

RZ/A1H グループ

R01AN1888JJ0103

Rev.1.03

ダイレクトメモリアクセスコントローラ (DMAC_RM) サンプルドライバ(PFV 付属)

2017.08.24

要旨

本アプリケーションノートでは、RZ/A1H のダイレクト メモリ アクセス コントローラ (DMAC) の機能を使用するサンプルドライバについて説明します。

本書は、PFV サンプルドライバに付属する DMAC を制御するプログラムについて説明します。

DMAC_RM プログラムの特長を以下に示します。

- DMAC のレジスターモード、または、リンク・モードで、DMA を起動、および、中止をします。
- DMA 拡張リソースセクターを使用することもできます。

対象デバイス

RZ/A1H グループ

RZ/A1M グループ

RZ/A1L グループ

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	9
6. ソフトウェア説明	10
6.1 動作概要	10
6.1.1 使用準備	11
6.2 メモリマップ	12
6.2.1 サンプルプログラムのセクション配置	13
6.2.2 MMU の設定	18
6.2.3 例外処理ベクタテーブル	19
6.3 使用割込み一覧	20
6.4 基本型	21
6.5 定数/列挙体/エラーコード	22
6.5.1 errnum_t 型 - エラーコード	22
6.5.2 dmac_bit_count_t	23
6.5.3 dmac_transfer_mode_t	23
6.5.4 dmac_ack_mode_t	23
6.5.5 dmac_edge_or_level_t	23
6.5.6 dmac_request_direction_t	23
6.5.7 dmac_pulse_or_level_t	24
6.5.8 dmac_priority_mode_t	24
6.5.9 dmac_interrupt_line_t	24
6.5.10 dmac_interrupt_lines_t	24
6.5.11 dmac_lm_header_t	24
6.6 構造体/共用体	26
6.6.1 dmac_register_mode_config_t	26
6.6.2 dmac_shared_config_t	29
6.6.3 dmac_rm_config_t	30
6.6.4 dmac_lm_descriptor_t	30
6.6.5 dmac_lm_descriptor_config_t	30
6.7 変数一覧	32
6.8 関数	33
6.8.1 一覧	33
6.8.2 DMAC を使用する関数	35
6.8.3 初期化前でも使える DMAC に関する関数	39
6.8.4 ドライバ移植層の関数	40
6.9 補足	42
6.9.1 RZ/A1H RTX BSP との連携	42
6.9.2 フラグド構造体パラメーター	42
7. サンプルコード	43
8. 参考ドキュメント	43
ホームページとサポート窓口	44

改訂記録	45
製品ご使用上の注意事項	46
ご注意書き	47

1. 仕様

表 1.1 に使用する周辺機能と用途を、図 1.1 にサンプルコード実行時の動作環境を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
ダイレクト メモリ アクセス コントローラ (DMAC) Ch(0~15)	データの転送メモリコントローラ
割込みコントローラ (INTC)	DMAC の割込み制御
シリアル コミュニケーション インターフェース (SCIF) (UART)	デバッグ情報表示

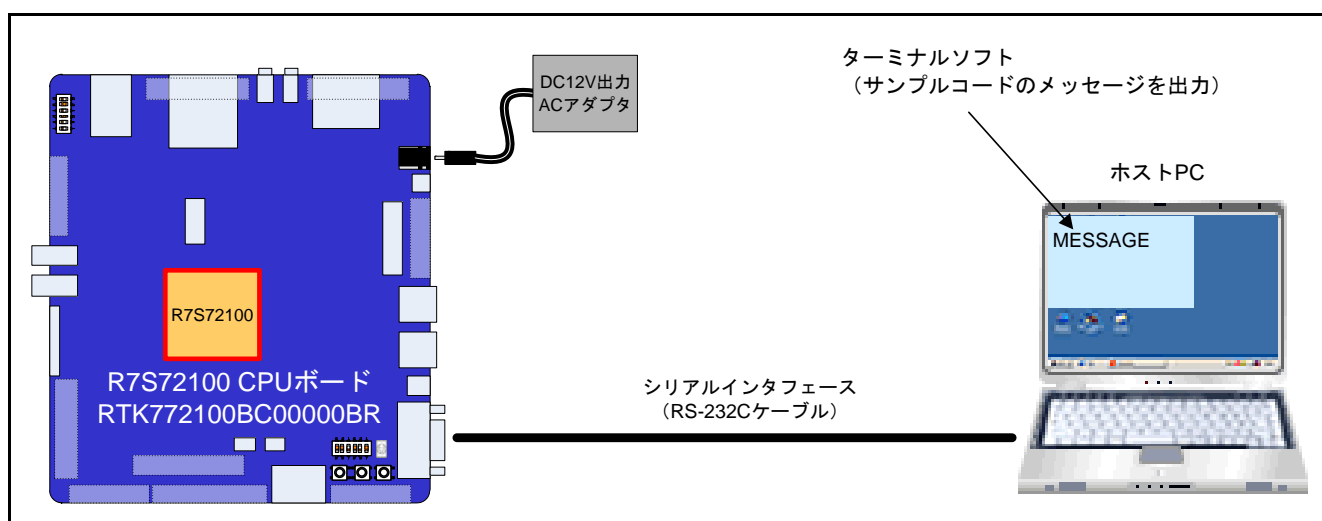


図 1.1 動作環境

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 (Iφ) : 400MHz 画像処理クロック (Gφ) : 266.67MHz 内部バスクロック (Bφ) : 133.33MHz 周辺クロック 1 (P1φ) : 66.67MHz 周辺クロック 0 (P0φ) : 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
ARM	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]
IAR	統合開発環境	IAR Embedded Workbench for ARM 7.80.4.12495
	C コンパイラ	
Renesas gcc	統合開発環境	e2 studio (Version: 5.3.0.023)
	C コンパイラ	GNUARM-NONE-EABI v16.01
動作モード		ブートモード 0 (CS0 空間 16 ビットブート)
ターミナルソフトの通信設定		・通信速度 : 115200bps ・データ長 : 8 ビット ・パリティ : なし ・ストップビット長 : 1 ビット ・フロー制御 : なし
使用ボード		GENMAI ボード ・R7S72100 CPU ボード RTK772100BC00000BR ・R7S72100 CPU ボード用オプションボード RTK7721000B00000BR
使用デバイス (ボード上で使用する機能)		・シリアルインタフェース (Dsub-9 コネクタ)

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 (R01AN1646JJ)
- RZ/A1H グループ レジスター定義ヘッダファイル iodef.h (R01AN1860JJ)
- RZ/A1H グループ OS 移植層 (OSPL) サンプルプログラム (R01AN1887JJ)
- RZ/A1H グループ ピクセル フォーマット コンバータ (PFV) サンプルドライバ (R01AN1880JJ)

