
SH7730 グループ

RJJ06B1065-0200

Rev.2.00

DMAC 内蔵周辺モジュールリクエストモード設定例

2010.09.06

要旨

本アプリケーションノートは、DMA 転送要求信号で SCIF の TXI (送信 FIFO データエンプティ)、RXI (受信 FIFO データフル) を使用した DMAC 内蔵周辺モジュールリクエストモードの設定方法について説明しています。

動作確認デバイス

SH7730

目次

1. はじめに.....	2
2. 応用例の説明.....	4
3. 参考プログラム例.....	24
4. 実行結果.....	53
5. 参考ドキュメント.....	54

1. はじめに

1.1 仕様

- DMAC0 で、SCIF0 の TXI (送信 FIFO データエンプティ) DMA 転送要求信号によって、SDRAM 上のデータを SCIF0.SCFTDR へ転送します。
- その際に、SCIF0 から SCIF1 へ調歩同期式でデータ送信します。
- SCIF1 でデータ受信後、DMAC1 で、RXI (受信 FIFO データフル) DMA 転送要求信号によって SCIF1.SCFRDR のデータを IL メモリに転送します。

【注】 仕様について

SDRAM IL メモリへのデータコピーのみが目的であれば、CPU、DMAC を通常使用することで行いますが、本応用例では、DMA 転送要求信号で SCIF の TXI (送信 FIFO データエンプティ)、RXI (受信 FIFO データフル) を使用した DMAC 内蔵周辺モジュールリクエストモードの設定方法について説明するため、あえて上記のことを行っています。

1.2 使用機能

- ダイレクトメモリアクセスコントローラ (DMAC0) のチャンネル 0
- ダイレクトメモリアクセスコントローラ (DMAC1) のチャンネル 1
- FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF0) のチャンネル 0
- FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF1) のチャンネル 1

1.3 適用条件

- 評価ボード: アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
外付けメモリ (エリア 0): NOR 型フラッシュメモリ 4M バイト
Spansion 製 S29AL032D70TFI04
(エリア 3): SDR-SDRAM 32 M バイト (16 M バイト × 2 個)
Samsung 製 K4S281632F-UC75
- マイコン: SH7730 (R8A77301)
- 動作周波数: CPU クロック: 266.66MHz
SuperHyway バスクロック: 133.33MHz
バスクロック: 66.66MHz
周辺クロック: 33.33MHz
- エリア 0 バス幅: 16 ビット固定 (MD3 端子 = Low レベル)
- クロック動作モード: モード 2 (MD0 端子 = Low レベル, MD1 端子 = High レベル)
- エンディアン: ビッグエンディアン (MD5 端子 = Low レベル)
- ツールチェーン: ルネサス テクノロジ製 SuperH RISC engine Standard Toolchain Ver.9.2.0.0
- コンパイルオプション: High-performance Embedded Workshop でのデフォルト設定
(-cpu=sh4a -include="\$ (PROJDIR)¥inc" -object="\$ (CONFIGDIR)¥\$ (FILELEAF).obj"
-debug -optimize=0 -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

本資料の参考プログラムは、「SH7730 グループ アプリケーションノート SH7730 初期設定例 (RJJ06B0864)」の設定条件で動作確認しています。

DMAC の使用方法については、「SH7730 グループ アプリケーションノート DMAC メモリ間転送例 (RJJ06B0936)」に記載しています。

SCIF 調歩同期式 送受信設定例については、「SH7730 グループ アプリケーションノート SCIF 調歩同期式 送受信設定例 (RJJ06B0954)」に記載しています。

これらも合わせてご参照ください。

2. 応用例の説明

本応用例では、DMAC の内蔵周辺モジュールリクエストの使用方法を説明するため、DMAC の内蔵周辺モジュールリクエストと SCIF の機能を利用し、SDRAM 上の 128 バイト分のデータを IL メモリに転送しています。

2.1 SH7730 の内蔵周辺モジュールリクエストモード概要

内蔵周辺モジュールリクエストモードでは、内蔵周辺モジュールからの DMA 転送要求信号によって転送が実行されます。

DMA 転送要求信号は、DMARS0/1/2 にて設定する SCIF0/1/2/3/4/5、IrDA0/1、SIOF、SIM からの送信データエンプティ転送要求と受信データフル転送要求、ADC、CMT0/1/2/3/4 からの転送要求があります。

内蔵周辺モジュールリクエストモード選択時に、DMA 転送許可状態 (DE = 1、DME = 1、TE = 0、AE = 0、NMIF = 0) であると、転送要求信号によって転送が実行されます。

転送要求を SCIF の送信データエンプティ転送要求に設定した場合、転送先を当該 SCIF のトランスミットデータレジスタとする必要があります。同様に転送要求を SCIF の受信データフル転送要求に設定した場合、転送元を当該 SCIF のレシーブデータレジスタとする必要があります。

これらは SIOF、SIM も同様です。

また、内蔵周辺モジュールによっては、受信 FIFO トリガ数を転送要求に設定できます。受信 FIFO トリガ条件が満たされない場合は、受信 FIFO にデータが残ることがあります。そのため、個別に CPU などから受信 FIFO の残りデータを読み出す必要があります。もしくは、受信 FIFO トリガ数を 1 に設定して受信 FIFO にデータが残らないように対応する必要があります。

本応用例では、DMAC0 に SCIF0 送信データエンプティ転送要求、DMAC1 に SCIF1 受信データフル転送要求を設定します。

表 1 内蔵周辺モジュールリクエストモードの選択

DMARS		DMA 転送 要求元	DMA 転送要求信号	転送元	転送先	バス モード
MID	RID					
001000	01	SCIF0 送信部	TXI (送信 FIFO データエンプティ)	任意	SCFTDR0	サイクル スチール
001001	10	SCIF1 受信部	RXI (受信 FIFO データフル)	SCFRDR1	任意	サイクル スチール

【注】 上記以外の内蔵周辺モジュールリクエストモードの詳細については、「SH7730 グループ ハードウェアマニュアル (RJJ09B0339)」の「ダイレクトメモリアクセスコントローラ (DMAC) 表 12.7 RS[3:0] ビットによる内蔵周辺モジュールリクエストモードの選択」の章を参照してください。

2.2 本応用例のデータ転送フロー

本応用例での SDRAM 上のデータが IL メモリに転送されるフローを以下に記載します。

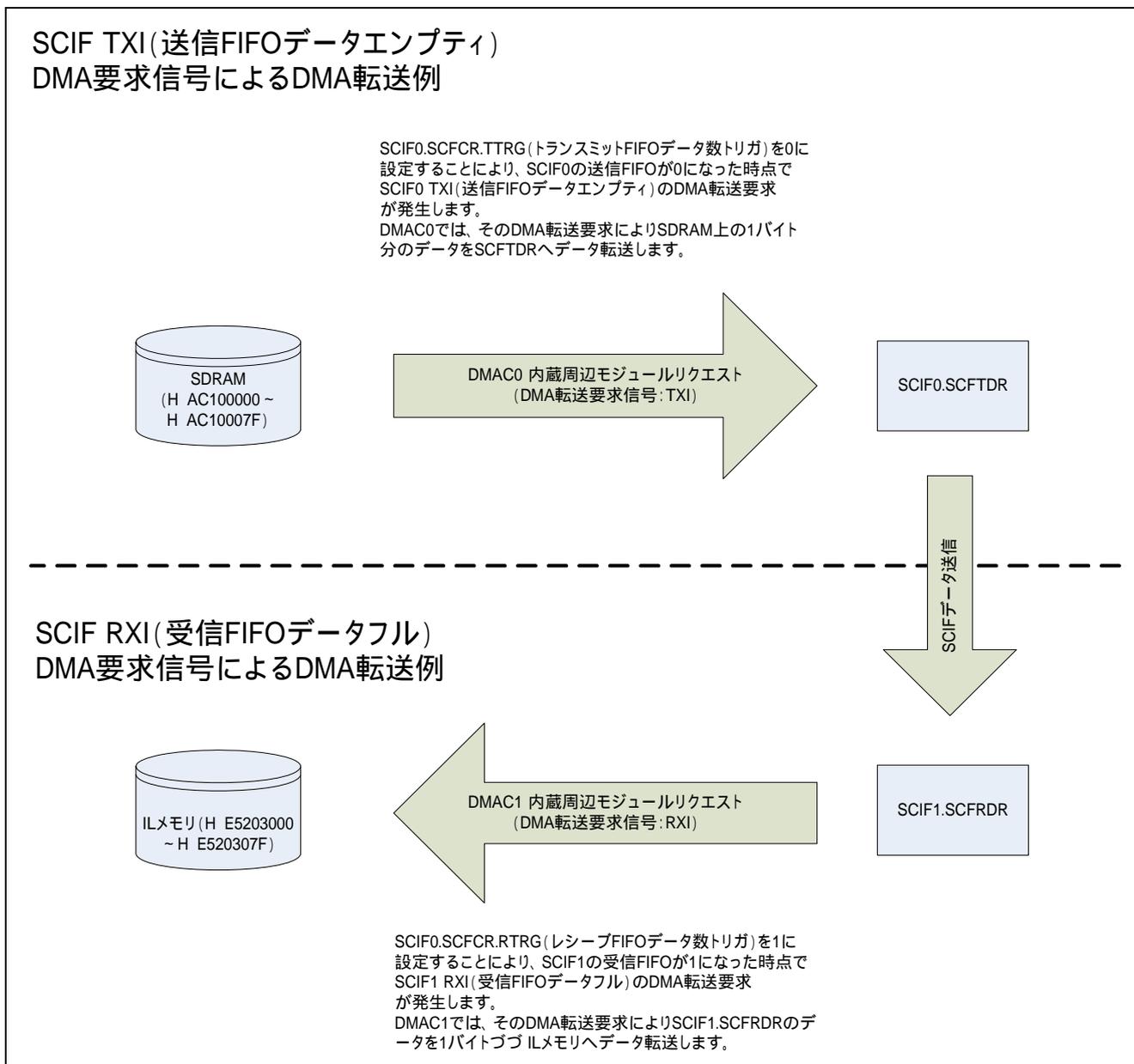


図 1 本応用例のデータ転送フロー

2.3 本応用例の動作環境

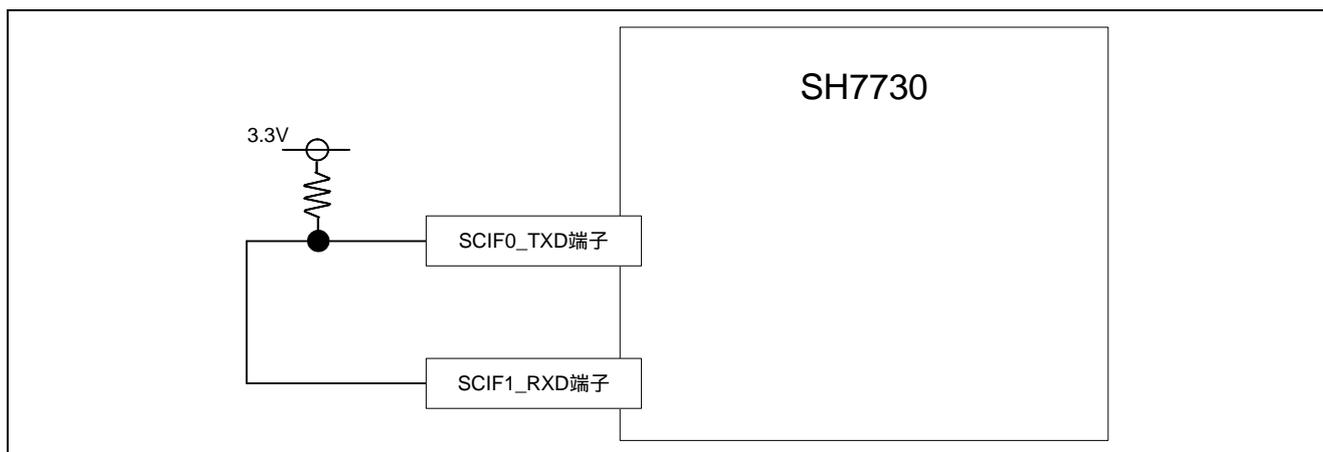


図 2 本応用例の動作環境

2.4 参考プログラムの処理フロー

2.4.1 参考プログラムメイン処理

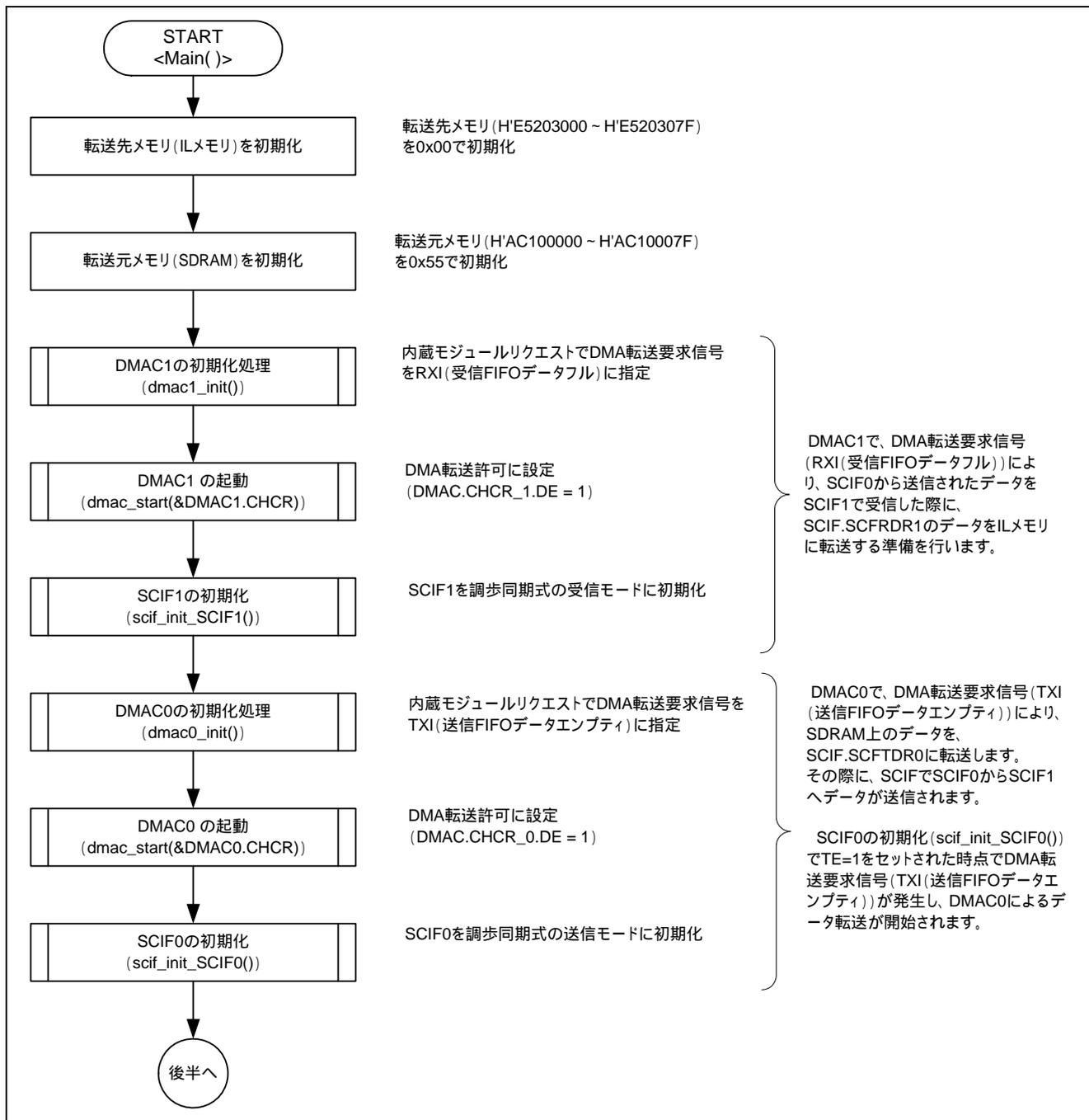


図3 参考プログラムメイン処理フロー (前半)

