQ == 0ul){
158              /* ---- wait data req ---- */
159          }
160          SSI0.SSICR.BIT.MUEN = 1ul; /* mute start */
161      }
162  }
163  /* End of File */

```

図 29 サンプルプログラムリスト"ssi.c" (4)

```

1  /*"FILE COMMENT"*****
2  *
3  *      System Name : SH7263 Sample Program
4  *      File Name   : buf.c
5  *      Contents    : リングバッファ
6  *      Version     : 1.00.00
7  *      Model       : R0K572630D001BR
8  *      CPU         : SH7263
9  *      Compiler    : SHC9.1.1.0
10 *      note        :
11 *
12 *          <注意事項>
13 *          本サンプルプログラムはすべて参考資料であり、
14 *          その動作を保証するものではありません。
15 *          本サンプルプログラムはお客様のソフトウェア開発時の
16 *          技術参考資料としてご利用ください。
17 *
18 *      The information described here may contain technical inaccuracies or
19 *      typographical errors. Renesas Technology Corporation and Renesas Solutions
20 *      assume no responsibility for any damage, liability, or other loss rising
21 *      from these inaccuracies or errors.
22 *
23 *      Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
24 *      AND Renesas Solutions Corp. All Rights Reserved
25 *
26 *      history      : 2008.05.09 ver.1.00.00
27 *"FILE COMMENT END"*****/
28 #include <string.h>
29 #include "iodefine.h"
30 #include "src.h"
31
32 #pragma section SDRAM
33 /* ==== サンプリングレート変更後の格納バッファ ==== */
34 static unsigned long SrcOutBuffer[SRC_OUT_BUF_NUM][SRC_OUT_BUF_SIZE];
35
36 #pragma section
37 static int BufArea[3]={0,0,0};
38 static int FlgSrcStop = 0;
39
40 static void src_out_cnt_up(int* cnt,int max);
41 static void src_trans_cmplt_proc(void);
42 static void ssi_trans_cmplt_proc(void);
43 static void clr_trans_cmplt_proc(void);
44
45 unsigned long* get_buf_adr_proc(int module);
46 void (*trans_cmplt_hdr[])(void) = {
47     clr_trans_cmplt_proc,
48     src_trans_cmplt_proc,
49     ssi_trans_cmplt_proc,
50 };
51

```

図 30 サンプルプログラムリスト"buf.c" (1)

```

52  /*"FUNC COMMENT"*****
53  * Outline      : バッファクリア
54  *-----
55  * Include      : #include <string.h>
56  *-----
57  * Declaration  : void src_out_buf_clr(void);
58  *-----
59  * Function     : バッファ全領域ををクリアします。
60  *-----
61  * Argument     : void
62  *-----
63  * Return Value: void
64  *-----
65  * Notice      :
66  *"FUNC COMMENT END"*****/
67  void src_out_buf_clr(void)
68  {
69      memset(SrcOutBuffer,0x00,sizeof(SrcOutBuffer));    /* バッファクリア */
70  }
71  /*"FUNC COMMENT"*****
72  * Outline      : バッファアドレス取得
73  *-----
74  * Include      :
75  *-----
76  * Declaration  : unsigned long* get_buf_adr_proc(int module);
77  *-----
78  * Function     : バッファアドレスを返します。
79  *-----
80  * Argument     : int module      : BUF_SRC SRC 出力
81  *               :                : BUF_SSI SSI 送信
82  *               :                : BUF_CLR バッファクリア
83  *-----
84  * Return Value: adr : バッファアドレス
85  *-----
86  * Notice      :
87  *"FUNC COMMENT END"*****/
88  unsigned long* get_buf_adr_proc(int module)
89  {
90      unsigned long *adr;
91
92      adr = SrcOutBuffer[BufArea[module]];                /* アドレスの取得 */
93      src_out_cnt_up(&BufArea[module],SRC_OUT_BUF_NUM);  /* BufArea[module]のカウンタアップ */
94
95      return(adr);
96  }
    
```

図 31 サンプルプログラムリスト"buf.c" (2)

