

RZ/A1H グループ

SSI サンプルプログラム

R01AN2076JJ0101
Rev.1.01
2014.7.16

要旨

本アプリケーションノートでは、RZ/A1H のシリアルサウンドインタフェース(以下、SSI)機能を使用し Audio Codec(WM8978)と全二重通信を行うサンプルプログラムについて説明します。

SSI プログラムの特長を以下に示します。

- ・ SSI をマスタトランシーバモードに設定し、Audio Codec(WM8978)と全二重通信を行います。
- ・ 再生/停止の例として、5 秒毎に SSI の動作(動作開始、停止)切り替えを行います。

対象デバイス

RZ/A1H

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. 仕様.....	4
2. 動作確認条件	5
3. 関連アプリケーションノート	6
4. 周辺機能説明	7
5. ハードウェア説明	8
5.1 ハードウェア構成例	8
5.2 使用端子一覧	8
6. ソフトウェア説明	9
6.1 動作概要	9
6.2 メモリマップ	10
6.2.1 サンプルプログラムのセクション配置	11
6.2.2 MMU の設定	14
6.2.3 例外処理ベクタテーブル	15
6.3 使用割り込み一覧	16
6.4 固定幅整数一覧	16
6.5 定数/エラーコード一覧	17
6.6 構造体/共用体一覧	23
6.7 変数一覧	25
6.8 関数一覧	28
6.9 関数仕様	29
6.9.1 R_SSI_RamInit	29
6.9.2 R_SSI_SsiInit	29
6.9.3 R_SSI_DmaInit	30
6.9.4 R_SSI_SsiStart	31
6.9.5 R_SSI_DmaStart	32
6.9.6 R_SSI_SsiStop	33
6.9.7 R_SSI_SsiDmaStop	34
6.9.8 R_SSI_SsiDmaAdsSet	35
6.10 フローチャート	36
6.10.1 メイン処理	36
6.10.2 メイン(SSI 動作中)処理	38
6.10.3 メイン(SSI 停止)処理	38
6.10.4 メイン(SSI 再開始)処理	39
6.10.5 DMA 転送終了割り込み処理	40
6.10.6 OS タイマ割り込み処理	40
6.10.7 SSI 管理情報の初期化処理	41
6.10.8 SSI 初期設定処理	42
6.10.9 DMAC 初期設定処理	43
6.10.10 SSI 起動処理	44
6.10.11 DMAC 起動処理	45
6.10.12 SSI 停止処理	46
6.10.13 SSI 送受信停止処理	47
6.10.14 バッファ登録処理	48
6.10.15 SSI エラー割り込み処理	50
6.10.16 DMA 管理情報の初期化処理	52
6.10.17 キュー初期化処理	52
6.10.18 キュー設定処理	53
6.10.19 DMA 転送許可処理	53
6.10.20 DMA 転送停止処理	54
6.10.21 DMA(バッファ→SSI)初期化処理	55

6.10.22	DMA 転送元アドレス更新処理.....	56
6.10.23	DMA(SSI→バッファ)初期化処理.....	57
6.10.24	DMA 転送先アドレス更新処理.....	58
6.10.25	DMA 転送終了処理.....	59
7.	サンプルコード.....	60
8.	参考ドキュメント.....	60

1. 仕様

表 1.1 に使用する周辺機能と用途を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
SSI チャンネル 0	SSI 送信および受信
NTC (割り込み ID : 172)	SSI エラー(送信アンダフローエラー、受信オーバーフローエラー)検知
INTC (割り込み ID : 41)	DMA 転送終了(SSI→バッファ)検知
INTC (割り込み ID : 42)	DMA 転送終了(バッファ→SSI)検知
INTC (割り込み ID : 135)	再生停止の動作切り替えタイミング生成
DMAC チャンネル 0	オーディオデータの転送(SSI→バッファ)
DMAC チャンネル 1	オーディオデータの転送(バッファ→SSI)
OSTM チャンネル 1	再生/停止の動作切り替えのための時間計測用
RSPI チャンネル 4	Audio Codec (WM8978)の制御

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2.1 動作確認条件

項目	内容
使用マイコン	RZ/A1H
動作周波数	CPU クロック (I ϕ) : 400MHz 画像処理クロック (G ϕ) : 266.67MHz 内部バスクロック (B ϕ) : 133.33MHz 周辺クロック 1 (P1 ϕ) : 66.67MHz 周辺クロック 0 (P0 ϕ) : 33.33MHz
動作電圧	電源電圧 (I/O) : 3.3V 電源電圧 (内部) : 1.18V
統合開発環境	ARM [®] 統合開発環境 ARM Development Studio 5 (DS-5 [™]) Version 5.16
C コンパイラ	ARM C/C++ Compiler/Linker/Assembler Ver.5.03 [Build 102] コンパイラオプション(ディレクトリパスの追加は除く) -O3 -Ospace --cpu=Cortex-A9 --littleend --arm --apcs=/interwork --no_unaligned_access --fpu=vfpv3_fp16 -g --asm
動作モード	ブートモード 0 (CS0 空間 16 ビットブート)
使用ボード	・ R7S72100 CPU ボード : RTK772100BC00000BR
使用デバイス (ボード上で使用する機能)	<R7S72100 CPU ボード : RTK772100BC00000BR 上> ・ システム設定用ディップスイッチ SW2-2 機能設定 : ON (P4_[7:4]を SSIF 端子として使用) ・ ジャンパ JP10 : 1-2 Short(P4_1 を SSL40 出力端子として使用)

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- RZ/A1H グループ 初期設定例 (R01AN1864JJ)
- RZ/A1H グループ レジスタ定義ヘッダ・ファイル iodef.h (R01AN1860JJ)
- RZ/A1H グループ ルネサスシリアルペリフェラルインタフェース EEPROM リードライト例 (R01AN1806JJ)

4. 周辺機能説明

SSI、DMAC、INTC、RSPI、OSTM についての基本的な内容は、RZ/A1H グループ・ユーザーズマニュアルハードウェア編に記載しています。

5. ハードウェア説明

5.1 ハードウェア構成例

図 5.1 にハードウェア構成例を示します。

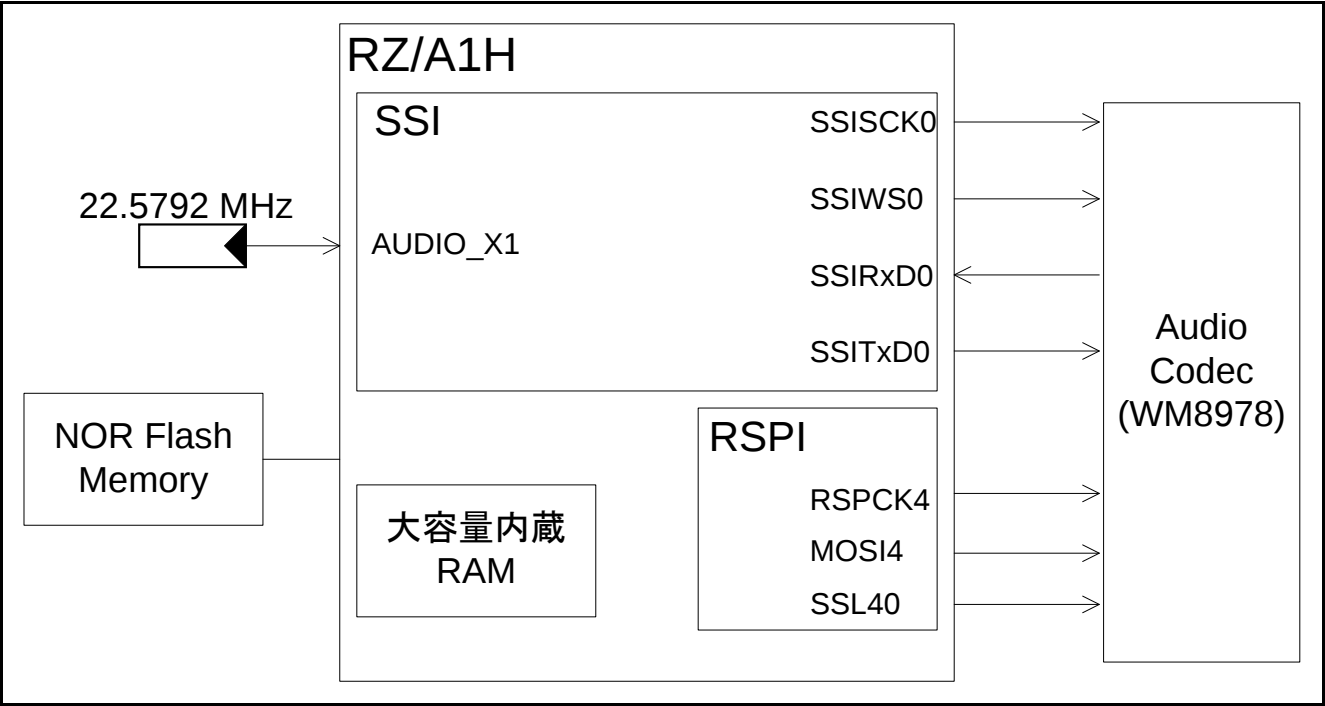


図 5.1 ハードウェア構成例

5.2 使用端子一覧

表 5.1 に使用端子と機能を示します。

表 5.1 使用端子と機能

端子名	入出力	内容
AUDIO_CLK	入力	未使用
AUDIO_X1	入力	オーバサンプルクロック入力
AUDIO_X2	出力	未使用
SSISCK0	出力	シリアルビットクロック
SSIWS0	出力	ワード選択
SSITxD0	出力	シリアルデータ出力
SSIRxD0	入力	シリアルデータ入力
RSPCK4	出力	クロック出力
MOSI4	出力	マスタ送出データ
SSL40	出力	スレーブセレクト

6. ソフトウェア説明

6.1 動作概要

図 6.1 にサンプルコードの動作概要図を示します。SSIRxD0 端子から受信したオーディオデータは SSI チャンネル 0 のレシーブデータレジスタ (SSIRDR) を経由して、受信 FIFO データレジスタ (SSIFRDR) に転送されます。SSIFRDR レジスタにデータが転送されると、SSI は受信データフル割り込みによる DMAC 転送要求を使用して DMAC チャンネル 0 を起動します。DMAC チャンネル 0 の転送先には 441 サンプル (1764byte) 分のバッファ (大容量内蔵 RAM) を設定しています。バッファは 4 面構成で、DMA 転送完了ごとに切り替えながら連続受信を行います。

SSI の送信データエンプティ割り込みによる DMAC 転送要求を使用して DMAC チャンネル 1 を起動し、バッファ (大容量内蔵 RAM) に保存したオーディオデータを SSI チャンネル 0 の送信 FIFO データレジスタ (SSIFTDR) へ転送します。SSIFTDR レジスタに転送されたデータは、トランスミットデータレジスタ (SSITDR) の空きを検出すると SSITDR レジスタに転送され、SSITxD0 端子から出力されます。バッファ 1 面の転送が完了する度にバッファを切り替えながら連続送信を行います。

上記の全二重通信を 5 秒間行い、その後、全二重通信を 5 秒間停止します。以後、その処理を繰り返します。

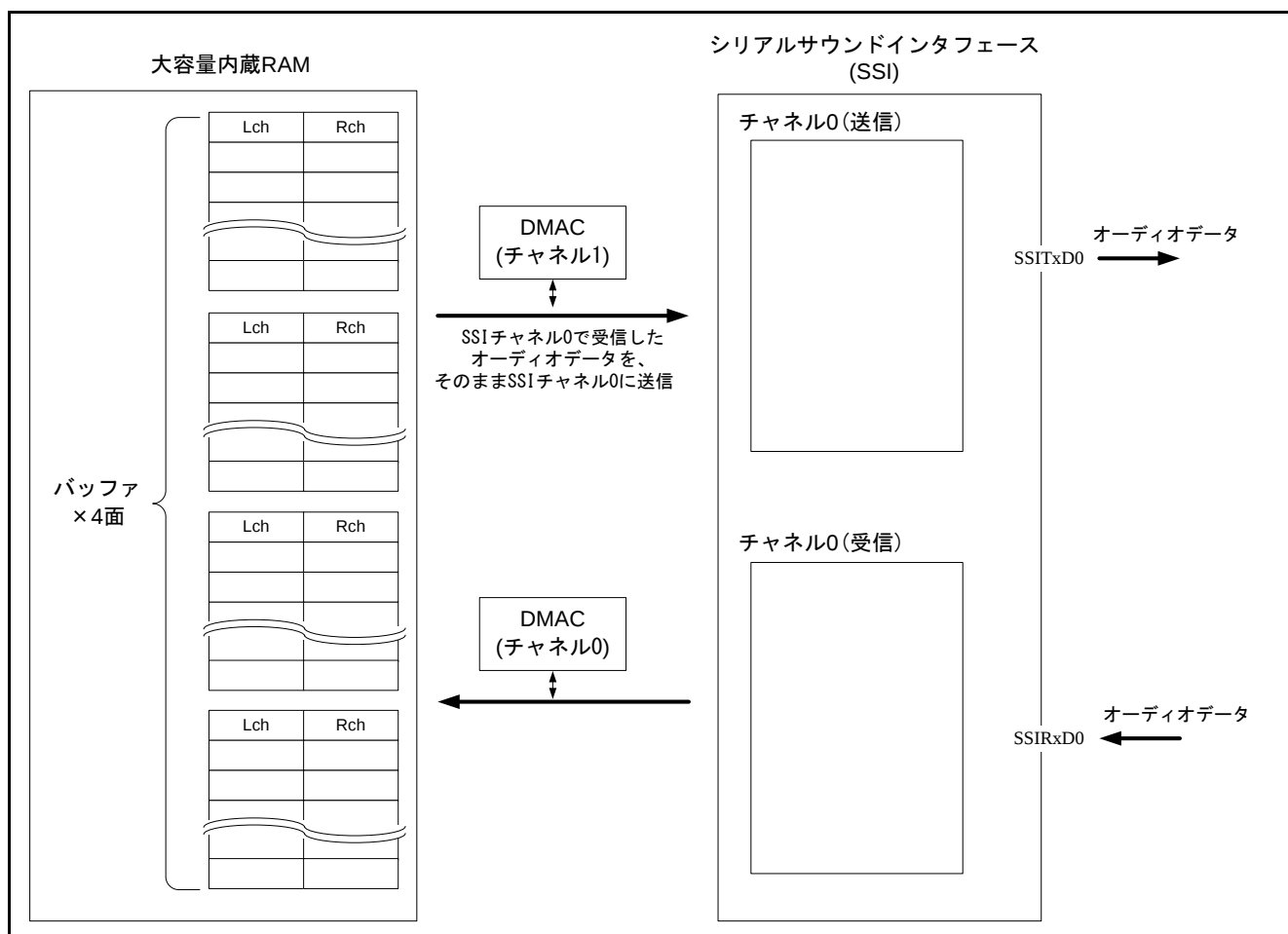


図 6.1 動作概要

6.2 メモリマップ

図 6.2 メモリマップに、RZ/A1H グループのアドレス空間と R7S72100 CPU ボード(RTK772100BC00000BR)のメモリマッピングを示します。

参考プログラムでは、ROM 領域を使用するコードおよびデータを CS0 空間に接続した NOR フラッシュメモリに配置し、RAM 領域を使用するコードおよびデータを大容量内蔵 RAM に配置するようにしています。

		RZ/A1H グループの アドレス空間	R7S72100 CPU ボード メモリマッピング
ミラー空間	H'FFFF FFFF	その他 (2550MB)	その他 (2550MB)
	H'60A0 0000	大容量内蔵 RAM (10MB)	大容量内蔵 RAM ミラー空間
	H'6000 0000	SPI マルチ I/O バス 空間 2 (64MB)	SPI マルチ I/O バス ミラー空間 2
	H'5C00 0000	SPI マルチ I/O バス 空間 1 (64MB)	SPI マルチ I/O バス ミラー空間 1
	H'5800 0000	CS5 空間 (64MB) CS4 空間 (64MB)	CS5 ミラー空間 CS4 ミラー空間
	H'5000 0000	CS3 空間 (64MB)	CS3 ミラー空間
	H'4C00 0000	CS2 空間 (64MB)	CS2 ミラー空間
	H'4800 0000	CS1 空間 (64MB)	CS1 ミラー空間
	H'4400 0000	CS0 空間 (64MB)	CS0 ミラー空間
	H'4000 0000	その他 (502MB)	その他 (502MB)
	H'20A0 0000	大容量内蔵 RAM (10MB)	大容量内蔵 RAM (10MB)
	H'2000 0000	SPI マルチ I/O バス 空間 2 (64MB)	シリアルフラッシュ メモリ (64MB)
	H'1C00 0000	SPI マルチ I/O バス 空間 1 (64MB)	シリアルフラッシュ メモリ (64MB)
	H'1800 0000	CS5 空間 (64MB) CS4 空間 (64MB)	ユーザ領域
通常空間	H'1000 0000	CS3 空間 (64MB)	SDRAM (64MB)
	H'0C00 0000	CS2 空間 (64MB)	SDRAM (64MB)
	H'0800 0000	CS1 空間 (64MB)	NOR フラッシュ メモリ (64MB)
	H'0400 0000	CS0 空間 (64MB)	NOR フラッシュ メモリ (64MB)
	H'0000 0000		

図 6.2 メモリマップ

6.2.1 サンプルプログラムのセクション配置

サンプルプログラムでは、割り込み処理の高速化のため、例外処理ベクタテーブルと IRQ 割り込みハンドラを大容量内蔵 RAM 上に配置して、これらの処理を大容量内蔵 RAM 上で実行するようにしています。例外処理ベクタテーブルおよび IRQ 割り込みハンドラのプログラムコードの NOR フラッシュメモリ領域から大容量内蔵 RAM 領域への転送処理、初期値なしデータセクションのゼロクリア処理、および初期値ありデータセクションの初期化処理は、スキッタローディング機能を使用して行っています。スキッタローディング機能の詳細は、ARM より提供される「ARM コンパイラツールチェーン／リンクの使用／イメージの構造と生成の章」を参照してください。

表 6.1 および表 6.2 に参考プログラムで使用するセクションを示し、図 6.3 にサンプルプログラムの初期状態のセクション配置 (ロードビュー) と、スキッタローディング機能を使用後のセクション配置 (実行ビュー) を示します。

表 6.1 使用するセクション

領域の名前	内容	タイプ	ロード領域	実行領域
VECTOR_TABLE	例外処理ベクタテーブル	Code	FALSH	FLASH
RESET_HANDLER	リセットハンドラ処理のプログラムコード領域 この領域は以下のセクションから構成されています ・ INITCA9CACHE (L1 キャッシュ設定) ・ INIT_TTB (MMU 設定) ・ RESET_HANDLER (リセットハンドラ)	Code	FALSH	FLASH
CODE_BASIC_SETUP	動作周波数とフラッシュメモリ最適化のためのプログラムコード領域	Code	FLASH	FLASH
InRoot	この領域は C 標準ライブラリなどのルート領域に配置するセクションから構成されています	Code および RO Data	FLASH	FLASH
CODE_FPU_INIT	NEON および VFP 初期設定のプログラムコード領域 この領域は以下のセクションから構成されています ・ CODE_FPU_INIT ・ FPU_INIT	Code	FLASH	FLASH
CODE_RESET	ハードウェア初期設定のプログラムコード領域 この領域は以下のセクションから構成されています ・ CODE_RESET (スタートアップ処理) ・ INIT_VBAR (ベクタベース設定)	Code	FLASH	FLASH
CODE_IO_REGRW	IO レジスタのリード/ライト関数のプログラムコード領域	Code	FLASH	FLASH
CODE	デフォルトのプログラムコード領域 C ソースでセクション名を定義しない Code タイプのセクションは、すべてこの領域に配置されます	Code	FLASH	FLASH
CONST	デフォルトの定数データ領域 C ソースでセクション名を定義しない RO Data タイプのセクションは、すべてこの領域に配置されます	RO Data	FLASH	FLASH

表 6.2 使用するセクション (2/2)