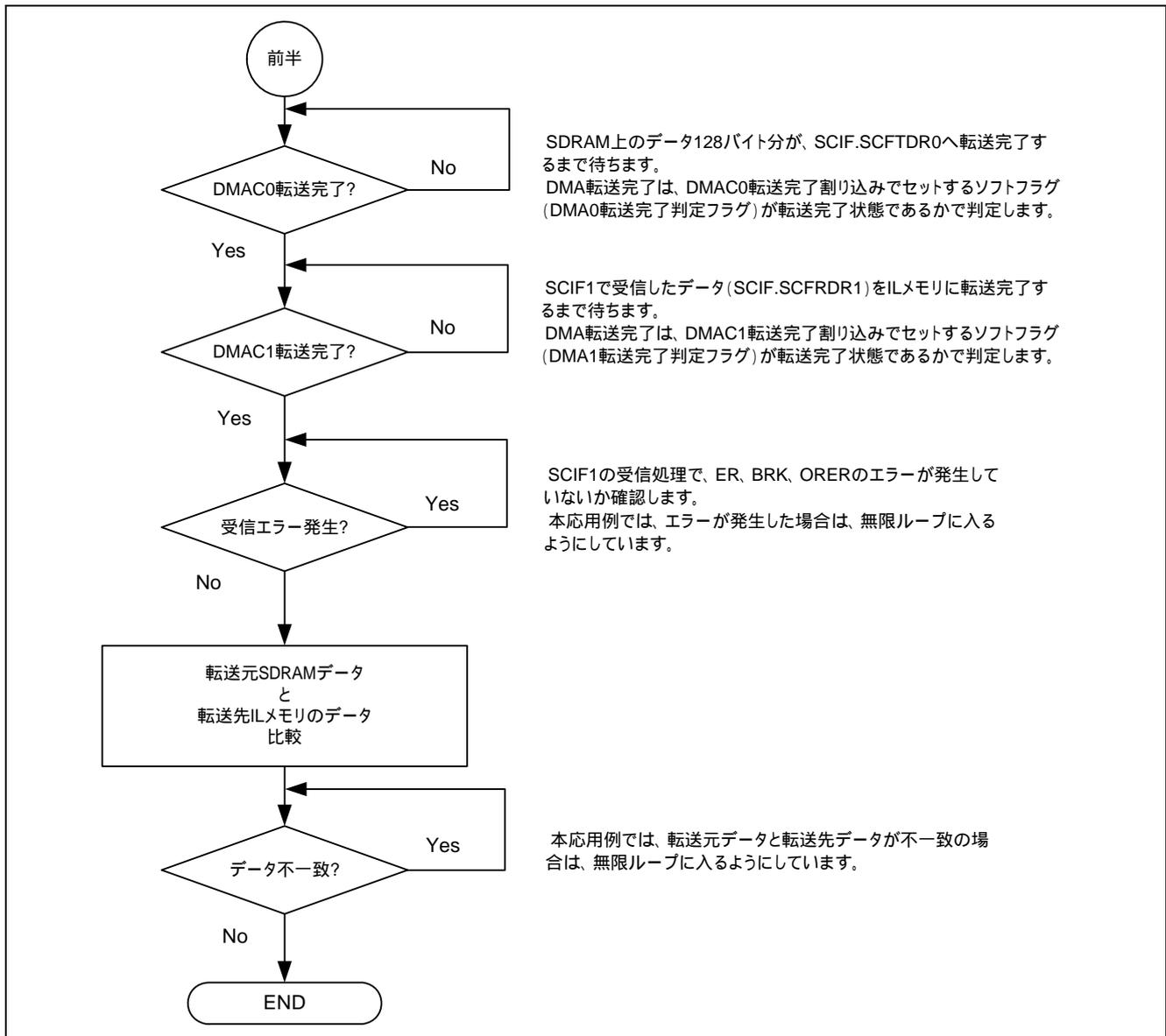


図4 参考プログラムメイン処理フロー (後半)