4. 周辺機能説明

DMAC 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 5.2 にブロック図を示します。

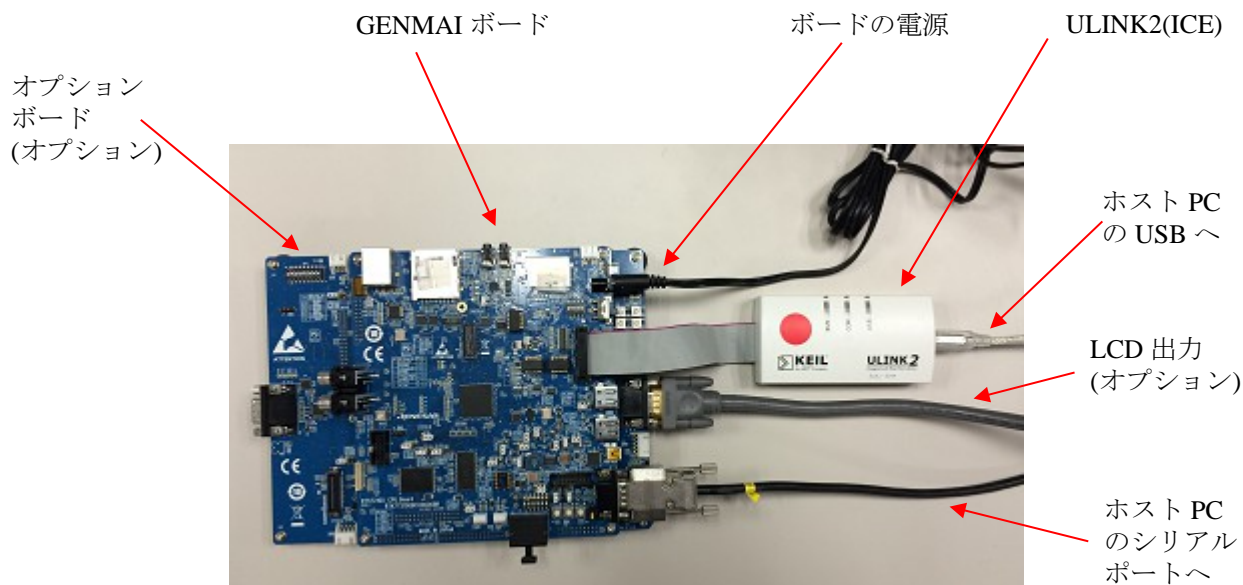


図 5.1 ハードウェア構成例

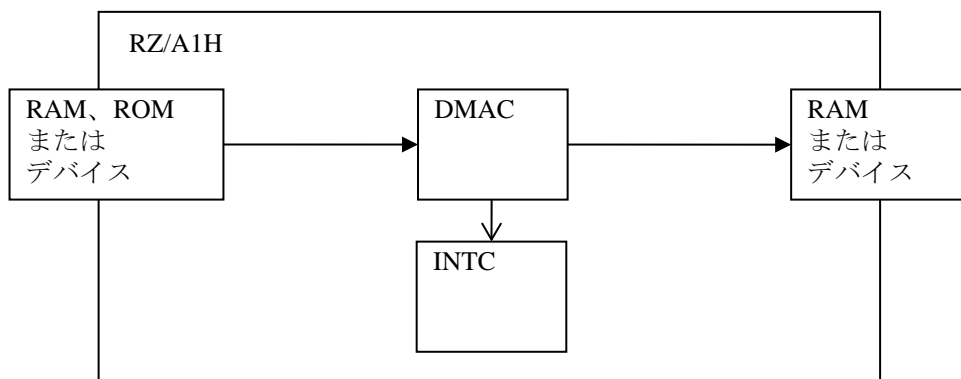


図 5.2 ブロック図

5.2 使用端子一覧

表 5.1 に使用端子と機能を示します。

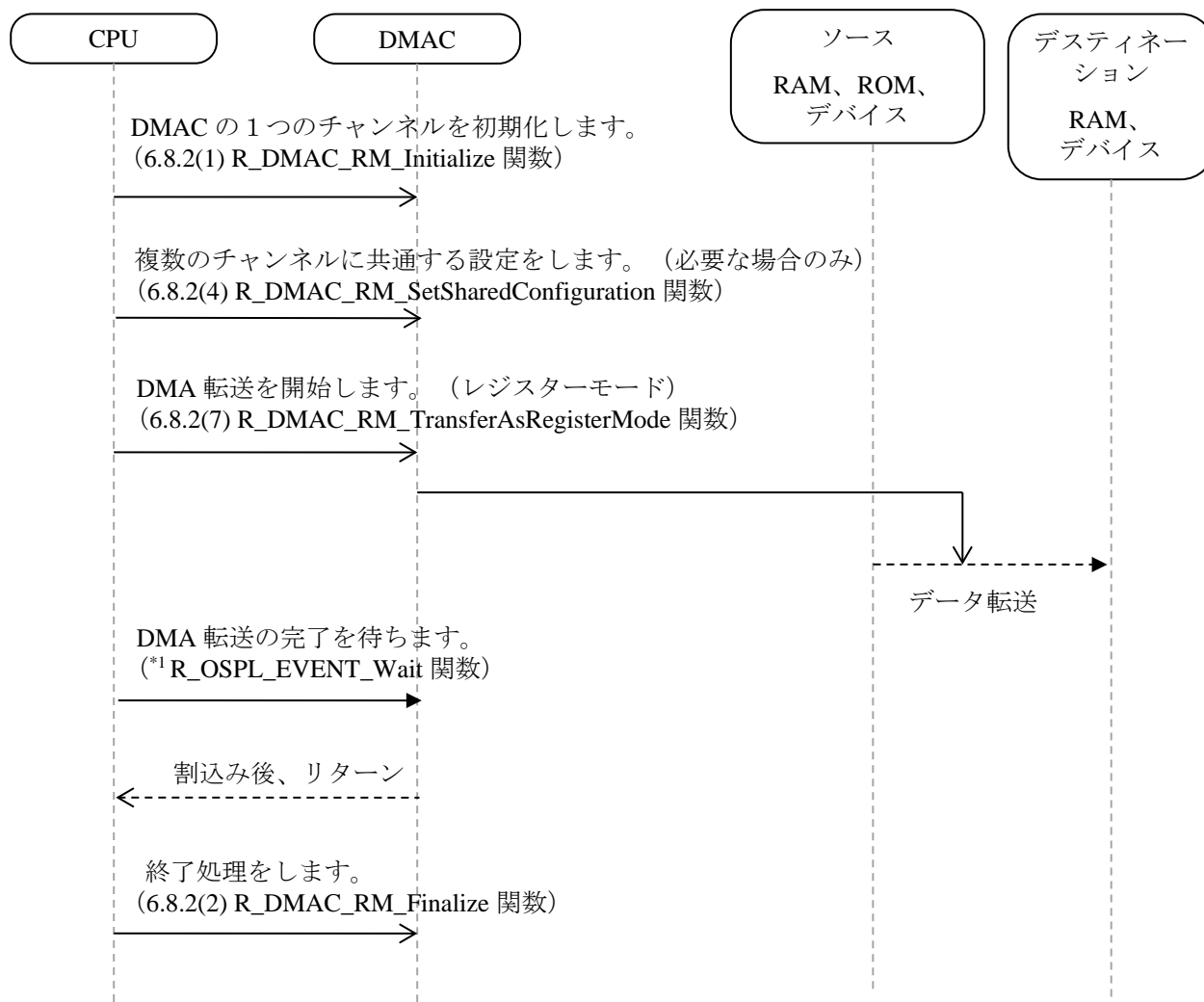
表 5.1 使用端子と機能

端子名	入出力	内容
なし		

6. ソフトウェア説明

6.1 動作概要

図 6.1 に動作シーケンスを示します。



*1 R_OSPL_EVENT_Wait 関数については、RZ/A1H グループ OS 移植層 OSPL (R01AN1887JJ) を参照。

図 6.1 動作シーケンス

6.1.1 使用準備

本サンプルプログラムの実行準備を説明します。

- 1) ホスト PC にてターミナルソフトを起動し、次のように設定します。(Tera Term の場合)



図 6.2 シリアルポートの設定

6.2 メモリマップ

図 6.3 に、RZ/A1H グループのアドレス空間と GENMAI ボード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		GENMAI ボードメモ リマッピング
ミラー空間	H'FFFF FFFF	その他 (2550MB)
	H'60A0 0000	大容量内蔵 RAM (10MB)
	H'6000 0000	大容量内蔵 RAM ミラー空間
	H'5C00 0000	SPI マルチ I/O バス 空間 2 (64MB)
	H'5800 0000	SPI マルチ I/O バス ミラー空間 2
		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
		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
通常空間	H'4000 0000	その他 (502MB)
		その他 (502MB)
	H'20A0 0000	大容量内蔵 RAM (10MB)
	H'2000 0000	大容量内蔵 RAM (10MB)
	H'1C00 0000	シリアルフラッシュ メモリ (64MB)
	H'1800 0000	シリアルフラッシュ メモリ (64MB)
		CS5 空間 (64MB) CS4 空間 (64MB)
		ユーザー領域
	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	NOR フラッシュ メモリ (64MB)
	H'0000 0000	NOR フラッシュ メモリ (64MB)

図 6.3 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 割込みハンドラーを大容量内蔵 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 割込みハンドラー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／リンカの使用／イメージの構造と生成の章」を参照してください。

表 6.1 に参考プログラムで使用するセクションを示し、

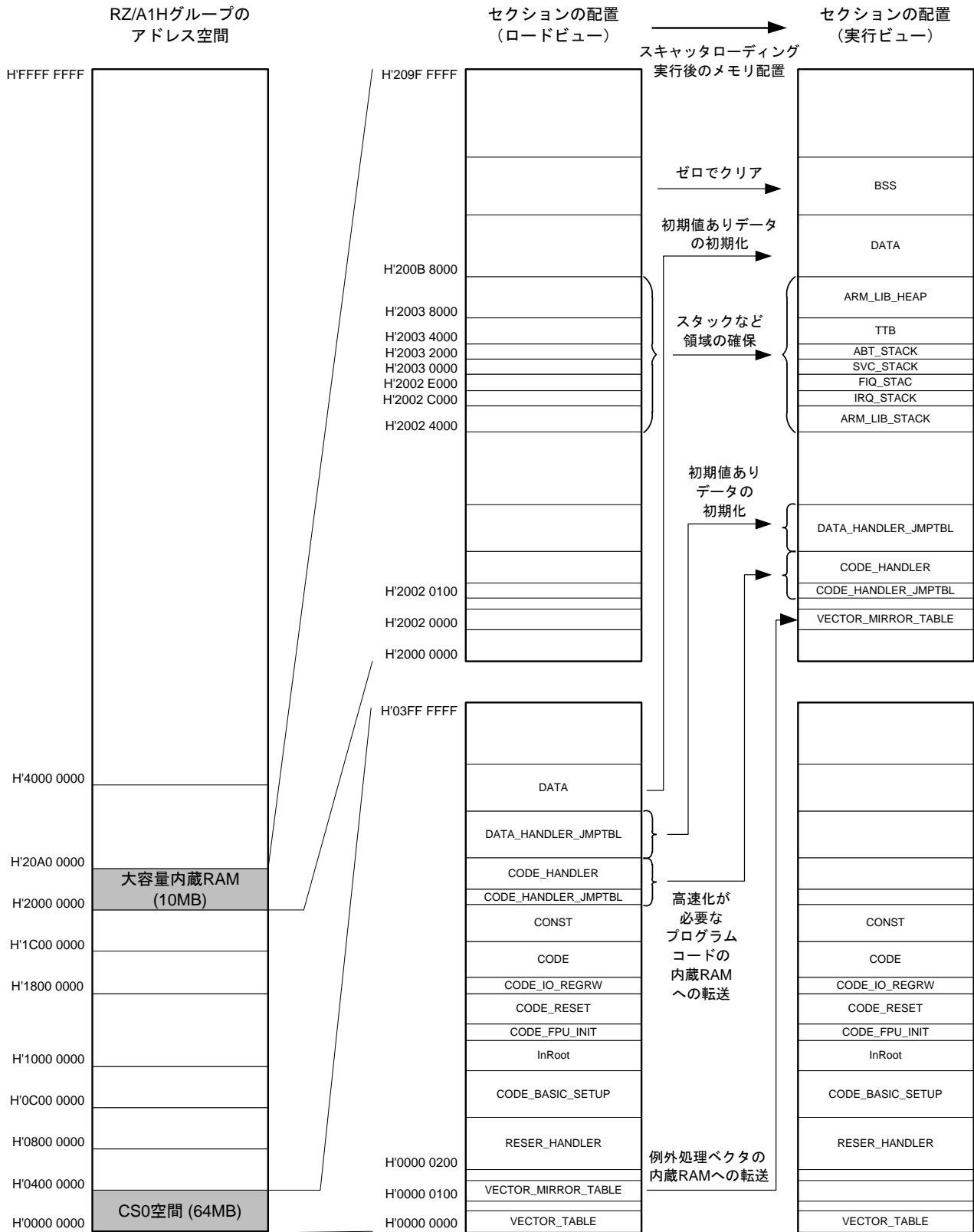


図 6.4 にサンプルプログラムの初期状態のセクション配置 (ロードビュー) と、スキッタローディング機能を使用後のセクション配置 (実行ビュー) を示します。

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	FLASH	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	FLASH	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
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	FLASH	LRAM
CODE_HANDLER_JMPTBL	IRQ 割込みハンドラのユーザー定義関数のプログラムコード領域	Code	FLASH	LRAM
CODE_HANDLER	IRQ 割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	FLASH	LRAM