領域の名前	内容	タイプ	ロード領域	実行領域
VECTOR_MIRROR_TABLE	例外処理ベクタテーブル (大容量内蔵 RAM に転送して実行するためのセクション)	Code	FALSH	LRAM
CODE_HANDLER_JMPTBL	IRQ 割り込みハンドラのユーザ定義関数のプログラムコード領域	Code	FLASH	LRAM
CODE_HANDLER	IRQ 割り込みハンドラのプログラムコード領域 この領域は以下のセクションから構成されています ・ CODE_HANDLER ・ IRQ_FIQ_HANDLER	Code	FALSH	LRAM
DATA_HANDLER_JMPTBL	IRQ 割り込みハンドラのユーザ定義関数の登録テーブルデータ領域	RW Data	FLASH	LRAM
ARM_LIB_STACK	アプリケーションスタック領域	ZI Data	—	LRAM
IRQ_STACK	IRQ モードのスタック領域	ZI Data	—	LRAM
FIQ_STACK	FIQ モードのスタック領域	ZI Data	—	LRAM
SVC_STACK	スーパーバイザ (SVC) モードのスタック領域	ZI Data	—	LRAM
ABT_STACK	アボート (ABT) モードのスタック領域	ZI Data	—	LRAM
TTB	MMU 変換テーブル領域	ZI Data	—	LRAM
ARM_LIB_HEAP	アプリケーションヒープ領域	ZI Data	—	LRAM
DATA	デフォルトの初期値ありデータ領域 C ソースでセクション名を定義しない RW Data タイプのセクションは、すべてこの領域に配置されます	RW Data	FLASH	LRAM
BSS	デフォルトの初期値なしデータ領域 C ソースでセクション名を定義しない ZI Data タイプのセクションは、すべてこの領域に配置されます	ZI Data	—	LRAM
BSS_AUDIO_OUT_BUF	オーディオデータを格納するセクションとなります。	ZI Data	—	LRAM

- 【注】 1. 表中のロード領域および実行領域において、FLASH は NOR フラッシュメモリの領域を、LRAM は大容量内蔵 RAM の領域を表します。
2. セクションの名前は基本的に領域と同じ名前にしていますが、RESET_HANDLER、InRoot、CODE_FPU_INIT、CODE_RESET、CODE、CONST、CODE_HANDLER、DATA、BSS の各領域は複数のセクションから構成されています。領域とセクションについては、ARM コンパイラツールチェーンのマニュアルを参照してください。

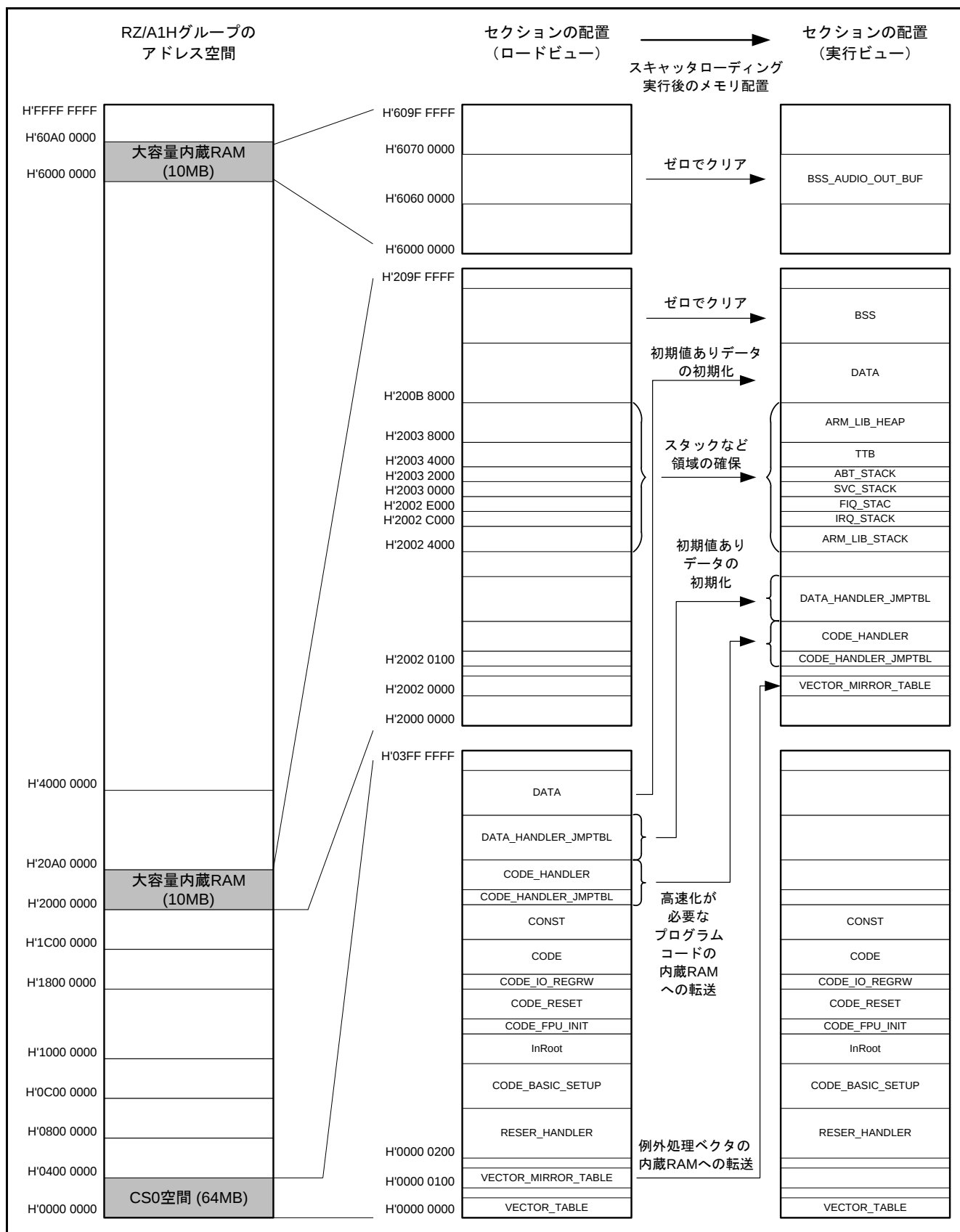


図 6.3 セクション配置

6.2.2 MMU の設定

RZ/A1H R7S72100 CPU ボード (RTK772100BC00000BR) で使用するハードウェア資源のメモリマップにあわせて、H'0000 0000 番地から 1MB 単位で 4GB の領域を MMU で管理するように設定しています (ttb_init.s ファイルで設定しています)。システムにあわせてカスタマイズする場合は、最少単位を 1MB としてください。

表 6.3 に、サンプルプログラムの MMU の設定を示します。

表 6.3 MMU の設定

定義名	内容	アドレス	サイズ	メモリタイプ
M_SIZE_NOR	CS0、CS1 空間 (NOR フラッシュメモリ)	H'0000 0000 ～ H'07FF FFFF	128MB	L1 キャッシュ有効、 ノーマルメモリ
M_SIZE_SDRAM	CS2、CS3 空間 (SDRAM)	H'0800 0000 ～ H'0FFF FFFF	128MB	L1 キャッシュ有効、 ノーマルメモリ
M_SIZE_CS45	CS4、CS5 空間	H'1000 0000 ～ H'17FF FFFF	128MB	ストロングリオーダメモリ (L1 キャッシュ無効)
M_SIZE_SPI	SPI マルチ IO バス空間 1 および 2 (シリアルフラッシュメモリ)	H'1800 0000 ～ H'1FFF FFFF	128MB	L1 キャッシュ有効、 ノーマルメモリ
M_SIZE_RAM	大容量内蔵 RAM 空間	H'2000 0000 ～ H'209F FFFF	10MB	L1 キャッシュ有効、 ノーマルメモリ
M_SIZE_IO_1	内蔵周辺モジュールおよび 予約エリア	H'20A0 0000 ～ H'3FFF FFFF	502MB	ストロングリオーダメモリ (L1 キャッシュ無効)
M_SIZE_NOR_M	CS0、CS1 ミラー空間	H'4000 0000 ～ H'47FF FFFF	128MB	L1 キャッシュ無効、 ノーマルメモリ
M_SIZE_SDRAM_M	CS2、CS3 ミラー空間	H'4800 0000 ～ H'4FFF FFFF	128MB	L1 キャッシュ無効、 ノーマルメモリ
M_SIZE_CS45_M	CS4、CS5 ミラー空間	H'5000 0000 ～ H'57FF FFFF	128MB	ストロングリオーダメモリ (L1 キャッシュ無効)
M_SIZE_SPI_M	SPI マルチ IO バス ミラー空間 1 および 2	H'5800 0000 ～ H'5FFF FFFF	128MB	L1 キャッシュ無効、 ノーマルメモリ
M_SIZE_RAM_M	大容量内蔵 RAM ミラー空間	H'6000 0000 ～ H'609F FFFF	10MB	L1 キャッシュ無効、 ノーマルメモリ
M_SIZE_IO_2	内蔵周辺モジュールおよび 予約エリア	H'60A0 0000 ～ H'FFFF FFFF	2550MB	ストロングリオーダメモリ (L1 キャッシュ無効)

6.2.3 例外処理ベクタテーブル

RZ/A1H には 7 種類の例外処理（リセット、未定義命令、ソフトウェア割り込み、プリフェッチアボート、データアボート、IRQ、FIQ）があり、ブートモード 0 の場合、リセット解除後に例外処理のベクタテーブルは H'0000 0000 番地から 32 バイトの領域（H'0000 0000 番地～H'0000 001F 番地）に配置されます。例外処理のベクタテーブルには、各例外処理への分岐命令を記述します。

図 6.4 に例外処理ベクタテーブルの記述例として、サンプルコードの例外処理ベクタテーブル内容を示します。

```
vector_table
LDR pc, =reset_handler      ; 0x0000_0000 : Reset exception
LDR pc, =undefined_handler  ; 0x0000_0004 : Undefined instructions exception
LDR pc, =svc_handler        ; 0x0000_0008 : Software interrupts exceptions
LDR pc, =prefetch_handler   ; 0x0000_000c : Prefetch abort exception
LDR pc, =abort_handler      ; 0x0000_0010 : Data abort exception
LDR pc, =reserved_handler   ; 0x0000_0014 : Reserved
LDR pc, =irq_handler        ; 0x0000_0018 : IRQ exception
LDR pc, =fiq_handler        ; 0x0000_001c : FIQ exception
```

図 6.4 例外処理ベクタテーブルの記述例

6.3 使用割り込み一覧

表 6.4 にサンプルコードで使用する割り込みを示します。

表 6.4 サンプルコードで使用する割り込み

割り込み(要因 ID)	優先度	処理概要
SSI0(172)	4	SSI エラー(送信アンダフローエラー、受信オーバーフローエラー)を検知すると割り込みが発生します。本割り込みにて SSI モジュールのソフトリセット処理を行い、SSI の再設定を行います。その後、SSI 送受信動作を行います。
DMAINT0(41)	5	オーディオデータ転送(SSI→バッファ)終了で割り込みが発生します。周期は約 10ms となり、本割り込みにて次バッファの先頭アドレスを DMAC Destination Address レジスタへ設定します。
DMAINT1 (42)	5	オーディオデータ転送(バッファ→SSI)終了で割り込みが発生します。周期は約 10ms となり、本割り込みにて次バッファの先頭アドレスを DMAC Source Address レジスタへ設定します。
OSTMI1 (135)	6	5 秒毎に割り込みが発生します。本割り込みをトリガに再生停止処理を行います。

6.4 固定幅整数一覧

表 6.5 にサンプルコードで使用する固定幅整数を示します。

表 6.5 サンプルコードで使用する固定幅整数

シンボル	内容
char_t	8 ビット文字
bool_t	論理型。値は true (1) , false (0)
int_t	高速な整数、符号あり、本サンプルコードでは 32 ビット整数。
int8_t	8 ビット整数、符号あり (標準ライブラリにて定義)
int16_t	16 ビット整数、符号あり (標準ライブラリにて定義)
int32_t	32 ビット整数、符号あり (標準ライブラリにて定義)
int64_t	64 ビット整数、符号あり (標準ライブラリにて定義)
uint8_t	8 ビット整数、符号なし (標準ライブラリにて定義)
uint16_t	16 ビット整数、符号なし (標準ライブラリにて定義)
uint32_t	32 ビット整数、符号なし (標準ライブラリにて定義)
uint64_t	64 ビット整数、符号なし (標準ライブラリにて定義)
float32_t	32 ビット浮動小数 ("__ARM_NEON__ "を指定時、標準ライブラリにて定義)
float64_t	64 ビット浮動小数 (標準ライブラリにて定義) ("__ARM_NEON__ "を指定時、標準ライブラリにて定義)
float128_t	128 ビット浮動小数

6.5 定数/エラーコード一覧

表 6.6 から表 6.10 にサンプルコードで使用する定数、表 6.11 にサンプルコードのエラーコードエラー! 参照元が見つかりません。を示します。

表 6.6 サンプルコードで使用する定数(1)

定数名	設定値	内容
AUDIO_BUFF_SIZE	1764	オーディオバッファバイトサイズ
SAMPLE_MODE_IDLE	0	アイドル状態
SAMPLE_MODE_PLAY	1	再生状態
SAMPLE_MODE_MAX	2	状態数
SAMPLE_EVENT_OFF	0	タイマイイベントなし
SAMPLE_EVENT_ON	1	タイマイイベントあり
SAMPLE_EVENT_MAX	2	タイマイイベント数
SSI_CHANNEL0	0	SSI チャンネル 0 本定数を用いて使用 SSI チャンネル番号を指定します。
SSI_CHANNEL1	1	SSI チャンネル 1 本定数を用いて使用 SSI チャンネル番号を指定します。
SSI_CHANNEL2	2	SSI チャンネル 2 本定数を用いて使用 SSI チャンネル番号を指定します。
SSI_CHANNEL3	3	SSI チャンネル 3 本定数を用いて使用 SSI チャンネル番号を指定します。
SSI_CHANNEL4	4	SSI チャンネル 4 本定数を用いて使用 SSI チャンネル番号を指定します。
SSI_CHANNEL5	5	SSI チャンネル 5 本定数を用いて使用 SSI チャンネル番号を指定します。
SSI_CHANNEL_MAX	6	SSI チャンネル数
SSI_RX	1	SSI 受信(SSI→バッファ)動作 本定数を用いて SSI 受信動作を指定します。
SSI_TX	2	SSI 送信(バッファ→SSI)動作 本定数を用いて SSI 送信動作を指定します。
SSI_FULLDUPLEX	3	SSI 全二重動作 本定数を用いて SSI 全二重動作を指定します。
SSI_ID_SSII0	172	SSI チャンネル 0 割り込み ID 本定数を用いて使用する SSI の割り込み ID を指定します。
SSI_ID_SSII1	175	SSI チャンネル 1 割り込み ID 本定数を用いて使用する SSI の割り込み ID を指定します。
SSI_ID_SSII2	178	SSI チャンネル 2 割り込み ID 本定数を用いて使用する SSI の割り込み ID を指定します。
SSI_ID_SSII3	180	SSI チャンネル 3 割り込み ID 本定数を用いて使用する SSI の割り込み ID を指定します。
SSI_ID_SSII4	183	SSI チャンネル 4 割り込み ID 本定数を用いて使用する SSI の割り込み ID を指定します。
SSI_ID_SSII5	185	SSI チャンネル 5 割り込み ID 本定数を用いて使用する SSI の割り込み ID を指定します。

表 6.7 サンプルコードで使用する定数(2)

定数名	設定値	内容
SSI_SSICR_TUIEN_INIT_VALUE	H'00000000	SSI コントロールレジスタの TUIEN ビットの初期設定値(送信アンダフロー割り込み禁止)
SSI_SSICR_TOIEN_INIT_VALUE	H'00000000	SSI コントロールレジスタの TOIEN ビットの初期設定値(送信オーバフロー割り込み禁止)
SSI_SSICR_RUIEN_INIT_VALUE	H'00000000	SSI コントロールレジスタの RUIEN ビットの初期設定値(受信アンダフロー割り込み禁止)
SSI_SSICR_ROIEN_INIT_VALUE	H'00000000	SSI コントロールレジスタの ROIEN ビットの初期設定値(受信オーバフロー割り込み禁止)
SSI_SSICR_IEN_INIT_VALUE	H'00000000	SSI コントロールレジスタの IEN ビットの初期設定値(アイドルモード割り込み禁止)
SSI_SSICR_MUEN_INIT_VALUE	H'00000000	SSI コントロールレジスタの MUEN ビットの初期設定値(ミュート状態でない)
SSI_SSICR_TEN_INIT_VALUE	H'00000000	SSI コントロールレジスタの TEN ビットの初期設定値(送信動作を禁止)
SSI_SSICR_REN_INIT_VALUE	H'00000000	SSI コントロールレジスタの REN ビットの初期設定値(受信動作を禁止)
SSI_SSICR0_CKS_VALUE	H'00000000	SSI チャンネル 0 コントロールレジスタの CKS ビットの 設定値(オーバサンプルクロック : AUDIO_X1) 本設定は drv_ssif_user.h に定義していますので、必要 に応じて変更してください
SSI_SSICR0_CHNL_VALUE	H'00000000	SSI チャンネル 0 コントロールレジスタの CHNL ビット の設定値(チャンネル : 各システムワードは 1 チャンネルで 構成) 本設定は drv_ssif_user.h に定義していますので、必要 に応じて変更してください
SSI_SSICR0_DWL_VALUE	H'00080000	SSI チャンネル 0 コントロールレジスタの DWL ビットの 設定値(データワード長 : 16 ビット) 本設定は drv_ssif_user.h に定義していますので、必要 に応じて変更してください
SSI_SSICR0_SWL_VALUE	H'00030000	SSI チャンネル 0 コントロールレジスタの SWL ビットの 設定値(システムワード長 : 32 ビット) 本設定は drv_ssif_user.h に定義していますので、必要 に応じて変更してください
SSI_SSICR0_SCKD_VALUE	H'00008000	SSI チャンネル 0 コントロールレジスタの SCKD ビット の設定値(シリアルビットクロック方向 : マスタモード) 本設定は drv_ssif_user.h に定義していますので、必要 に応じて変更してください
SSI_SSICR0_SWSD_VALUE	H'00004000	SSI チャンネル 0 コントロールレジスタの SWSD ビット の設定値(シリアル WS 方向 : マスタモード) 本設定は drv_ssif_user.h に定義していますので、必要 に応じて変更してください
SSI_SSICR0_SCKP_VALUE	H'00000000	SSI チャンネル 0 コントロールレジスタの SCKP ビット の設定値(シリアルビットクロック極性 : SSIWS と SSIDATA は SSISCK の立下りエッジで変化) 本設定は drv_ssif_user.h に定義していますので、必要 に応じて変更してください

表 6.8 サンプルコードで使用する定数(3)

定数名	設定値	内容
SSI_SSICR0_SWSP_VALUE	H'00000000	SSI チャンネル 0 コントロールレジスタの SWSP ビットの設定値(シリアル WS 極性: SSIWS は第 1 チャンネルではローレベル、第 2 チャンネルではハイレベル) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSICR0_SPDP_VALUE	H'00000000	SSI チャンネル 0 コントロールレジスタの SPDP ビットの設定値(シリアルパディング極性: ローレベル) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSICR0_SDТА_VALUE	H'00000000	SSI チャンネル 0 コントロールレジスタの SDТА ビットの設定値(シリアルデータアライメント: シリアルデータ、パディングビットの順に送受信) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSICR0_PDТА_VALUE	H'00000000	SSI チャンネル 0 コントロールレジスタの PDТА ビットの設定値(パラレルデータアライメント: パラレルデータ(SSITDR、SSIRDR)の下位側を先行して送受信) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSICR0_DEL_VALUE	H'00000000	SSI チャンネル 0 コントロールレジスタの DEL ビットの設定値(シリアルデータディレイ: SSIWS と SSIDATA 間で 1 クロックサイクルの遅延) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSICR0_CKDV_VALUE	H'00000030	SSI チャンネル 0 コントロールレジスタの CKDV ビットの設定値(シリアルオーバーサンプルクロック分周比: $AUDIO\phi/8$) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSICR1_CKS_VALUE	H'00000000	SSI チャンネル 1 コントロールレジスタの CKS ビットの設定値(オーバーサンプルクロック: $AUDIO_X1$) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSICR1_CHNL_VALUE	H'00000000	SSI チャンネル 1 コントロールレジスタの CHNL ビットの設定値(チャンネル: 各システムワードは 1 チャンネルで構成) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSICR1_DWL_VALUE	H'00080000	SSI チャンネル 1 コントロールレジスタの DWL ビットの設定値(データワード長: 16 ビット) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSICR1_SWL_VALUE	H'00030000	SSI チャンネル 1 コントロールレジスタの SWL ビットの設定値(システムワード長: 32 ビット) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください

表 6.9 サンプルコードで使用する定数(4)