2.4.2 DMAC0 の初期化処理

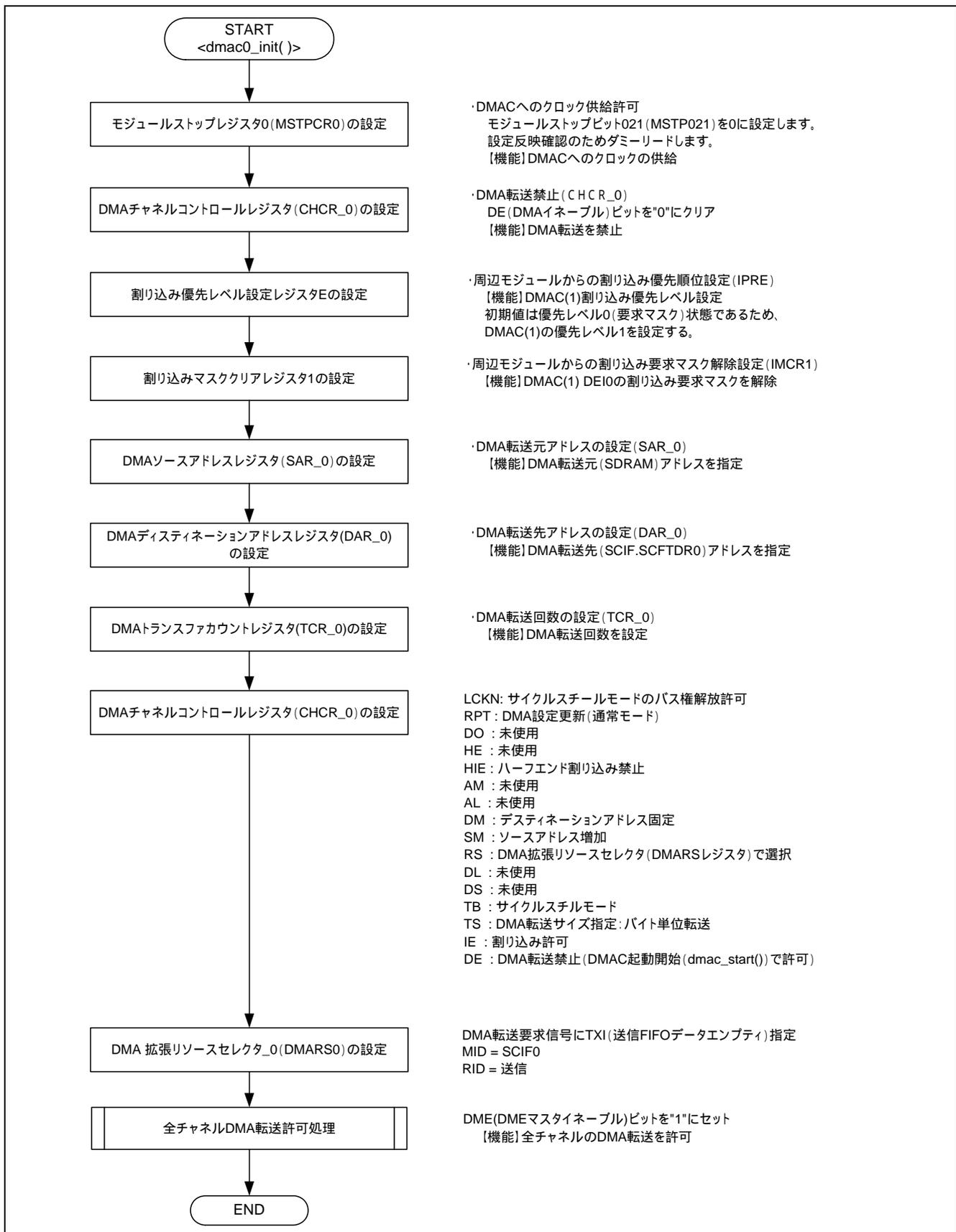


図 5 DMAC0 の初期化処理フロー

2.4.3 DMAC1 の初期化処理

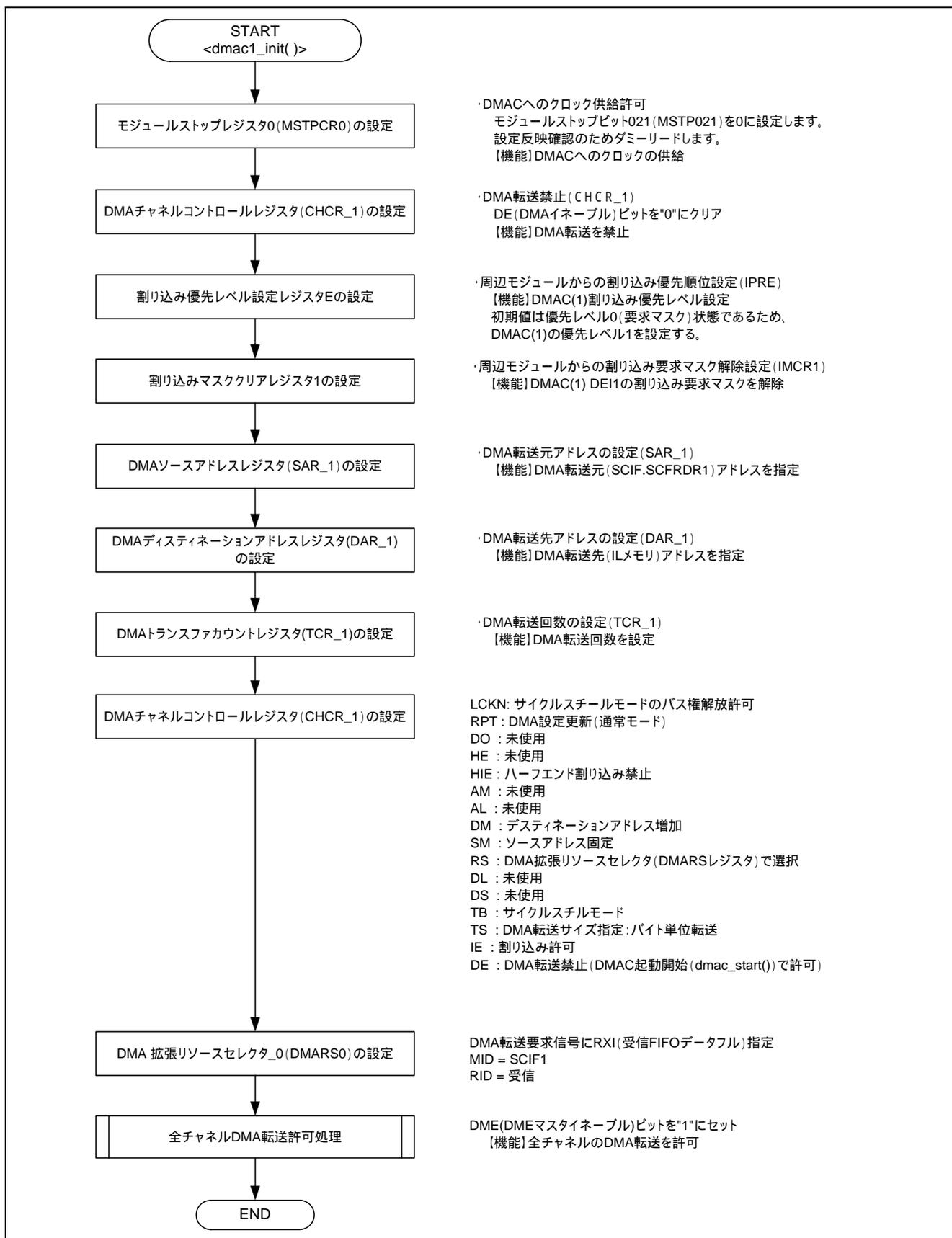


図 6 DMAC1 の初期化処理フロー

2.4.4 DMAC 起動処理

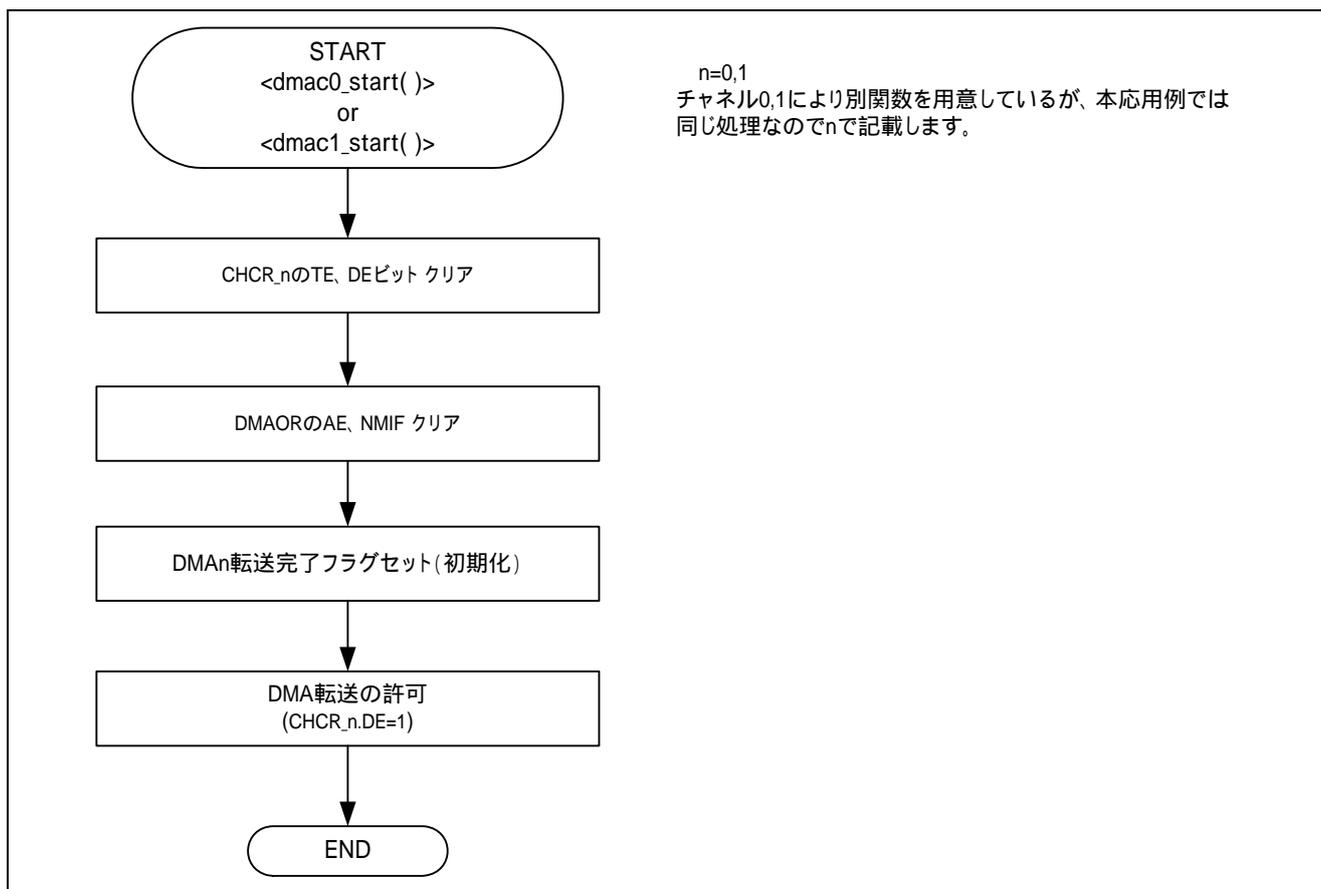


図7 DMAC 起動処理フロー

2.4.5 DMAC 停止処理

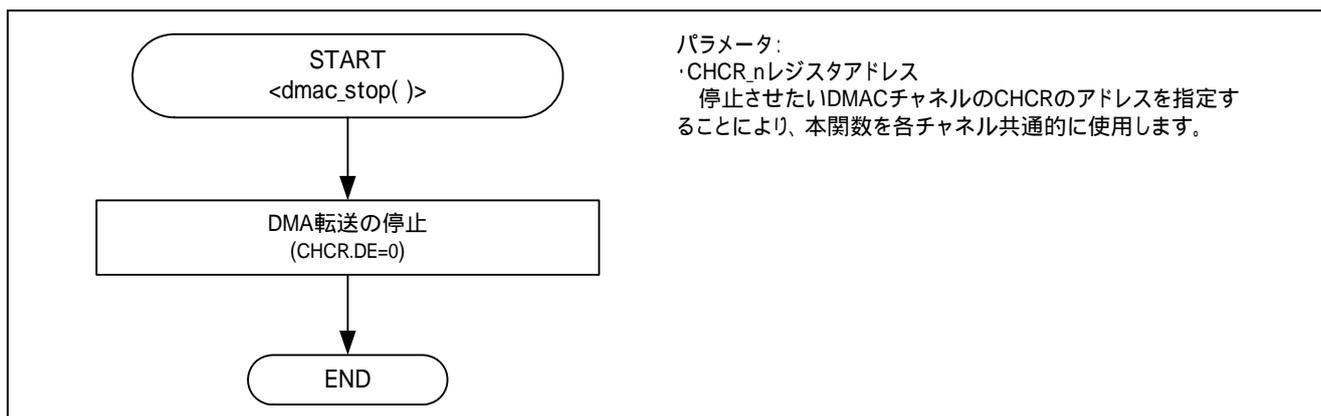


図8 DMAC 停止処理フロー

2.4.6 DMAC 転送終了割り込み処理

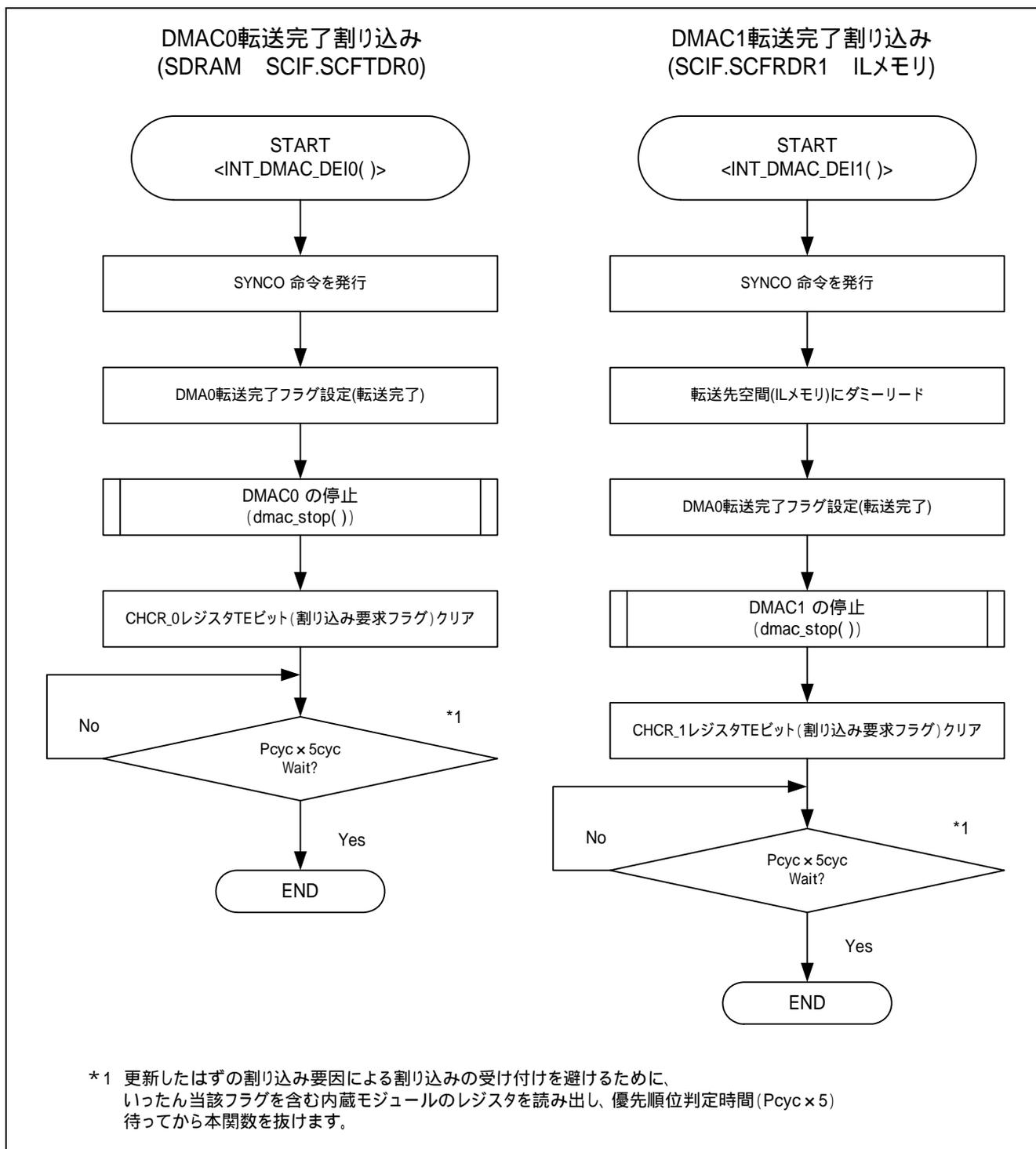


図9 DMAC 転送終了割り込み処理フロー

2.4.7 SCIF0 初期化処理 (調歩同期式の送信モードに初期化)

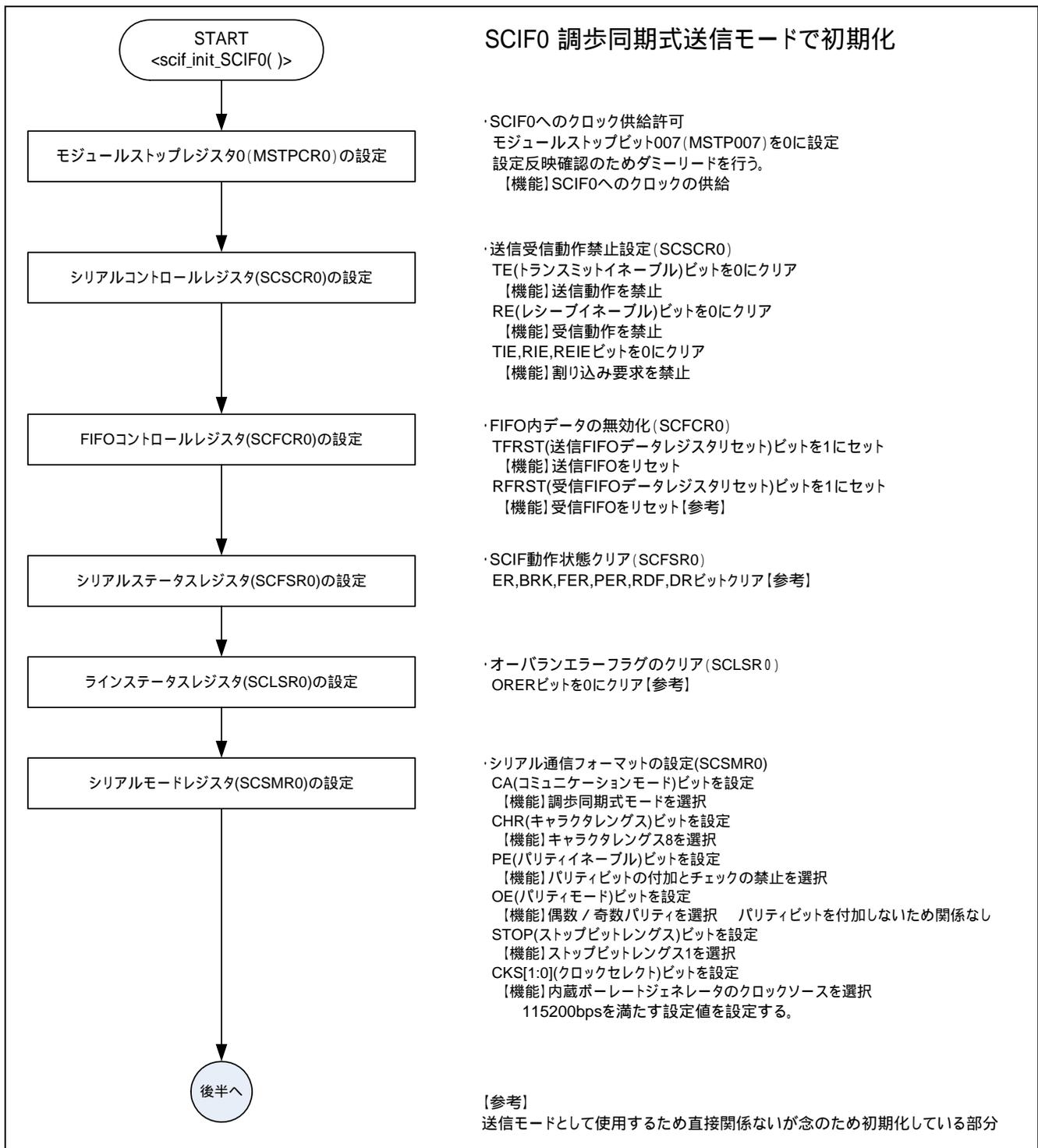


図 10 SCIF0 初期化処理フロー (前半)

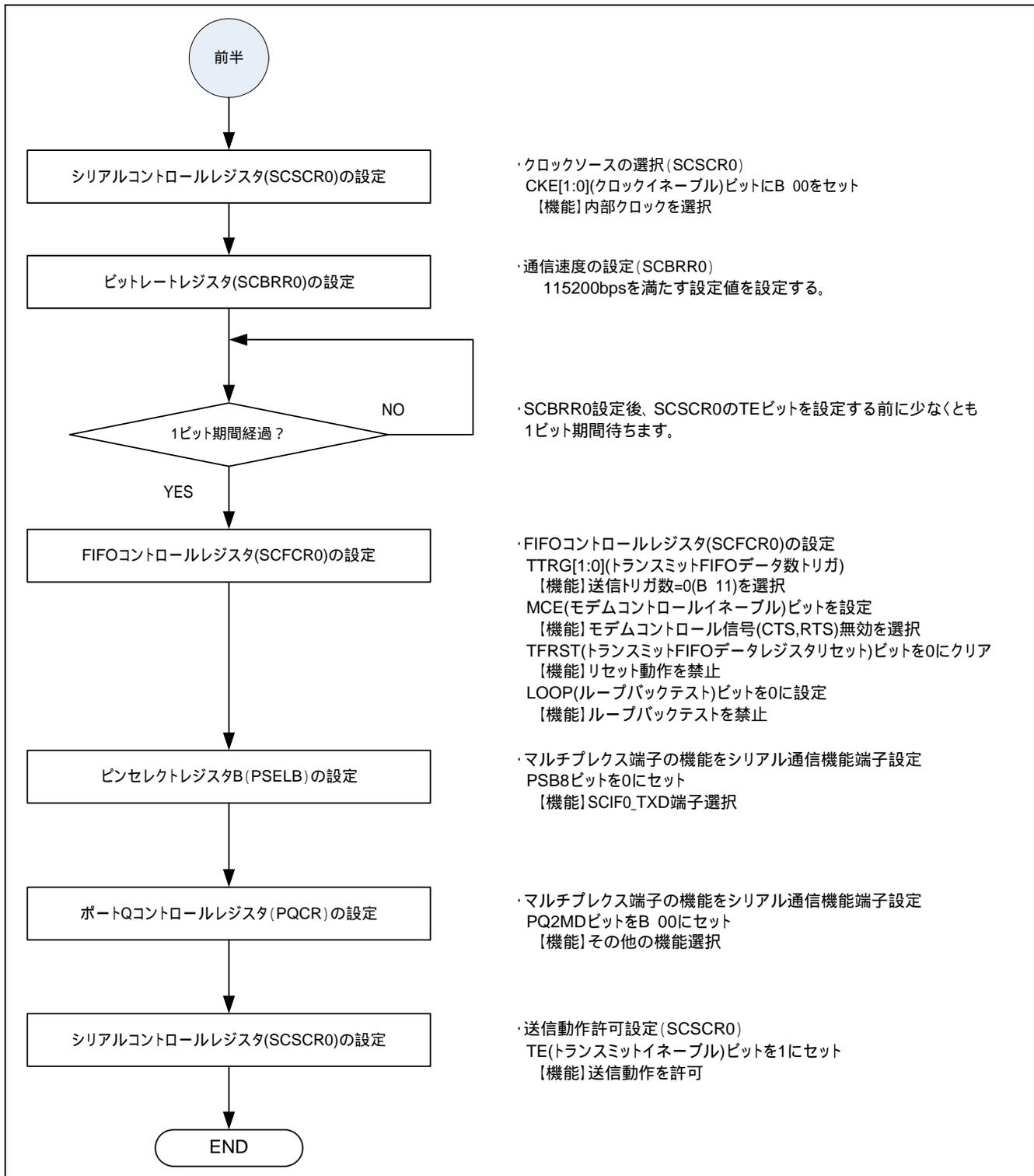


図 11 SCIF0 初期化処理フロー (後半)

2.4.8 SCIF1 初期化処理 (調歩同期式の受信モードに初期化)

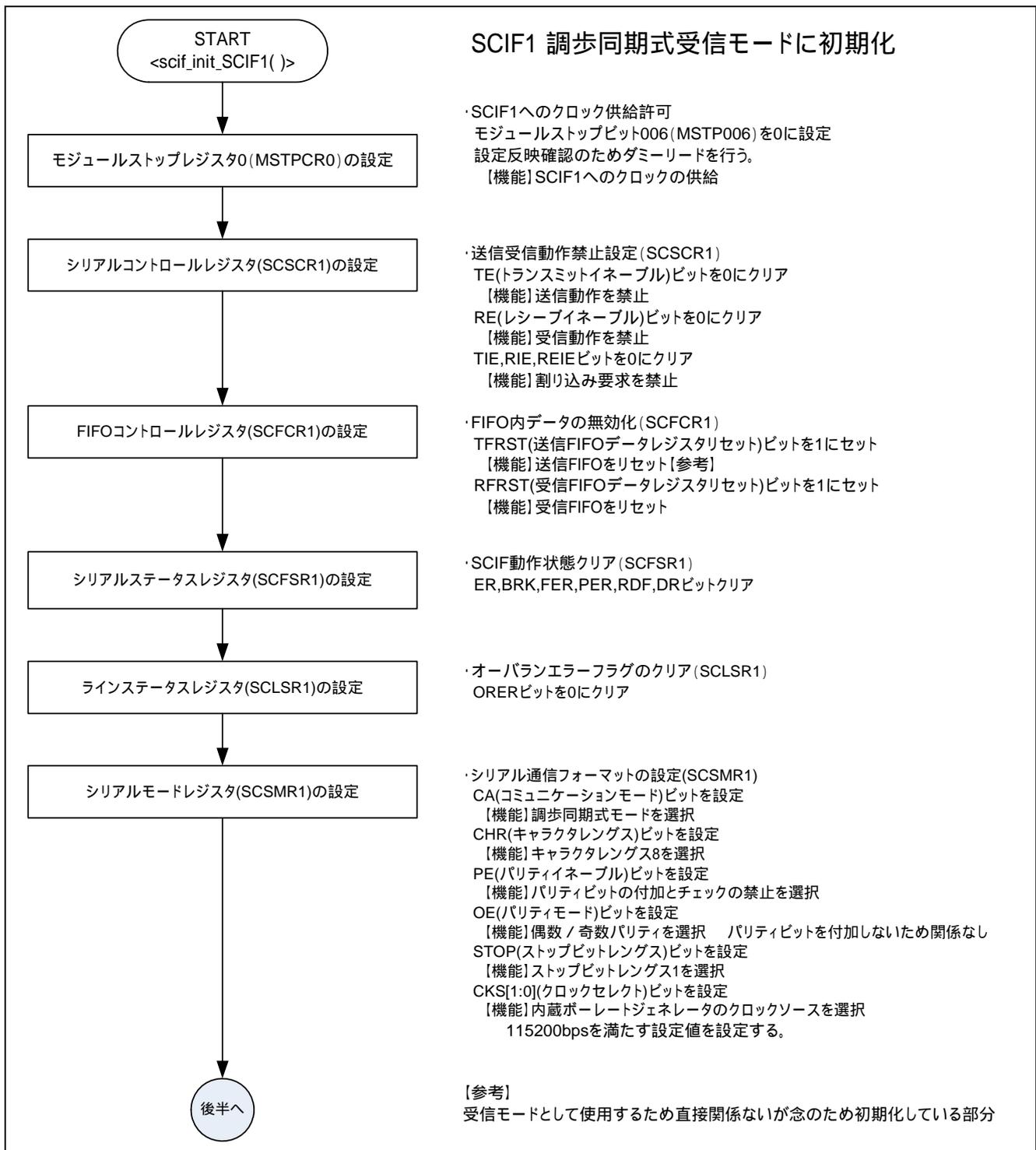


図 12 SCIF1 初期化処理フロー (前半)