DATA_HANDLER_JMPTBL	IRQ 割込みハンドラーのユーザー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

- 【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。

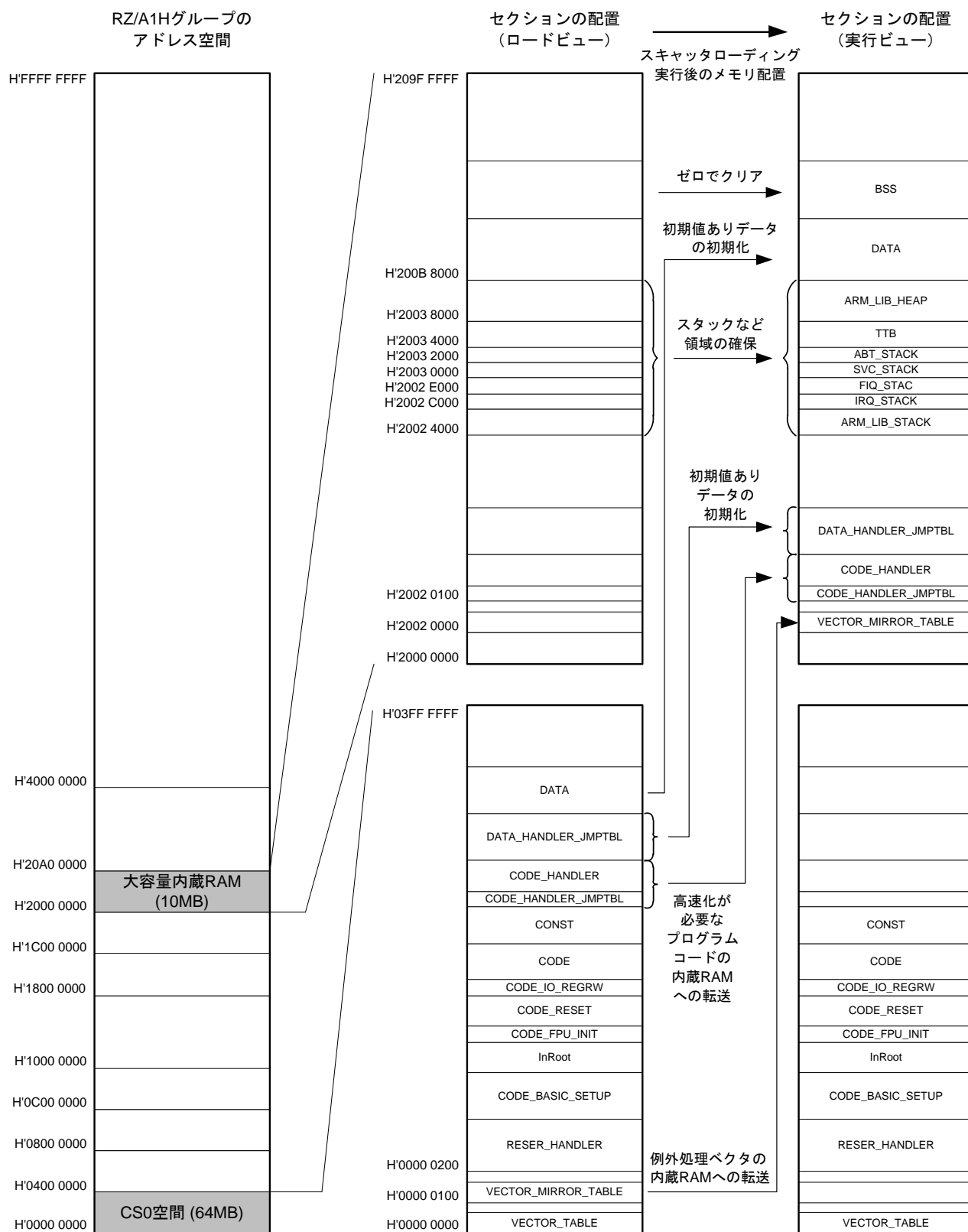


図 6.4 セクション配置

6.2.2 MMU の設定

RZ/A1H GENMAI ボード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 6.2 に、サンプルプログラムの MMU の設定を示します。

表 6.2 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 6.5 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 6.5 例外処理ベクタテーブルの記述例

6.3 使用割込み一覧

表 6.3 にサンプルコードで使用する割込みを示します。

表 6.3 サンプルコードで使用する割込み

割込み(要因 ID)	優先度	処理概要
DMAINT0～ DMAINT#	INTERRUPT_LEVEL_OF_INPUT_DMACH(=20) INTERRUPT_LEVEL_OF_OUTPUT_DMACH(=19)	DMA 転送完了割込みの受信

6.4 基本型

表 6.4 にサンプルコードで使用する基本型を示します。

表 6.4 サンプルコードで使用する基本型

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)。sizeof(bool_t) == sizeof(int)
bool8_t	8 ビット論理型。値は true (1), false (0)
bool16_t	16 ビット論理型。値は true (1), false (0)
bool32_t	32 ビット論理型。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)
int_fast8_t	少なくとも 8 ビットある、高速な整数、符号あり
int_fast16_t	少なくとも 16 ビットある、高速な整数、符号あり
int_fast32_t	少なくとも 32 ビットある、高速な整数、符号あり
uint_fast8_t	少なくとも 8 ビットある、高速な整数、符号なし
uint_fast16_t	少なくとも 16 ビットある、高速な整数、符号なし
uint_fast32_t	少なくとも 32 ビットある、高速な整数、符号なし
uintptr_t	ポインタと同じビット数の符号なし整数型、または、物理アドレス
size_t	ポインタと同じビット数の符号なし整数型、または、バイトサイズ
ssize_t	ポインタと同じビット数の符号あり整数型
ptrdiff_t	ポインタと同じビット数の符号あり整数型、アドレスの差分
bit_flags_fast32_t	uint_fast32_t 型の ビットフラグ型 (ビットフィールド型)
bit_flags32_t	uint32_t 型の ビットフラグ型 (ビットフィールド型)
bit_flags16_t	uint16_t 型の ビットフラグ型 (ビットフィールド型)
bit_flags8_t	uint8_t 型の ビットフラグ型 (ビットフィールド型)
errnum_t	エラーコード、0=エラーなし。参考(6.5.1)
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/列挙体/エラーコード

章	型名	内容
6.5.1	errnum_t	エラーコード
6.5.2	dmac_bit_count_t	DMA トランスファーサイズ
6.5.3	dmac_transfer_mode_t	DMA 転送モード
6.5.4	dmac_ack_mode_t	DMAACK 出力モード
6.5.5	dmac_edge_or_level_t	DMA 要求を、信号のレベルで検出するかエッジで検出するか
6.5.6	dmac_request_direction_t	DMAREQ が、ソース側かデスティネーション側のどちらであるか
6.5.7	dmac_pulse_or_level_t	DMA 割込み、DMAERR をパルスで出力するか、レベルで出力するか
6.5.8	dmac_priority_mode_t	チャンネル間の転送優先順位制御モード
6.5.9	dmac_interrupt_line_t	DMAC から発生した割込み線
6.5.10	dmac_interrupt_lines_t	dmac_interrupt_line_t 型のビットフラグ
6.5.11	dmac_lm_header_t	リンク・モードで使用するディスクリプターのヘッダの値
-	r_ospl_axi_cache_attribute_t	AXI バスの先にある L2 キャッシュのキャッシュ属性 * ¹
-	r_ospl_axi_protection_t	AXI バスの先にある L2 キャッシュの保護属性 * ¹

6.5.1 errnum_t 型 - エラーコード

シンボル	設定値	内容
0	0	エラーなし
E_OTHERS	1	その他のエラー
E_FEW_ARRAY	2	固定長配列が不足したときのエラー
E_FEW_MEMORY	3	ヒープメモリ不足
E_FIFO_OVER	4	キューに入りきらなかったエラー
E_NOT_FOUND_SYMBOL	5	シンボルが定義されていない
E_NO_NEXT	6	次のリスト要素がない
E_ACCESS_DENIED	7	読み込み書き込み拒否エラー
E_NOT_IMPLEMENT_YET	8	未実装
E_ERRNO	0x0E(=14)	errno を参照
E_LIMITATION	0x0F(=15)	暫定の制限事項
E_STATE	0x10(=16)	現在の状態では実行できないというエラー
E_NOT_THREAD	0x11(=17)	スレッドではないエラー、割込みコンテキストから呼べない
E_PATH_NOT_FOUND	0x12(=18)	ファイルやフォルダーが見つからない
E_BAD_COMMAND_ID	0x16(=22)	コマンド ID が範囲外
E_TIME_OUT	0x17(=23)	タイムアウト
E_STACK_OVERFLOW	0x1C(=28)	スタック オーバーフロー
E_NO_DEBUG_TLS	0x1D(=29)	デバッグ用ワーク領域がない
E_EXIT_TEST	0x1E(=30)	テストの停止要求

¹ RZ/A1H グループ OS 移植層 OSPL (R01AN1887JJ)

6.5.2 dmac_bit_count_t

DMA トランスファーサイズ。

シンボル	値	内容
DMAC_8_BIT	0	8 ビット（初期値）
DMAC_16_BIT	1	16 ビット
DMAC_32_BIT	2	32 ビット
DMAC_64_BIT	3	64 ビット
DMAC_128_BIT	4	128 ビット
DMAC_256_BIT	5	256 ビット
DMAC_512_BIT	6	512 ビット
DMAC_1024_BIT	7	1024 ビット