定数名	設定値	内容
SSI_SSICR1_SCKD_VALUE	H'00008000	SSI チャンネル 1 コントロールレジスタの SCKD ビットの設定値(シリアルビットクロック方向: マスタモード) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSIFCR_TIE_INIT_VALUE	H'00000000	SSI FIFO コントロールレジスタの TIE ビットの初期設定値(送信インタラプトイネーブル: 送信データエンプティ割り込み (TXI) 要求を禁止)
SSI_SSIFCR_RIE_INIT_VALUE	H'00000000	SSI FIFO コントロールレジスタの RIE ビットの初期設定値(受信インタラプトイネーブル: 受信データフル割り込み (RXI) 要求を禁止)
SSI_SSIFCR_TFRST_INIT_VALUE	H'00000002	SSI FIFO コントロールレジスタの TFRST ビットの初期設定値(送信 FIFO データレジスタリセット: リセット動作を許可)
SSI_SSIFCR_RFRST_INIT_VALUE	H'00000001	SSI FIFO コントロールレジスタの RFRST ビットの初期設定値(受信 FIFO データレジスタリセット: リセット動作を許可)
SSI_SSIFCR0_TTRG_VALUE	H'00000000	SSI チャンネル 0 FIFO コントロールレジスタの TTRG ビットの初期設定値(送信データ数トリガ: 7) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSIFCR0_RTRG_VALUE	H'00000000	SSI チャンネル 0 FIFO コントロールレジスタの RTRG ビットの初期設定値(受信データ数トリガ: 1) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_SSITDMR_RXDMUTE_INIT_VALUE	H'00000000	SSI TDM モードレジスタの RXDMUTE ビットの初期設定値(受信ダイレクトデータ MUTE 設定: 受信したデータをそのまま出力します)
SSI_SSITDMR_TDM_INIT_VALUE	H'00000000	SSI TDM モードレジスタの TDM ビットの初期設定値(DM モード: TDM モードを禁止)
SSI_SSITDMR0_CONT_VALUE	H'00000000	SSI チャンネル 0 TDM モードレジスタの CONT ビットの初期設定値(WS コンティニューモード: WS コンティニューモードを禁止) 本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください
SSI_ERR_INT_LV	4	SSI エラー(送信アンダエラー、受信オーバーフローエラー)割り込みレベル
DMA_TRNEND_INT_LV	5	DMA 転送終了割り込みレベル
SSI_IDLE_WAIT	45	SSI 停止時のアイドル遷移待ち時間(フェールセーフ機能) SSI を停止するにあたり SSI モジュールがアイドルに遷移するまで待つ必要がありますが、アイドルへの遷移失敗を考慮し、フェールセーフの待ち時間を本定義で設定します。本設定は drv_ssif_user.h に定義していますので、必要に応じて変更してください。

表 6.10 サンプルコードで使用する定数(5)

定数名	設定値	内容
DMA_CHANNEL0	0	DMA チャンネル 0 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL1	1	DMA チャンネル 1 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL2	2	DMA チャンネル 2 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL3	3	DMA チャンネル 3 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL4	4	DMA チャンネル 4 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL5	5	DMA チャンネル 5 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL6	6	DMA チャンネル 6 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL7	7	DMA チャンネル 7 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL8	8	DMA チャンネル 8 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL9	9	DMA チャンネル 9 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL10	10	DMA チャンネル 10 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL11	11	DMA チャンネル 11 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL12	12	DMA チャンネル 12 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL13	13	DMA チャンネル 13 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL14	14	DMA チャンネル 14 本定数を用いて使用 DMA チャンネル番号を指定します。
DMA_CHANNEL15	15	DMA チャンネル 15 本定数を用いて使用 DMA チャンネル番号を指定します。
NO_SIDE	0	DMA Next0 レジスタを更新する場合、本定数を用いて更新指定を行います。
N1_SIDE	1	DMA Next1 レジスタを更新する場合、本定数を用いて更新指定を行います。
DMA_TRANSFER_SIZE_16	H'00000001	DMA 転送サイズを 16 ビット
DMA_TRANSFER_SIZE_32	H'00000002	DMA 転送サイズを 32 ビット
DMA_STOP_WAIT	200	DMA 停止時の DMA 停止待ち時間(フェールセーフ機能) DMA を停止するにあたり DMA モジュールが停止するまで待つ必要がありますが、DMA が停止出来なかった場合に備え、フェールセーフの待ち時間を本定義で設定します。本定義は、ユーザにより調整してください。

表 6.11 サンプルコードのエラーコード

定数名	設定値	内容
DEVDRV_SUCCESS	0	処理が成功した場合の戻り値
DEVDRV_ERROR	-1	処理が失敗した場合の戻り値
DEVDRV_QUEUE_EMPTY	-10	キューにバッファが登録されていない場合の戻り値
DEVDRV_QUEUE_FULL	-11	キューがフル(バッファをキューに登録できない)の場合の戻り値

6.6 構造体/共用体一覧

図 6.5 から図 6.7 にサンプルコードで使用する構造体/共用体を示します。

```
typedef struct{                                メイン処理で状態に応じて実行する関数ポインタ
    int32_t (* SampleFunc)(void); /* Function pointer */
} sample_ssif_play_t;
```

図 6.5 サンプルコードで使用する構造体/共用体(メイン)

```
typedef struct{                                SSI モジュールのレジスタアドレス
    volatile uint32_t *ssicr; /* control register(SSICR) */
    volatile uint32_t *ssifcr; /* FIFO control register(SSIFCR) */
    volatile uint32_t *ssisr; /* status register(SSISR) */
    volatile uint32_t *ssifsr; /* FIFO status register(SSIFSR) */
    volatile uint32_t *ssiftdr; /* Tx FIFO data register(SSIFTDR) */
    volatile uint32_t *ssifrdrr; /* Rx FIFO data register(SSIFRDR) */
    volatile uint32_t *ssitdmr; /* TDM mode register(SSITDMR) */
} ssif_reg_t;

typedef struct                                SSI モジュールで使用する DMAC チャネル番号
{
    uint32_t txdma_channel; /* DMA registration situation */
    uint32_t rxdma_channel; /* DMA registration situation */
} ssi_regist_info_t;
```

図 6.6 サンプルコードで使用する構造体/共用体(SSi モジュール)

```

typedef struct{
    volatile uint32_t *n0sa;      /* Next0 Source Address Register(N0SA) */
    volatile uint32_t *n0da;      /* Next0 Destination Address Register(N0DA) */
    volatile uint32_t *n0tb;      /* Next0 Transaction Byte Register(N0TB) */
    volatile uint32_t *n1sa;      /* Next1 Source Address Register(N1SA) */
    volatile uint32_t *n1da;      /* Next1 Destination Address Register(N1DA) */
    volatile uint32_t *n1tb;      /* Next1 Transaction Byte Register(N1TB) */
    volatile uint32_t *chstat;     /* Channel Status Register(CHSTAT) */
    volatile uint32_t *chctrl;     /* Channel Control Register(CHCTRL) */
    volatile uint32_t *chcfg;      /* Channel Configuration Register(CHCFG) */
    volatile uint32_t *chitvl;     /* Channel Interval Register(chitvl) */
    volatile uint32_t *chext;      /* Channel Extension Register(CHEXT) */
    volatile uint32_t *dstat_en;   /* DMA Status EN Register(DSTAT_EN) */
    volatile uint32_t *dctrl;      /* DMA Control Register(DCTRL) */
    volatile uint32_t *dmars;      /* DMA Expansion resource selector(DMARS) */
} dma_reg_t;

typedef struct {
    uint32_t direct;              /* Transfer direction */
    uint32_t module_name;         /* Name of transmit peripheral module */
    uint32_t module_channel;      /* Channel number of the DMA */
    bool_t transfer_continue;     /* Transfer continuation */
    fp_type user_trnend_func;     /* Callback interrupt function */
} dma_regist_info_t;

```

図 6.7 サンプルコードで使用する構造体/共用体(DMA モジュール)

6.7 変数一覧

表 6.12 と表 6.13 に static 型変数エラー! 参照元が見つかりません。を、表 6.14 に const 型変数エラー! 参照元が見つかりません。を示します。

表 6.12 static 型変数(1)

型	変数名	内容	使用関数
uint32_t	audio_out_buf	Audio Codec から受信したデータを格納するバッファです。Audio Codec へは、このバッファの受信したデータを送信します。 本バッファは 4 面構成で送受信が完了する度にバッファを切り替えながら連続送受信を行います。	Sample_Main Sample_ssif_StartPlay Sample_ssif_Play
uint32_t	mute_buf	ミュートデータを格納するバッファです。 Audio Codec にミュートデータを送信する場合、本バッファからミュートデータを送信します。	DMA_DmaInit DMA_DmaTxInit DMA_TxRegSet DMA_CheckDmaMute DmaInt
uint32_t	g_txbuf_side	audio_out_buf の送信面を管理している変数です。 送信面は、2 面→3 面→0 面→1 面→2 面→3 面→0 面・・・と送信面を切り替えながら送信処理を行います。	Sample_Set_TxbufSide Sample_Get_TxbufSide Sample_Updata_TxbufSide
uint32_t	g_rxbuf_side	audio_out_buf の受信面を管理している変数です。 受信面は、0 面→1 面→2 面→3 面→0 面→1 面→2 面・・・と受信面を切り替えながら受信処理を行います。	Sample_Set_RxbufSide Sample_Get_RxbufSide Sample_Updata_RxbufSide
uint32_t	sample_rx_flag	audio_out_buf の 1 面分(1764 バイト)にオーディオデータが転送終了したことを示す変数です。	Sample_Main Sample_ssif_StartPlay Sample_ssif_Play Sample_RxDMA_Interrupt
uint32_t	sample_timer_event	再生/停止動作の切り替え処理を本変数にて管理します。 OSTM 割り込み(5 秒)にて本変数をセットし、本変数はメイン側で確認します。 メイン処理は本変数に応じて再生停止を実行します。	Sample_Clear_eventFlag Sample_Get_eventFlag Sample_OSTM1_Interrupt

表 6.13 static 型変数(2)

ssi_regist_info_t	ssi_regist_info	SSI モジュールで使用する DMA チャネル番号を本変数で管理します。	R_SSI_RamInit R_SSI_DmaInit R_SSI_DmaStart R_SSI_SsiDmaStop R_SSI_SsiDmaAdsSet SsiErrInt
dma_regist_info_t	dma_regist	SSI に関連づけられた DMA モジュール管理情報を本変数で管理します。	DMA_DmaInit DMA_DmaAdsSet DMA_DmaStart DMA_DmaTxInit DMA_DmaRxInit DMA_TxRegSet DMA_RxRegSet DmaInt
queue_inf_t	queue	ユーザにて指定される送受信バッファの情報※を本変数で管理します。 ※送受信バッファの先頭アドレス、送受信バッファのバイトサイズ等の情報 DMA Next レジスタにセットする値は、本変数に格納されているバッファアドレス、転送サイズをセットします。	DMA_DmaAdsSet DMA_QueueInit DMA_DmaStop DMA_TxRegSet DMA_RxRegSet QueueSet DmaInt

表 6.14 const 型変数

型	変数名	内容	使用関数
ssif_reg_t	Ssif	各 SSI モジュールのレジスタアドレスを格納します。	R_SSI_Ssilnit R_SSI_SsiStart R_SSI_SsiStop R_SSI_SsiDmaAdsSet SsiErrInt
uint16_t	ssi_int_id	各 SSI モジュールの割り込み ID を格納します。	R_SSI_Ssilnit R_SSI_SsiStart R_SSI_SsiStop
uint32_t	ssi_ssicr_reg	各 SSI コントロールレジスタに設定する初期設定値を格納します。	R_SSI_Ssilnit R_SSI_SsiStop SsiErrInt
uint32_t	ssi_ssifcr_reg	各 SSI FIFO コントロールレジスタに設定する初期設定値を格納します。	R_SSI_Ssilnit R_SSI_SsiStop SsiErrInt
uint32_t	ssi_ssitdmr_reg	各 SSI TDM モードレジスタに設定する初期設定値を格納します。	R_SSI_Ssilnit R_SSI_SsiStop
uint8_reg_bit_inf_t	ssi_swrstcr1_bit_tbl	各 SSI モジュールのソフトウェアリセットビットのマスク、シフト量を格納します。	R_SSI_Ssilnit R_SSI_SsiStop SsiErrInt
uint32_reg_bit_inf_t	ssi_ssifcr_bit_tbl	SSI FIFO コントロールレジスタの TIE、RIE、TFRST、RFRST ビットのマスク、シフト量を格納します。	R_SSI_SsiStart R_SSI_SsiStop
uint32_reg_bit_inf_t	ssi_ssizr_bit_tbl	SSI ステータスレジスタの TUIRQ、ROIQR、IIRQ ビットのマスク、シフト量を格納します。	R_SSI_SsiStart R_SSI_SsiStop
uint32_reg_bit_inf_t	ssi_ssicr_bit_tbl	SSI コントロールレジスタの TUIEN、ROIEN、TEN、REN ビットのマスク、シフト量を格納します。	R_SSI_SsiStart R_SSI_SsiStop R_SSI_SsiDmaAdsSet SsiErrInt
dma_reg_t	g_dma_reg_tbl	各 DMA モジュールのレジスタアドレスを格納します。	DMA_DmaStart DMA_DmaStop DMA_DmaTxInit DMA_DmaRxInit DMA_TxRegSet DMA_RxRegSet DmaInt
uint16_t	dma_ssif_dmars_reg	各 DMA 拡張リソースセクタレジスタに設定する初期設定値を格納します。	DMA_DmaTxInit DMA_DmaRxInit
uint16_t	dma_int_id	各 DMA モジュールの割り込み ID を格納します。	DMA_DmaStart DMA_DmaStop

6.8 関数一覧

表 6.15 に関数一覧表を示します。

表 6.15 関数一覧表

関数名	ページ番号
R_SSI_RamInit	29
R_SSI_Ssilnit	29
R_SSI_DmaInit	30
R_SSI_SsiStart	31
R_SSI_DmaStart	32
R_SSI_SsiStop	33
R_SSI_SsiDmaStop	34
R_SSI_SsiDmaAdsSet	35

6.9 関数仕様

サンプルコードの関数仕様を示します。

6.9.1 R_SSI_RamInit

R_SSI_RamInit	
概 要	SSI ドライバで使用する管理情報の初期化
ヘッダ	drv_ssif.h drv_ssif_user.h
宣 言	void R_SSI_RamInit(void)
説 明	本関数では、下記の処理を行います。 ・ SSI ドライバで使用する管理情報の初期化を行います。
引 数	本 API は、他 API を実行する前に必ず実行してください。
リターン値	なし

6.9.2 R_SSI_Ssilnit

R_SSI_Ssilnit	
概 要	SSI の初期化処理を行います。
ヘッダ	drv_ssif.h drv_ssif_user.h
宣 言	int32_t R_SSI_Ssilnit(uint32_t ssi_channel)
説 明	本関数は、引数「ssi_channel」で指定した SSI の初期化処理を実施します。 ・ SSI のモジュールスタンバイを解除します。 ・ SSI に対してソフトウェアリセットを行い、その後、ソフトウェアリセットを解除します。 ・ SSI の端子(SSISCK、SSIWS 及び SSIDATA)設定を行います。本設定は、drv_ssif_user.c にユーザ関数を記載していますので使用する SSI に応じて記載してください。 ・ SSI の初期設定(システムワード長、データ長、転送サイズ等)を行います。本初期設定値は、ssif_user.h にマクロとして記載していますので使用する設定に応じて変更してください。 ・ SSI の割り込み(送信アンダフロー、受信オーバフロー)の設定を行います。
引 数	uint32_t ssi_channel 初期化を行う SSI チャンネル番号
リターン値	DEVDRV_SUCCESS : 初期化成功 DEVDRV_ERROR : 初期化失敗 ・ 引数にて設定された SSI のチャンネル番号が存在しない。

6.9.3 R_SSI_Dmalnit

R_SSI_Dmalnit

概 要	DMAC の初期化処理を行います。	
ヘッダ	drv_ssif.h drv_ssif_user.h	
宣 言	int32_t R_SSI_Dmalnit(uint32_t ssi_channel, uint32_t dma_channel, uint32_t direct, fp_type func)	
説 明	本関数では、SSI の送受信処理にて使用する DMAC の初期設定を行います。 引数 direct により送受信の方向を選択し、direct が「SSI_TX」の場合、送信方向(バッファ→SSI)として、引数 dma_channel により指定された DMAC の初期設定を行います。また direct が「SSI_RX」の場合は受信方向(SSI→バッファ)として、引数 dma_channel により指定された DMAC の初期設定を行います。 DMAC の設定は以下の通りです。 ・ DMA モードをレジスタ・モードに設定します。 ・ DMA 転送モードをシングル転送モードに設定します。 ・ DMA トランスファ・サイズを 32 ビット設定に設定します。 ・ DMAC の ACK、DREQ 信号をレベル設定にします。 ・ DMA (バッファ→SSI)起動要因は SSI 送信データエンプティに設定します。 ・ DMA (SSI→バッファ)起動要因は SSI 受信データフルに設定します。 ・ DMA 転送終了割り込みの設定を行います。 ・ 引数 func により指定されたコールバック関数を転送完了後コールするように登録処理を行います。	
引 数	uint32_t ssi_channel	初期化を行う SSI チャンネル番号(0-5)
	uint32_t dma_channel	初期化を行う DMA チャンネル番号(0-15)
	uint32_t direct	送受信の選択 (SSI_TX : 送信、SSI_RX : 受信)
	void *func	転送完了時に実行するコールバック関数
リターン値	DEVDRV_SUCCESS : 初期化成功 DEVDRV_ERROR : 初期化失敗 ・ SSI、DMAC のチャンネル番号が存在しない。 ・ 既に初期設定が完了している DMAC に対し、再設定を行っている。	

6.9.4 R_SSI_SsiStart

R_SSI_SsiStart

概 要	SSI を起動します。	
ヘッダ	drv_ssif.h drv_ssif_user.h	
宣 言	int32_t R_SSI_SsiStart(uint32_t ssi_channel, uint32_t direct)	
説 明	本関数では、SSI の起動設定を行います。 引数 direct により送受信の方向を選択し、direct が「SSI_TX」の場合、送信方向として、引数 ssi_channel により指定された SSI の送信許可設定を行います。direct が「SSI_RX」の場合、受信方向として、引数 ssi_channel により指定された SSI の受信許可設定を行います。 引数で指定した SSI に対し以下の設定を行います。 direct が「SSI_TX」の場合： ・送信 FIFO を初期化します。 ・エラー割り込みステータス(送信アンダフロー)をクリアします。 ・エラー割り込み(送信アンダフロー)を許可します。 ・送信データエンプティ割り込みを許可します。 ・送信処理を許可します。 direct が「SSI_RX」の場合： ・受信 FIFO を初期化します。 ・エラー割り込みステータス(受信オーバフロー)をクリアします。 ・エラー割り込み(受信オーバフロー)を許可します。 ・受信フル割り込みを許可します。 ・SSI の受信処理を許可します。 direct が「SSI_RX SSI_TX」の場合： ・送受信 FIFO を初期化します。 ・エラー割り込みステータス(受信オーバフロー、送信アンダフロー)をクリアします。 ・エラー割り込み(受信オーバフロー、送信アンダフロー)を許可します。 ・送信データエンプティ、受信フル割り込みを許可します。 ・SSI の送受信処理を許可します。	
引 数	uint32_t ssi_channel	初期化を行う SSI チャンネル番号(0-5)
	uint32_t direct	送受信の選択 (SSI_TX : 送信、SSI_RX : 受信)
リターン値	DEVDRV_SUCCESS : 起動成功 DEVDRV_ERROR : 起動失敗 ・SSI のチャンネル番号が存在しない。 ・direct に「SSI_TX」、「SSI_RX」以外のマクロが指定された。	

6.9.5 R_SSI_DmaStart

R_SSI_DmaStart

概 要	DMAC を起動します。
ヘッダ	drv_ssif.h drv_ssif_user.h
宣 言	int32_t R_SSI_DmaStart(uint32_t ssi_channel, uint32_t direct)
説 明	本関数は DMAC を起動し、引数が送信設定(direct : SSI_TX)時は、SSI ヘデータ転送の開始を許可します。受信設定(direct : SSI_RX)時は、SSI からデータ受信を許可します。

送信設定(direct : SSI_TX)の場合は、本関数を実行する前に R_SSI_SsiDmaAdsSet 関数を実行し、バッファをキューに登録してください。R_SSI_SsiDmaAdsSet 関数を実行せず、本関数を実行した場合は、無音データを出力します。

受信設定(direct : SSI_RX)の場合は、本関数を実行する前に R_SSI_SsiDmaAdsSet 関数を実行し、バッファをキューに登録してください。R_SSI_SsiDmaAdsSet 関数を実行せず、本関数を実行した場合は、戻り値がエラーとなり受信動作を開始しません。

また、本関数を実行する前に、必ず R_SSI_Ssilnit 関数、R_SSI_Dmalnit 関数を用い SSI、DMAC の初期化を行ってください。

引 数	uint32_t ssi_channel 初期化を行う SSI チャンネル番号(0-5)
	uint32_t direct 送受信の選択 (SSI_TX : 送信、SSI_RX : 受信)
リターン値	DEVDRV_SUCCESS : 起動成功 DEVDRV_ERROR : 起動失敗 ・ SSI のチャンネル番号が存在しない。 ・ direct に「SSI_TX」、「SSI_RX」以外のマクロが指定された。 ・ 初期化関数により DMAC への初期設定が行われていない。 ・ 受信指定の場合、R_SSI_SsiDmaAdsSet 関数にてバッファの登録が行われていない。