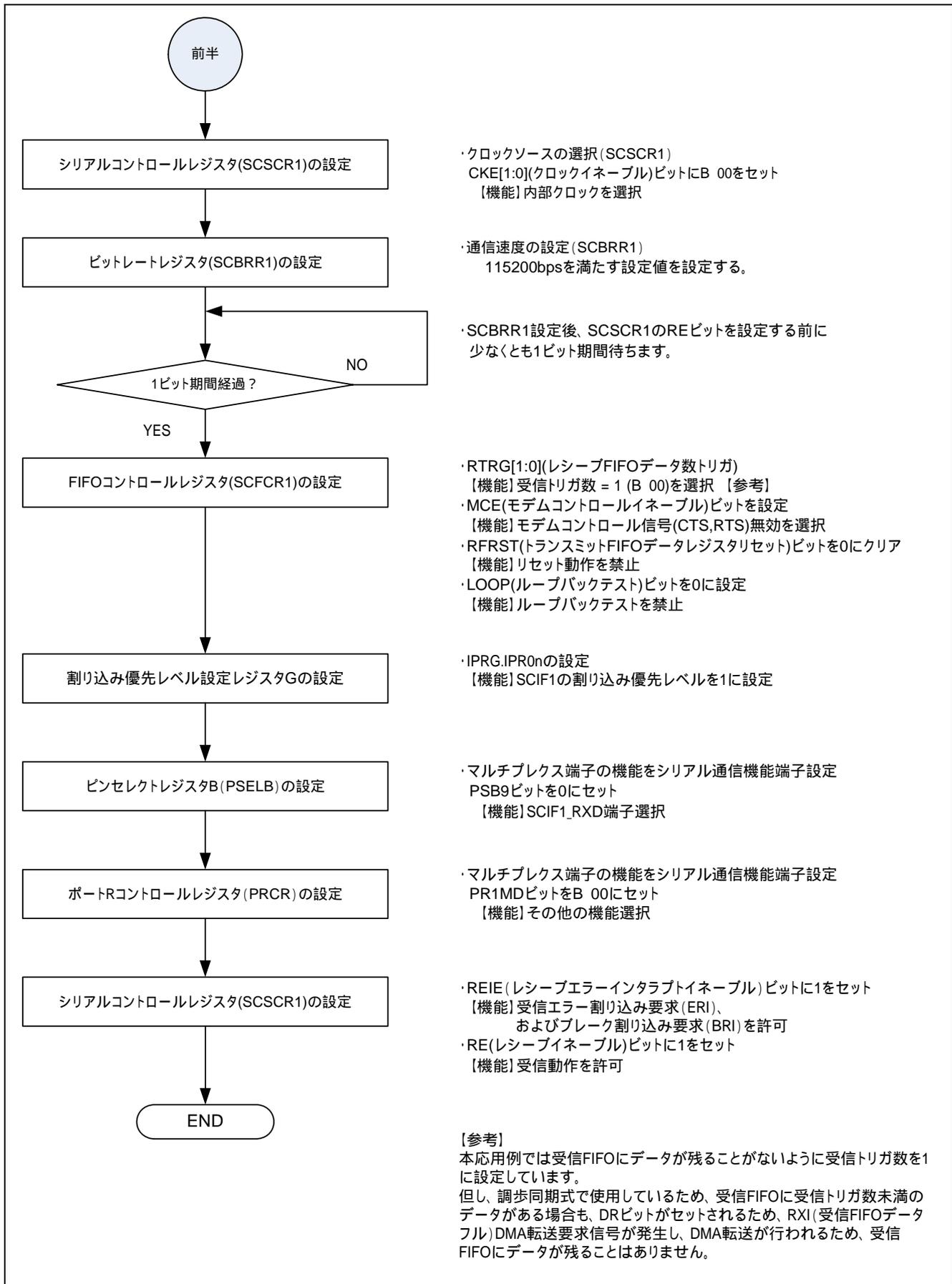


図 13 SCIF1 初期化処理フロー (後半)

2.5 参考プログラムのレジスタ設定値

以下に本参考プログラムのレジスタ設定値を示します。

2.5.1 DMAC0 関連レジスタ設定値

表 2 参考プログラムの DMAC0 関連レジスタ設定値

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_0 (CHCR_0)	H'FE00802C	H'00000000	<ul style="list-style-type: none"> 初期化処理時 DE = "0": DMA 転送禁止
		H'4000D804	<ul style="list-style-type: none"> 初期化処理時 LCKN = "1": バス権の解放許可 RPT = "B'000": 通常モード TS = "B'0000": バイト単位 TS[3:2]と TS[1:0]に設定する HIE = "0": ハーフエンド割り込み禁止 DM = "B'11": 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 IE = "1": DMA 転送時割り込み許可 <p>その他ビットについては、本応用例の目的とは関係ないビットであるため初期値のままとします。</p>
		H'4000D801	<ul style="list-style-type: none"> DMA 転送開始時 DE = "1": DMA 転送許可
		H'4000D802	<ul style="list-style-type: none"> DMA 転送完了時 DE = "0": DMA 転送禁止 <p>転送完了時 TE ビットがセットされています。</p>
DMA ソースアドレスレジスタ_0 (SAR_0)	H'FE008020	H'AC100000	<ul style="list-style-type: none"> 初期化処理時 転送元の開始アドレス: SDRAM の指定アドレス設定
DMA デスティネーションアドレスレジスタ_0 (DAR_0)	H'FE008024	H'FFE0000C	<ul style="list-style-type: none"> 初期化処理時 転送先の開始アドレス: SCIF0.SCFTDR のアドレス指定
DMA トランスファカウントレジスタ_0 (TCR_0)	H'FE008028	H'00000080	<ul style="list-style-type: none"> 初期化処理時 転送回数: 128 回 (H'80) 転送サイズがバイト単位のため
DMA 拡張リソースセクタ_0 (DMARS0)	H'FE009000	H'0021	<ul style="list-style-type: none"> 初期化処理時 MID = SCIF0 RID = 送信
DMA オペレーションレジスタ (DMAOR)	H'FE008060	H'0001	<ul style="list-style-type: none"> 初期化処理時 DME = "1": 全チャンネル DMA 転送許可

2.5.2 DMAC1 関連レジスタ設定値

表3 参考プログラムのDMAC1 関連レジスタ設定値

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'FE00803C	H'00000000	<ul style="list-style-type: none"> 初期化処理時 DE = "0": DMA 転送禁止
		H'40007804	<ul style="list-style-type: none"> 初期化処理時 LCKN = "1": バス権の解放許可 RPT = "B'000": 通常モード TS = "B'0000": バイト単位 TS[3:2]と TS[1:0]に設定する HIE = "0": ハーフエンド割り込み禁止 DM = "B'01": デスティネーションアドレス増加 SM = "B'11": 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 IE = "1": 転送終了時割り込み許可 その他ビットについては、本応用例の目的とは関係ないビットであるため初期値のままとします。
		H'40007801	<ul style="list-style-type: none"> DMA 転送開始時 DE = "1": DMA 転送許可
		H'40007802	<ul style="list-style-type: none"> DMA 転送完了時 DE = "0": DMA 転送禁止 転送完了時 TE ビットがセットされています。
DMA ソースアドレスレジスタ_1 (SAR_1)	H'FE008030	H'FFE10014	<ul style="list-style-type: none"> 初期化処理時 転送元の開始アドレス: SCIF1.SCFRDR のアドレス指定
DMA デスティネーションアドレスレジスタ_1 (DAR_1)	H'FE008034	H'E5203000	<ul style="list-style-type: none"> 初期化処理時 転送先の開始アドレス: IL メモリの指定アドレス設定
DMA トランスファカウントレジスタ_1 (TCR_1)	H'FE008038	H'00000080	<ul style="list-style-type: none"> 初期化処理時 転送回数: 128 回 (H'80) 転送サイズがバイト単位のため
DMA 拡張リソースセクタ_0 (DMARS0)	H'FE009000	H'2600	<ul style="list-style-type: none"> 初期化処理時 MID = SCIF1 RID = 受信
DMA オペレーションレジスタ (DMAOR)	H'FE008060	H'0001	<ul style="list-style-type: none"> 初期化処理時 DME = "1": 全チャンネル DMA 転送許可

2.5.3 SCIF0 関連レジスタ設定値

表 4 参考プログラムの SCIF0 関連レジスタ設定値

レジスタ名	アドレス	設定値	機能と設定値
シリアルコントロールレジスタ (SCSCR0)	H'FFE00008	H'0000	<ul style="list-style-type: none"> 初期化処理 TE = "0": 送信動作を禁止 RE = "0": 受信動作を禁止
		H'0020	<ul style="list-style-type: none"> 初期化処理 CKE[1:0] = "B'00": 内部クロック/SCK 端子は入力端子 TE = "1": 送信動作を許可
FIFO コントロールレジスタ (SCFCR0)	H'FFE00018	H'0006	<ul style="list-style-type: none"> 初期化処理 TFRST = "1": トランスミット FIFO データレジスタリセット動作を許可 RFRST = "1": レシーブ FIFO データレジスタリセット動作を許可
		H'0030	<ul style="list-style-type: none"> 初期化処理 TFRST = "0": トランスミット FIFO データレジスタリセット動作を禁止 RFRST = "0": レシーブ FIFO データレジスタリセット動作を禁止 TTRG[1:0] = "B'11": 送信 FIFO データ数のトリガ送信トリガ数 = 0 #define により変更可能
シリアルステータスレジスタ (SCFSR0)	H'FFE00010	H'0060	<ul style="list-style-type: none"> 初期化処理 ER, BRK, FER, PER, RDF, DR ビットクリア
ラインステータスレジスタ (SCLSR0)	H'FFE00024	H'0000	<ul style="list-style-type: none"> 初期化処理 ORER ビットクリア
シリアルモードレジスタ (SCSMR0)	H'FFE00000	H'0000	<ul style="list-style-type: none"> 初期化処理 CA = "0": 調歩同期式モード CHR = "0": 8 ビットデータ PE = "0": パリティビットの付加を禁止 STOP = "0": 1 ストップビット CKS[1:0] = "0": Pϕクロック #define により変更可能
ビットレートレジスタ (SCBRR0)	H'FFE00004	H'08	<ul style="list-style-type: none"> 初期化処理 115.2Kbps 設定 #define により変更可能
ピンセレクトレジスタ B (PSELB)	H'A4050102	H'00	<ul style="list-style-type: none"> 初期化処理 PSB8 = "0": SCIF0_TXD の選択
ポート Q コントロールレジスタ (PQCR)	H'A405001A	H'000A	<ul style="list-style-type: none"> 初期化処理 PQ2MD[1:0] = "B'00": その他の機能選択

2.5.4 SCIF1 関連レジスタ設定値

表 5 参考プログラムの SCIF1 関連レジスタ設定値

レジスタ名	アドレス	設定値	機能と設定値
シリアルコントロールレジスタ (SCSCR1)	H'FFE10008	H'0000	<ul style="list-style-type: none"> 初期化処理 TE = "0": 送信動作を禁止 RE = "0": 受信動作を禁止
		H'0018	<ul style="list-style-type: none"> 初期化処理 CKE[1:0] = "B'00": 内部クロック/SCK 端子は入力端子 REIE = "1": レシーブエラーインタラプトイネーブルを許可 RE = "1": 受信動作を許可
FIFO コントロールレジスタ (SCFCR1)	H'FFE10018	H'0006	<ul style="list-style-type: none"> 初期化処理 RFRST = "1": レシーブ FIFO データレジスタリセット動作を許可 TFRST = "1": トランスミット FIFO データレジスタリセット動作を許可
		H'0000	<ul style="list-style-type: none"> 初期化処理 RFRST = "0": レシーブ FIFO データレジスタリセット動作を禁止 TFRST = "0": トランスミット FIFO データレジスタリセット動作を禁止 RTRG[1:0] = "B'00": レシーブ FIFO データ数のトリガ 受信トリガ数 = 1 #define により変更可能
ラインステータスレジスタ (SCLSR1)	H'FFE10024	H'0000	<ul style="list-style-type: none"> 初期化処理 ORER ビットクリア
シリアルモードレジスタ (SCSMR1)	H'FFE10000	H'0000	<ul style="list-style-type: none"> 初期化処理 CA = "0": 調歩同期式モード CHR = "0": 8 ビットデータ PE = "0": パリティビットの付加を禁止 STOP = "0": 1 ストップビット CKS[1:0] = "0": Pϕ クロック #define により変更可能
ビットレートレジスタ (SCBRR1)	H'FFE10004	H'08	<ul style="list-style-type: none"> 初期化処理 115.2Kbps 設定 #define により変更可能
シリアルステータスレジスタ (SCFSR1)	H'FFE10010	H'0060	<ul style="list-style-type: none"> 初期化処理 ER, BRK, FER, PER, RDF, DR ビットクリア
ピンセレクトレジスタ B (PSELB)	H'A4050102	H'00	<ul style="list-style-type: none"> 初期化処理 PSB9 = "0": SCIF1_RXD の選択
ポート R コントロールレジスタ (PRCR)	H'A405001C	H'0022	<ul style="list-style-type: none"> 初期化処理 PR1MD [1:0] = "0": その他の機能選択
割り込み優先レベル設定レジスタ G (IPRG)	H'A4080018	H'0100	<ul style="list-style-type: none"> 初期化処理 IPR1 = "B'0001": 優先レベル 1 設定

2.6 参考プログラムのマクロ定義

表 6 参考プログラムの DMAC 関連マクロ定義

マクロ定義	設定値	機能
D_DMAM_SDRAM_SRC_ADR	H'AC100000	転送元の外部 SDRAM の先頭アドレス この値を変更することにより転送元のアドレスが変更可能です。 【注】 本参考プログラムでは、左記のマクロを利用して SDRAM 領域の初期化を行っています。 そのためキャッシュにアクセスしないようにキャッシング不可空間のアドレスを指定してください。
D_DMAM_IL_SRC_ADR	H'E5203000	転送先の IL メモリ先頭アドレス この値を変更することにより転送先のアドレスが変更可能です。
D_DMAM_TRANS_SIZE	128	転送サイズ (128 バイト) この値を変更することにより転送サイズが変更可能です。

表 7 参考プログラムの SCIF 関連マクロ定義

マクロ定義	設定値	機能
D_SCIF_DATA_NUM_SND_TRIGGER_CH0	0	トランスミット FIFO データ数トリガの設定値 設定値を以下 D_SCIF_SEND_FIFO_TRIGGER 0, 2, 4, 8 から選択可能。 本参考プログラムでは D_SCIF_SEND_FIFO_TRIGGER0 を設定
D_SCIF_DATA_NUM_RCV_TRIGGER_CH1	1	レシーブ FIFO データ数トリガの設定値 設定値を以下 D_SCIF_RCV_FIFO_TRIGGER 1, 2, 8, 14 から選択可能。 本参考プログラムでは D_SCIF_RCV_FIFO_TRIGGER1 を設定
D_SCIF_SEND_FIFO_TRIGGER0	0	トランスミット FIFO データ数トリガ: 0
D_SCIF_SEND_FIFO_TRIGGER2	2	トランスミット FIFO データ数トリガ: 2
D_SCIF_SEND_FIFO_TRIGGER4	4	トランスミット FIFO データ数トリガ: 4
D_SCIF_SEND_FIFO_TRIGGER8	8	トランスミット FIFO データ数トリガ: 8
D_SCIF_RCV_FIFO_TRIGGER1	1	レシーブ FIFO データ数トリガ: 1
D_SCIF_RCV_FIFO_TRIGGER4	4	レシーブ FIFO データ数トリガ: 4
D_SCIF_RCV_FIFO_TRIGGER8	8	レシーブ FIFO データ数トリガ: 8
D_SCIF_RCV_FIFO_TRIGGER14	14	レシーブ FIFO データ数トリガ: 14
D_SCIF_SCBRR_1200	0	ビットレート指定値: 1200bps *
D_SCIF_SCBRR_2400	1	ビットレート指定値: 2400bps *
D_SCIF_SCBRR_4800	2	ビットレート指定値: 4800bps *
D_SCIF_SCBRR_9600	3	ビットレート指定値: 9600bps *
D_SCIF_SCBRR_19200	4	ビットレート指定値: 19200bps *
D_SCIF_SCBRR_31250	5	ビットレート指定値: 31250bps *
D_SCIF_SCBRR_38400	6	ビットレート指定値: 38400bps *
D_SCIF_SCBRR_57600	7	ビットレート指定値: 57600bps *
D_SCIF_SCBRR_115200	8	ビットレート指定値: 115200bps *

【注】 * ビットレート指定値のビットレートを満たす SCSMR の CKS[1:0]、SCBRR 設定値、1bit wait 用の値が、あらかじめ T_SCIF_CKS_SCBRR_SET_INFO scif_cks_scbrr_tbl[] テーブルに格納されています。