```

97  /*"FUNC COMMENT"*****
98  * Outline      : SRC 出力 割り込み処理
99  *-----
100 * Include      : #include <iodef.h>
101 *-----
102 * Declaration  : static void src_trans_cmplt_proc(void);
103 *-----
104 * Function     : 次バッファ内のデータをクリア中の場合, SRC 出力の DMA 転送
105 *              : を禁止し, 変数 FlgSrcStop に 1 をセットします。
106 *              : 最初のバッファに PCM データを格納後, SSI の DMA 転送を許可します。
107 *-----
108 * Argument    : void
109 *-----
110 * Return Value: void
111 *-----
112 * Notice      :
113 *"FUNC COMMENT END"*****/
114 static void src_trans_cmplt_proc(void)
115 {
116     if(BufArea[BUF_SRC] == BufArea[BUF_CLR]){ /* 次バッファ内のデータをクリア中の場合 */
117         DMAC.CHCR2.BIT.DE = 0ul; /* DMA ch2(割り込み要因:SRC 出力)を停止 */
118         FlgSrcStop = 1;
119     }
120     if(DMAC.CHCR1.BIT.DE == 0ul){ /* 初回のみ */
121         DMAC.CHCR1.BIT.DE = 1ul; /* 1 つ目のバッファに対して SRC 出力が完了後 */
122                                 /* SSI 送信の DMA を許可します */
123     }
124 }
125 }
126 /*"FUNC COMMENT"*****
127 * Outline      : SSI 送信 割り込み処理
128 *-----
129 * Include      : #include <iodef.h>
130 *-----
131 * Declaration  : static void ssi_trans_cmplt_proc(void);
132 *-----
133 * Function     : 全データの送信が完了すると
134 *              : SSI をミュート状態に移行します。
135 *-----
136 * Argument    : void
137 *-----
138 * Return Value: void
139 *-----
140 * Notice      :
141 *"FUNC COMMENT END"*****/
142 static void ssi_trans_cmplt_proc(void)
143 {
144     if(BufArea[BUF_SRC] == BufArea[BUF_SSI]){ /* 終了判定 */
145         io_ssi0_set_mute(); /* ミュート設定 */
146     }
147     if(DMAC.CHCR4.BIT.DE == 0ul){
148         DMAC.CHCR4.BIT.DE = 1ul; /* バッファクリアの DMA を許可します */
149     }
150 }
151 }

```

図 32 サンプルプログラムリスト"buf.c" (3)

```

152  /*"FUNC COMMENT"*****
153  * Outline      : バッファクリア 割り込み処理
154  *-----
155  * Include      : #include <iodef.h>
156  *-----
157  * Declaration  : static void clr_trans_cmplt_proc(void);
158  *-----
159  * Function     : 次バッファ内のデータを送信中の場合、バッファクリアの
160  *               : DMA 転送を禁止します。
161  *               : SRC 出力が停止している場合、SRC 出力を再開し、オーバーフロー
162  *               : 割り込み要求ビットをクリアします。
163  *-----
164  * Argument    : void
165  *-----
166  * Return Value: void
167  *-----
168  * Notice      :
169  *"FUNC COMMENT END"*****/
170  static void clr_trans_cmplt_proc(void)
171  {
172  if(BufArea[BUF_SSI] == BufArea[BUF_CLR]){ /* 次バッファ内のデータを送信中の場合 */
173     DMAC.CHCR4.BIT.DE = 0ul; /* DMA ch4(割り込み要因:ソフトウェア)を停止 */
174 }
175 if(FlgSrcStop == 1){ /* SRC 出力が停止している場合 */
176     FlgSrcStop = 0;
177     DMAC.CHCR2.BIT.DE = 1ul; /* DMA ch2(割り込み要因:SRC 出力)を開始 */
178     io_src_ovf_clr(); /* SRC 出力オーバーフロー割り込み要求フラグクリア */
179 }
180 }
181 /*"FUNC COMMENT"*****
182  * Outline      : カウントアップ
183  *-----
184  * Include      :
185  *-----
186  * Declaration  : static void src_out_cnt_up(int* cnt,int max);
187  *-----
188  * Function     : 0 から変数 max の間でカウントアップします。
189  *-----
190  * Argument    : int* cnt : カウント格納 RAM
191  *               : int max : 最大値
192  *-----
193  * Return Value: void
194  *-----
195  * Notice      :
196  *"FUNC COMMENT END"*****/
197  static void src_out_cnt_up(int* cnt,int max)
198  {
199  if(*cnt < max - 1){
200     *cnt += 1;
201 }
202 else{
203     *cnt = 0;
204 }
205 }
206

```

図 33 サンプルプログラムリスト"buf.c" (4)