6.9.6 R_SSI_SsiStop

R_SSI_SsiStop

概 要	SSI を停止します。
ヘッダ	drv_ssif.h drv_ssif_user.h
宣 言	int32_t R_SSI_SsiStop(uint32_t ssi_channel)
説 明	本関数は、引数「ssi_channel」で指定した SSI の処理を停止します。

トランスミッタ（送信）動作の場合：

- ・送信処理を停止します。
- ・送信データエンプティ割り込みを禁止します。
- ・エラー割り込み(送信アンダフローエラー)を禁止します。
- ・SSI がアイドルに遷移するまで待ちます。
- ・引数で指定した SSI に対しソフトウェアリセットを行います。

レシーバ（受信）動作の場合：

- ・受信処理を停止します。
- ・受信フル割り込みを禁止します。
- ・エラー割り込み(受信オーバフローエラー)を禁止します。
- ・SSI がアイドルに遷移するまで待ちます。
- ・引数で指定した SSI に対しソフトウェアリセットを行います。

トランシーバ（全二重通信）動作の場合：

- ・送受信処理を停止します。
- ・送信データエンプティ、受信フル割り込みを禁止します。
- ・エラー割り込み(送信アンダフローエラー、受信オーバフローエラー)を禁止します。
- ・SSI がアイドルに遷移するまで待ちます。
- ・引数で指定した SSI に対しソフトウェアリセットを行います。

引 数	uint32_t ssi_channel	初期化を行う SSI チャンネル番号(0-5)
リターン値	DEVDRV_SUCCESS	停止成功
	DEVDRV_ERROR	停止失敗
		・SSI のチャンネル番号が存在しない。

6.9.7 R_SSI_SsiDmaStop

R_SSI_SsiDmaStop

概 要	SSI の送受信停止処理を行います。
ヘッダ	drv_ssif.h drv_ssif_user.h
宣 言	int32_t R_SSI_SsiDmaStop(uint32_t ssi_channel)
説 明	本関数は、引数「ssi_channel」で指定した SSI の処理を停止します。

トランスミッタ（送信）動作の場合：

- ・ 引数で指定した SSI に関連づけられている DMAC(バッファ→SSI)を停止します。その後、Source Address レジスタにミュートバッファをセットし、ミュートデータの出力を開始します。

レシーバ（受信）動作の場合：

- ・ 受信処理を停止します。
- ・ 受信フル割り込みを禁止します。
- ・ エラー割り込み(受信オーバーフローエラー)を禁止します。
- ・ SSI がアイドルに遷移するまで待ちます。
- ・ 引数で指定した SSI に対しソフトウェアリセットを行います。

トランシーバ（全二重通信）動作の場合：

- ・ 送受信処理を停止します。
- ・ 送信データエンプティ、受信フル割り込みを禁止します。
- ・ エラー割り込み(送信アンダフローエラー、受信オーバーフローエラー)を禁止します。
- ・ SSI がアイドルに遷移するまで待ちます。
- ・ 引数で指定した SSI に対しソフトウェアリセットを行います。

引 数	uint32_t ssi_channel	初期化を行う SSI チャンネル番号(0-5)
リターン値	DEVDRV_SUCCESS	停止成功
	DEVDRV_ERROR	停止失敗
		・ SSI モジュールのチャンネル番号が存在しない。

6.9.8 R_SSI_SsiDmaAdsSet

R_SSI_SsiDmaAdsSet

概 要	バッファをキューに登録します。	
ヘッダ	drv_ssif.h drv_ssif_user.h	
宣 言	int32_t R_SSI_SsiDmaAdsSet(const uint32_t ssi_channel, const uint32_t direct, volatile const void * const buf, const uint32_t buf_size)	
説 明	本関数は、送受信バッファをキューに登録します。 引数が送信設定(direct : SSI_TX)の場合は、キューに送信バッファを登録し、受信設定(direct : SSI_RX)の場合は、キューに受信バッファを登録します。 送受信に各 4 段のキューがあります。 送信設定(direct : SSI_TX)の場合は、キューに登録されている送信バッファの転送が完了するまでに本関数を用いて送信バッファをキューに登録してください。登録が間に合わない場合は、ミュートデータを出力します。 受信設定(direct : SSI_RX)の場合は、キューに登録されている受信バッファがフルになる前に次受信バッファをキューに登録してください。登録が間に合わない場合は、最後に登録された受信バッファにデータを上書きします。	
引 数	uint32_t ssi_channel	初期化を行う SSI チャンネル番号(0-5)
	uint32_t direct	送受信の選択 (SSI_TX : 送信、SSI_RX : 受信)
	void * const buf,	送受信バッファの先頭アドレス
	uint32_t buf_size	送受信バッファのサイズ
リターン値	DEVDRV_SUCCESS : キュー登録成功 DEVDRV_ERROR : キュー登録失敗 ・ SSI のチャンネル番号が存在しない。 DEVDRV_QUEUE_FULL : キューがフルのため、登録失敗	

6.10 フローチャート

6.10.1 メイン処理

図 6.8、図 6.9 にメイン処理エラー! 参照元が見つかりません。のフローチャートを示します。

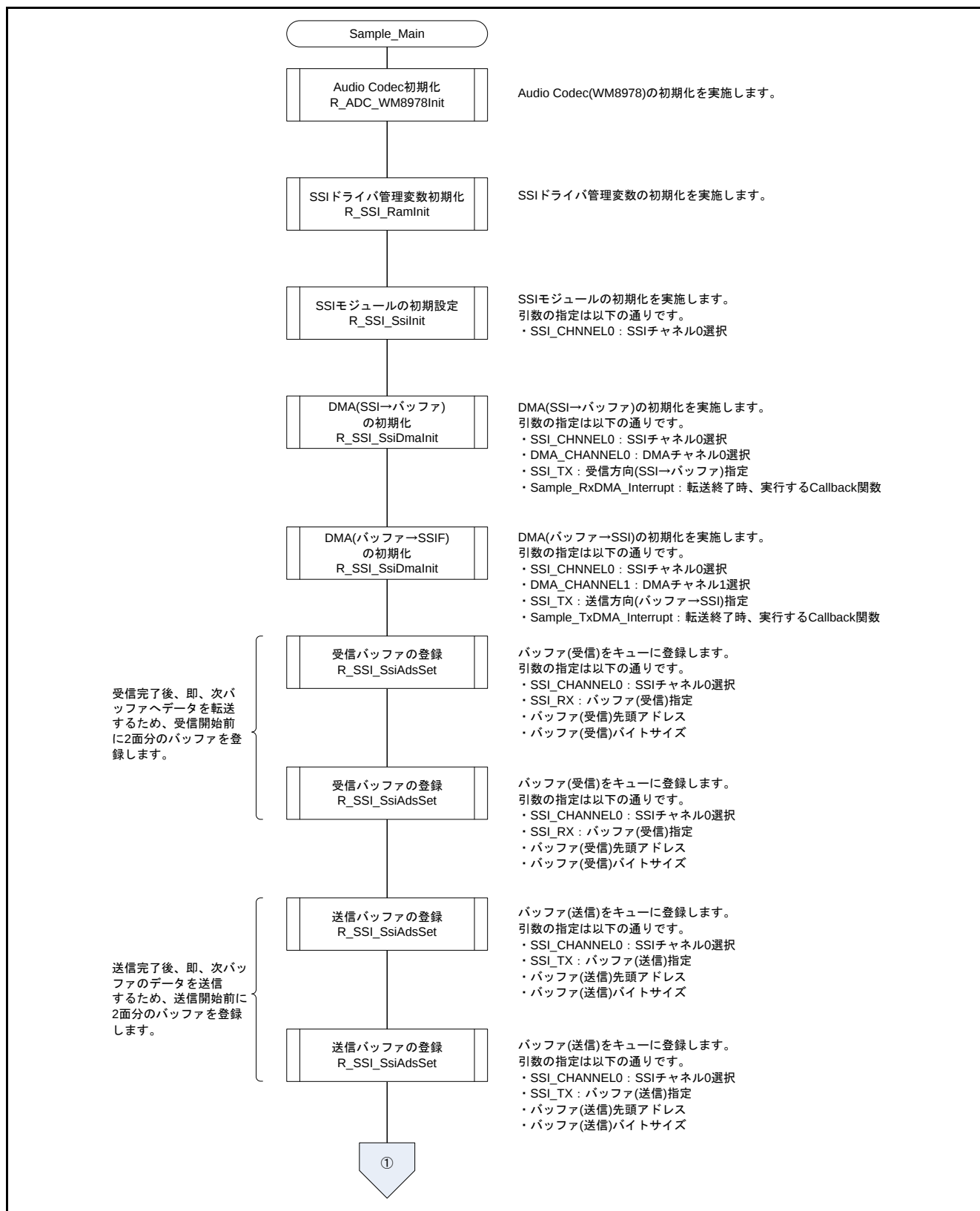


図 6.8 メイン処理(1)

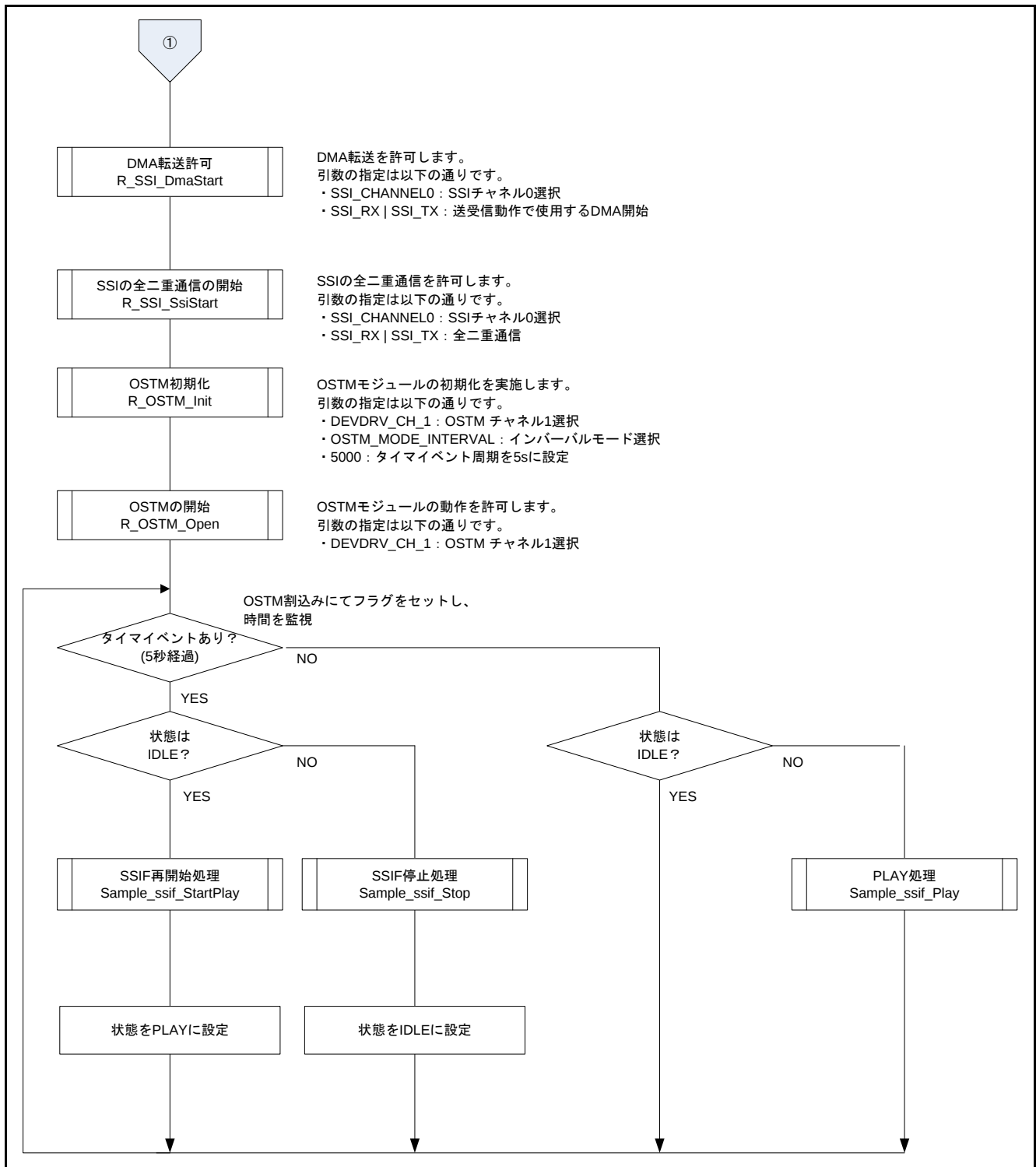


図 6.9 メイン処理(2)