テーブルの設定値は、 $f_{\phi} = 266.66\text{MHz}$ 、 $P_{\phi} = 33.33\text{MHz}$ で動作する場合を前提としております。

そのため、 f_{ϕ} 、 P_{ϕ} を変更する場合はテーブルの設定値も変更してください。

このテーブルに格納されている SCSMR の CKS[1:0]、SCBRR 設定値については、「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 22 章 FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF) 22.3.8 ビットレートレジスタ (SCBRR)」を参照ください。

2.7 SCIF 割り込み要因と DMAC 注意事項

- 割り込み要因の判定
SCIFはチャンネルごとに、送信 FIFO データエンプティ割り込み (TXI) 要求、受信エラー割り込み (ERI) 要求、受信 FIFO データフル割り込み (RXI) 要求、ブ레이크割り込み (BRI) 要求の 4 種類の割り込み要因を持っています。しかし、INTEVT のコードは 1 チャンネルに 1 つしか割り当てられていないため、割り込み要因の判定はソフトウェアで行う必要があります。
- 送信 FIFO データエンプティ DMA 転送要求
各割り込み要因は、チャンネルごと独立に、SCSCR の TIE、RIE、REIE ビットで、許可または禁止設定できます。TIE ビットにより TXI が許可されている場合、SCFSR の TDFE フラグが 1 にセットされると、TXI 割り込み要求と送信 FIFO データエンプティ DMA 転送要求が発生します。TIE ビットにより TXI が禁止されている場合、TDFE フラグが 1 にセットされると、送信 FIFO データエンプティ DMA 転送要求のみが発生します。送信 FIFO データエンプティ DMA 転送要求で、DMAC を起動してデータ転送を行うことができます。
- 受信 FIFO データフル DMA 転送要求
RIE ビットにより RXI が許可されている場合、SCFSR の RDF フラグまたは DR フラグが 1 にセットされると、RXI 割り込み要求と受信 FIFO データフル DMA 転送要求が発生します。RIE ビットにより RXI が禁止されている場合、RDF フラグまたは DR フラグが 1 にセットされると、受信 FIFO データフル DMA 転送要求のみが発生します。受信 FIFO データフル DMA 転送要求で、DMAC を起動してデータ転送を行うことができます。なお、DR フラグが 1 にセットされたことによる RXI 割り込みまたは受信 FIFO データフル DMA 転送要求は、調歩同期式モード時のみ発生します。
- BRI 割り込み要求
SCFSR の BRK フラグまたは SCLSR の ORER フラグが 1 にセットされると、BRI 割り込み要求が発生します。
- DMAC と SCIF の設定順序
DMAC を使って送受信を行う場合は、先に DMAC を設定し、イネーブル状態にしてから SCIF の設定を行ってください。
- RXI、TXI 割り込み要求 使用禁止
割り込みコントローラへ SCIF の RXI、TXI 割り込み要求を出さない設定にしてください。割り込み要求を出す設定にした場合、割り込みコントローラへの割り込み要求は割り込み処理プログラムとは無関係に DMAC によってクリアされます。
- ERI 割り込み要求のみ発行
SCSCR の RIE ビットを 0 に設定し、REIE ビットを 1 に設定すると、RXI 割り込み要求を出さずに ERI、BRI 割り込み要求だけを出すことができます。

3. 参考プログラム例

(1) サンプルプログラムリスト "sh7730.c"

```

1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. 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 * TECHNOLOGY CORP. 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 THE 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. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name : sh7730.c
32 * Abstract : SH7730 DMAC 内蔵モジュールへの転送例
33 * Version : Ver 1.00
34 * Device : SH7730
35 * Tool-Chain : High-performance Embedded Workshop (Version 4.04.01.001)
36 * : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description : SH7730 DMAC 内蔵モジュールへの転送例のサンプルプログラムです。
40 * :
41 * Operation :
42 * Limitation :
43 * :
44 *****/
45 * History : 26.JUNE.2009 Ver. 1.00 First Release
46 /*"FILE COMMENT END"*****
47
48 #include <machine.h>
49 #include "iodefine.h"
50 #include "dmac.h"
51 #include "scif.h"
52
53 /* ==== プロトタイプ宣言 ==== */
54 void main(void);
55
56 /*"FUNC COMMENT"*****
57 * ID :
58 * Outline : サンプルプログラムメイン
59 * Include :
60 * Declaration : void main(void)
61 * Description : SCIF0 TXI (送信 FIFO データエンプティ) の
62 * : DMA 転送要求信号によって SDRAM 上のデータを

```

```

63 *          : SCIF0.SCFDR へ転送します。
64 *          : SCIF0 SCIF1 へ調歩同期式でデータ送信します。
65 *          : SCIF1 では、RXI (受信 FIFO データフル) の
66 *          : DMA 転送要求信号によって SCIF1.SCFDR のデータを
67 *          : IL メモリに転送します。
68 *          : 転送完了後、SCIF1 の受信エラー状態を確認し
69 *          : 問題がなければ、
70 *          : SDRAM 上のデータと IL メモリ上の
71 *          : データが一致することを確認します。
72 *          :
73 * Limitation      : セクションの重なり
74 *          : 参考サンプルプログラムでは、データの
75 *          : 転送元の開始アドレスと転送先の
76 *          : 開始アドレスを明確にするため、
77 *          : 絶対番地で割り付けています。
78 *          : 絶対番地で割り付ける場合は、
79 *          : 使用しているセクションと重ならないよう
80 *          : にしてください。
81 *          :
82 *          : キャッシュを有効にして DMA 転送を行う場合の注意事項
83 *          : キャッシュメモリと外部メモリの内容が一致していない場合
84 *          : DMAC で指定するアドレスは、物理アドレスであるため、
85 *          : キャッシュの内容でなく、外部メモリの内容を転送する
86 *          : ことになり、意図しない状況を生じさせる可能性
87 *          : があります。
88 *          : そのため、DMA 転送する際には、キャッシュの無効化や
89 *          : 書き戻し等によりコヒーレンスを保つ
90 *          : 必要があります。
91 *          : 本プログラムでは、メモリアクセスにおいて、
92 *          : キャッシング不可空間でアクセスすることにより、
93 *          : キャッシュにアクセスせず対応しています。
94 *          :
95 * Argument        : none
96 * Return Value    : none
97 * Calling Functions :
98 * "FUNC COMMENT END"*****/
99 void main(void)
100 {
101     int i;
102     unsigned short dummy;
103     unsigned char *ptr1; /* 転送元メモリ用 */
104     unsigned char *ptr2; /* 転送先メモリ用 */
105
106     /* 転送先のメモリ初期化 */
107     ptr2 = D_DMAL_IL_DST_ADR;
108
109     for(i=0; i < D_DMAL_TRANS_SIZE; i++)
110     {
111         *ptr2++ = 0x00; /* 転送先メモリを 0x00 クリア */
112     }
113
114     /* 転送元のメモリ初期化 */
115     ptr1 = D_DMAL_SDRAM_SRC_ADR;
116
117     for(i=0; i < D_DMAL_TRANS_SIZE; i++)
118     {
119         *ptr1++ = 0x55; /* 転送元メモリに 0x55 をセット */
120     }
121
122     /* DMAL の初期化処理 */
123     dmal_init();
124
125     /* DMAL の起動 */
126     dmal_start();
127
128     /* SCIF1 を調歩同期式の受信モードに初期化 */
129

```

```
130     scif_init_SCIF1();
131
132     /* DMAC0 の初期化処理 */
133     dmac0_init();
134
135     /* DMAC0 の起動 */
136     dmac0_start();
137
138     /* SCIF0 を調歩同期式の送信モードに初期化 */
139     scif_init_SCIF0();
140
141     /* DMAC0 転送完了待ち(SDRAM SCIF0) */
142     /* (転送完了した際に以下処理を抜けます) */
143     while(D_DMACE_TRANS_END != dmac0_get_trans_end_flg()){
144     }
145
146     /* DMAC1 転送完了待ち(SCIF1 ILメモリ) */
147     /* (転送完了した際に以下処理を抜けます) */
148     while(D_DMACE_TRANS_END != dmac1_get_trans_end_flg()){
149     }
150
151     /* SCIF 受信エラー状態確認 */
152     /* SCIF1 受信失敗であれば、以下の無限ループに入る */
153     while(D_SCIF_RCV_NO_ERROR != scif_get_rcv_error());
154
155     /* 転送先メモリと転送元メモリのデータ一致確認 */
156     ptr1 = D_DMACE_SDRAM_SRC_ADR;
157     ptr2 = D_DMACE_IL_DST_ADR;
158
159     /* 転送サイズ分比較する */
160     for(i=0; i < D_DMACE_TRANS_SIZE; i++)
161     {
162         if(*ptr1++ != *ptr2++){
163             while(1){
164                 /* 転送先メモリと転送元メモリのデータが */
165                 /* 不一致(意図しないケース) */
166
167             }
168         }
169     }
170
171     /* ここまで処理が実行されることにより、転送先メモリと転送元メモリの */
172     /* データが一致していることが確認できる */
173     while(1){
174         /* ==== Program end ==== */
175     }
176
177 }
178
179 /* End of File */
```

(2) サンプルプログラムリスト "dmac.c"

```

1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. 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 * TECHNOLOGY CORP. 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 THE 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. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name : dmac.c
32 * Abstract : SH7730 DMAC 内蔵モジュールへの転送例
33 * Version : Ver 1.00
34 * Device : SH7730
35 * Tool-Chain : High-performance Embedded Workshop (Version 4.04.01.001)
36 * : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description : SH7730 DMAC 内蔵モジュールへの転送例のサンプルプログラムです。
40 * :
41 * Operation :
42 * Limitation :
43 * :
44 *****/
45 * History : 26.JUNE.2009 Ver. 1.00 First Release
46 /*"FILE COMMENT END"*****
47 #include <machine.h>
48 #include "iodefine.h"
49 #include "dmac.h"
50
51 /* 内部関数宣言 */
52 static unsigned long dmac_get_tcr(unsigned long i_size, unsigned long i_mode);
53 static void dmac_set_DMAOR_trans_ok(void);
54
55 /* DMA 転送完了フラグ */
56 T_Dmac_Trans_Chk g_dmac0_trans_end_chk = D_DMACH0_TRANS_INIT; /* DMACH0 用 */
57 T_Dmac_Trans_Chk g_dmac1_trans_end_chk = D_DMACH1_TRANS_INIT; /* DMACH1 用 */
58
59 /*"FUNC COMMENT"*****
60 * ID :
61 * Outline : DMACH0 初期設定
62 * Include :

```

```

63 * Declaration          : void dmac0_init(void)
64 * Description          : DMAC0 レジスタ初期設定を行います。
65 *                     : 内蔵モジュールリクエストで DMA 転送要求信号
66 *                     : を TXI (送信 FIFO データエンブティ) に指定
67 *                     : MID=B'001000 RID=B'01 設定
68 *                     : 転送元
69 *                     : SDRAM 0xAC100000 番地から 128 バイト分データ
70 *                     : 転送先
71 *                     : SCFTDR0 レジスタ
72 *                     :
73 * Argument             : none
74 * Return Value         : none
75 * Calling Functions    :
76 * "FUNC COMMENT END" *****/
77 void dmac0_init(void)
78 {
79     unsigned long dummy;
80
81     /* ==== モジュールストップレジスタ 0 設定 ==== */
82     LOWP.MSTPCR0 &= ~0x00200000; /* DMAC 起動 */
83
84     dummy = LOWP.MSTPCR0; /* MSTPCR0 への設定反映確認のためダミーリード */
85
86     /* DMA 転送完了フラグセット (初期化) */
87     g_dmac0_trans_end_chk = D_DMACH_TRANS_INIT;
88
89     /* ==== DMA チャンネル 0 コントロールレジスタの設定 ==== */
90     DMAC0.CHCR = (DMAC0.CHCR & 0xFFFFFFF0); /* DMA 禁止 */
91
92     /* ==== 割り込み優先レベル設定 ==== */
93     INTC0.IPRE |= 0x1000;
94
95     /* ==== 割り込みマスク要求の解除 ==== */
96     INTC0.IMCR1 |= 0x01;
97
98     /* ==== DMA ソースアドレスレジスタ_0 の設定 ==== */
99     DMAC0.SAR = D_DMACH_SDRAM_SRC_ADDR; /* SDRAM アドレスを指定 */
100
101     /* ==== DMA デスティネーションアドレスレジスタ_0 の設定 ==== */
102     DMAC0.DAR = &SCIF0.SCFTDR; /* SCIF0.SCFTDR のアドレス指定 */
103
104     /* ==== DMA トランスファカウンタレジスタの設定 ==== */
105     DMAC0.TCR = dmac_get_tcr(D_DMACH_TRANS_SIZE, D_DMACH_DMA_SIZE_BYTE); /* バイト単位転送 */
106
107     /* ==== DMA チャンネルコントロールレジスタ_0 設定 ==== */
108     DMAC0.CHCR = 0x4000D804;
109     /*
110         bit31      : reserve          : 0
111         bit30      : LCKN             : 1----- サイクルスチールモードのバス権解放許可
112         bit29-28   : reserve          : 0
113         bit27-25   : RPT              : 0----- DMA 設定更新 (通常モード)
114         bit24      : reserve          : 0
115         bit23      : DO over run0     : 0----- 未使用
116         bit22      : reserve          : 0
117         bit21-20   : TS               : 0----- DMA 転送サイズ指定 (bit4-3 と合わせて設定)
118         bit19      : HE               : 0----- 未使用
119         bit18      : HIE              : 0----- ハーフエンド割り込み禁止
120         bit17      : AM               : 0----- 未使用
121         bit16      : AL               : 0----- 未使用
122         bit15-14   : DM               : B'11----- デスティネーションアドレス固定
123         bit13-12   : SM               : B'01----- ソースアドレス増加
124         bit11-8    : RS               : B'1000----- DMA 拡張リソースセクタ (DMARS レジスタ) で選択
125         bit7       : DL               : 0 ----- 未使用
126         bit6       : DS               : 0 ----- 未使用

```

```

127         bit5      : TB          : 0----- サイクルスチルモード
128         bit4-3    : TS          : B'00----- バイト単位転送
129         bit2      : IE          : 1----- 割り込み許可
130         bit1      : TE          : 0
131         bit0      : DE          : 0----- DMA 転送禁止 (DMAC 起動開始 (dmac_start()) で許可)
132     */
133
134     /* DMA 拡張リソースセクタ_0 の設定 */
135     DMACR.DMARS0.BIT.COMID = 0x08; /* MID = SCIF0 */
136     DMACR.DMARS0.BIT.CORID = 0x01; /* RID = 送信 */
137
138     /* DMA オペレーションレジスタの設定 */
139     /* 全チャンネル DMA 転送を許可処理 */
140     dmac_set_DMAOR_trans_ok();
141
142 }
143
144 /*"FUNC COMMENT"*****
145 * ID :
146 * Outline : DMAC1 初期設定
147 * Include :
148 * Declaration : void dmac1_init(void)
149 * Description : DMAC1 レジスタ初期設定を行います。
150 * : 内蔵モジュールリクエストで DMA 転送要求信号
151 * : を RXI (受信 FIFO データフル) に指定
152 * : MID=B'001001 RID=B'10 設定
153 * : 転送元
154 * : SCFRDR1 レジスタ
155 * : 転送先
156 * : IL メモリ 0xE5203000 番地から 128 バイト分データ
157 * :
158 * Argument : none
159 * Return Value : none
160 * Calling Functions :
161 *"FUNC COMMENT END"*****/
162 void dmac1_init(void)
163 {
164     unsigned long dummy;
165
166     /* ==== モジュールストップレジスタ 0 設定 ==== */
167     LOWP.MSTPCR0 &= ~0x00200000; /* DMAC 起動 */
168
169     dummy = LOWP.MSTPCR0; /* MSTPCR0 への設定反映確認のためダミーリード */
170
171     /* DMA 転送完了フラグセット (初期化) */
172     g_dmac1_trans_end_chk = D_DMAL_TRANS_INIT;
173
174     /* ==== DMA チャンネル 1 コントロールレジスタの設定 ==== */
175     DMAC1.CHCR = (DMAC1.CHCR & 0xFFFFFFF); /* DMA 禁止 */
176
177     /* ==== 割り込み優先レベル設定 ==== */
178     INTC0.IPRE |= 0x1000;
179
180     /* ==== 割り込みマスク要求の解除 ==== */
181     INTC0.IMCR1 |= 0x02;
182
183     /* ==== DMA ソースアドレスレジスタ_1 の設定 ==== */
184     DMAC1.SAR = &SCIF1.SCFRDR; /* SCIF1.SCFRDR のアドレス指定 */
185
186     /* ==== DMA デスティネーションアドレスレジスタ_1 の設定 ==== */
187     DMAC1.DAR = D_DMAL_IL_DST_ADR; /* IL メモリアドレスを指定 */
188
189     /* ==== DMA トランスファカウンタレジスタの設定 ==== */
190     DMAC1.TCR = dmac_get_tcr(D_DMAL_TRANS_SIZE, D_DMAL_DMA_SIZE_BYTE); /* バイト単位転送 */