6.5.3 dmac_transfer_mode_t

DMA 転送モード。

シンボル	値	内容
DMAC_SINGLE_TRANSFER_0	0	シングル転送モード（初期値）
DMAC_BLOCK_TRANSFER_1	1	ブロック転送モード

6.5.4 dmac_ack_mode_t

DMAACK 出力モード。

シンボル	値	内容
DMAC_ACK_MODE_RESET_VALUE	0	（初期値）
DMAC_ACK_LEVEL_MODE_001	1	レベルモード
DMAC_ACK_BUS_CYCLE_MODE_010	2	バスサイクルモード
DMAC_ACK_NOT_OUTPUT_100	4	DMAACK を出力しない

6.5.5 dmac_edge_or_level_t

DMA 要求を、信号のレベルで検出するかエッジで検出するか。

シンボル	値	内容
DMAC_EDGE_0	0	エッジで検出します（初期値）
DMAC_LEVEL_1	1	レベルで検出します

6.5.6 dmac_request_direction_t

DMAREQ が、ソース側かデスティネーション側のどちらであるか。

シンボル	値	内容
DMAC_DMAREQ_SOURCE_0	0	ソース側、DMAACK はリード時にアクティブ（初期値）

DMAC_DMAREQ_DESTINATION_1	1	デスティネーション側、DMAACK はライト時にアクティブ
---------------------------	---	-------------------------------

6.5.7 dmac_pulse_or_level_t

DMA 割込みや、DMAERR をパルスで出力するか、レベルで出力するか。

シンボル	値	内容
DMAC_PULSE	0	パルス出力（初期値）
DMAC_LEVEL	1	レベル出力

6.5.8 dmac_priority_mode_t

チャンネル間の転送優先順位制御モード。

シンボル	値	内容
DMAC_PRIORITY_FIXED	0	固定優先順位モード（初期値）
DMAC_PRIORITY_ROUNDROBIN	1	ラウンドロビンモード

参考：RZ/A1H グループ ユーザーズマニュアル ハードウェア編 - 9.7.2 DMA チャンネルの優先順位制御

6.5.9 dmac_interrupt_line_t

DMAC から発生した割込み線。

シンボル	値	内容
DMAC_INTERRUPT_LINE_TRANSFER_END	0x00000020	DMA トランザクションが完了
DMAC_INTERRUPT_LINE_ERROR	0x00000010	DMA 転送中に、ERROR レスポンスを受信

6.5.10 dmac_interrupt_lines_t

dmac_interrupt_line_t 型のビットフラグ。

6.5.11 dmac_lm_header_t

リンク・モードで使用するディスクリプターのヘッダの値。

シンボル	値	内容
F_DMAC_LM_DESCRIPTOR_INTERRUPT_MASK	0x00000008	通常、0 に設定します。 0 = F_DMAC_LM_LINK_VALID (LV) = 0 のときに割込みが発生します。 1 = 割込みが発生しません。 DMA 転送終了時の割込みの有無は、ディスクリプター内の CHCFG_n レジスタの DEM ビットで設定してください。割込みコールバック関数で、DMAC の停止やリセットをしているときは、リストの最後以外では割込みが発生させないようにしてください。

F_DMAC_LM_WRITE_BACK_DISABLE	0x00000004	<p>通常、繰り返すディスクリプターなら 1、そうでないなら 0 に設定します。</p> <p>0 = DMA 転送終了時に、F_DMAC_LM_LINK_VALID (LV) = 0 にします。</p> <p>1 = F_DMAC_LM_LINK_VALID (LV) = 0 にしません。</p> <p>0 にすると、LV の値で、DMA 転送が完了したディスクリプターかどうかを確認できます。</p>
F_DMAC_LM_LINK_END	0x00000002	<p>リストの最後以外のディスクリプターなら 0、最後なら 1 に設定します。</p> <p>0 = next_descriptor_link_address が有効。 次のディスクリプターを引き続き使う。</p> <p>1 = 現在のディスクリプターの DMA 転送で終了する。</p>
F_DMAC_LM_LINK_VALID	0x00000001	<p>通常、1 に設定します。</p> <p>1 = ディスクリプターとして使い、DMA 転送を起動します。</p> <p>0 = DMA 転送を起動しません。</p> <p>F_DMAC_LM_WRITE_BACK_DISABLE (WBD) = 0 なら、DMA 転送終了時に F_DMAC_LM_LINK_VALID (LV) = 0 になります。</p>

6.6 構造体/共用体

章	型名	概要
6.6.1	dmac_register_mode_config_t	R_DMAC_RM_StartAsRegisterMode 関数のパラメーター
6.6.2	dmac_shared_config_t	R_DMAC_RM_SetSharedConfiguration 関数のパラメーター
6.6.3	dmac_rm_config_t	レジスターモードの設定
6.6.4	dmac_lm_descriptor_t	リンク・モードのディスクリプター
6.6.5	dmac_lm_descriptor_config_t	R_DMAC_RM_StartAsLinkMode 関数のパラメーター
-	r_ospl_async_t	非同期関数に指定する通知設定 ^{*2}
-	dmac_rm_async_status_t	r_ospl_async_status_t 型と同じ ^{*2}

6.6.1 dmac_register_mode_config_t

概 要	R_DMAC_RM_StartAsRegisterMode 関数のパラメーター
ヘッダ	r_dmac_rm.h
説 明	
メンバー変数	bit_flags32_t flags

フラグド構造体パラメーター。参考 : (6.9.1)
F_DMAC_NEXT0_SOURCE_PHYSICAL_ADDRESS
F_DMAC_NEXT0_DESTINATION_PHYSICAL_ADDRESS
F_DMAC_NEXT0_TRANSFER_BYTE
F_DMAC_NEXT1_SOURCE_PHYSICAL_ADDRESS
F_DMAC_NEXT1_DESTINATION_PHYSICAL_ADDRESS
F_DMAC_NEXT1_TRANSFER_BYTE
F_DMAC_IS_SOURCE_ADDRESS_FIXED
F_DMAC_SOURCE_BIT_COUNT
F_DMAC_IS_DESTINATION_ADDRESS_FIXED
F_DMAC_DESTINATION_BIT_COUNT
F_DMAC_IS_RELOAD_NEXT_REGISTER_SET
F_DMAC_IS_AUTOMATIC_SWITCH_NEXT_NUM
F_DMAC_NEXT_REGISTER_SET_NUMBER
F_DMAC_IS_SWEEP_BUFFER
F_DMAC_IS_DISABLE_NEXT_END_INTERRUPT
F_DMAC_TRANSFER_MODE_TM
F_DMAC_IS_DMA_ACK_OUTPUT_MODE_AM
F_DMAC_REQUEST_RESOURCE_MID
F_DMAC_REQUEST_RESOURCE_RID
F_DMAC_REQUEST_EDGE_OR_LEVEL_LVL
F_DMAC_IS_REQUEST_RAISING_OR_HIGH_HIEN
F_DMAC_IS_REQUEST_FALLING_OR_LOW_LOEN
F_DMAC_REQUEST_DIRECTION_REQD

² RZ/A1H グループ OS 移植層 OSPL (R01AN1887JJ)

	F_DMAC_INTERVAL_COUNT F_DMAC_SOURCE_AXI_CACHE_ATTRIBUTE F_DMAC_SOURCE_AXI_PROTECTION F_DMAC_DESTINATION_AXI_CACHE_ATTRIBUTE F_DMAC_DESTINATION_AXI_PROTECTION
uintptr_t next0_source_physical_address	N0SA_n レジスターに設定する値。 次に実行する DMA 転送元アドレス(Next0) 必須。
uintptr_t next0_destination_physical_address	N0DA_n レジスターに設定する値。 次に実行する DMA 転送先アドレス(Next0) 必須。
size_t next0_transfer_byte	N0TB_n レジスターに設定する値。 次に実行する DMA の総転送バイト数(Next0) 必須。
uintptr_t next1_source_physical_address	N1SA_n レジスターに設定する値。 次に実行する DMA 転送元アドレス(Next1)
uintptr_t next1_destination_physical_address	N1DA_n レジスターに設定する値。 次に実行する DMA 転送先アドレス(Next1)
size_t next1_transfer_byte	N1TB_n レジスターに設定する値。 次に実行する DMA の総転送バイト数(Next1)
bool8_t is_source_address_fixed	CHCFG_n レジスターの SAD ビットに設定する値。 false なら DMA 転送元アドレスを+1 する。 省略時は、DMA 転送先アドレス+1 する。
dmac_bit_count_t source_bit_count	CHCFG_n レジスターの SDS ビットに設定する値。 転送元 DMA トランスファーサイズ。参照(6.5.2) 省略時は、DMAC_8_BIT。
bool8_t is_destination_address_fixed	CHCFG_n レジスターの DAD ビットに設定する値。 false なら DMA 転送先アドレス+1 する。 省略時は、DMA 転送先アドレス+1 する。
dmac_bit_count_t destination_bit_count	CHCFG_n レジスターの DDS ビットに設定する値。 転送先 DMA トランスファーサイズ。参照(6.5.2) 省略時は、DMAC_8_BIT。
bool8_t is_reload_next_register_set	CHCFG_n レジスターの REN ビットに設定する値。 DMA トランザクション完了後に、続けて、 next_register_set_number (RSEL ビット) で選択されている Next レジスターセットで DMA 転送を行うかどうか。 省略時は、false。
bool8_t is_automatic_switch_next_num	CHCFG_n レジスターの RSW ビットに設定する値。 DMA トランザクション終了後に、 next_register_set_number (RSEL ビット) を自動で反転する かどうか。 省略時は、false。