6.10.2 メイン(SSi 動作中)処理

図 6.10 にメイン(SSi 動作中)処理のフローチャートを示します。

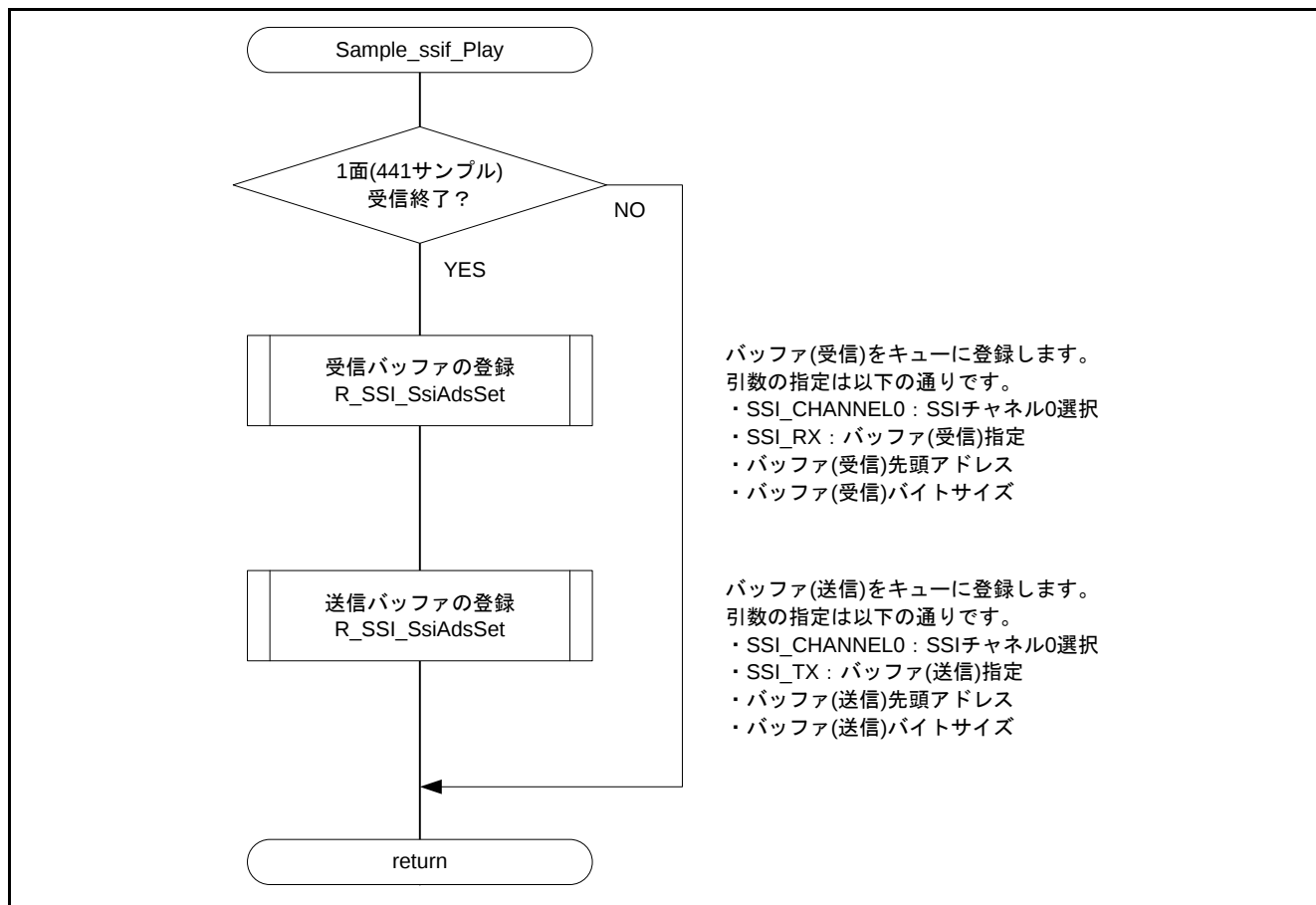


図 6.10 メイン(SSi 動作中)処理

6.10.3 メイン(SSi 停止)処理

図 6.11 にメイン(SSi 停止)処理のフローチャートを示します。

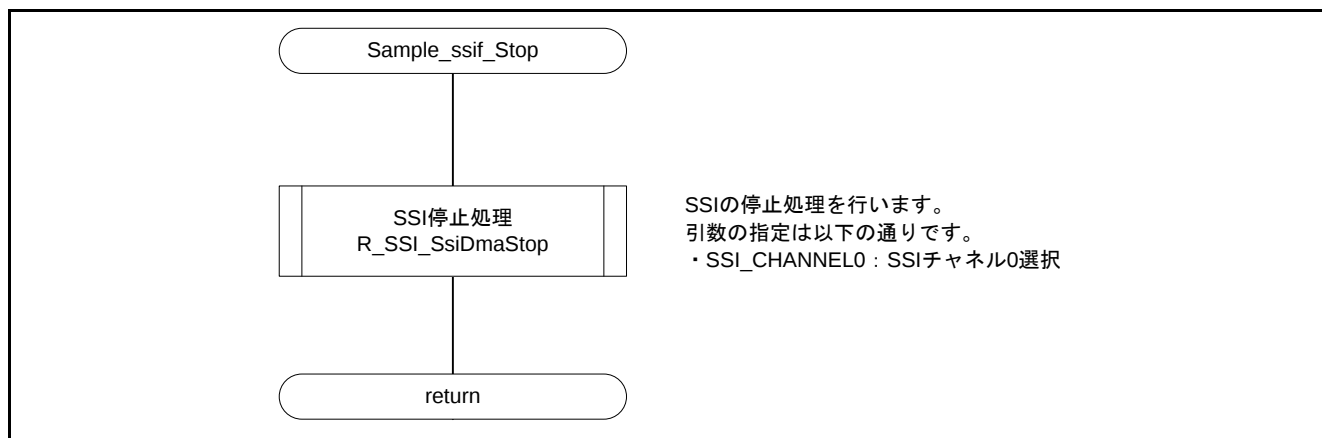


図 6.11 メイン(SSi 停止)処理

6.10.4 メイン(SSI 再開始)処理

図 6.12 にメイン(SSI 再開始)処理のフローチャートを示します。

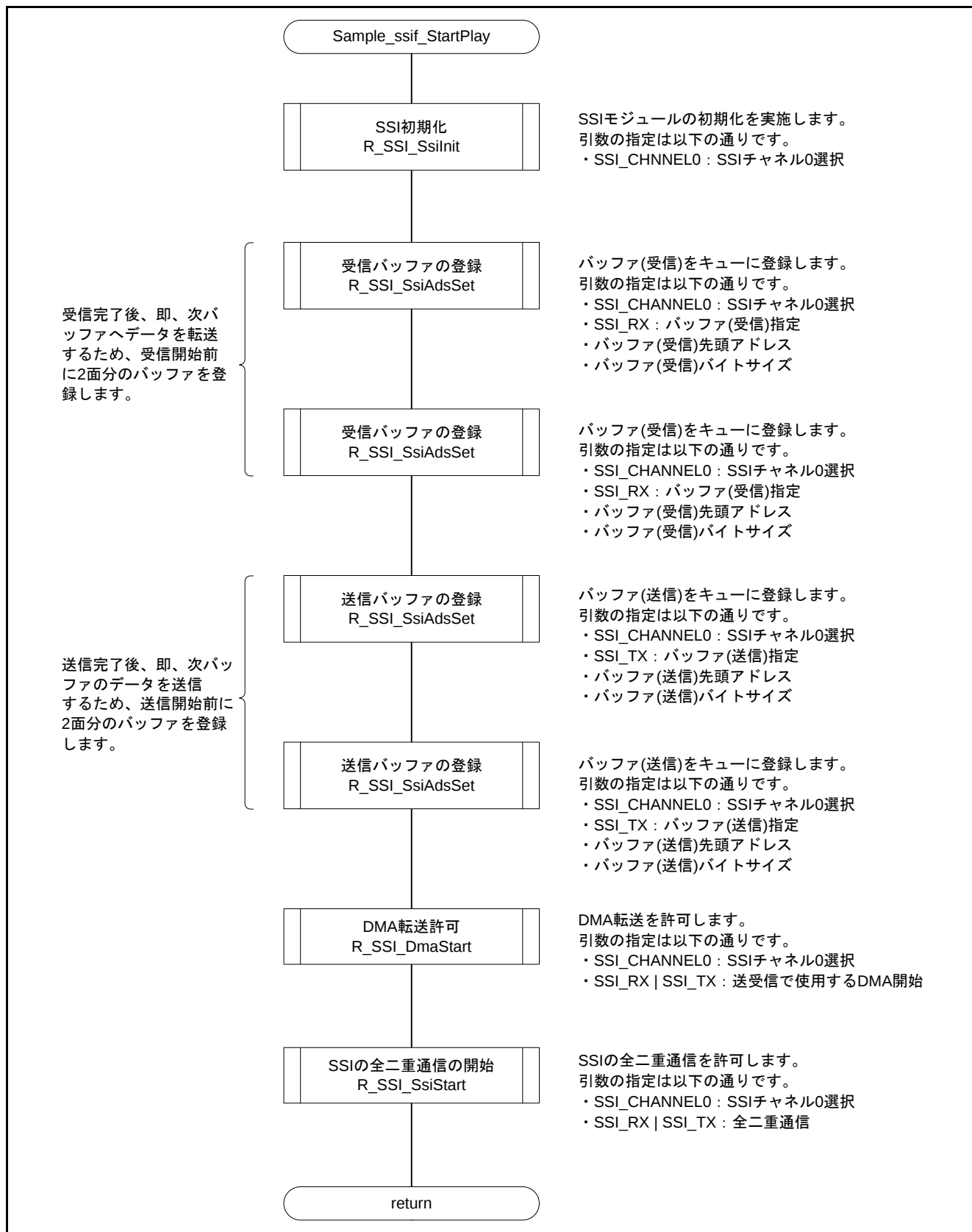


図 6.12 メイン(SSI 再開始)処理

6.10.5 DMA 転送終了割り込み処理

図 6.13 に DMA 転送終了割り込み処理のフローチャートを示します。

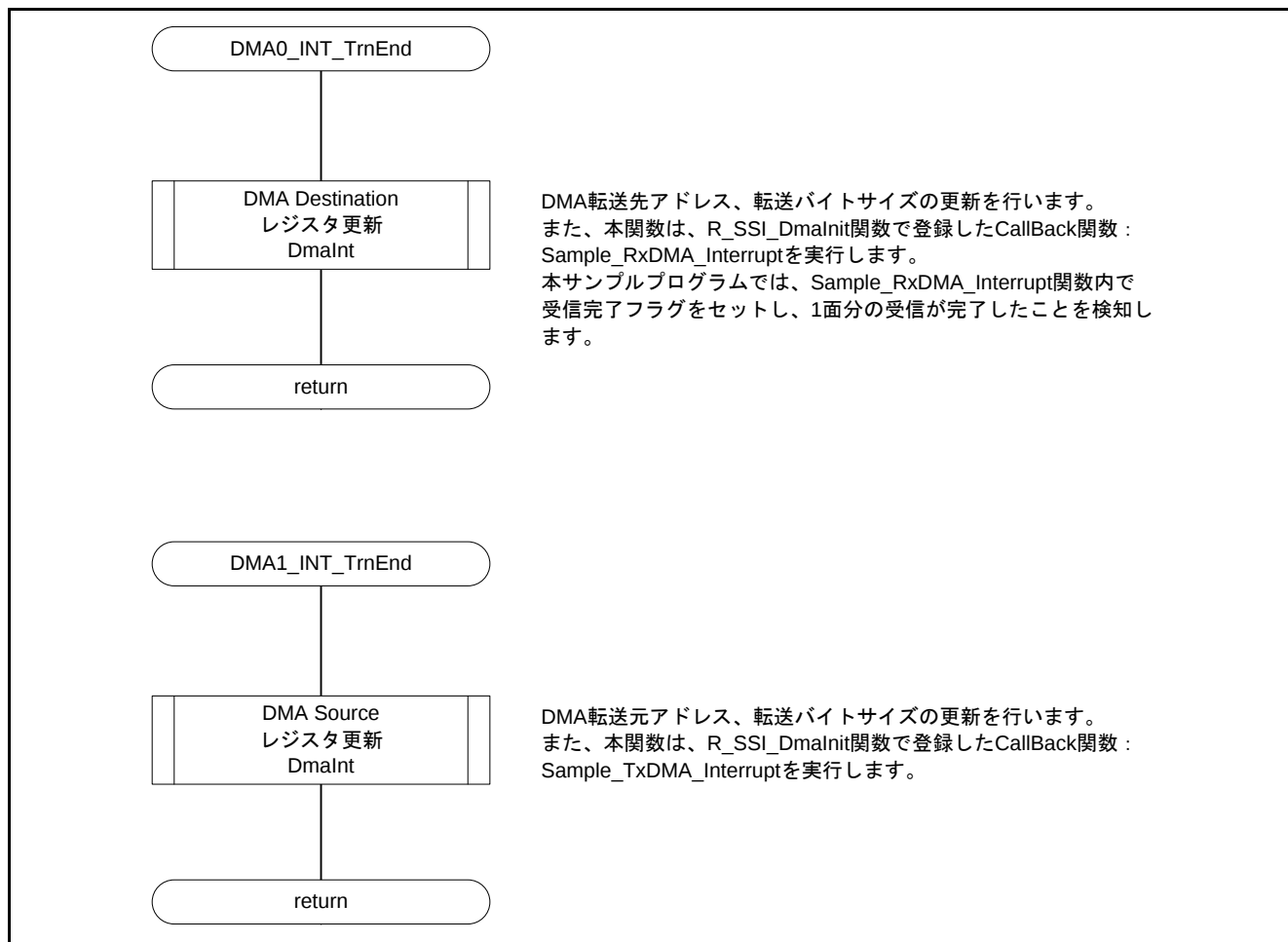


図 6.13 DMA 転送終了割り込み処理

6.10.6 OS タイマ割り込み処理

図 6.14 に OS タイマ割り込み処理のフローチャートを示します。

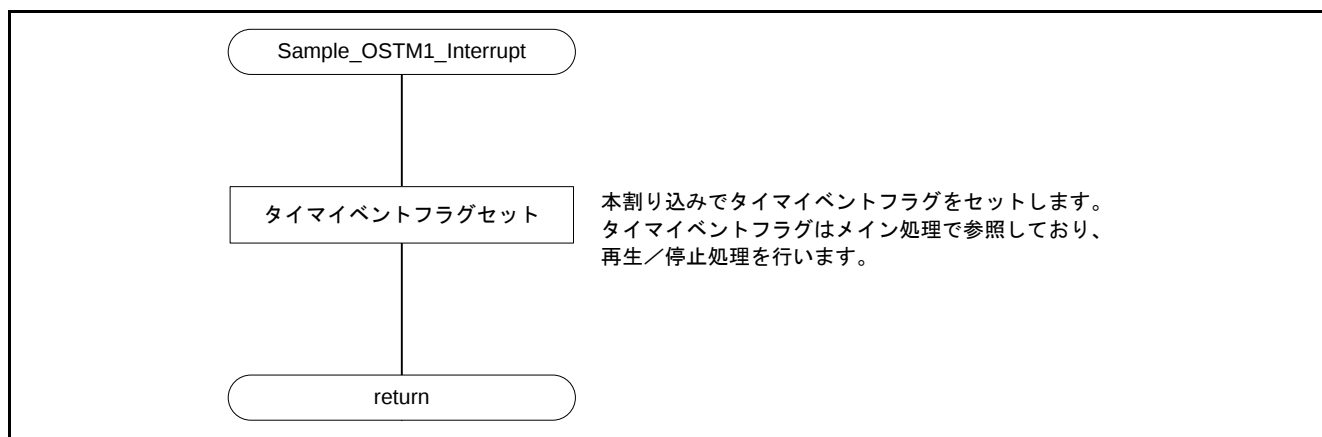


図 6.14 OS タイマ割り込み処理

6.10.7 SSI 管理情報の初期化処理

図 6.15 に SSI 管理情報の初期化設定処理のフローチャートを示します。

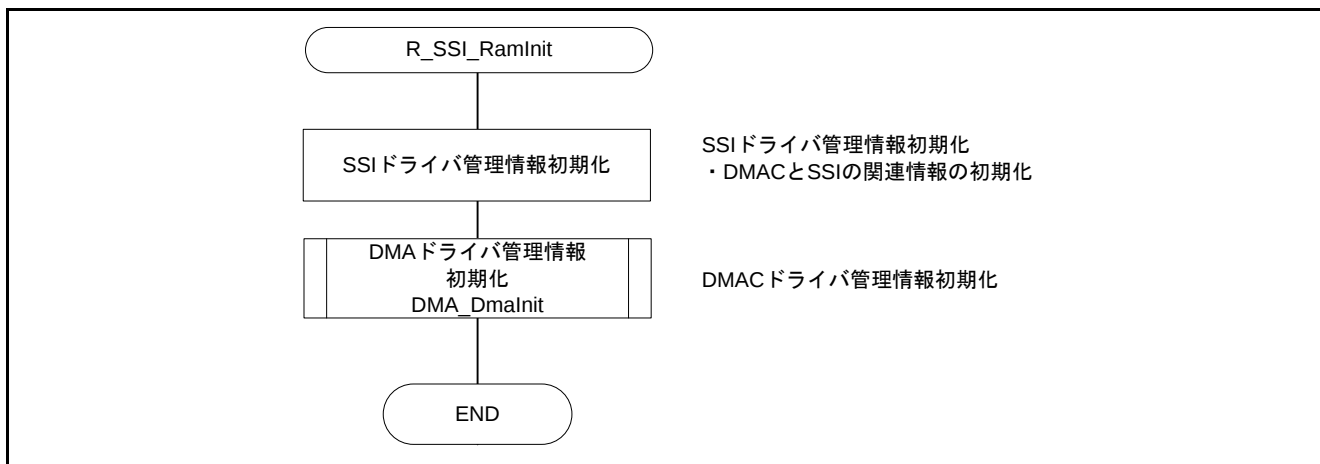


図 6.15 SSI 管理情報の初期化設定処理

6.10.8 SSI 初期設定処理

図 6.16 に SSI 初期設定処理のフローチャートを示します。