```

```

191
192 /* ==== DMA チャンネルコントロールレジスタ_1 設定 ==== */
193 DMAC1.CHCR = 0x40007804;
194 /*
195     bit31      : reserve      : 0
196     bit30      : LCKN         : 1----- サイクルスチルモードのバス権解放許可
197     bit29-28   : reserve      : 0
198     bit27-25   : RPT          : 0----- DMA 設定更新 (通常モード)
199     bit24      : reserve      : 0
200     bit23      : DO over run0 : 0----- 未使用
201     bit22      : reserve      : 0
202     bit21-20   : TS           : 0----- DMA 転送サイズ指定(bit4-3 と合わせて設定)
203     bit19      : HE           : 0----- 未使用
204     bit18      : HIE          : 0----- ハーフエンド割り込み禁止
205     bit17      : AM           : 0----- 未使用
206     bit16      : AL           : 0----- 未使用
207     bit15-14   : DM           : B'01----- デスティネーションアドレス増加
208     bit13-12   : SM           : B'11----- ソースアドレス固定
209     bit11-8     : RS           : B'1000----- DMA 拡張リソースセクタ (DMARS レジスタ) で選択
210     bit7        : DL           : 0 ----- 未使用
211     bit6        : DS           : 0 ----- 未使用
212     bit5        : TB           : 0----- サイクルスチルモード
213     bit4-3     : TS           : B'00----- バイト単位転送
214     bit2        : IE           : 1----- 割り込み禁止
215     bit1        : TE           : 0
216     bit0        : DE           : 0----- DMA 転送禁止 (DMAC 起動開始 (dmac_start()) で許可)
217 */
218
219 /* DMA 拡張リソースセクタ_0 の設定 */
220 DMACR.DMARS0.BIT.C1MID = 0x09; /* MID = SCIF1 */
221 DMACR.DMARS0.BIT.C1RID = 0x02; /* RID = 受信 */
222
223 /* DMA オペレーションレジスタの設定 */
224 /* 全チャンネル DMA 転送を許可処理 */
225 dmac_set_DMAOR_trans_ok();
226
227 }
228
229 /*"FUNC COMMENT"*****
230 * ID :
231 * Outline : DMAC0 起動開始
232 * Include :
233 * Declaration : void dmac0_start()
234 * Description : DMA0 転送を実行します。
235 * :
236 * Argument : none
237 * Return Value : none
238 * Calling Functions :
239 *"FUNC COMMENT END"*****/
240 void dmac0_start(void)
241 {
242     /* ==== DE,TE のクリア ==== */
243     DMAC0.CHCR &= ~0x00000003; /* DMA 転送禁止 */
244
245     /* ==== DMA オペレーションレジスタの設定 ==== */
246     DMAC.DMAOR.WORD &= 0xffff9; /* AE,NMIF ビットクリア */
247
248     /* DMA 転送完了フラグセット (初期化) */
249     g_dmac0_trans_end_chk = D_DMAL_TRANS_INIT;
250
251     /* ==== DMA の実行 ==== */
252     DMAC0.CHCR |= 0x00000001; /* DMA 転送許可 */
253 }
254

```

```

255 /*"FUNC COMMENT"*****
256 * ID :
257 * Outline : DMAC1 起動開始
258 * Include :
259 * Declaration : void dmacl_start()
260 * Description : DMA1 転送を実行します。
261 * :
262 * Argument : none
263 * Return Value : none
264 * Calling Functions :
265 /*"FUNC COMMENT END"*****/
266 void dmacl_start(void)
267 {
268     /* ==== DE,TE のクリア ==== */
269     DMAC1.CHCR &= ~0x00000003; /* DMA 転送禁止 */
270
271     /* ==== DMA オペレーションレジスタの設定 ==== */
272     DMAC.DMAOR.WORD &= 0xffff9; /* AE,NMIF ビットクリア */
273
274     /* DMA 転送完了フラグセット (初期化) */
275     g_dmacl_trans_end_chk = D_DMAL_TRANS_INIT;
276
277     /* ==== DMA の実行 ==== */
278     DMAC1.CHCR |= 0x00000001; /* DMA 転送許可 */
279 }
280
281 /*"FUNC COMMENT"*****
282 * ID :
283 * Outline : DMAC 停止処理
284 * Include :
285 * Declaration : void dmac_stop(unsigned long *io_pDMACn_CHCR)
286 * Description : DMA 転送を停止します。
287 * :
288 * Argument : unsigned long *io_pDMACn_CHCR :
289 * : 対象となるチャンネルの CHCR_n
290 * Return Value : none
291 * Calling Functions :
292 /*"FUNC COMMENT END"*****/
293 void dmac_stop(unsigned long *io_pDMACn_CHCR)
294 {
295     /* ==== DMA の停止 ==== */
296     *io_pDMACn_CHCR &= 0xffffffe; /* DE ビットクリア */
297 }
298
299 /*"FUNC COMMENT"*****
300 * ID :
301 * Outline : 転送回数を算出処理
302 * Include :
303 * Declaration : unsigned long dmac_get_tcr
304 * : (unsigned long i_size,
305 * : unsigned long i_mode)
306 * Description : DMAC の転送回数を算出します。
307 * :
308 * Argument : unsigned long i_size : 転送サイズ (バイト)
309 * : unsigned long i_mode : 転送モード
310 * : 以下のモードを指定します。
311 * : D_DMAL_DMA_SIZE_BYTE(0x0000)
312 * : バイト転送
313 * : D_DMAL_DMA_SIZE_WORD(0x0001)
314 * : ワード転送
315 * : D_DMAL_DMA_SIZE_LONG(0x0002)
316 * : ロング転送
317 * : D_DMAL_DMA_SIZE_LONGx2(0x0003)
318 * : 8 バイト単位転送

```

```

319 *          : D_DMACH_DMA_SIZE_LONGx4(0x0004)
320 *          :   16 バイト単位転送
321 *          : D_DMACH_DMA_SIZE_LONGx8(0x0005)
322 *          :   32 バイト単位転送
323 *          : D_DMACH_DMA_SIZE_LONGx2x2(0x0006)
324 *          :   16 バイト 2 分割 (8 バイト単位 x 2)
325 *          : D_DMACH_DMA_SIZE_LONGx4x2(0x0007)
326 *          :   32 バイト 2 分割 (16 バイト単位 x 2)
327 *          :
328 * Limitation      : 本関数の転送サイズに 0 を指定する場合
329 *                  : はご注意ください。
330 *                  : 0 指定した場合は、転送回数は 16777216 回
331 *                  : (最大回数)となります。
332 *                  :
333 * Return Value    : unsigned long : 転送回数
334 * Calling Functions :
335 * "FUNC COMMENT END"*****
336 unsigned long dmach_get_tcr(unsigned long i_size, unsigned long i_mode)
337 {
338     unsigned long     count = 0;
339
340     /* パラメータ注意 */
341     /* 転送サイズが 0 の場合は転送回数は 16777216 回となる */
342     if(i_size <= 0)
343     {
344         return count;
345     }
346
347     switch(i_mode)
348     {
349         case D_DMACH_DMA_SIZE_BYTE:
350         case D_DMACH_DMA_SIZE_WORD:
351         case D_DMACH_DMA_SIZE_LONG:
352         case D_DMACH_DMA_SIZE_LONGx2:
353         case D_DMACH_DMA_SIZE_LONGx4:
354         case D_DMACH_DMA_SIZE_LONGx8:
355             count = i_size >> i_mode;
356             break;
357
358         case D_DMACH_DMA_SIZE_LONGx2x2:
359             count = i_size >> 4u;
360             break;
361
362         case D_DMACH_DMA_SIZE_LONGx4x2:
363             count = i_size >> 5u;
364             break;
365
366         default:
367             break;
368     }
369
370     return     count;
371 }
372
373
374 /*"FUNC COMMENT"*****
375 * ID          :
376 * Outline      : 全チャンネル DMA 転送許可処理
377 * Include      :
378 * Declaration  : void dmach_set_DMAOR_trans_ok(void)
379 * Description  : AE,NMIF ビットクリア
380 *              : DME ビットに 1 設定
381 * Argument    : none
382 * Return Value : none

```

```

383 * Calling Functions      :
384 * "FUNC COMMENT END"*****
385 void dmac_set_DMAOR_trans_ok(void)
386 {
387     /* ==== DMA オペレーションレジスタの設定 ==== */
388     DMAC.DMAOR.WORD &= 0xffff9u; /* AE,NMIF ビットクリア */
389
390     /* 全チャンネルDMA 転送を許可 */
391     if(DMAC.DMAOR.BIT.DME == 0)
392     {
393         DMAC.DMAOR.BIT.DME = 1;
394     }
395 }
396
397 /* "FUNC COMMENT"*****
398 * ID                      :
399 * Outline                  : DMA0 転送完了フラグ設定処理
400 * Include                  :
401 * Declaration              : void dmac0_chk_trans_end(void)
402 *
403 * Description              : DMA0 転送完了フラグを設定します。
404 *                          : 本関数は DMAC0 転送終了割り込み関数からコールされる
405 *                          : ことを想定しています。
406 *
407 * Argument                 : none
408 * Return Value             : none
409 * Calling Functions        :
410 * "FUNC COMMENT END"*****
411 void dmac0_set_trans_end_flg(void)
412 {
413     volatile unsigned long    dummy;
414
415     /* SYNCO 命令を発行 */
416     synco();
417
418     /* DMA 転送完了フラグ設定(転送完了) */
419     g_dmac0_trans_end_chk = D_DMAL_TRANS_END;
420 }
421
422
423 /* "FUNC COMMENT"*****
424 * ID                      :
425 * Outline                  : DMA0 転送完了フラグ設定処理
426 * Include                  :
427 * Declaration              : void dmac1_chk_trans_end(void)
428 *
429 * Description              : DMA1 転送完了フラグを設定します。
430 *                          : 本関数は DMAC1 転送終了割り込み関数からコールされる
431 *                          : ことを想定しています。
432 *                          : DMA トランスファカウントレジスタ (TCR)
433 *                          : の値が 0 になり DMA 最終転送の実行を開始するとき
434 *                          : TE ビットは 1 にセットされます。
435 *                          : 最終転送完了を確認するには TE=1 にセットされた後
436 *                          : (DMA 転送終了による割り込み発生後)
437 *                          : SYNCO 命令を発行した後に、
438 *                          : 転送先空間にダミーリードを行います。
439 *
440 * Argument                 : none
441 * Return Value             : none
442 * Calling Functions        :
443 * "FUNC COMMENT END"*****
444 void dmac1_set_trans_end_flg(void)
445 {
446     volatile unsigned char    dummy;

```

```
447 volatile unsigned long *pdst_add;
448
449 /* SYNCO 命令を発行 */
450 synco();
451
452 /* 転送先空間にダミーリード */
453 pdst_add = D_DMAL_IL_DST_ADR;
454 dummy = *pdst_add;
455
456 /* DMA 転送完了フラグ設定(転送完了) */
457 g_dmac1_trans_end_chk = D_DMAL_TRANS_END;
458
459 }
460
461 /*"FUNC COMMENT"*****
462 * ID :
463 * Outline : DMA0 転送完了判定フラグ取得
464 * Include :
465 * Declaration : T_Dmac_Trans_Chk dmac0_get_trans_end_flg(void)
466 * :
467 * Description : DMA0 転送完了判定フラグを取得します。
468 * :
469 * Argument : none
470 * Return Value : T_Dmac_Trans_Chk : DMA0 転送完了判定フラグ
471 * Calling Functions :
472 *"FUNC COMMENT END"*****/
473 T_Dmac_Trans_Chk dmac0_get_trans_end_flg(void)
474 {
475 /* DMA0 転送完了フラグ */
476 return g_dmac0_trans_end_chk;
477 }
478
479 /*"FUNC COMMENT"*****
480 * ID :
481 * Outline : DMA1 転送完了判定フラグ取得
482 * Include :
483 * Declaration : T_Dmac_Trans_Chk dmac1_get_trans_end_flg(void)
484 * :
485 * Description : DMA1 転送完了判定フラグを取得します。
486 * :
487 * Argument : none
488 * Return Value : T_Dmac_Trans_Chk : DMA1 転送完了判定フラグ
489 * Calling Functions :
490 *"FUNC COMMENT END"*****/
491 T_Dmac_Trans_Chk dmac1_get_trans_end_flg(void)
492 {
493 /* DMA1 転送完了フラグ */
494 return g_dmac1_trans_end_chk;
495 }
496
497 /* End of File */
```

(3) サンプルプログラムリスト "scif.c"

```
1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. 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 * TECHNOLOGY CORP. 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 THE 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. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name : scif.c
32 * Abstract : SH7730 DMAC 内蔵モジュールへの転送例
33 * Version : Ver 1.00
34 * Device : SH7730
35 * Tool-Chain : High-performance Embedded Workshop (Version 4.04.01.001)
36 * : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description : SH7730 DMAC 内蔵モジュールへの転送例のサンプルプログラムです。
40 * :
41 * Operation :
42 * Limitation :
43 * :
44 *****/
45 * History : 26.JUNE.2009 Ver. 1.00 First Release
46 /*"FILE COMMENT END"*****
47
48 #include <machine.h>
49 #include <stdio.h>
50 #include <stdlib.h>
51 #include "iodefine.h"
52 #include "scif.h"
53
54 /* ==== レジスタ(CKS、SCBRR)設定値テーブル ==== */
55 /* 注意
56 以下の設定値は I =266.66MHz、P =33.33MHz で動作している場合の設定値
57 となります。I、P を変更する場合は以下の設定値も変更してください
```

```

58 */
59 const static T_SCIF_CKS_SCBRR_SET_INFO gc_scif_cks_scbrr_tbl[] =
60 {
61     /* SCBRR, CKS, 1bitwait */
62     { 216, 1, 222222}, /* 1200bps */
63     { 108, 1, 111111}, /* 2400bps */
64     { 216, 0, 55556}, /* 4800bps */
65     { 108, 0, 27778}, /* 9600bps */
66     { 53, 0, 13889}, /* 19200bps */
67     { 32, 0, 8533}, /* 31250bps */
68     { 26, 0, 6944}, /* 38400bps */
69     { 17, 0, 4630}, /* 57600bps */
70     { 8, 0, 2315} /*115200bps */
71 };
72
73 /* ==== レジスタ(SCFCR.TTRG) 設定値テーブル ==== */
74 const static T_SCIF_SCFCR_TTRG_SET gc_scif_ttrg_tbl[] =
75 {
76     { D_SCIF_SEND_FIFO_TRIGGER8, 0x00}, /* トランスミット FIFO データ数トリガ = 8 */
77     { D_SCIF_SEND_FIFO_TRIGGER4, 0x01}, /* トランスミット FIFO データ数トリガ = 4 */
78     { D_SCIF_SEND_FIFO_TRIGGER2, 0x02}, /* トランスミット FIFO データ数トリガ = 2 */
79     { D_SCIF_SEND_FIFO_TRIGGER0, 0x03} /* トランスミット FIFO データ数トリガ = 0 */
80 };
81
82 /* ==== レジスタ(SCFCR.RTRG) 設定値テーブル ==== */
83 const static T_SCIF_SCFCR_RTRG_SET scif_rtrg_tbl[] =
84 {
85     { D_SCIF_RCV_FIFO_TRIGGER1, 0x00}, /* レシーブ FIFO データ数トリガ = 1 */
86     { D_SCIF_RCV_FIFO_TRIGGER4, 0x01}, /* レシーブ FIFO データ数トリガ = 4 */
87     { D_SCIF_RCV_FIFO_TRIGGER8, 0x02}, /* レシーブ FIFO データ数トリガ = 8 */
88     { D_SCIF_RCV_FIFO_TRIGGER14, 0x03} /* レシーブ FIFO データ数トリガ = 14 */
89 };
90
91 /* ==== プロトタイプ宣言 ==== */
92 static unsigned char scif_get_clock_select(T_SCIF_bit_rate_type i_type_bps);
93 static unsigned char scif_get_SCBRR(T_SCIF_bit_rate_type i_type_bps);
94 static unsigned char scif_get_snd_trigger_num(T_SCIF_send_fifo_trigger i_trigger_num);
95 static unsigned char scif_get_rcv_trigger_num(T_SCIF_rcv_fifo_trigger i_trigger_num);
96 static void scif_wait_lbit(unsigned long i_bit_rate);
97
98 /* SCIF 受信エラー状態フラグ */
99 T_SCIF_rcv_error g_rcv_error = D_SCIF_RCV_NO_ERROR;
100
101 /*"FUNC COMMENT"*****
102 * ID :
103 * Outline : SCIF0 初期化処理
104 * Include :
105 * Declaration : void scif_init_SCIF0(void)
106 * Description : SCIF0 を調歩同期式送信モードで初期化します。
107 * : 主に以下の設定をしています。
108 * :
109 * : TIE 割り込み要求禁止
110 * : RIE 割り込み要求禁止
111 * : TE 送信動作を禁止 許可
112 * : RE 受信動作を禁止
113 * : REIE 割り込み要求禁止
114 * : CKE 内部クロック選択
115 * :