int8_t next_register_set_number	CHCFG_n レジスタの RSEL ビットに設定する値。 0(Next0) or 1(Next1)。 省略時は、0。
bool8_t is_sweep_buffer	CHCFG_n レジスタの SBE ビットに設定する値。 DMA トランザクション中にイネーブルを 0 にクリアした場合、すでにリードしてバッファに取り込んでいるデータを掃き出して（ライトして）停止するか否か。 省略時は、false。
bool8_t is_disable_next_end_interrupt	CHCFG_n レジスタの DEM ビットに設定する値。 次の DMA 割込みをマスクするかどうか。 省略時は、false。
dmac_transfer_mode_t transfer_mode_TM	CHCFG_n レジスタの TM ビットに設定する値。 DMA 転送モード。参照(6.5.3) 省略時は、request_resource_MID が指定されていなければ、メモリ to メモリ転送などのオートリクエストモードを想定した DMAC_BLOCK_TRANSFER_1。 request_resource_MID が指定されていれば省略できない。
uint8_t is_DMA_ack_output_mode_AM	CHCFG_n レジスタの AM ビットに設定する値。 DMAACK 出力モード。参照(6.5.4) 省略時は、request_resource_MID が指定されていなければ、メモリ to メモリ転送などのオートリクエストモードを想定した DMAC_ACK_NOT_OUTPUT_100。 request_resource_MID が指定されていれば省略できない。
int_t request_resource_MID	DMARSn レジスタの MID ビットに設定する値。 DMA 転送を開始する割込みを発生するデバイス。 ハードウェアマニュアル 7.5 割込み ID を参照。 省略時はデバイスと連携しないで、メモリ to メモリ転送などのオートリクエストモードで動作します。
int_t request_resource_RID	DMARSn レジスタの RID ビットに設定する値。 DMA 転送を開始する割込みを発生するデバイス。 ハードウェアマニュアル 7.5 割込み ID を参照。 request_resource_MID が指定されていれば必須。
dmac_edge_or_level_t request_edge_or_level_LVL	CHCFG_n レジスタの LVL ビットに設定する値。 割込み信号の形状。 ハードウェアマニュアル 7.5 割込み ID を参照。 request_resource_MID が指定されていれば必須。
bool8_t is_request_raising_or_high_HIEN	CHCFG_n レジスタの HIEN ビットに設定する値。 割込み信号の形状。 ハードウェアマニュアル 7.5 割込み ID を参照。 request_resource_MID が指定されていれば必須。
bool8_t is_request_falling_or_low_LOEN	CHCFG_n レジスタの LOEN ビットに設定する値。 割込み信号の形状。 ハードウェアマニュアル 7.5 割込み ID を参照。 request_resource_MID が指定されていれば必須。
dmac_request_direction_t	CHCFG_n レジスタの REQD ビットに設定する値。 割込みを発生するデバイスの入力側か出力側か。

request_direction_REQ D	ハードウェアマニュアル 7.5 割込み ID を参照。 request_resource_MID が指定されていれば必須。
int_fast32_t interval_count	CHITVL_n レジスタの ITVL ビットに設定する値。 インターバル・カウント。1 回のリードまたはライトが完了してから次にリードまたはライトまで待つカウント数。単位は、Bφクロック。省略時は 0。
r_ospl_axi_cache_attribute_t source_AXI_cache_attribute	CHEXT_n レジスタの SCA ビットに設定する値。 DMAC がリードするときの AXI バスの先にある L2 キャッシュのキャッシュ属性。省略時は、内部バス用の R_OSPL_AXI_CACHE_ZERO。
r_ospl_axi_protection_t source_AXI_protection	CHEXT_n レジスタの SPR ビットに設定する値。 DMAC がリードするときの AXI バスの先にある L2 キャッシュの保護属性。省略時は、内部バス用の R_OSPL_AXI_PROTECTION_ZERO。
r_ospl_axi_cache_attribute_t destination_AXI_cache_attribute	CHEXT_n レジスタの DCA ビットに設定する値。 DMAC がライトするときの AXI バスの先にある L2 キャッシュのキャッシュ属性。省略時は、内部バス用の R_OSPL_AXI_CACHE_ZERO。
r_ospl_axi_protection_t destination_AXI_protection	CHEXT_n レジスタの DPR ビットに設定する値。 DMAC がライトするときの AXI バスの先にある L2 キャッシュの保護属性。省略時は、内部バス用の R_OSPL_AXI_PROTECTION_ZERO。

6.6.2 dmact_shared_config_t

概 要	R_DMAM_RM_SetSharedConfiguration 関数のパラメーター										
ヘッダ	r_dmam_rm.h										
説 明	参照：ハードウェアマニュアルの DCTRL_0_7 レジスタ。										
メンバー変数	<table border="1"> <tr> <td>bit_flags_fast32_t flags</td><td>フラグド構造体パラメーター。参考：(6.9.1) F_DMAM_LINK_WRITE_BACK_CACHE, F_DMAM_LINK_WRITE_BACK_PROTECTION, F_DMAM_LINK_DESCRIPTOR_CACHE, F_DMAM_LINK_DESCRIPTOR_PROTECTION, F_DMAM_OUTPUT_INTERRUPT_PULSE_OR_LEVEL, F_DMAM_PRIORITY_MODE</td></tr> <tr> <td>r_ospl_axi_cache_attribute_t link_write_back_cache</td><td>リンク・モードのディスクリプターライトバック時に AWCACHE[3:0]に出力する値。 初期値は、R_OSPL_AXI_CACHE_ZERO。省略時は、 設定値を変更しない。</td></tr> <tr> <td>r_ospl_axi_protection_t link_write_back_protection</td><td>リンク・モードのディスクリプターライトバック時に AWPROT[2:0]に出力する値。 初期値は、R_OSPL_AXI_PROTECTION_ZERO。省略 時は、設定値を変更しない。</td></tr> <tr> <td>r_ospl_axi_cache_attribute_t link_descriptor_cache</td><td>リンク・モードのディスクリプターロード時に ARCACHE[3:0]に出力する値。 初期値は、R_OSPL_AXI_CACHE_ZERO。省略時は、 設定値を変更しない。</td></tr> <tr> <td>r_ospl_axi_protection_t link_descriptor_protection</td><td>リンク・モードのディスクリプターロード時に ARPROT[2:0]に出力する値。</td></tr> </table>	bit_flags_fast32_t flags	フラグド構造体パラメーター。参考：(6.9.1) F_DMAM_LINK_WRITE_BACK_CACHE, F_DMAM_LINK_WRITE_BACK_PROTECTION, F_DMAM_LINK_DESCRIPTOR_CACHE, F_DMAM_LINK_DESCRIPTOR_PROTECTION, F_DMAM_OUTPUT_INTERRUPT_PULSE_OR_LEVEL, F_DMAM_PRIORITY_MODE	r_ospl_axi_cache_attribute_t link_write_back_cache	リンク・モードのディスクリプターライトバック時に AWCACHE[3:0]に出力する値。 初期値は、R_OSPL_AXI_CACHE_ZERO。省略時は、 設定値を変更しない。	r_ospl_axi_protection_t link_write_back_protection	リンク・モードのディスクリプターライトバック時に AWPROT[2:0]に出力する値。 初期値は、R_OSPL_AXI_PROTECTION_ZERO。省略 時は、設定値を変更しない。	r_ospl_axi_cache_attribute_t link_descriptor_cache	リンク・モードのディスクリプターロード時に ARCACHE[3:0]に出力する値。 初期値は、R_OSPL_AXI_CACHE_ZERO。省略時は、 設定値を変更しない。	r_ospl_axi_protection_t link_descriptor_protection	リンク・モードのディスクリプターロード時に ARPROT[2:0]に出力する値。
bit_flags_fast32_t flags	フラグド構造体パラメーター。参考：(6.9.1) F_DMAM_LINK_WRITE_BACK_CACHE, F_DMAM_LINK_WRITE_BACK_PROTECTION, F_DMAM_LINK_DESCRIPTOR_CACHE, F_DMAM_LINK_DESCRIPTOR_PROTECTION, F_DMAM_OUTPUT_INTERRUPT_PULSE_OR_LEVEL, F_DMAM_PRIORITY_MODE										
r_ospl_axi_cache_attribute_t link_write_back_cache	リンク・モードのディスクリプターライトバック時に AWCACHE[3:0]に出力する値。 初期値は、R_OSPL_AXI_CACHE_ZERO。省略時は、 設定値を変更しない。										
r_ospl_axi_protection_t link_write_back_protection	リンク・モードのディスクリプターライトバック時に AWPROT[2:0]に出力する値。 初期値は、R_OSPL_AXI_PROTECTION_ZERO。省略 時は、設定値を変更しない。										
r_ospl_axi_cache_attribute_t link_descriptor_cache	リンク・モードのディスクリプターロード時に ARCACHE[3:0]に出力する値。 初期値は、R_OSPL_AXI_CACHE_ZERO。省略時は、 設定値を変更しない。										
r_ospl_axi_protection_t link_descriptor_protection	リンク・モードのディスクリプターロード時に ARPROT[2:0]に出力する値。										

	初期値は、R_OSPL_AXI_PROTECTION_ZERO。省略時は、設定値を変更しない。
dmac_pulse_or_level_t output_interrupt_pulse_or_level	DMA 割込み、DMAERR をパルスで出力するか、レベルで出力するか。参照：(6.5.7) 初期値は、DMAC_PULSE。省略時は、設定値を変更しない。
dmac_priority_mode_t priority_mode	チャンネル間の転送優先順位制御モード。参照：(6.5.8) 初期値は、DMAC_PRIORITY_FIXED。省略時は、設定値を変更しない。