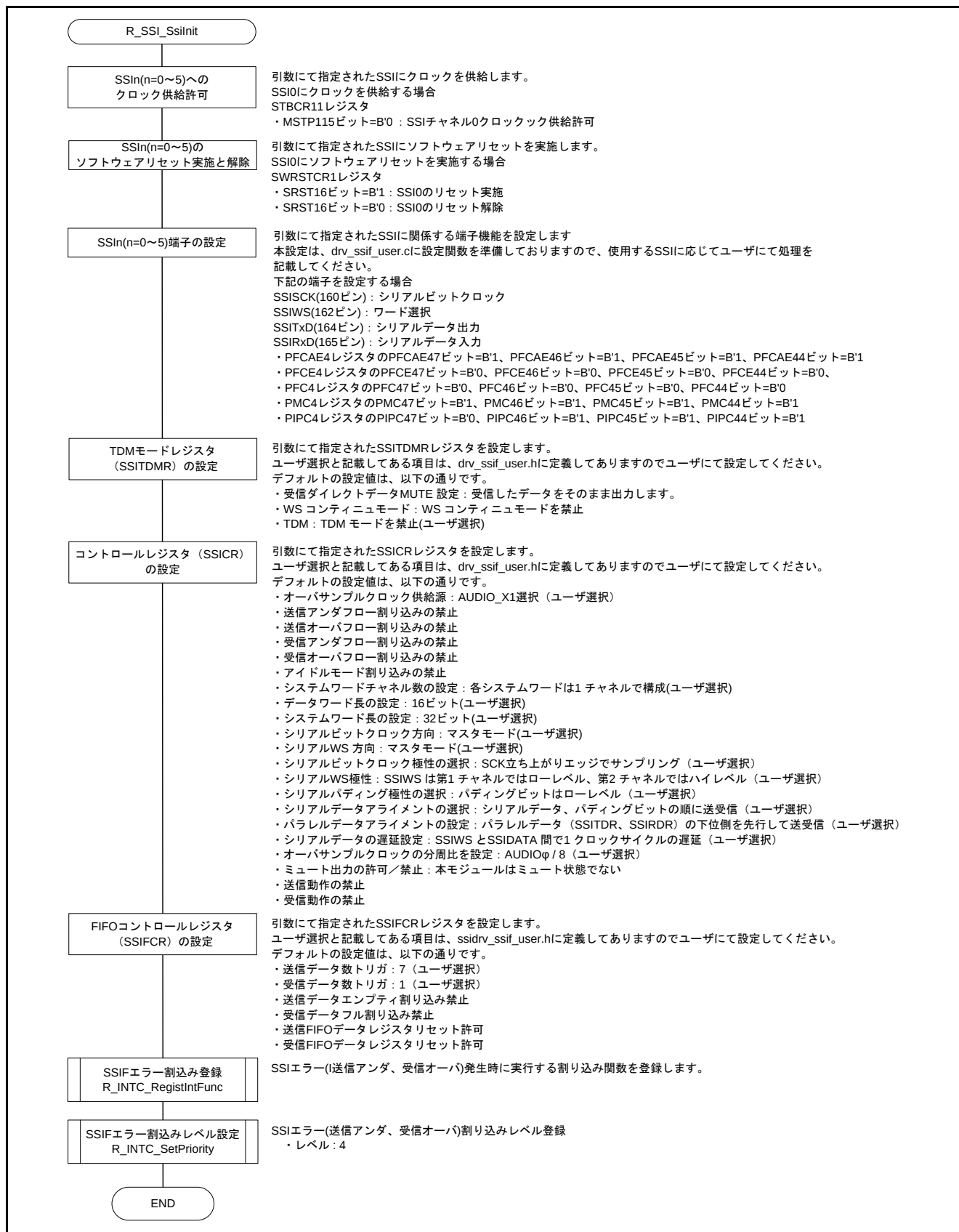


図 6.16 SSI 初期設定処理

6.10.9 DMAC 初期設定処理

図 6.17 に DMAC 初期化設定処理のフローチャートを示します。

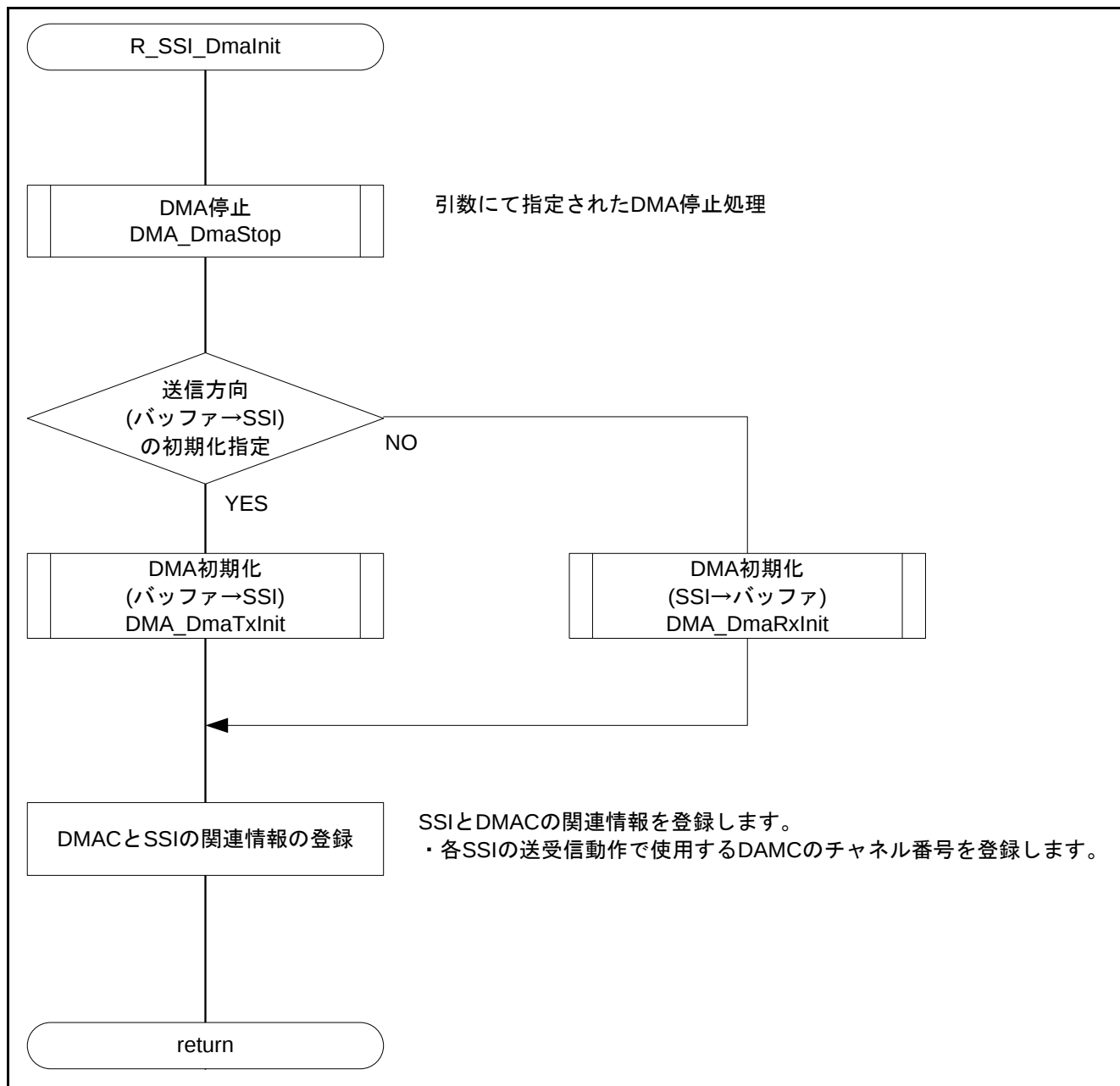


図 6.17 DMAC 初期化設定処理

6.10.10 SSI 起動処理

図 6.18 に SSI 起動処理のフローチャートを示します。

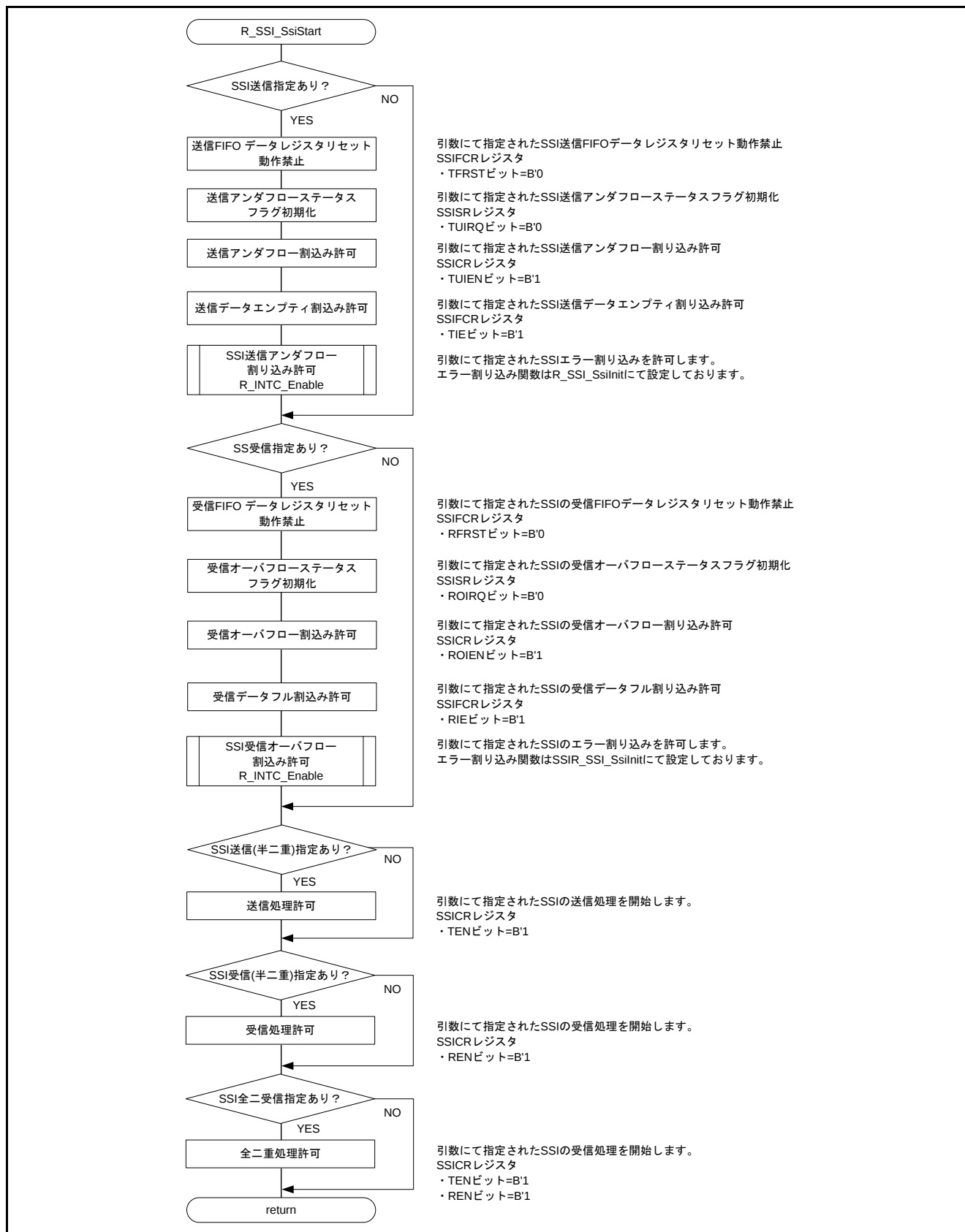


図 6.18 SSI 起動処理

6.10.11 DMAC 起動処理

図 6.19 に DMAC 起動処理のフローチャートを示します。

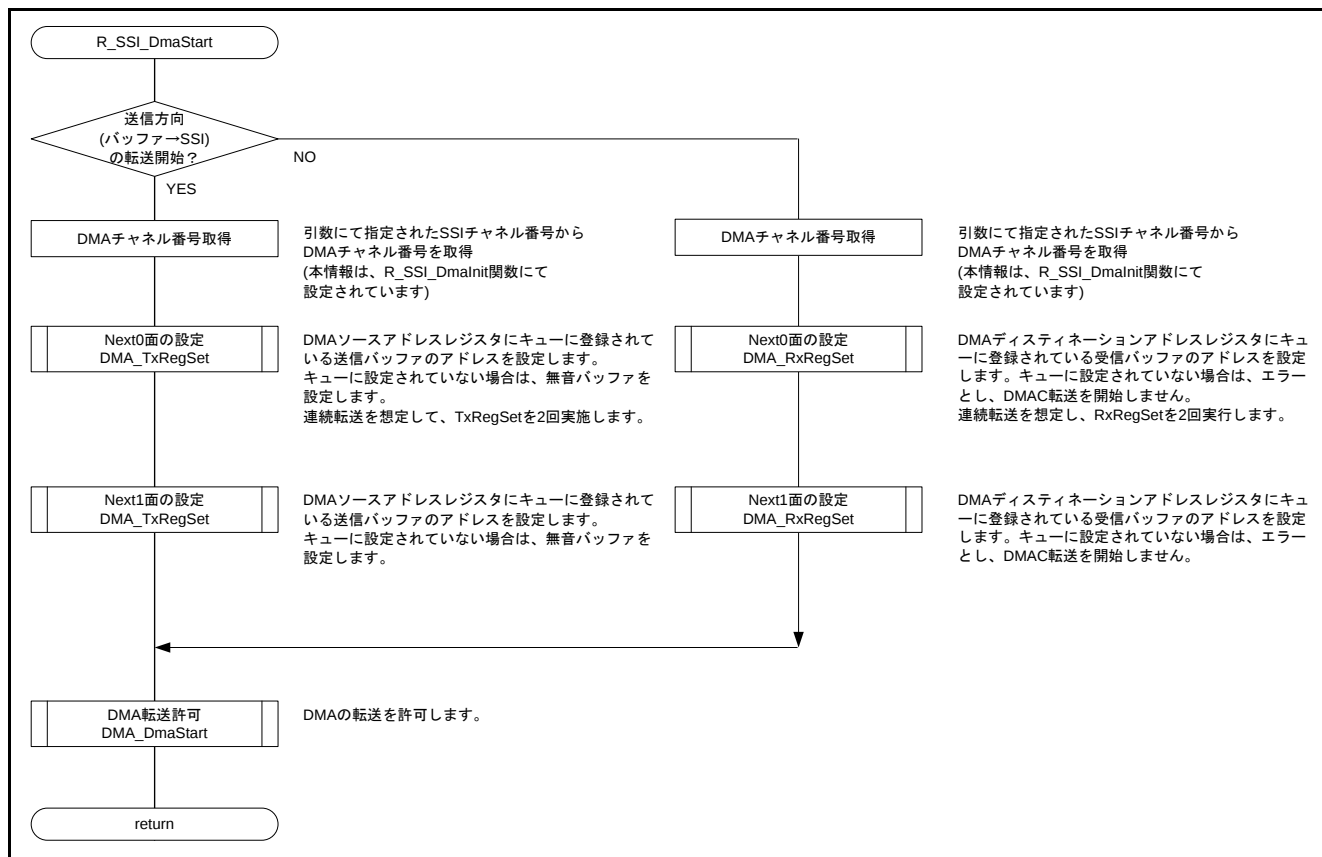


図 6.19 DMAC 起動処理

6.10.12 SSI 停止処理

図 6.20 に SSI 停止処理のフローチャートを示します。

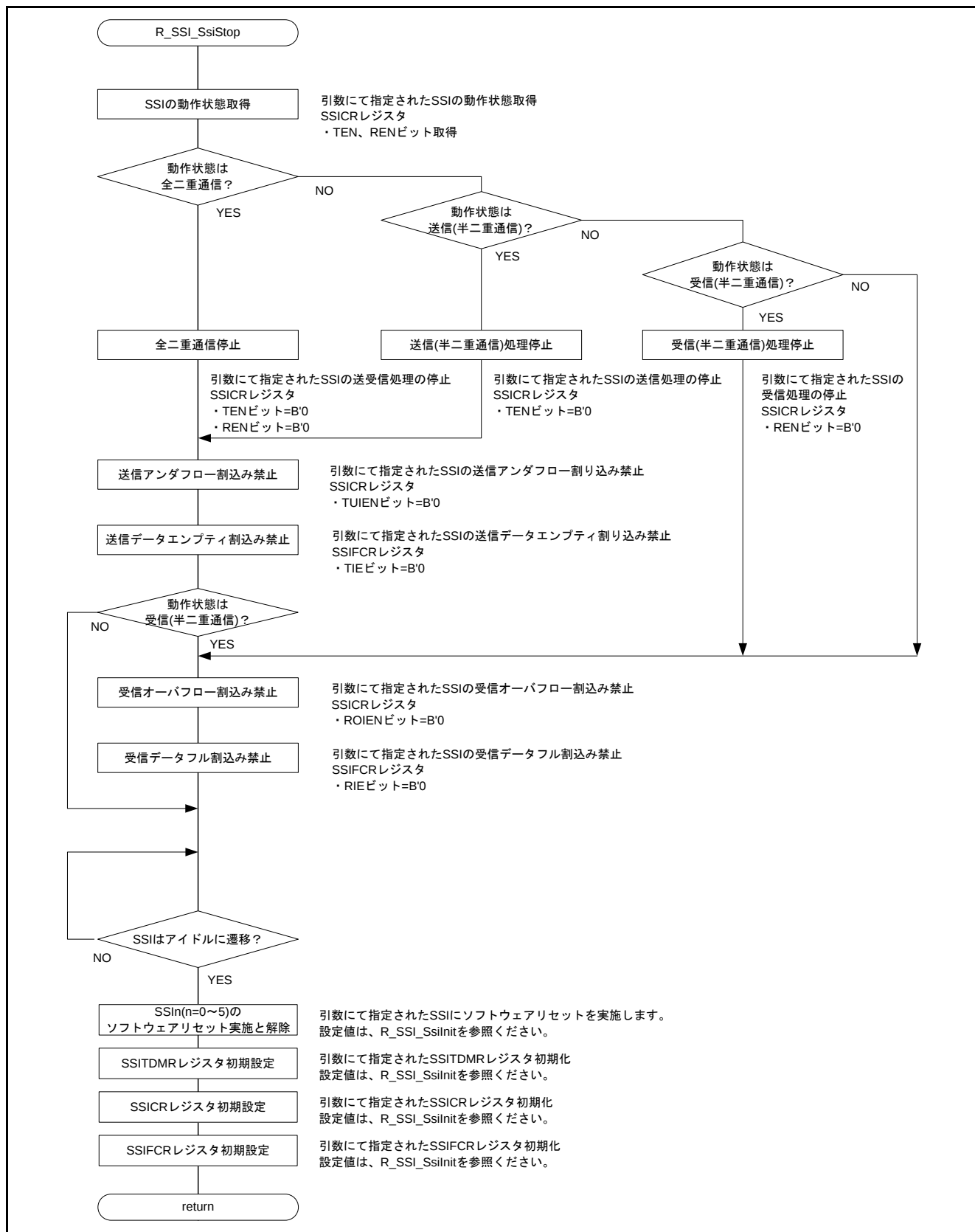


図 6.20 SSI 停止処理

6.10.13 SSI 送受信停止処理

図 6.21 に SSI 送受信停止処理のフローチャートを示します。

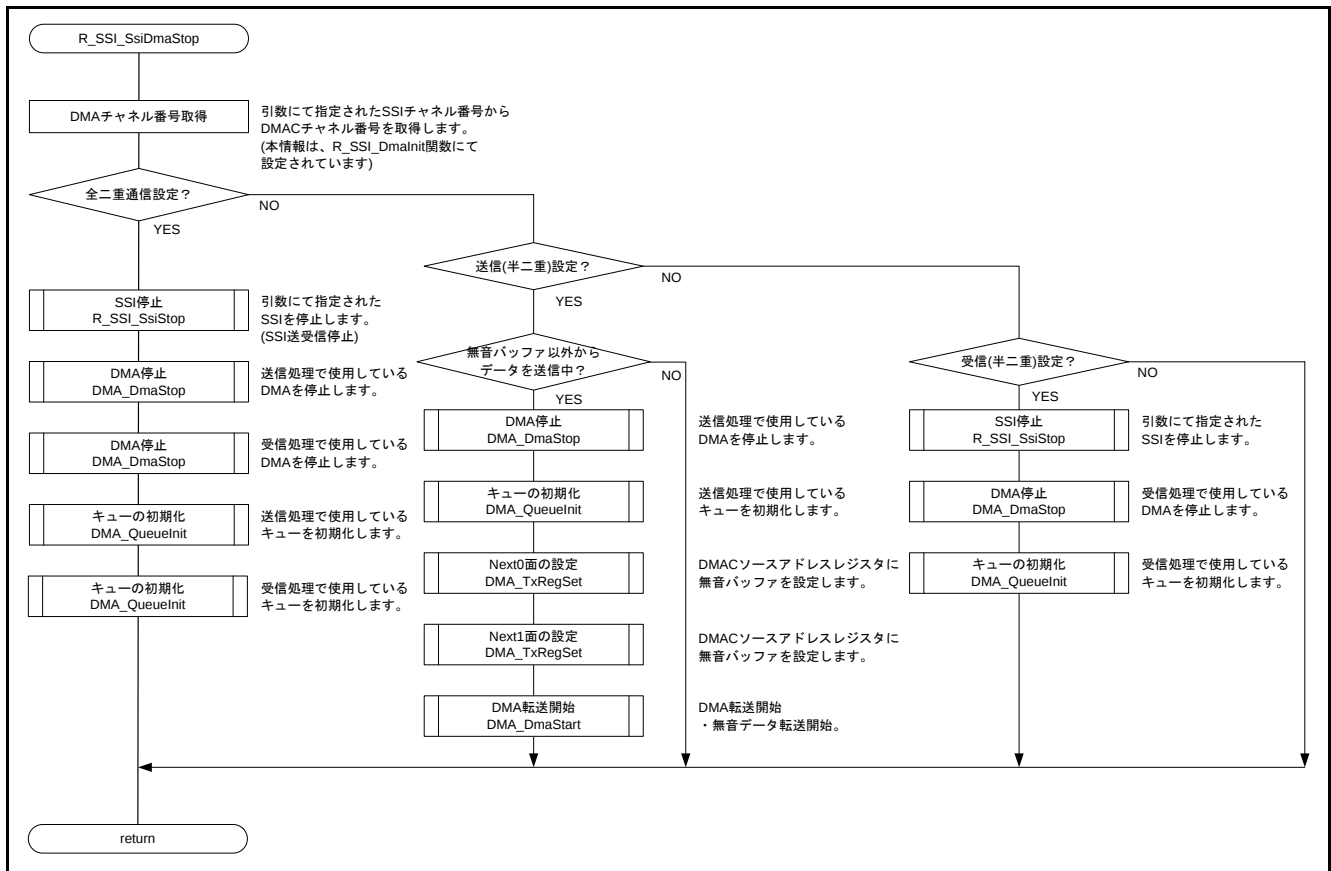


図 6.21 SSI 送受信停止処理

6.10.14 バッファ登録処理

図 6.22、図 6.23 にバッファ登録処理のフローチャートを示します。

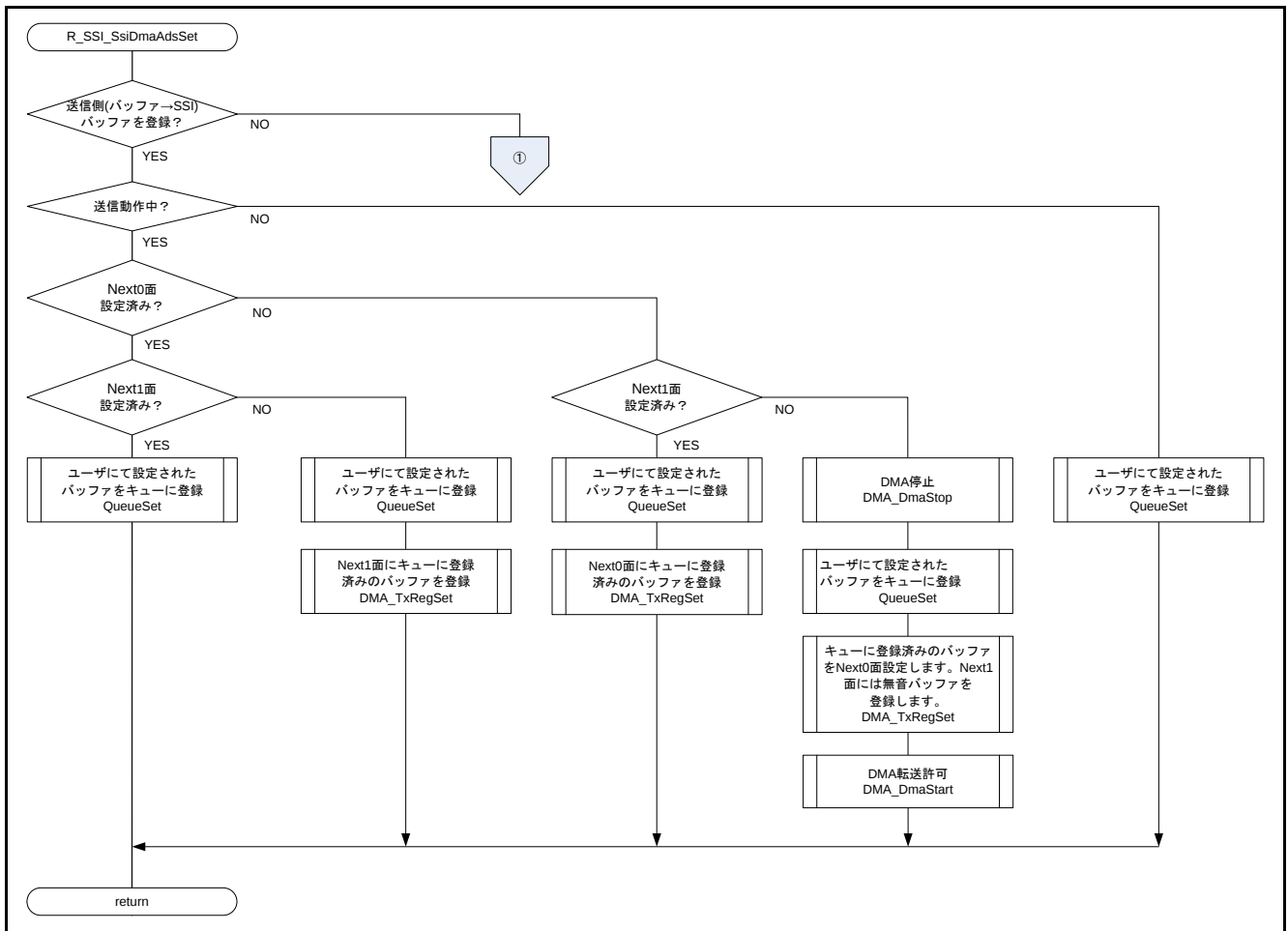


図 6.22 バッファ登録処理(1)

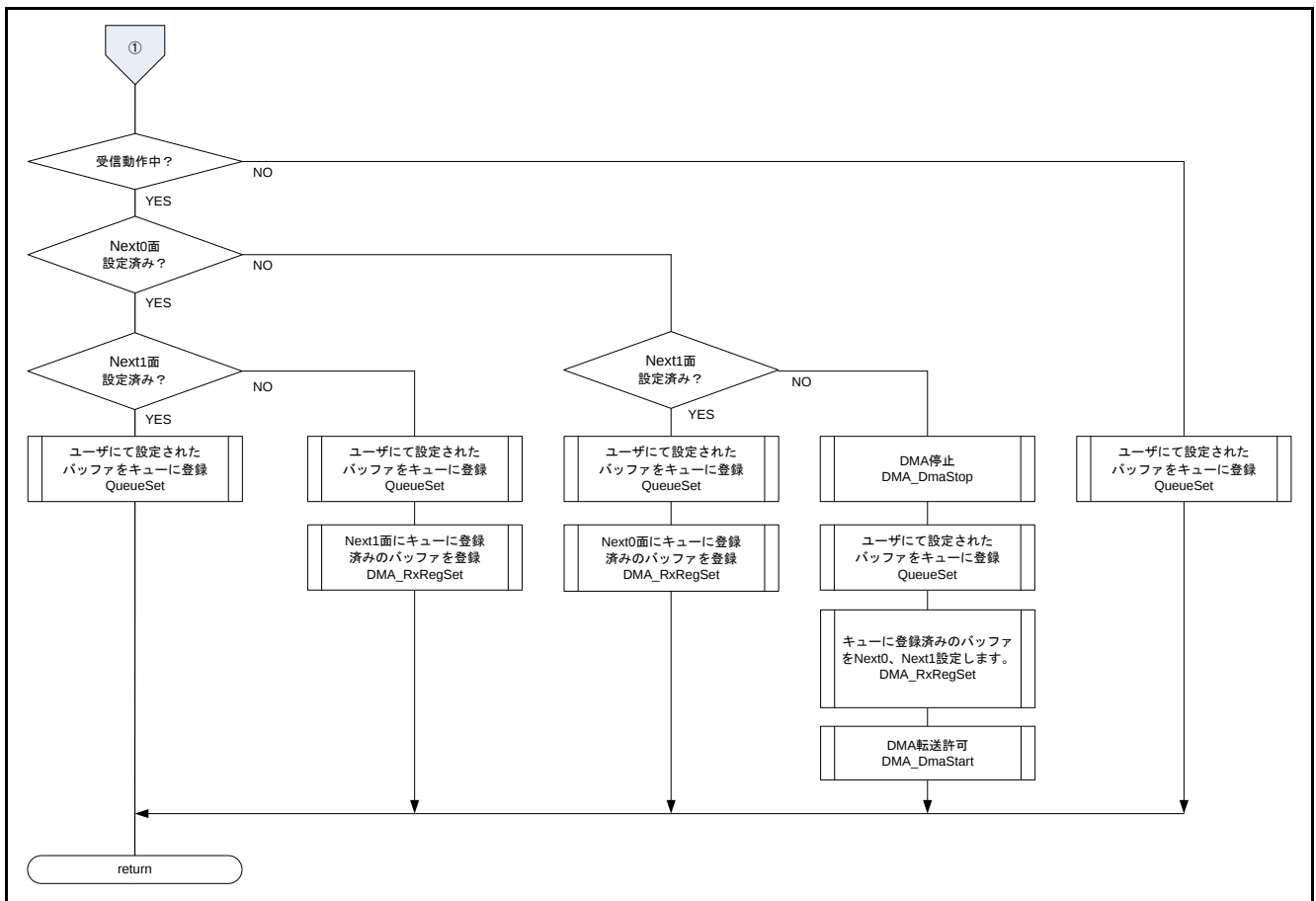


図 6.23 バッファ登録処理(2)