```

```

116 *           : SCBRR
117 *           : gc_scif_cks_scbrr_tbl[]より設定
118 *           : 本参考プログラムでは、ビットレートは、
119 *           : 115200bps で指定します。
120 *           : TTRG トランスミット FIFO データ数トリガ 0 設定
121 *           :
122 * Limitation : ボーレート設定値は内部クロックを使った
123 *           : 周辺モジュール用動作周波数 P =33.33MHz
124 *           : の場合です。
125 *           : 他のクロックを使用する場合はビットレート
126 *           : 設定値を変更してください。
127 *           : gc_scif_cks_scbrr_tbl[]の設定値を
128 *           : 修正してください。
129 *           :
130 * Argument   : none
131 * Return Value : none
132 * Calling Functions :
133 * "FUNC COMMENT END"*****/
134 void scif_init_SCIF0(void)
135 {
136     unsigned long    dummy;
137
138     /* ==== SCIF0 の初期設定 ==== */
139     /* ==== モジュールストップレジスタの設定 ==== */
140     LOWP.MSTPCR0 &= ~0x00000080;
141
142     dummy = LOWP.MSTPCR0; /* 設定反映確認のためダミリード */
143
144     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
145     SCIF0.SCSCR.WORD = 0x0000; /* SCIF0 送/受信動作停止 */
146
147     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
148     SCIF0.SCFCR.BIT.TFRST = 1; /* 送信 FIFO リセット */
149     SCIF0.SCFCR.BIT.RFRST = 1; /* 受信 FIFO リセット */
150
151     /* ==== シリアルステータスレジスタ(SCFSR)の初期化 ==== */
152     SCIF0.SCFSR.WORD &= ~0xff9f; /* ER, BRK, FER, PER, RDF, DR ビットクリア */
153
154     /* ==== ラインステータスレジスタ(SCLSR)の設定 ==== */
155     SCIF0.SCLSR.BIT.ORER = 0; /* ORER ビットクリア */
156
157     /* ==== シリアルモードレジスタ(SCSMR)の設定 ==== */
158     /* コミュニケーションモード : 調歩同期式 */
159     /* キャラクタレングス : 8 ビットデータ */
160     /* パリティイネーブル : 付加とチェックを禁止 */
161     /* ストップビットレングス : 1 ストップビット */
162     SCIF0.SCSMR.WORD = 0x0000;
163
164     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
165     SCIF0.SCSCR.BIT.CKE = 0; /* B'00:内部クロック */
166
167     /* ==== クロックセレクト : ビットレートの設定値により決定 ==== */
168     SCIF0.SCSMR.BIT.CKS = scif_get_clock_select(D_SCIF_SCBRR_ACT);
169
170     /* ==== ビットレートレジスタ(SCBRR)の設定 : ビットレートの設定値により決定 ==== */
171     SCIF0.SCBRR = scif_get_SCBRR(D_SCIF_SCBRR_ACT);
172
173     /* ==== 1 ビット期間経過待ち ==== */

```

```

174     scif_wait_lbit(D_SCIF_SCBRR_ACT);
175
176     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
177     /* モデムコントロールイネーブル : 禁止 */
178     /* 送信 FIFO データレジスタリセット : 禁止 */
179     /* ループバックテスト : 禁止 */
180     SCIF0.SCFCR.WORD = 0x0000;
181
182     /* 送信 FIFO データ数トリガ : データ数 0 */
183     /* 送信 FIFO データ数トリガは D_SCIF_DATA_NUM_SND_TRIGGER_CH0 の設定値になります */
184     SCIF0.SCFCR.BIT.TTRG = scif_get_snd_trigger_num(D_SCIF_DATA_NUM_SND_TRIGGER_CH0);
185
186     /* ==== ピンセレクトレジスタ B (PSELB) ==== */
187     PFC.PSELB.BIT.PSB8 = 0; /* SCIF0_TXD の選択 */
188
189     /* ==== ポート Q コントロールレジスタ (PQCR) の設定 ==== */
190     PFC.PQCR.BIT.PQ2MD = 0; /* その他の機能選択 */
191
192     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
193     SCIF0.SCSCR.BIT.TE = 1; /* SCIF0 送信動作を許可する */
194
195 }
196
197 /*"FUNC COMMENT"*****
198 * ID :
199 * Outline : SCIF1 初期化処理
200 * Include :
201 * Declaration : void scif_init_SCIF1(void)
202 * Description : SCIF1 を調歩同期式受信モードで初期化します。
203 * : 主に以下の設定をしています。
204 * :
205 * : TIE 割り込み要求禁止
206 * : RIE 割り込み要求禁止
207 * : TE 送信動作を禁止
208 * : RE 受信動作を禁止 許可
209 * : REIE 割り込み要求許可
210 * : CKE 内部クロック
211 * :
212 * : SCBRR
213 * : gc_scif_cks_scbrr_tbl[]より設定
214 * : 本参考プログラムでは、ビットレートは、
215 * : 115200bps で指定します。
216 * : RTRG レシーブ FIFO データ数トリガ 1 設定
217 * :
218 * Limitation : ボーレート設定値は内部クロックを使った
219 * : 周辺モジュール用動作周波数 P =33MHz
220 * : の場合です。
221 * : 他のクロックを使用する場合はビットレート
222 * : 設定値を変更してください。
223 * : gc_scif_cks_scbrr_tbl[]の設定値を
224 * : 修正してください。
225 * :
226 * Argument : none
227 * Return Value : none
228 * Calling Functions :
229 *"FUNC COMMENT END"*****/
230 void scif_init_SCIF1(void)
231 {

```

```
232 unsigned long dummy;
233
234 /* ==== SCIF1 の初期設定 ==== */
235 /* ==== モジュールストップレジスタの設定 ==== */
236 LOWP.MSTPCR0 &= ~0x00000040;
237
238 dummy = LOWP.MSTPCR0; /* 設定反映確認のためダミリード */
239
240 /* SCIF 受信エラー状態フラグ初期化 */
241 g_rcv_error = D_SCIF_RCV_NO_ERROR;
242
243 /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
244 SCIF1.SCSCR.WORD = 0x0000; /* SCIF1 送/受信動作停止 */
245
246 /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
247 SCIF1.SCFCR.BIT.TFRST = 1; /* 送信 FIFO リセット */
248 SCIF1.SCFCR.BIT.RFRST = 1; /* 受信 FIFO リセット */
249
250 /* ==== シリアルステータスレジスタ(SCFSR)の設定 ==== */
251 SCIF1.SCFSR.WORD &= ~0xff9f; /* ER, BRK, FER, PER, RDF, DR ビットクリア */
252
253 /* ==== ラインステータスレジスタ(SCLSR)の設定 ==== */
254 SCIF1.SCLSR.BIT.OPER = 0; /* OPER ビットクリア */
255
256 /* ==== シリアルモードレジスタ(SCSMR)の設定 ==== */
257 /* コミュニケーションモード : 調歩同期式 */
258 /* キャラクタレンクス : 8 ビットデータ */
259 /* パリティイネーブル : 付加とチェックを禁止 */
260 /* ストップビットレンクス : 1 ストップビット */
261 SCIF1.SCSMR.WORD = 0x00;
262
263 /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
264 SCIF1.SCSCR.BIT.CKE = 0; /* B'00:内部クロック */
265
266 /* ==== クロックセレクト : ビットレートの設定値により決定 ==== */
267 SCIF1.SCSMR.BIT.CKS = scif_get_clock_select(D_SCIF_SCBRR_ACT);
268
269 /* ==== ビットレートレジスタ(SCBRR)の設定 : ビットレートの設定値により決定 ==== */
270 SCIF1.SCBRR = scif_get_SCBRR(D_SCIF_SCBRR_ACT);
271
272 /* ==== 1 ビット期間経過待ち ==== */
273 scif_wait_lbit(D_SCIF_SCBRR_ACT);
274
275 /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
276 /* モデムコントロールイネーブル : 禁止 */
277 /* 送信 FIFO データレジスタリセット : 禁止 */
278 /* ループバックテスト : 禁止 */
279 SCIF1.SCFCR.WORD = 0x0000;
280
281 /* 受信 FIFO データ数トリガ : データ数 1 */
282 /* 受信 FIFO データ数トリガは D_SCIF_DATA_NUM_RCV_TRIGGER_CH1 の設定値になります */
283 SCIF1.SCFCR.BIT.RTRG = scif_get_rcv_trigger_num(D_SCIF_DATA_NUM_RCV_TRIGGER_CH1);
284
285 /* ==== 割り込みマスククリアレジスタ 5 の設定 ==== */
286 INTC0.IMCR5 |= 0x02;
287
288 /* ==== 割り込み優先レベル設定 (優先レベル 1) ==== */
289 INTC0.IPRG = INTC0.IPRG | 0x0100;
```

```

290
291 /* ==== ピンセレクトレジスタ B (PSELB) ==== */
292 PFC.PSELB.BIT.PSB9 = 0; /* SCIF1_RXD の選択 */
293
294 /* ==== ポート R コントロールレジスタ (PRCR) の設定 ==== */
295 PFC.PRCR.BIT.PR1MD = 0; /* その他の機能選択 */
296
297 /* ==== シリアルコントロールレジスタ (SCSCR) の設定 ==== */
298 SCIF1.SCSCR.BIT.REIE = 1; /* ERI, BRI 割り込み有効 */
299
300 /* ==== シリアルコントロールレジスタ (SCSCR) の設定 ==== */
301 SCIF1.SCSCR.BIT.RE = 1; /* SCIF0 受信動作を許可する */
302
303 }
304
305 /*"FUNC COMMENT"*****
306 * ID :
307 * Outline : 受信時エラーフラグ対応処理
308 * Include :
309 * Declaration : void scif_rcv_error(void)
310 * Description : 本関数は SCIF1 割り込み処理 (INT_SCIF_SCIF1())
311 * : からコールされます。
312 * : 本応用例では、SCSCR.RIE を許可にせず、
313 * : SCSCR.REIE のみを有効にしています。
314 * : よって、受信エラー割り込み (ERI) 要求 or
315 * : ブレーク割り込み (BRI) 要求
316 * : が発生した際のみコールされます。
317 * : SCIF 受信エラー状態フラグをエラーありに
318 * : セットします。
319 * :
320 * Argument : none
321 * Return Value : none
322 * Calling Functions :
323 *"FUNC COMMENT END"*****/
324 void scif_rcv_error(void)
325 {
326
327 /* SCIF 受信エラー状態フラグをエラー状態にセット */
328 g_rcv_error = D_SCIF_RCV_ERROR;
329
330 do
331 {
332 /* ==== ORER (オーバラン) エラーの時 ==== */
333 /* ORER (オーバラン) エラーの時に行いたい処理を定義してください */
334 if (SCIF1.SCLSR.BIT.ORER == 1)
335 {
336 SCIF1.SCLSR.BIT.ORER = 0; /* 割り込み要求のクリア */
337 break;
338 }
339
340 /* ==== BRK (ブレーク) エラーの時 ==== */
341 /* BRK (ブレーク) エラーの時に行いたい処理を定義してください */
342 if (SCIF1.SCFSR.BIT.BRK == 1)
343 {
344 SCIF1.SCFSR.BIT.BRK = 0; /* 割り込み要求のクリア */
345 break;
346 }
347

```

```

348     /* ==== ER(レシーブ)エラーの時 ==== */
349     /* ER(レシーブ)エラーの時にやりたい処理を定義してください */
350     if(SCIF1.SCFSR.BIT.ER == 1)
351     {
352         SCIF1.SCFSR.BIT.ER = 0;           /* 割り込み要求のクリア */
353         break;
354     }
355
356     }while(0);
357
358     return;
359
360 }
361
362 /*"FUNC COMMENT"*****
363 * ID          :
364 * Outline     : SCIF 受信エラー状態取得
365 * Include     :
366 * Declaration : T_SCIF_rcv_error scif_get_rcv_error(void)
367 * Description : SCIF 受信エラー状態が取得できます。
368 *            :
369 * Argument    : none
370 * Return Value : T_SCIF_rcv_error
371 *            : D_SCIF_RCV_NO_ERROR エラーなし
372 *            : D_SCIF_RCV_ERROR   エラーあり
373 * Calling Functions :
374 *"FUNC COMMENT END"*****/
375 T_SCIF_rcv_error scif_get_rcv_error(void)
376 {
377     return g_rcv_error;
378 }
379
380 /*"FUNC COMMENT"*****
381 * ID          :
382 * Outline     : クロックセレクト設定値取得
383 * Include     :
384 * Declaration : unsigned char scif_get_clock_select
385 *            : (T_SCIF_bit_rate_type i_type_bps)
386 * Description : ビットレートに対応するクロックセレクト
387 *            : CKS[1:0]設定値を取得する。
388 *            :
389 * Argument    : T_SCIF_bit_rate_type i_type_bps :
390 *            : ビットレート指定値
391 * Return Value : unsigned char : クロックセレクト設定値
392 * Calling Functions :
393 *"FUNC COMMENT END"*****/
394 unsigned char scif_get_clock_select(T_SCIF_bit_rate_type i_type_bps)
395 {
396     return gc_scif_cks_scbrr_tbl[i_type_bps].mscsmr_cks;
397 }
398
399 /*"FUNC COMMENT"*****
400 * ID          :
401 * Outline     : ビットレートレジスタ設定値(SCBRR)取得
402 * Include     :
403 * Declaration : unsigned char scif_get_SCBRR
404 *            : (T_SCIF_bit_rate_type i_type_bps)
405 * Description : ビットレートに対応するビットレート

```

```

406 *           : レジスタ設定値を取得します。
407 *           :
408 * Argument   : T_SCIF_bit_rate_type i_type_bps :
409 *           : ビットレート指定値
410 * Return Value : unsigned char : ビットレートレジスタ設定値
411 * Calling Functions :
412 * "FUNC COMMENT END"*****/
413 unsigned char scif_get_SCBRR(T_SCIF_bit_rate_type i_type_bps)
414 {
415     return      gc_scif_cks_scbrr_tbl[i_type_bps].mscbrr;
416 }
417
418 /*"FUNC COMMENT"*****
419 * ID           :
420 * Outline      : 送信データ数トリガ設定値取得処理
421 * Include      :
422 * Declaration  : unsigned char scif_get_snd_trigger_num
423 *              : (T_SCIF_send_fifo_trigger i_trigger_num)
424 * Description  : 引数で指定したデータ数トリガに対応する
425 *              : SCFCR.TTRG 設定値を取得する。
426 *              :
427 * Argument     : T_SCIF_send_fifo_trigger i_trigger_num
428 *              : データ数トリガ
429 * Return Value : unsigned char :
430 *              : レジスタ設定値(SCFCR.TTRG に設定)
431 * Calling Functions :
432 * "FUNC COMMENT END"*****/
433 unsigned char scif_get_snd_trigger_num(T_SCIF_send_fifo_trigger i_trigger_num)
434 {
435     unsigned char  ret = 0;
436     int            i = 0;
437     int            count = 0;
438
439     count = sizeof(gc_scif_ttrg_tbl) / sizeof(T_SCIF_SCFCR_TTRG_SET);
440
441     for(i = 0; i < count; i++)
442     {
443         if(gc_scif_ttrg_tbl[i].mtrigger_num == i_trigger_num)
444         {
445             /* ==== SCFCR.TTRG 設定 ==== */
446             ret = gc_scif_ttrg_tbl[i].mttrg;
447             break;
448         }
449     }
450
451     return ret;
452 }
453
454 /*"FUNC COMMENT"*****
455 * ID           :
456 * Outline      : 受信データ数トリガ設定値取得処理
457 * Include      :
458 * Declaration  : unsigned char scif_get_rcv_trigger_num
459 *              : (T_SCIF_rcv_fifo_trigger i_trigger_num)
460 * Description  : 引数で指定したデータ数トリガに対応する
461 *              : SCFCR.RTRG 設定値を取得する。
462 *              :
463 * Argument     : T_SCIF_rcv_fifo_trigger i_trigger_num :