6.6.3 dmac_rm_config_t

概要	レジスターモードの設定				
ヘッダ	r_dmac_rm.h				
説明	R_DMAMC_RM_InitializeEx 関数に指定します。				
メンバー変数	<table border="1"> <tr> <td>bit_flags32_t flags</td><td>フラグド構造体パラメーター。参考：(6.9.1) F_DMAMC_CHANNEL_NUM</td></tr> <tr> <td>int_fast32_t channel_num</td><td>DMAC のチャンネル番号。 R_DMAMC_RM_InitializeEx 関数の channel_num 引数の値が入ります。ただし、channel_num 引数が R_OSPL_UNLOCKED_CHANNEL だったときは、未使用だったチャンネル番号が入ります。</td></tr> </table>	bit_flags32_t flags	フラグド構造体パラメーター。参考：(6.9.1) F_DMAMC_CHANNEL_NUM	int_fast32_t channel_num	DMAC のチャンネル番号。 R_DMAMC_RM_InitializeEx 関数の channel_num 引数の値が入ります。ただし、channel_num 引数が R_OSPL_UNLOCKED_CHANNEL だったときは、未使用だったチャンネル番号が入ります。
bit_flags32_t flags	フラグド構造体パラメーター。参考：(6.9.1) F_DMAMC_CHANNEL_NUM				
int_fast32_t channel_num	DMAC のチャンネル番号。 R_DMAMC_RM_InitializeEx 関数の channel_num 引数の値が入ります。ただし、channel_num 引数が R_OSPL_UNLOCKED_CHANNEL だったときは、未使用だったチャンネル番号が入ります。				

6.6.4 dmac_lm_descriptor_t

概要	リンク・モードのディスクリプター																
ヘッダ	r_dmac_lm.h																
説明	R_DMAMC_LM_DESCRIPTOR_Initialize 関数で、すべてのメンバー変数を設定できます。																
メンバー変数	<table border="1"> <tr> <td>bit_flags32_t header</td><td>ディスクリプターの各種属性。 dmac_lm_header_t 型の値の論理和</td></tr> <tr> <td>uintptr_t source_physical_address</td><td>N0SA_n レジスターに設定する値。 DMA の転送元アドレス</td></tr> <tr> <td>uintptr_t destination_physical_address</td><td>N0DA_n レジスターに設定する値。 DMA の転送先アドレス</td></tr> <tr> <td>uint32_t transfer_byte</td><td>N0TB_n レジスターに設定する値。 DMA の総転送バイト数</td></tr> <tr> <td>uint32_t CHCFG_n</td><td>CHCFG_n レジスターに設定する値。</td></tr> <tr> <td>uint32_t CHITVL_n</td><td>CHITVL_n レジスターに設定する値。</td></tr> <tr> <td>uint32_t CHEXT_n</td><td>CHEXT_n レジスターに設定する値。</td></tr> <tr> <td>uintptr_t next_descriptor_link_address</td><td>次のディスクリプターのアドレス</td></tr> </table>	bit_flags32_t header	ディスクリプターの各種属性。 dmac_lm_header_t 型の値の論理和	uintptr_t source_physical_address	N0SA_n レジスターに設定する値。 DMA の転送元アドレス	uintptr_t destination_physical_address	N0DA_n レジスターに設定する値。 DMA の転送先アドレス	uint32_t transfer_byte	N0TB_n レジスターに設定する値。 DMA の総転送バイト数	uint32_t CHCFG_n	CHCFG_n レジスターに設定する値。	uint32_t CHITVL_n	CHITVL_n レジスターに設定する値。	uint32_t CHEXT_n	CHEXT_n レジスターに設定する値。	uintptr_t next_descriptor_link_address	次のディスクリプターのアドレス
bit_flags32_t header	ディスクリプターの各種属性。 dmac_lm_header_t 型の値の論理和																
uintptr_t source_physical_address	N0SA_n レジスターに設定する値。 DMA の転送元アドレス																
uintptr_t destination_physical_address	N0DA_n レジスターに設定する値。 DMA の転送先アドレス																
uint32_t transfer_byte	N0TB_n レジスターに設定する値。 DMA の総転送バイト数																
uint32_t CHCFG_n	CHCFG_n レジスターに設定する値。																
uint32_t CHITVL_n	CHITVL_n レジスターに設定する値。																
uint32_t CHEXT_n	CHEXT_n レジスターに設定する値。																
uintptr_t next_descriptor_link_address	次のディスクリプターのアドレス																

6.6.5 dmac_lm_descriptor_config_t

概要	R_DMAMC_RM_StartAsLinkMode 関数のパラメーター		
ヘッダ	r_dmac_lm.h		
説明			
メンバー変数	<table border="1"> <tr> <td>bit_flags_fast32_t flags</td><td>フラグド構造体パラメーター。参考：(6.9.1) F_DMAMC_AXI_CACHE_ATTRIBUTE, F_DMAMC_AXI_PROTECTION</td></tr> </table>	bit_flags_fast32_t flags	フラグド構造体パラメーター。参考：(6.9.1) F_DMAMC_AXI_CACHE_ATTRIBUTE, F_DMAMC_AXI_PROTECTION
bit_flags_fast32_t flags	フラグド構造体パラメーター。参考：(6.9.1) F_DMAMC_AXI_CACHE_ATTRIBUTE, F_DMAMC_AXI_PROTECTION		

r_ospl_axi_cache_attribute_t AXI_cache_attribute	リンク・モードで実行するときのディスクリプターに DMAC がアクセスするときの AXI バスの L2 キャッシュ属性。
r_ospl_axi_protection_t AXI_protection	リンク・モードで実行するときのディスクリプターに DMAC がアクセスするときの AXI バスの保護属性。

6.7 変数一覧

表 6.5 に static 型変数を、表 6.6 に const 型変数を示します。

表 6.5 static 型変数

型	変数名	内容	使用関数
r_dmac_rm_channel_t	gs_dmac_rm_channel	DMAC	各種

表 6.6 const 型変数

型	変数名	内容	使用関数
なし			

6.8 関数

6.8.1 一覧

章	分類
(1)	DMAC を使用する関数
(2)	初期化前でも使える DMAC に関する関数
(3)	ドライバ移植層の関数

(1) DMAC を使用する関数の一覧

章	関数名	概要
6.8.2(1)	R_DMAMC_RM_Initialize	DMAC を初期化します
6.8.2(2)	R_DMAMC_RM_Finalize	DMAC の終了処理をします
6.8.2(3)	R_DMAMC_RM_InitializeEx	DMAC を初期化します。オプション付き
6.8.2(4)	R_DMAMC_RM_SetSharedConfiguration	各チャンネルで共通の設定をします
6.8.2(5)	R_DMAMC_RM_LockChannel	DMAC の 1 つのチャンネルをロックします
6.8.2(6)	R_DMAMC_RM_UnlockChannel	DMAC のチャンネルをロック解除します
6.8.2(7)	R_DMAMC_RM_TransferAsRegisterMode	レジスターモードで DMA 転送を実行します (同期実行)
6.8.2(8)	R_DMAMC_RM_StartAsRegisterMode	レジスターモードで DMA 転送を開始します (非同期実行)
6.8.2(9)	R_DMAMC_RM_StopAsRegisterMode	レジスターモードで始めた DMA 転送を中止します
6.8.2(10)	R_DMAMC_RM_TransferAsLinkMode	リンク・モードで DMA 転送を実行します (同期実行)
6.8.2(11)	R_DMAMC_RM_StartAsLinkMode	リンク・モードで DMA 転送を開始します (非同期実行)
6.8.2(12)	R_DMAMC_RM_StopAsLinkMode	リンク・モードで始めた DMA 転送を中止します
6.8.2(13)	R_DMAMC_RM_OnInterrupting	割り込みを受信します
6.8.2(14)	R_DMAMC_RM_OnInterrupted	割り込み応答処理を行います
6.8.2(15)	R_DMAMC_RM_GetAsyncStatus	割り込みや非同期処理の状況を示す構造体へのポインターを取得します

(2) 初期化前でも使える DMAC に関する関数の一覧

章	関数名	概要
6.8.3(1)	R_DMAMC_RM_STATIC_GetID_FromInterruptID	割り込み番号から、DMAC のチャンネル番号を取得します
6.8.3(2)	R_DMAMC_RM_STATIC_GetInterruptID	DMAC のチャンネル番号から、割り込み番号を取得します
6.8.3(3)	R_DMAMC_RM_STATIC_GetDMA_BitCountSymbol	DMA トランスファーサイズのシンボルを取得します
6.8.3(4)	R_DMAMC_LM_DESCRIPTOR_Initialize	リンク・モードのディスクリプターを作成します。

(3) ドライバ移植層の一覧

章	関数名	概要
6.8.4(1)	R_DMAMC_RM_SetDefaultAsync	r_ospl_async_t 型の構造体のデフォルト値を設定します
6.8.4(2)	R_DMAMC_RM_OnInitialize	ドライバ移植層の初期化をします
6.8.4(3)	R_DMAMC_RM_Onfinalize	ドライバ移植層の終了処理をします
6.8.4(4)	R_DMAMC_RM_SetInterruptCallbackCaller	込みコールバック関数を呼び出すオブジェクトを、ドライバの移植層に登録します
6.8.4(5)	R_DMAMC_RM_OnEnableInterrupt	割込みを許可します
6.8.4(6)	R_DMAMC_RM_OnDisableInterrupt	割込みを禁止します
6.8.4(7)	R_DMAMC_RM_OnInterruptDefault	デフォルトの割込みコールバック関数

6.8.2 DMAC を使用する関数

(1) R_DMAMC_RM_Initialize

概 要	DMAC を初期化します。	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_Initialize(int_fast32_t dmac_channel);	
説 明	内部変数を初期化します。 ステータスレジスタをクリアします。(SWRST ビット)	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号
リターン値	エラーコード。エラーなし=0	

(2) R_DMAMC_RM_Finalize

概 要	DMAC の終了処理をします	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_Finalize(int_fast32_t dmac_channel, errnum_t e);	
説 明	内部変数の終了処理をします。 DMA 転送が動いているときは、中止します。	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号
	errnum_t e	これまでに発生したエラーコード。 エラー無し=0
リターン値	エラーコードまたは e。 0=「成功かつ e=0」	