6.10.15 SSI エラー割り込み処理

図 6.24、図 6.25 に SSI エラー割り込み処理のフローチャートを示します。

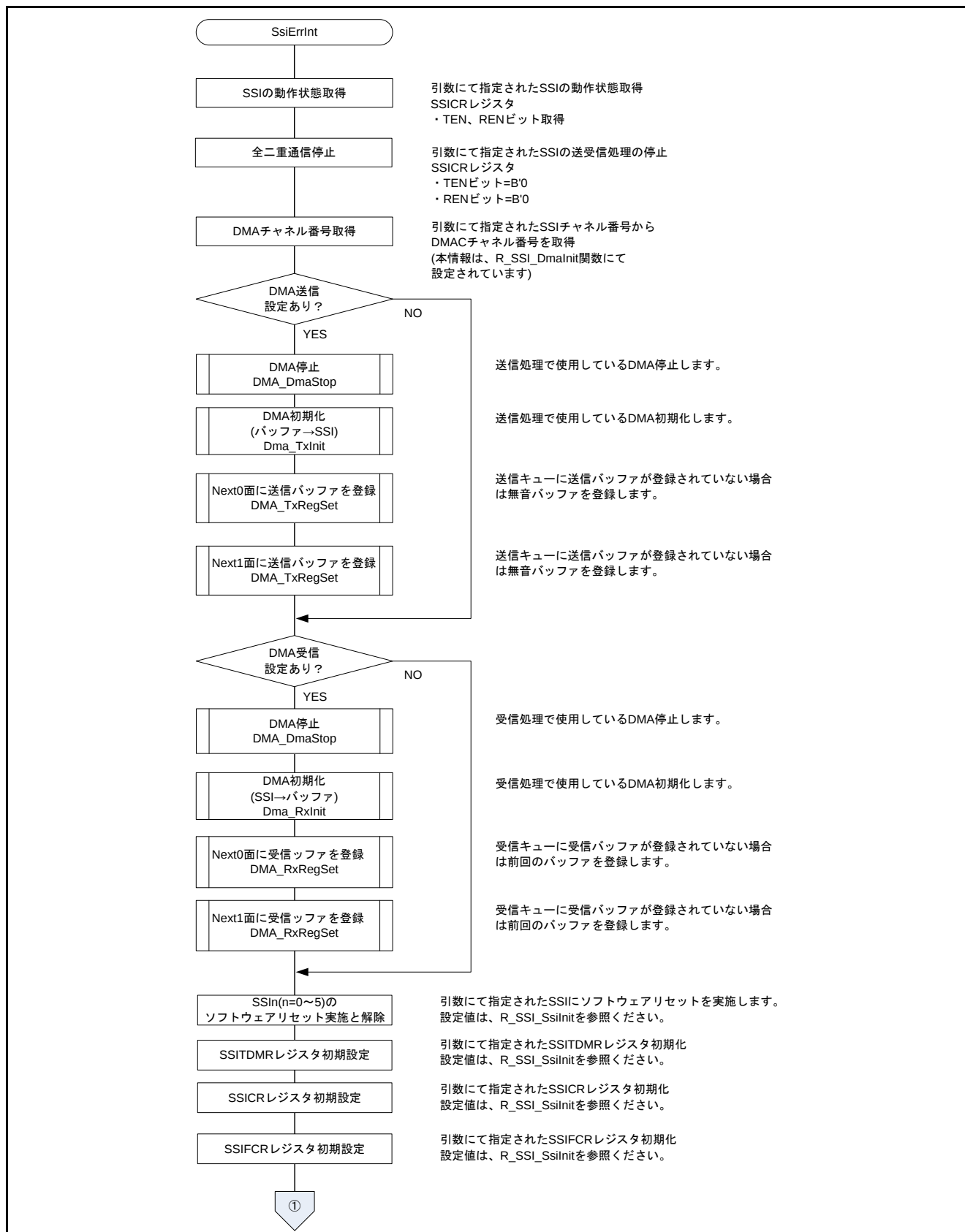


図 6.24 SSI エラー割り込み処理(1)

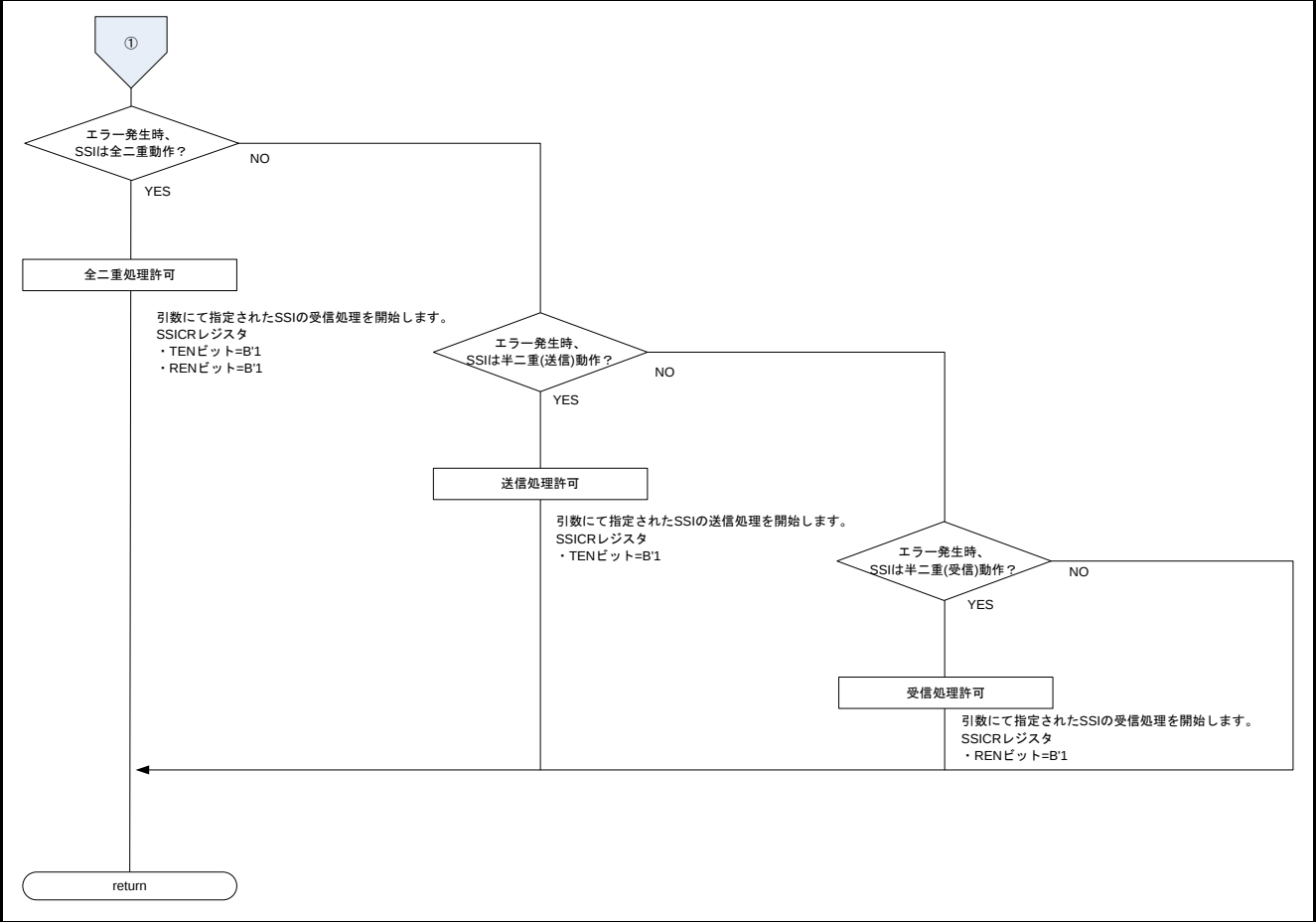


図 6.25 SSI エラー割り込み処理(2)