```

```

464 *           : データ数トリガ
465 * Return Value       : unsigned char :
466 *           : レジスタ設定値(SCFCR.RTRG に設定)
467 * Calling Functions   :
468 * "FUNC COMMENT END"*****/
469 unsigned char scif_get_rcv_trigger_num(T_SCIF_rcv_fifo_trigger i_trigger_num)
470 {
471     unsigned char    ret = 0;
472     int              i = 0;
473     int              count = 0;
474
475     count = sizeof(scif_rtrg_tbl) / sizeof(T_SCIF_SCFCR_RTRG_SET);
476
477     for(i = 0; i < count; i++)
478     {
479         if(scif_rtrg_tbl[i].mtrigger_num == i_trigger_num)
480         {
481             /* ==== SCFCR.RTRG 設定 ==== */
482             ret = scif_rtrg_tbl[i].mrtrg;
483             break;
484         }
485     }
486
487     return ret;
488 }
489 }
490
491 /*"FUNC COMMENT"*****
492 * ID           :
493 * Outline      : 1 ビット期間経過待ち処理
494 * Include      :
495 * Declaration  : void scif_wait_lbit(
496 *               : unsigned long i_bit_rate)
497 *               :
498 * Description  : 指定したビットレートに対応した
499 *               : 1 ビット期間以上の十分な時間経過後に
500 *               : 本関数を扱います。
501 *               :
502 *               : gc_scif_cks_scbrn_tbl テーブルに設定され
503 *               : ている cpu 実行回数分 nop 処理を行い、
504 *               : 1 ビット期間以上の経過を待ちます。
505 *               :
506 * Limitation   : I を変更した場合は、
507 *               : gc_scif_cks_scbrn_tbl テーブルの設定値
508 *               : も修正ください。
509 *               :
510 * Argument     : long i_bit_rate : 動作ビットレート
511 *               :
512 * Return Value : none
513 * Calling Functions :
514 * "FUNC COMMENT END"*****/
515 static void scif_wait_lbit(unsigned long i_bit_rate)
516 {
517     unsigned long    cpu_count;
518     unsigned long    i;
519
520     /* 1 ビット期間以上の十分な待ち時間分の cpu 実行回数を取得 */
521     cpu_count = gc_scif_cks_scbrn_tbl[i_bit_rate].cpu_count_lbit;

```

```
522
523     for(i = 0; i < cpu_count; i++)
524     {
525         nop();
526     }
527
528     return;
529 }
530
531 /* End of File */
```

(4) サンプルプログラムリスト "intprg.c"

DMAC 転送終了割り込み、SCIF の受信エラー割り込みに対応する処理を定義します。

```
1  /*
2  以下の INTC_RESPONSEWAIT は、周辺モジュールの割り込み応答時間待ち用の値です。
3  int_responstime_wait() にパラメータ指定して使用ください。
4  int_responstime_wait() については、common.src を参照ください。
5
6  INT response wait Pck 5cycle
7  H'0E >= (1/Pcyc*5cyc)/(1/Icyc*3cyc)
8
9  注)
10 I =266.66MHz、P =33.33MHz 動作時の設定となっています。
11 これらを変更する場合はこちらの設定値も見直してください。
12 */
13 #define INTC_RESPONSEWAIT (0x0000000E)
14
15 ..途中省略..
16
17 /* H'800 DMAC DEI0 ; Illegal FPU -> Dummy Code H'880 */
18 void INT_DMACEI0(void)
19 {
20     unsigned long dummy;
21
22     /* DMA0 転送完了フラグ設定処理 */
23     dmac0_set_trans_end_flg();
24
25     /* DMACE0 の停止 */
26     dmac_stop(&DMACE0.CHCR);
27
28     /* CHCR レジスタ TE ビット (割り込み要求フラグ) クリア */
29     DMACE0.CHCR &= ~0x00000002;
30
31     /* 更新したはずの割り込み要因による割り込みの受け付けを避ける対応 */
32     dummy = DMACE0.CHCR;
33     int_responstime_wait(INTC_RESPONSEWAIT); /* 優先順位判定時間待ち */
34
35 }
36
37 /* H'820 DMAC DEI1 ; Illegal slot FPU -> Dummy Code H'8A0 */
38 void INT_DMACEI1(void)
39 {
40     unsigned long dummy;
41
42     /* DMA1 転送完了フラグ設定処理 */
43     dmac1_set_trans_end_flg();
44
45     /* DMACE1 の停止 */
46     dmac_stop(&DMACE1.CHCR);
47
48     /* CHCR レジスタ TE ビット (割り込み要求フラグ) クリア */
49     DMACE1.CHCR &= ~0x00000002;
50
51     /* 更新したはずの割り込み要因による割り込みの受け付けを避ける対応 */
52     dummy = DMACE1.CHCR;
53     int_responstime_wait(INTC_RESPONSEWAIT); /* 優先順位判定時間待ち */
54
55 }
56
```

```
57 ...途中省略...
58
59 /* H'C20 SCIF SCIF11 */
60 void INT_SCIF_SCIF11(void)
61 {
62     unsigned short dummy;
63
64     /* 受信時エラーフラグ対応処理 */
65     /* 本応用例では、SCSCR.RIE を許可にせず、SCSCR.REIE のみを有効にしています。
66        よって、受信エラー割り込み (ERI) 要求 or ブレーク割り込み (BRI) 要求
67        が発生した際にのみ本関数はコールされます。
68     */
69     scif_rcv_error();
70
71     /* 更新したはずの割り込み要因による割り込みの受け付けを避ける対応 */
72     dummy = SCIF1.SCF11.WORD;
73     int_responstime_wait(INTC_RESPONSEWAIT); /* 優先順位判定時間分待ち */
74
75 }
```

(5) サンプルプログラムリスト "vecttbl.src"

DMAC、SCIF に関連する割り込み実行時の割り込み優先度を設定しています。

DMAC0、DMAC1、SCIF1 に関連する割り込みの優先度を 1 に設定しているため、DMAC0、DMAC1、SCIF1 に関連する割り込み中に新たな DMAC0、DMAC1、SCIF1 に関連する割り込みが発生しないように優先度に 1 を設定しています。

```
1  ...途中省略...
2
3  ;DMAC(1)
4          ;H'800      DMAC DEI0
5          .data.b    H'10
6          ;H'820      DMAC DEI1
7          .data.b    H'10
8
9
10 ...途中省略...
11
12          ;H'C20     SCIF SCIF11
13          .data.b    H'10
14
15 ...途中省略...
```

(6) サンプルプログラムリスト "dmac.h"

```
1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. 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 * TECHNOLOGY CORP. 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 THE 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. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name : dmac.h
32 * Abstract : SH7730 DMAC 内蔵モジュールへの転送例
33 * Version : Ver 1.00
34 * Device : SH7730
35 * Tool-Chain : High-performance Embedded Workshop (Version 4.04.01.001)
36 * : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description : SH7730 DMAC 内蔵モジュールへの転送例のサンプルプログラムです。
40 * :
41 * Operation :
42 * Limitation :
43 * :
44 *****/
45 * History : 26.JUNE.2009 Ver. 1.00 First Release
46 /*"FILE COMMENT END"*****
47
48 /* ==== マクロ定義 ==== */
49 #define D_DMAM_SDRAM_SRC_ADR ((void *)0xac10000) /* 転送元の外部 SDRAM の先頭アドレス */
50 /* 【注】
51 本参考プログラムでは、上記マクロを利用して SDRAM 領域の初期化を行っています。
52 そのためキャッシュにアクセスしないようにキャッシング不可空間のアドレスを指定してください。
53 */
54 #define D_DMAM_IL_DST_ADR ((void *)0xe520300) /* 転送先の IL メモリ の先頭アドレス */
55
56 #define D_DMAM_TRANS_SIZE 128 /* 転送サイズ(128 バイト) */
57
```

```
58 /* dmac_get_tcr()で使用 */
59 #define D_DMACE_DMA_SIZE_BYTE 0x0000u
60 #define D_DMACE_DMA_SIZE_WORD 0x0001u
61 #define D_DMACE_DMA_SIZE_LONG 0x0002u
62 #define D_DMACE_DMA_SIZE_LONGx2 0x0003u
63 #define D_DMACE_DMA_SIZE_LONGx4 0x0004u
64 #define D_DMACE_DMA_SIZE_LONGx8 0x0005u
65 #define D_DMACE_DMA_SIZE_LONGx2x2 0x0006u
66 #define D_DMACE_DMA_SIZE_LONGx4x2 0x0007u
67
68 /* DMA 転送状態 */
69 typedef enum
70 {
71     D_DMACE_TRANS_INIT, /* DMA 転送初期状態、転送中状態 */
72     D_DMACE_TRANS_END /* DMA 転送完了状態 */
73 } T_Dmac_Trans_Chk;
74
75
76 /* ==== 関数宣言 ==== */
77 void dmac0_init(void);
78 void dmac1_init(void);
79 void dmac0_start(void);
80 void dmac1_start(void);
81 void dmac_stop(unsigned long *io_pDMACn_CHCR);
82 void dmac0_set_trans_end_flg(void);
83 void dmac1_set_trans_end_flg(void);
84 T_Dmac_Trans_Chk dmac0_get_trans_end_flg(void);
85 T_Dmac_Trans_Chk dmac1_get_trans_end_flg(void);
```

(7) サンプルプログラムリスト "scif.h"

```
1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. 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 * TECHNOLOGY CORP. 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 THE 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. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name : scif.h
32 * Abstract : SH7730 DMAC 内蔵モジュールへの転送例
33 * Version : Ver 1.00
34 * Device : SH7730
35 * Tool-Chain : High-performance Embedded Workshop (Version 4.04.01.001)
36 * : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description : SH7730 DMAC 内蔵モジュールへの転送例のサンプルプログラムです。
40 * :
41 * Operation :
42 * Limitation :
43 * :
44 *****/
45 * History : 26.JUNE.2009 Ver. 1.00 First Release
46 /*"FILE COMMENT END"*****
47
48 #ifndef __SCIF_DEF_H__
49 #define __SCIF_DEF_H__
50
51 /* ==== マクロ定義 ==== */
52
53 /* 戻り値 */
54 #define D_SCIF_RET_NG -1
55 #define D_SCIF_RET_OK 0
56
57 /* FIFO 設定可能数 */
```

```

58 #define D_SCIF_FIFO_NUM      16
59
60 /* 動作ビットレートの指定 */
61 /* ビットレートを変更する場合は T_SCIF_bit_rate_type から指定してください。 */
62 #define D_SCIF_SCBRR_ACT D_SCIF_SCBRR_115200
63
64 /* FIFO データ数トリガ */
65 /* 動作させたい FIFO のトリガ数を以下の定義している */
66 /* デファインから設定してください */
67 #define D_SCIF_DATA_NUM_SND_TRIGGER_CH0 D_SCIF_SEND_FIFO_TRIGGER0 /* 設定可能値 0,2,4,8 */
68 #define D_SCIF_DATA_NUM_RCV_TRIGGER_CH1 D_SCIF_RCV_FIFO_TRIGGER1 /* 設定可能値 1,4,8,14 */
69 */
70
71 /* 送信 FIFO データ数トリガ */
72 typedef enum
73 {
74     D_SCIF_SEND_FIFO_TRIGGER0 = 0, /* 送信 FIFO データトリガ数 0 */
75     D_SCIF_SEND_FIFO_TRIGGER2 = 2, /* 送信 FIFO データトリガ数 2 */
76     D_SCIF_SEND_FIFO_TRIGGER4 = 4, /* 送信 FIFO データトリガ数 4 */
77     D_SCIF_SEND_FIFO_TRIGGER8 = 8 /* 送信 FIFO データトリガ数 8 */
78 } T_SCIF_send_fifo_trigger;
79
80 /* 受信 FIFO データ数トリガ */
81 typedef enum
82 {
83     D_SCIF_RCV_FIFO_TRIGGER1 = 1, /* 受信 FIFO データトリガ数 1 */
84     D_SCIF_RCV_FIFO_TRIGGER4 = 4, /* 受信 FIFO データトリガ数 4 */
85     D_SCIF_RCV_FIFO_TRIGGER8 = 8, /* 受信 FIFO データトリガ数 8 */
86     D_SCIF_RCV_FIFO_TRIGGER14 = 14 /* 受信 FIFO データトリガ数 14 */
87 } T_SCIF_rcv_fifo_trigger;
88
89 /* ビットレート設定タイプ */
90 typedef enum
91 {
92     D_SCIF_SCBRR_1200,
93     D_SCIF_SCBRR_2400,
94     D_SCIF_SCBRR_4800,
95     D_SCIF_SCBRR_9600,
96     D_SCIF_SCBRR_19200,
97     D_SCIF_SCBRR_31250,
98     D_SCIF_SCBRR_38400,
99     D_SCIF_SCBRR_57600,
100    D_SCIF_SCBRR_115200
101 } T_SCIF_bit_rate_type;
102
103 /* SCIF 受信エラー状態 */
104 typedef enum
105 {
106     D_SCIF_RCV_NO_ERROR = 0, /* エラーなし */
107     D_SCIF_RCV_ERROR = 1 /* エラーあり */
108 } T_SCIF_rcv_error;
109
110 /* ==== 構造体定義 ==== */
111 /* SCBRR 設定情報 */
112 typedef struct {
113     unsigned char mscbrr;
114     unsigned char mscsmr_cks;
115     unsigned long cpu_count_lbit;

```

```
116 } T_SCIF_CKS_SCBRR_SET_INFO;
117
118 /* SCFCR.TTRG 設定用 */
119 typedef struct {
120     unsigned char    mtrigger_num;
121     unsigned char    mttrg;
122 } T_SCIF_SCFCR_TTRG_SET;
123
124 /* SCFCR.RTRG 設定用 */
125 typedef struct {
126     unsigned char    mtrigger_num;
127     unsigned char    mrtrg;
128 } T_SCIF_SCFCR_RTRG_SET;
129
130 /* ==== 関数宣言 ==== */
131 void scif_init_SCIF0(void);
132 void scif_init_SCIF1(void);
133 void scif_rcv_error();
134 T_SCIF_rcv_error scif_get_rcv_error(void);
135 void dmac0_chk_trans_end(void);
136 void dmac1_chk_trans_end(void);
137
138 #endif /* __SCIF_DEF_H__ */
```

4. 実行結果

上記参考プログラムの実行結果については、転送元 (SDRAM) のデータと転送先 (IL メモリ) データを比較処理するループ処理を抜けているため、DMAC の内蔵周辺モジュールリクエストと SCIF の機能を利用したデータ転送処理が問題なくできていることが確認できます。

また、図 14 のように、転送先 (IL メモリ) データを High-performance Embedded Workshop で出力しても DMAC の内蔵周辺モジュールリクエストと SCIF の機能を利用した転送処理が問題なくできていることが確認できます。

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII
E5203000	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUU
E5203010	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUU
E5203020	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUU
E5203030	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUU
E5203040	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUU
E5203050	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUU
E5203060	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUU
E5203070	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUU
E5203080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E5203090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E52030A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E52030B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

図 14 転送先メモリ (IL メモリ) の設定値

5. 参考ドキュメント

- ソフトウェアマニュアル
SH-4A ソフトウェアマニュアル (RJJ09B0090)
(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
SH7730 グループ ハードウェアマニュアル (RJJ09B0339)
(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 エレクトロニクスホームページ

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

お問合せ先

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

すべての商標および登録商標は、それぞれの所有者に帰属します。

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・電話番号は変更することがあります。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>