```

207  /*"FUNC COMMENT"*****
208  * Outline      : SRC 出力側バッファのブロックサイズを返します。
209  *-----
210  * Include      :
211  *-----
212  * Declaration : size_t src_out_buf_size(void);
213  *-----
214  * Function     : SRC 出力側バッファのブロックサイズを返します。
215  *-----
216  * Argument    : void
217  *-----
218  * Return Value: size_t size : SRC 出力側バッファのブロックサイズ
219  *-----
220  * Notice      :
221  *"FUNC COMMENT END"*****/
222  size_t src_out_buf_size(void)
223  {
224      return(sizeof(SrcOutBuffer[0]));
225  }
226  /* End of File */
    
```

図 34 サンプルプログラムリスト"buf.c" (5)

```

1  /*****
2  /*
3  /* FILE      :intprg.c
4  /* DATE      :Wed, Nov 21, 2007
5  /* DESCRIPTION :Interrupt Program
6  /* CPU TYPE   :SH7263
7  /*
8  /* This file is generated by Renesas Project Generator (Ver.4.5).
9  /*
10 /*****
11
12
13
14 #include <machine.h>
15 #include "vect.h"
16 #pragma section IntPRG
17
18 ~省略~
19
20427 // 112 DMAC1 TEI1
20428 void INT_DMAL1_TEI1(void)
20429 {
20430     extern void io_int_dma1(void);
20431     io_int_dma1();
20432 }
20433
20434 ~省略~
20435
20443 // 116 DMAC2 TEI2
20444 void INT_DMAL2_TEI2(void)
20445 {
20446     extern void io_int_dma2(void);
20447     io_int_dma2();
20448 }
20449
20450 ~省略~
20451
20461 // 120 DMAC3 TEI3
20462 void INT_DMAL3_TEI3(void)
20463 {
20464     extern void io_int_dma3(void);
20465     io_int_dma3();
20466 }
20467
20468 ~省略~
20469
20478 // 124 DMAC4 TEI4
20479 void INT_DMAL4_TEI4(void)
20480 {
20481     extern void io_int_dma4(void);
20482     io_int_dma4();
20483 }
20484
20485 ~省略~
20486
20987 // 214 SSI0
20988 void INT_SSI0(void)
20989 {
20990     extern void io_int_ssi0(void);
20991     io_int_ssi0();
20992 }
20993
    
```

図 35 サンプルプログラムリスト"intprg.c" (1)



```

~省略~
1167 // 244 SRC OVF
1168 void INT_SRC_OVF(void)
1169 {
1170     extern void io_int_src_out_ovr(void);
1171     io_int_src_out_ovr();
1172 }
1173
1174 // 245 SRC IDFI
1175 void INT_SRC_IDFI(void)
1176 {
1177     extern void io_int_src_out_full(void);
1178     io_int_src_out_full();
1179 }
1180
1181 // 246 SRC IDEI
1182 void INT_SRC_IDEI(void)
1183 {
1184     extern void io_int_src_in_emp(void);
1185     io_int_src_in_emp();
1186 }
1187 // 247 IEB IEBI
1188 void INT_IEB_IEBI(void)
1189 {
1190     /* sleep(); */
1191 }
1192 // 248 Dummy
1193 void Dummy(void)
1194 {
1195     /* sleep(); */
1196 }
1197
1198 /* End of File */
    
```

図 36 サンプルプログラムリスト"intprg.c" (2)

#### 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 (RJJ09B0086)  
(最新版をルネサス テクノロジ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  
SH7263 グループ ハードウェアマニュアル (RJJ09B0284)  
(最新版をルネサス テクノロジ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 テクノロジホームページ

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

お問合せ先

<http://japan.renesas.com/inquiry>[csc@renesas.com](mailto:csc@renesas.com)

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2009.01.19	—	初版発行

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。

## 本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりますは、事前に弊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、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
5. 本資料に記載した情報は、正確を期すため慎重に制作したものです。万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等については弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
  - 1) 生命維持装置。
  - 2) 人体に埋め込み使用するもの。
  - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
  - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 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弊社営業窓口までご照会ください。

D039444