6.10.16 DMA 管理情報の初期化処理

図 6.26 に DMA 管理情報の初期化処理のフローチャートを示します。

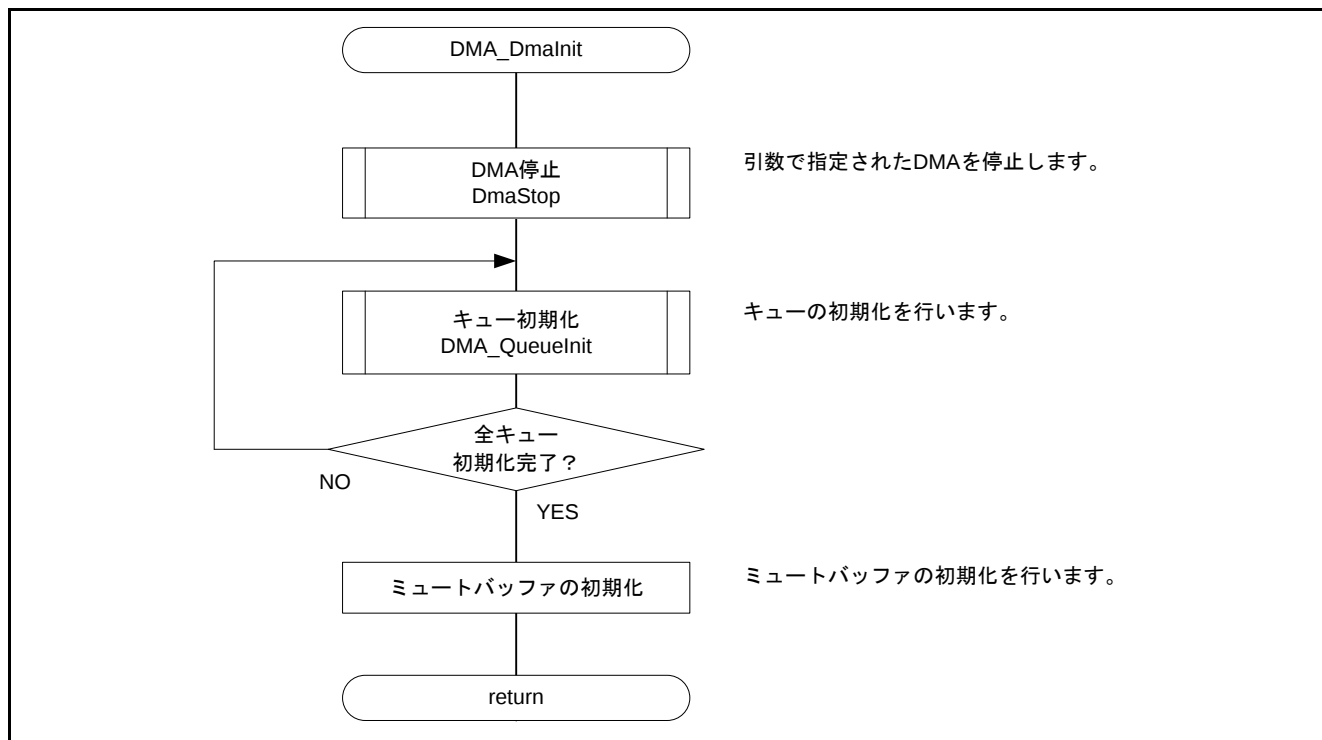


図 6.26 DMA 管理情報の初期化処理

6.10.17 キュー初期化処理

図 6.27 にキュー初期化処理のフローチャートを示します。

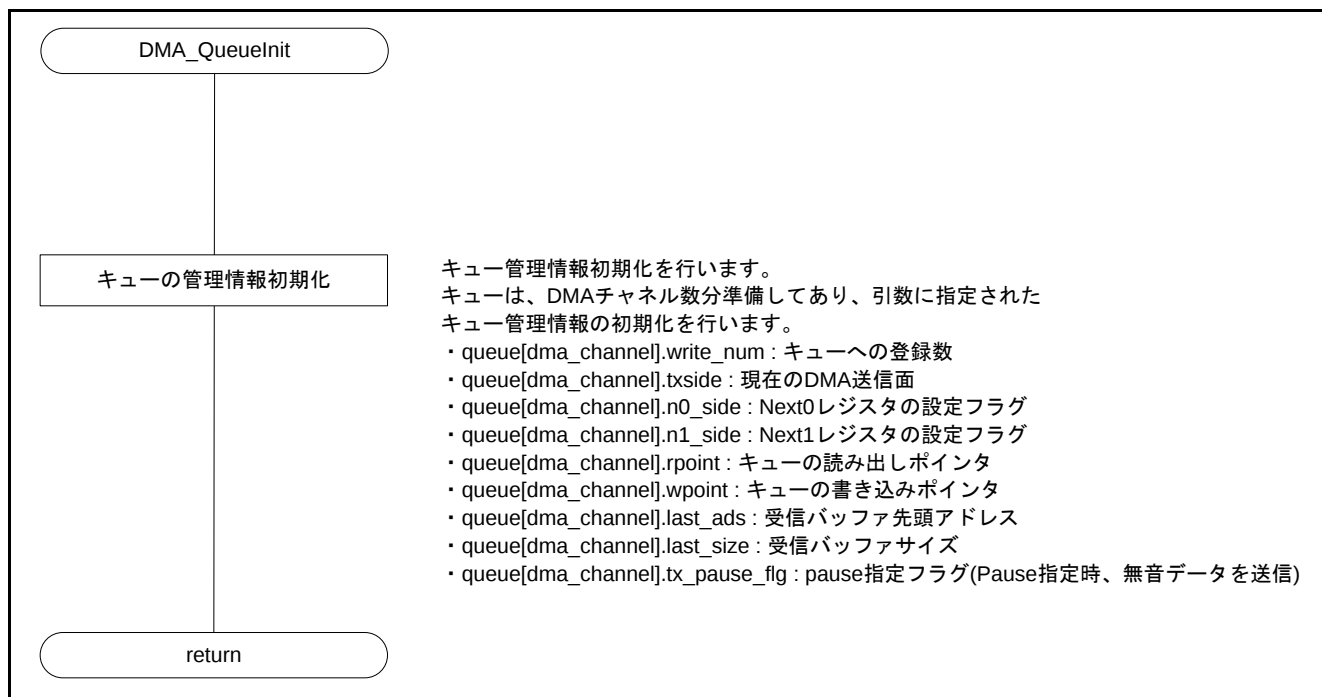


図 6.27 キュー初期化処理

6.10.18 キュー設定処理

図 6.28 にキュー設定処理のフローチャートを示します。

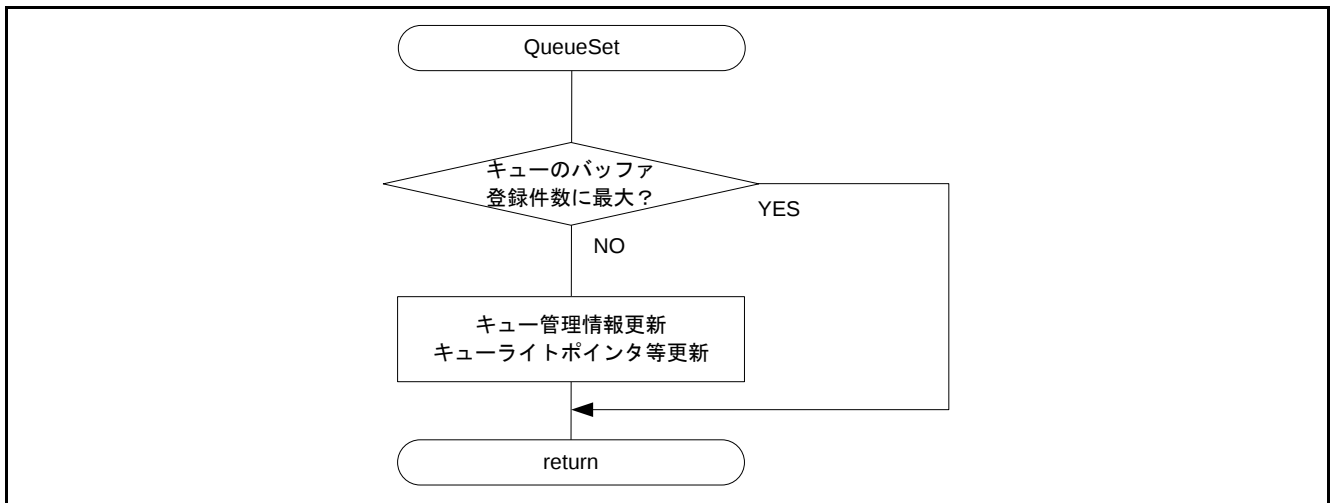


図 6.28 キュー設定処理

6.10.19 DMA 転送許可処理

図 6.29 に DMA 転送許可処理のフローチャートを示します。



図 6.29 DMA 転送許可処理

6.10.20 DMA 転送停止処理

図 6.30 に DMA 転送停止処理のフローチャートを示します。

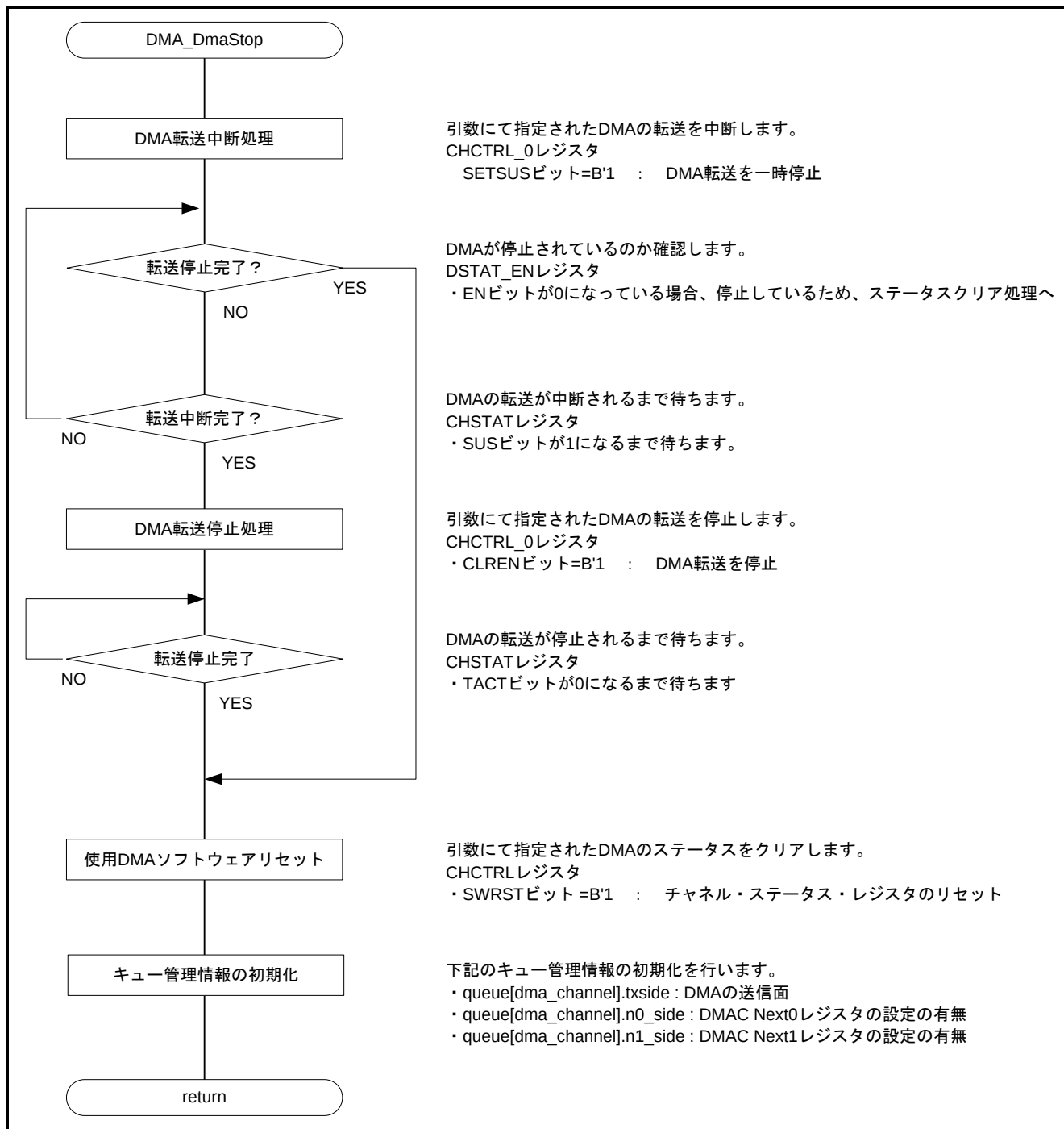


図 6.30 DMA 転送停止処理

6.10.21 DMA(バッファ→SSI)初期化处理

図 6.31 に DMA(バッファ→SSI)初期化处理のフローチャートを示します。

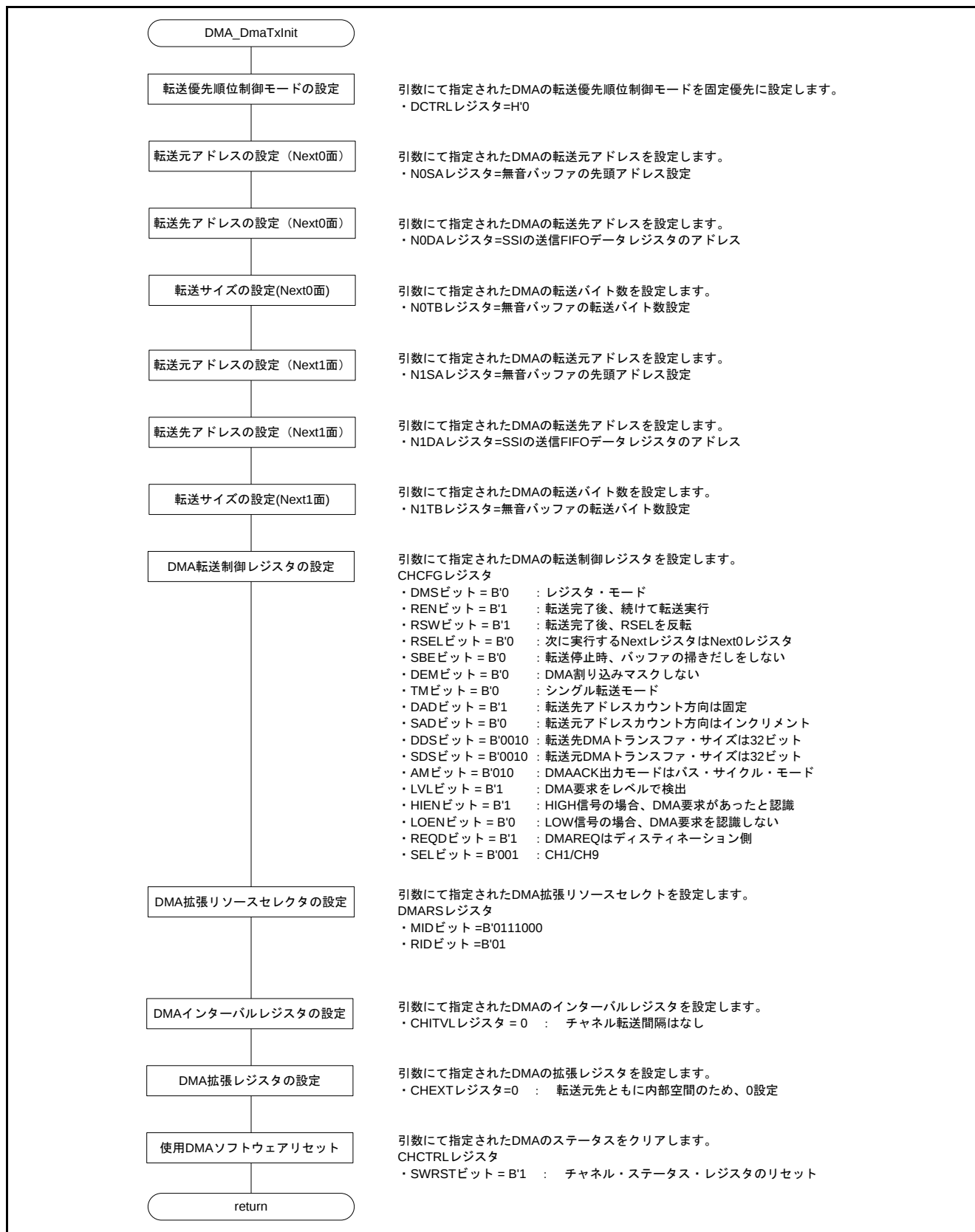


図 6.31 DMA(バッファ→SSI)初期化处理

6.10.22 DMA 転送元アドレス更新処理

図 6.32 に DMA 転送元アドレス更新処理のフローチャートを示します。

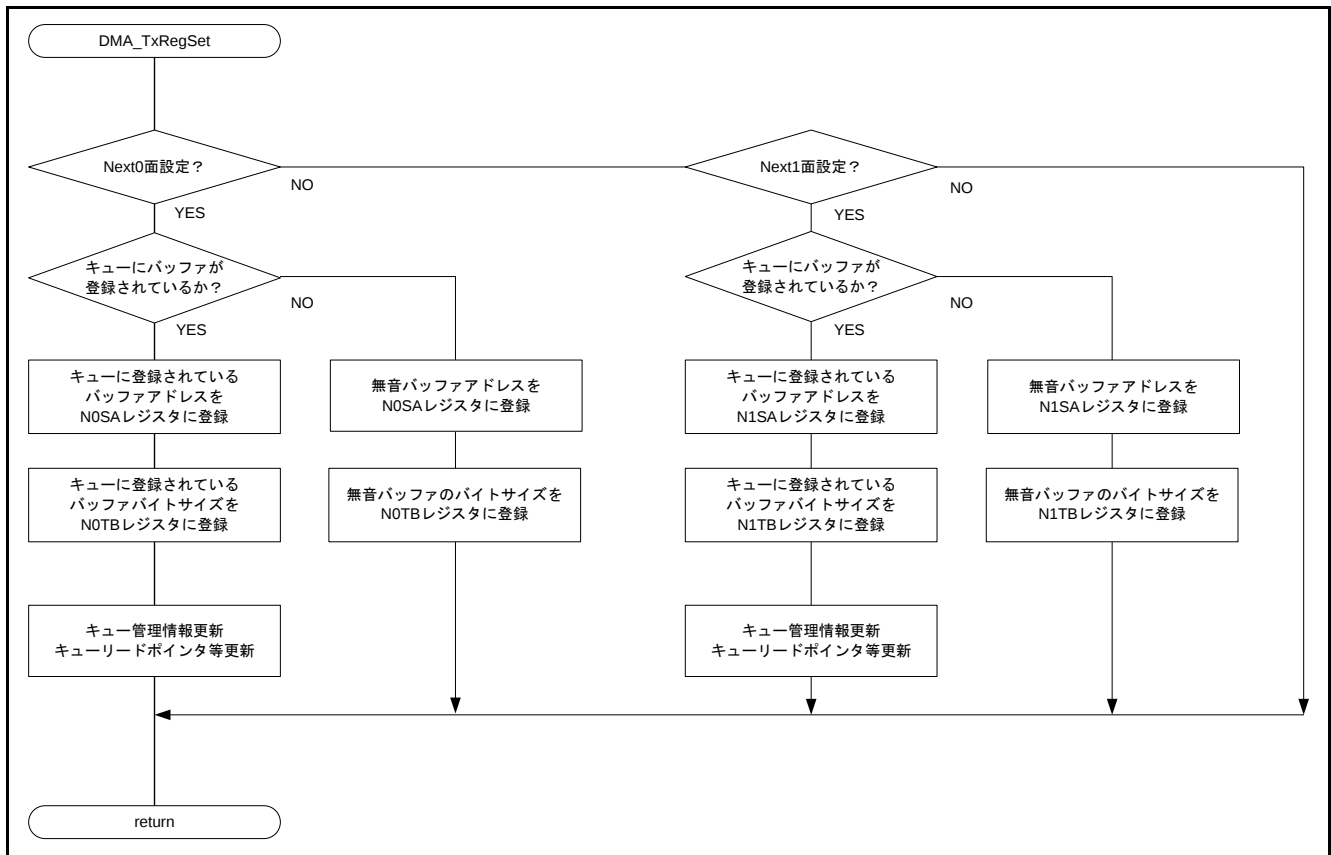


図 6.32 DMA 転送元アドレス更新処理

6.10.23 DMA(SS1→バッファ)初期化処理

図 6.33 に DMA(SS1→バッファ)初期化処理のフローチャートを示します。

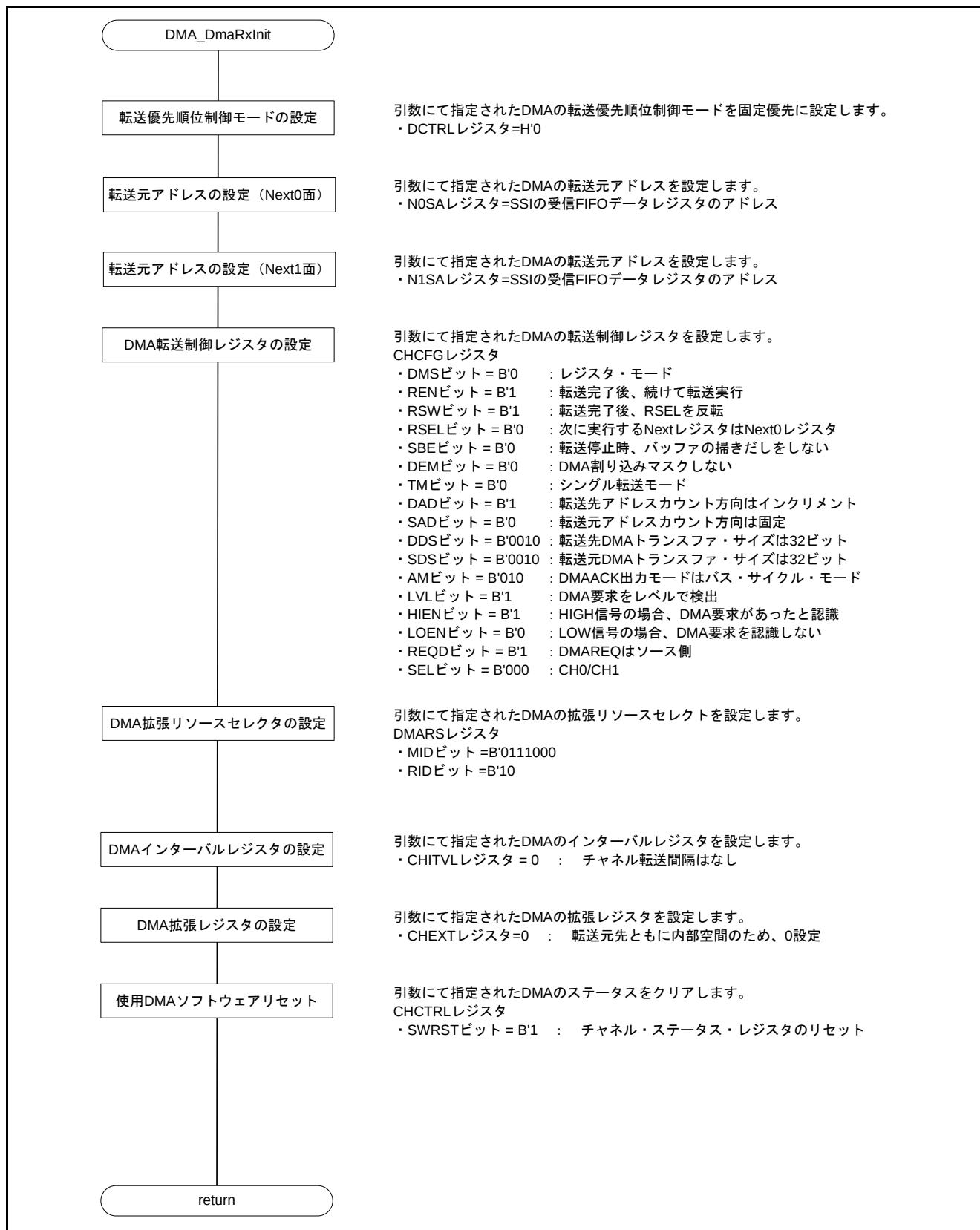


図 6.33 DMA(SS1→バッファ)初期化処理

6.10.24 DMA 転送先アドレス更新処理

図 6.34 に DMA 転送先アドレス更新処理のフローチャートを示します。

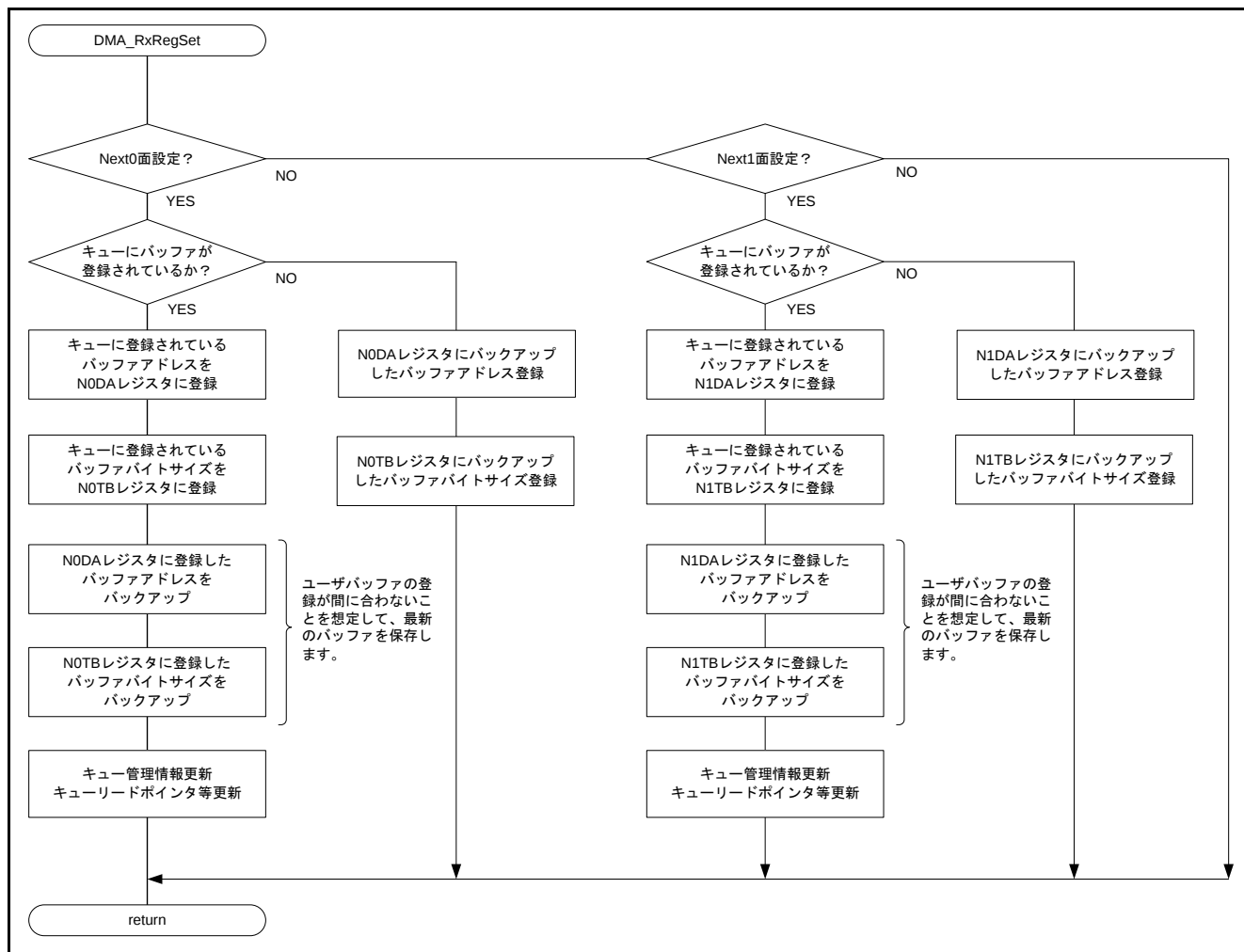


図 6.34 DMA 転送先アドレス更新処理

6.10.25 DMA 転送終了処理

図 6.35 に DMA 転送終了処理のフローチャートを示します。

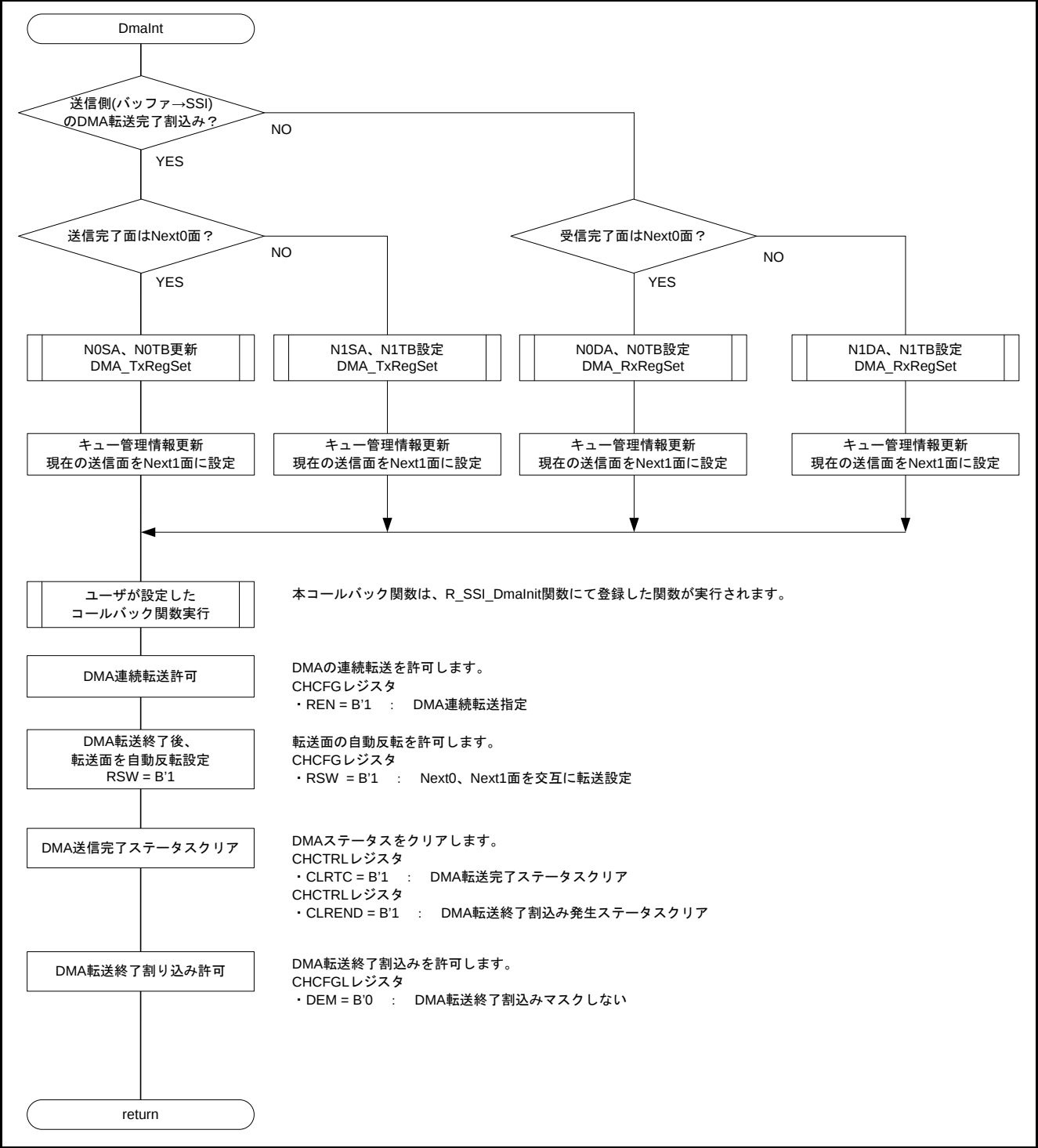


図 6.35 DMA 転送終了処理

7. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

8. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RZ/A1H グループ ユーザーズマニュアル ハードウェア編

(最新版をルネサス エレクトロニクスホームページから入手してください。)

R7S72100 CPU ボード (RTK772100BC00000BR) ユーザーズマニュアル

(最新版をルネサス エレクトロニクスホームページから入手してください。)

ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C

(最新版を ARM ホームページから入手してください。)

ARM Generic Interrupt Controller Architecture Specification Architecture version 1.0

(最新版を ARM ホームページから入手してください。)

テクニカルアップデート／テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

ユーザーズマニュアル：開発環境

ARM ソフトウェア開発ツール (ARM Compiler toolchain、ARM DS-5 等) に関しては、ARM ホームページから入手してください。

(最新版を ARM ホームページから入手してください。)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

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

お問合せ先

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

改訂記録	RZ/A1H グループ SSI サンプルプログラム アプリケーション ノート
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.5.28	—	初版発行
1.01	2014.7.16	—	複数コンパイラ対応

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。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

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

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

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

3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

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

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違うと、内部ROM、レイアウトパターンの相違などにより、電氣的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、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・システムの設計において、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て、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。
2. 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。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因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に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
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、産業用ロボット等
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、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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、社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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。
9. 本資料に記載されている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・システムに使用することはできません。また、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事務に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にご負担して頂きますのでご了承ください。
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禁じます。

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は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

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

■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/contact/>