(3) R_DMAMC_RM_InitializeEx

概 要	DMAC を初期化します。オプション付き	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_InitializeEx(int_fast32_t dmac_channel, dmac_rm_config_t* in_out_Config);	
説 明	内部変数を初期化します。 ステータスレジスタをクリアします。(SWRST ビット) dmac_rm_config_t 型のオプションを指定できます。	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号。 R_OSPL_UNLOCKED_CHANNEL を指定したときは、in_out_Config->channel_num に、未使用だったチャンネル番号が格納されます。
	dmac_rm_config_t* in_out_Config	NULL、または dmac_rm_config_t (6.6.3)。
リターン値	エラーコード。エラーなし=0	

(4) R_DMAMC_RM_SetSharedConfiguration

概 要	各チャンネルで共通の設定をします。	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_SetSharedConfiguration(int_fast32_t dmac_channel, dmac_shared_config_t* config);	
説 明		
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号。 0～7 を指定したときは、チャンネル 0～7 で共通の設定をします。 8～15 を指定したときは、チャンネル 8～15 で共通の設定をします。
	dmac_shared_config_t* config	設定。参照(6.6.2)
リターン値	エラーコード。エラーなし=0	

(5) R_DMAMC_RM_LockChannel

概 要	DMAC の 1 つのチャンネルを初期化しないで、ロックします。	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_LockChannel(int_fast32_t const dmac_channel, int_fast32_t* out_ChannelNum);	
説 明	初期化をしないでロックをすると、空いているチャンネルのリストから除外され、別のドライバを競合せずに使えるようになります。	

引 数	int_fast32_t const dmac_channel	ロックしようとする DMAC のチャンネル番号、または、R_OSPL_UNLOCKED_CHANNEL。
	int_fast32_t* out_ChannelNum	ロックされた DMAC のチャンネル番号。
リターン値	エラーコード。エラーなし=0	

(6) R_DMAMC_RM_UnlockChannel

概 要	DMAC のチャンネルの終了処理をしないで、ロック解除します	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_UnlockChannel(int_fast32_t const dmac_channel, errnum_t e);	
説 明		

引 数	int_fast32_t const dmac_channel	ロック解除しようとする DMAC のチャンネル番号
	errnum_t e	これまでに発生したエラーコード。エラー無し=0
リターン値	エラーコードまたは e。 0=「成功かつ e=0」	

(7) R_DMAMC_RM_TransferAsRegisterMode

概 要	レジスターモードで DMA 転送を実行します。（同期実行）	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_TransferAsRegisterMode(int_fast32_t dmac_channel, dmac_register_mode_config_t* config);	
説 明	転送が完了するまでリターンしない同期関数です。config->request_resource_MID が省略されたときは、オートリクエストモードであると想定して、CHCTRL_n レジスターの STG ビットに 1 をライトします。	

引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号
	dmac_register_mode_config_t* config	設定。参考：(6.6.1)
リターン値	エラーコード。エラーなし=0	

(8) R_DMAMC_RM_StartAsRegisterMode

概 要	レジスターモードで DMA 転送を開始します。（非同期実行）	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_StartAsRegisterMode(int_fast32_t dmac_channel, dmac_register_mode_config_t* config, r_ospl_async_t* Async);	
説 明	転送が完了する前にすぐにリターンする非同期関数です。それ以外は、R_DMAMC_RM_TransferAsRegisterMode 関数と同じです。 Async 引数の詳細は、RZ/A1H グループ OS 移植層 OSPL（R01AN1887JJ）にある R_DRIVER_TransferAsync 関数の説明を参照	

引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号
	dmac_register_mode_config_t* config	設定。参考：(6.6.1)
	r_ospl_async_t* Async	通知設定
リターン値	エラーコード。エラーなし=0	

(9) R_DMAMC_RM_StopAsRegisterMode

概 要	レジスタモードで始めた DMA 転送を中止します。	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_StopAsRegisterMode(int_fast32_t dmac_channel);	
説 明	中止したら、R_DMAMC_RM_StartAsRegisterMode 関数の Async 引数に指定したイベントは発生しなくなります。	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号
リターン値	エラーコード。エラーなし=0	

(10) R_DMAMC_RM_TransferAsLinkMode

概 要	リンク・モードで DMA 転送を実行します。（同期実行）	
ヘッダ	r_dmac_lm.h	
宣 言	errnum_t R_DMAMC_RM_TransferAsLinkMode(int_fast32_t dmac_channel, dmac_register_mode_config_t* a_config, uintptr_t first_descriptor_physical_address, dmac_lm_descriptor_config_t* descriptor_config);	
説 明	リスト構造でつながったディスクリプターに記述された DMA 転送を連続して実行します（リンク・モード）。 転送が完了するまでリターンしない同期関数です。 a_config->request_resource_MID が省略されたときは、オートリクエストモードであると想定して、CHCTRL_n レジスタの STG ビットに 1 をライトします。このライトは、2 つ目以降のディスクリプターを実行するときは必要ないためしていません。	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号
	dmac_register_mode_config_t* a_config	DMA 転送を開始する割り込みを発生する周辺機能に関する設定。以下のメンバー変数以外は無視します。 ● flags ● request_resource_MID ● request_resource_RID この引数に、参照するディスクリプターの 1 つを指定することもできます。
	uintptr_t first_descriptor_physical_address	最初に参照するディスクリプターの物理アドレス。 dmac_lm_descriptor_t* 型相当。
	dmac_lm_descriptor_config_t* descriptor_config	ディスクリプターに関する設定。参照 (6.6.4)
リターン値	エラーコード。エラーなし=0	

(11) R_DMAMC_RM_StartAsLinkMode

概 要	リンク・モードで DMA 転送を開始します。（非同期実行）	
ヘッダ	r_dmac_lm.h	
宣 言	errnum_t R_DMAMC_RM_StartAsLinkMode(int_fast32_t dmac_channel, dmac_register_mode_config_t* a_config, uintptr_t first_descriptor_physical_address, dmac_lm_descriptor_config_t* descriptor_config, r_ospl_async_t* Async);	
説 明	転送が完了する前にすぐにリターンする非同期関数です。それ以外は、R_DMAMC_RM_TransferAsLinkMode 関数と同じです。 Async 引数の詳細は、RZ/A1H グループ OS 移植層 OSPL（R01AN1887JJ）にある R_DRIVER_TransferAsync 関数の説明を参照	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号
	dmac_register_mode_config_t* a_config	DMA 転送を開始する割り込みを発生するデバイスに関する設定。以下のメンバー変数以外は無視します。

リターン値		<ul style="list-style-type: none"> ● flags ● request_resource_MID ● request_resource_RID 実行するディスクリプターの1つを作成したときに使用した <code>dmac_register_mode_config_t</code> 型構造体を指定することもできます。
	<code>uintptr_t</code> <code>first_descriptor_physical_address</code>	最初に実行するディスクリプターの物理アドレス。 <code>dmac_lm_descriptor_t*</code> 型相当。
	<code>dmac_lm_descriptor_config_t* descriptor_config</code>	ディスクリプターに関する設定。参照 (6.6.5)
	<code>r_ospl_async_t* Async</code>	通知設定
	エラーコード。エラーなし=0	

(12) R_DMAC_RM_StopAsLinkMode

概 要	リンク・モードで始めた DMA 転送を中止します。	
ヘッダ	<code>r_dmac_lm.h</code>	
宣 言	<code>errnum_t R_DMAC_RM_StopAsLinkMode(int_fast32_t dmac_channel);</code>	
説 明	中止したら、 <code>R_DMAC_RM_StartAsLinkMode</code> 関数の <code>Async</code> 引数に指定したイベントは発生しなくなります。	
引 数	<code>int_fast32_t dmac_channel</code>	DMAC のチャンネル番号
リターン値	エラーコード。エラーなし=0	

(13) R_DMAC_RM_OnInterrupting

概 要	割り込みを受信します。	
ヘッダ	<code>r_dmac_rm.h</code>	
宣 言	<code>errnum_t R_DMAC_RM_OnInterrupting(const r_ospl_interrupt_t* InterruptSource);</code>	
説 明	本関数は、割り込みステータスレジスタから、 <code>dmac_rm_async_status_t::InterruptFlags</code> 変数に割り込み通知を伝達し、割り込みをクリアします。 詳細は、RZ/A1H グループ OS 移植層 OSPL (R01AN1887JJ) にある <code>R_DRIVER_OnInterrupting</code> 関数の説明を参照	
引 数	<code>r_ospl_interrupt_t* InterruptSource</code>	割り込み発信元
リターン値	エラーコード。エラーなし=0	

(14) R_DMAC_RM_OnInterrupted

概 要	割り込み応答処理を行います。	
ヘッダ	<code>r_dmac_rm.h</code>	
宣 言	<code>errnum_t R_DMAC_RM_OnInterrupted(int_fast32_t const dmac_channel);</code>	
説 明	<code>R_DMAC_RM_OnInterrupting</code> 関数によって 1 に設定された <code>dmac_rm_async_status_t::InterruptFlags</code> 変数のビットを 0 にクリアして、割り込み 応答処理を行います。 詳細は、RZ/A1H グループ OS 移植層 OSPL (R01AN1887JJ) にある <code>R_DRIVER_OnInterrupted</code> 関数の説明を参照。	
引 数	<code>int_fast32_t dmac_channel</code>	DMAC のチャンネル番号。
リターン値	エラーコード。エラーなし=0	

(15) R_DMAC_RM_GetAsyncStatus

概 要	割り込みや非同期処理の状況を示す構造体へのポインターを取得します。	
ヘッダ	<code>r_dmac_rm.h</code>	

宣 言	errnum_t R_DMAMC_RM_GetAsyncStatus(int_fast32_t dmac_channel, const dmac_rm_async_status_t** out_Status);	
説 明	out_Status 引数に指定するポインター変数には、const 修飾子が必要です。	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号
	dmac_rm_async_status_t** out_Status	(出力) 割り込みや非同期処理の状況を示す構造体へのポインター
リターン値	エラーコード。エラーなし=0	

6.8.3 初期化前でも使える DMAC に関する関数

(1) R_DMAMC_RM_STATIC_GetID_FromInterruptID

概 要	割り込み番号から、DMAC のチャンネル番号を取得します。	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_STATIC_GetID_FromInterruptID(bsp_int_src_t interrupt_id, int_fast32_t* out_dmac_channel);	
説 明		
引 数	bsp_int_src_t interrupt_id	割り込み番号
	int_fast32_t* out_dmac_channel	(出力) DMAC のチャンネル番号
リターン値	エラーコード。エラーなし=0	

(2) R_DMAMC_RM_STATIC_GetInterruptID

概 要	DMAC のチャンネル番号から、割り込み番号を取得します。	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_STATIC_GetInterruptID(int_fast32_t dmac_channel, bsp_int_src_t* out_interrupt_id);	
説 明	DMAC の割り込みの状態は、本関数では取得できません。	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号。
	bsp_int_src_t* out_interrupt_id	(出力) 割り込み番号
リターン値	エラーコード。エラーなし=0	

(3) R_DMAMC_RM_STATIC_GetDMA_BitCountSymbol

概 要	DMA トランスファーサイズのシンボルを取得します。	
ヘッダ	r_dmac_rm.h	
宣 言	errnum_t R_DMAMC_RM_STATIC_GetDMA_BitCountSymbol(int_t bit_count, dmac_bit_count_t* out_symbol);	
説 明		
引 数	int_t bit_count	DMA トランスファーサイズ (バイト)
	dmac_bit_count_t* out_symbol	(出力) DMA トランスファーサイズのシンボル
リターン値	エラーコード。エラーなし=0	

(4) R_DMAMC_RM_DESCRIPTOR_Initialize

概 要	リンク・モードで使用するディスクリプターの 1 つを作成します。	
ヘッダ	r_dmac_lm.h	
宣 言	errnum_t R_DMAMC_RM_DESCRIPTOR_Initialize(dmac_lm_descriptor_t* self, int_fast32_t const dmac_channel, bit_flags32_t header, dmac_register_mode_config_t* config, uintptr_t next_descriptor_physical_address);	
説 明	本関数を呼び出した後、R_DMAMC_RM_TransferAsLinkMode 関数などでリンク・モードの DMA 転送を開始する前に、self 引数が指すメモリに対し、必要に応じてキャッシュのフラッシュをしてください。	

引 数	dmac_lm_descriptor_t* self	dmac_lm_descriptor_t 型構造体変数の仮想アドレス。 (出力) リンク・モードで使用するディスクリプター。
	int_fast32_t const dmac_channel	config 引数に指定した DMA 転送を実行する DMAC のチャンネル番号。
	bit_flags32_t header	dmac_lm_header_t 型の値の論理和。
	dmac_register_mode_config_t* config	DMA 転送に関する設定。参考：(6.6.1) ただし、F_DMAMC_NEXT1_* で始まるメンバー変数は、指定しないでください。
	uintptr_t next_descriptor_physical_address	self に指定したディスクリプターの DMA 転送を実行した後 に実行するディスクリプターの物理アドレス。
リターン値	エラーコード。エラーなし=0	

6.8.4 ドライバ移植層の関数

(1) R_DMAMC_RM_SetDefaultAsync

概 要	r_ospl_async_t 型の構造体のデフォルト値を設定します。	
ヘッダ	r_dmac_rm_pl.h	
宣 言	void R_DMAMC_RM_SetDefaultAsync(r_ospl_async_t* Async);	
説 明	r_ospl_async_t 型の構造体の Flags メンバー変数の値のうち、0 になっているビットに対応する、それぞれのメンバー変数をデフォルト値に設定します。 ReturnValue メンバー変数は、コールバック元のドライバで初期化します。	
引 数	r_ospl_async_t* Async	通知設定、NULL 不可
リターン値	なし	

(2) R_DMAMC_RM_OnInitialize

概 要	ドライバ移植層の初期化をします。	
ヘッダ	r_dmac_rm_pl.h	
宣 言	errnum_t R_DMAMC_RM_OnInitialize(int_fast32_t dmac_channel);	
説 明	必要ならクロックを供給する設定をしてください。 R_DMAMC_RM_Initialize 関数から呼ばれます。	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号。
リターン値	エラーコード。エラーなし=0	

(3) R_DMAMC_RM_OnFinalize

概 要	ドライバ移植層の終了処理をします。	
ヘッダ	r_dmac_rm_pl.h	
宣 言	errnum_t R_DMAMC_RM_OnFinalize(int_fast32_t dmac_channel, errnum_t e);	
説 明	必要ならクロックを停止する設定をしてください。 R_DMAMC_RM_Finalize 関数から呼ばれます。	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号。
	errnum_t e	これまでに発生したエラーコード。 エラー無し=0
リターン値	エラーコードまたは e。 0=「成功かつ e=0」	

(4) R_DMAMC_RM_SetInterruptCallbackCaller

概 要	割り込みコールバック関数を呼び出すオブジェクトを、ドライバの移植層に登録します。
ヘッダ	r_dmac_rm_pl.h

宣 言	errnum_t R_DMAM_RM_SetInterruptCallbackCaller(int_fast32_t dmac_channel, const r_ospl_caller_t* caller);	
説 明	<p>割り込みがある非同期処理を開始するたびに呼ばれます。</p> <p>割り込みハンドラーから、caller 引数の値を指定した R_OSPL_CallInterruptCallback 関数を呼び出すようにしてください。</p> <p>ドライバが割り込み禁止にしているときは、caller 引数が指す構造体の本体が存在しない可能性があります、その管理はドライバの責任です。</p> <p>本関数の内部で、チャンネル番号はチェックしません。</p>	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号。
	r_ospl_caller_t* caller	R_OSPL_CallInterruptCallback 関数に渡す値
リターン値	エラーコード。エラーなし=0	

(5) R_DMAM_RM_OnEnableInterrupt

概 要	割り込みを許可します。	
ヘッダ	r_dmac_rm_pl.h	
宣 言	void R_DMAM_RM_OnEnableInterrupt(int_fast32_t dmac_channel, dmac_interrupt_lines_t enables);	
説 明	本関数の内部で、チャンネル番号をチェックしません。	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号。
	dmac_interrupt_lines_t enables	許可する割り込み線の種類を 1 にしたビットフラグ
リターン値	なし	

(6) R_DMAM_RM_OnDisableInterrupt

概 要	割り込みを禁止します。	
ヘッダ	r_dmac_rm_pl.h	
宣 言	void R_DMAM_RM_OnDisableInterrupt(int_fast32_t dmac_channel, dmac_interrupt_lines_t disables1);	
説 明	<p>本関数の内部で、チャンネル番号をチェックしません。</p> <p>R_DMAM_RM_OnInitialize 関数が呼ばれる前に、本関数が呼ばれることがあります。</p>	
引 数	int_fast32_t dmac_channel	DMAC のチャンネル番号。
	dmac_interrupt_lines_t disables1	禁止する割り込み線の種類を 1 にしたビットフラグ
リターン値	なし	

(7) R_DMAM_RM_OnInterruptDefault

概 要	デフォルトの割り込みコールバック関数。	
ヘッダ	r_dmac_rm_pl.h	
宣 言	errnum_t R_DMAM_RM_OnInterruptDefault(const r_ospl_interrupt_t* interrupt_source, const r_ospl_caller_t* caller);	
説 明	<p>R_DMAM_RM_StartAsRegisterMode 関数の async 引数の .InterruptCallback メンバー変数が NULL のときに本関数がコールバックされます。</p> <p>本関数は、r_ospl_callback_t 型です。</p>	
引 数	r_ospl_interrupt_t* interrupt_source	割り込み発信元
	r_ospl_caller_t* caller	R_OSPL_CallInterruptCallback 関数に渡された値
リターン値	エラーコード。エラーなし=0	

6.9 補足

6.9.1 RZ/A1H RTX BSP との連携

本ドライバと、RZ/A1H RTX BSP の DMAC ドライバと同時に使用するときは、チャンネルのロックの一元化して管理する必要があります。IS_RZ_A1_BSP_USED を 1 に設定すれば、本ドライバの内部から、OSPL を経由して、BSP の DMAC ドライバのチャンネルのロック (R_DMA_Alloc 関数呼び出し) を行い、BSP で管理するようになります。

6.9.2 フラグド構造体パラメーター

構造体の中の Flags メンバー変数をビットフラグとして使い、ビットが 1 であれば、対応するメンバー変数を有効にするというコーディングパターンです。ビットが 0 ならば、メンバー変数の値は、デフォルトの値が設定されたものと同じとします。バージョンアップしたら構造体のメンバーが増えた場合でも、バイナリ互換にできます。

```
FuncA_ConfigClass config;  
  
config.Flags = F_FuncA_Param1 | F_FuncA_Param2;  
config.Param1 = 10;  
config.Param2 = 2;  
FuncA( &config );
```

Flags |= F_FuncA_Param3 が無いため、config.Param3 はデフォルト値。

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 (GENMAI) 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。)

R7S72100 CPU ボード用オプションボード RTK7721000B00000BR (GENMAI) 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

Rev.	発行日	改訂内容
1.03	2017.08.24	OSPL のバージョンを 1.60 に更新
1.01	2016.02.29	追加 : <ul style="list-style-type: none"> ● リンク・モード ● チャンネルのロック ● 未使用チャンネルの検索 (R_OSPL_UNLOCKED_CHANNEL) ● L2 キャッシュ対応 (F_DMAC_DESTINATION_AXI_CACHE_ATTRIBUTE など) 初期設定例 1.01 に更新。 内部で使用する OSPL をバージョン 0.96 に更新。 R_DMAC_RM_StopAsRegisterMode の状態遷移の不具合を対処。
1.00	2014.06.20	初版

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、レジスター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、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、電気的特性の範囲で、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生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品、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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 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、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。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）、交通制御（信号）、大規模通信機器、
金融端末基幹システム、各種安全制御装置等
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（宇宙、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に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損害が生じて、当社は一切その責任を負いません。
 6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
 9. 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、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 10. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）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、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
 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。
 12. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.3.0-1 2016.11)



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 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■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： <https://www.renesas.com/contact/>