

RH850/U2B Group

R01AN7038JJ0100

Rev.1.00

データCRC機能K（KCRC）

要旨

本アプリケーションノートは、RH850/U2Bx に搭載されているデータ CRC 機能 K（KCRC）の使用方法について説明します。

なお、本アプリケーションノートに掲載されているタスク例およびアプリケーション例は動作確認しておりますが、実際にご使用になる場合には、必ず動作確認の上ご使用くださいますようお願いいたします。

動作確認デバイス

RH850/U2Bx

目次

1.	はじめに	3
1.1	使用機能	3
2.	データCRC機能K (KCRC) の概要	4
2.1	CRC生成手順	5
3.	動作概要	6
3.1	動作例1 SPIフレームに格納したCRCによる送信データの検証	6
3.1.1	概要	6
3.1.2	使用機能の動作条件	8
3.1.3	ソフトウェア説明	10
3.1.4	動作シーケンス	20
3.2	動作例2 CAN FDフレームに格納したCRCによる送信データの検証	22
3.2.1	概要	22
3.2.2	使用機能の動作条件	23
3.2.3	ソフトウェア説明	25
3.2.4	動作シーケンス	35

1. はじめに

本アプリケーションノートでは、RH850/U2Bx のデータ CRC 機能 K (KCRC) の使用方法およびソフトウェアの作成例を掲載しています。

1.1 使用機能

本アプリケーションノートで使用する RH850/U2Bx のハードウェア機能を以下に示します。

- データ CRC 機能 K (KCRC)
- DMA (sDMAC)
- マルチチャネルシリアルペリフェラルインターフェース (MSPI)
- CANFD インタフェース (RSCANFD)

2. データ CRC 機能 K (KCRC) の概要

データ CRC 機能 K を利用して、CRC (巡回冗長検査) で保護された任意の長さとはさまざまなビット幅を持つデータストリームを検証または生成することができます。

- サポートする CRC 多項式

- 64 ビット CRC64ECMA

42F0E1EB A9EA3693H: $X^{64} + X^{62} + X^{57} + X^{55} + X^{54} + X^{53} + X^{52} + X^{47} + X^{46} + X^{45} + X^{40} + X^{39} + X^{38} + X^{37} + X^{35} + X^{33} + X^{32} + X^{31} + X^{29} + X^{27} + X^{24} + X^{23} + X^{22} + X^{21} + X^{19} + X^{17} + X^{13} + X^{12} + X^{10} + X^9 + X^7 + X^4 + X + 1$

- 32 ビット CRC32P4

F4ACFB13H: $X^{32} + X^{31} + X^{30} + X^{29} + X^{28} + X^{26} + X^{23} + X^{21} + X^{19} + X^{18} + X^{15} + X^{14} + X^{13} + X^{12} + X^{11} + X^9 + X^8 + X^4 + X + 1$

- 32 ビットイーサネット CRC

04C1 1DB7H: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X^1 + 1$

- 32-bit CRC32 Reversed Polynomial

EDB88320H: $X^{32} + X^{31} + X^{30} + X^{29} + X^{27} + X^{26} + X^{24} + X^{23} + X^{21} + X^{20} + X^{19} + X^{15} + X^9 + X^8 + X^5 + 1$

- 32 ビット CRC32C (Castagnoli)

1EDC 6F41H: $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$

- 16 ビット CCITT CRC

1021H: $X^{16} + X^{12} + X^5 + 1$

- 16 ビット Baicheva00:

90D9H: $X^{16} + X^{15} + X^{12} + X^7 + X^6 + X^4 + X^3 + 1$

- 15 ビット CRC15CAN

4599H: $X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$

- 16 ビット ARC

8005H: $X^{16} + X^{15} + X^2 + 1$

- 8 ビット SAE J1850 CRC

1DH: $X^8 + X^4 + X^3 + X^2 + 1$

- 8 ビット 0x2F CRC

2FH: $X^8 + X^5 + X^3 + X^2 + X + 1$

- 任意のデータ長ブロックに対する CRC 生成

- KCRC データレジスタの初期化後、KCRC 入力レジスタ (KCRCnDIN) へのすべての書き込みアクセスは、選択した多項式による新しい CRC を生成し、その結果を KCRC データレジスタ (KCRCnDOUT) に格納します。

2.1 CRC 生成手順

データ CRC 機能 K は任意のデータブロック長の CRC を生成します。データは、8 ビット単位、16 ビット単位または 32 ビット単位でデータ CRC 機能に転送されます。KCRCnDIN レジスタへの最初の書き込みアクセスを行う前に、KCRCnDOUT レジスタに初期開始値を設定する必要があります。

KCRCnDIN レジスタへの最後の書き込みアクセスを行った後、1 クロック後に KCRCnDOUT レジスタから結果を読み出せます。

図 2-1に、データCRC機能KにおけるCRC生成のフローチャートを示します。

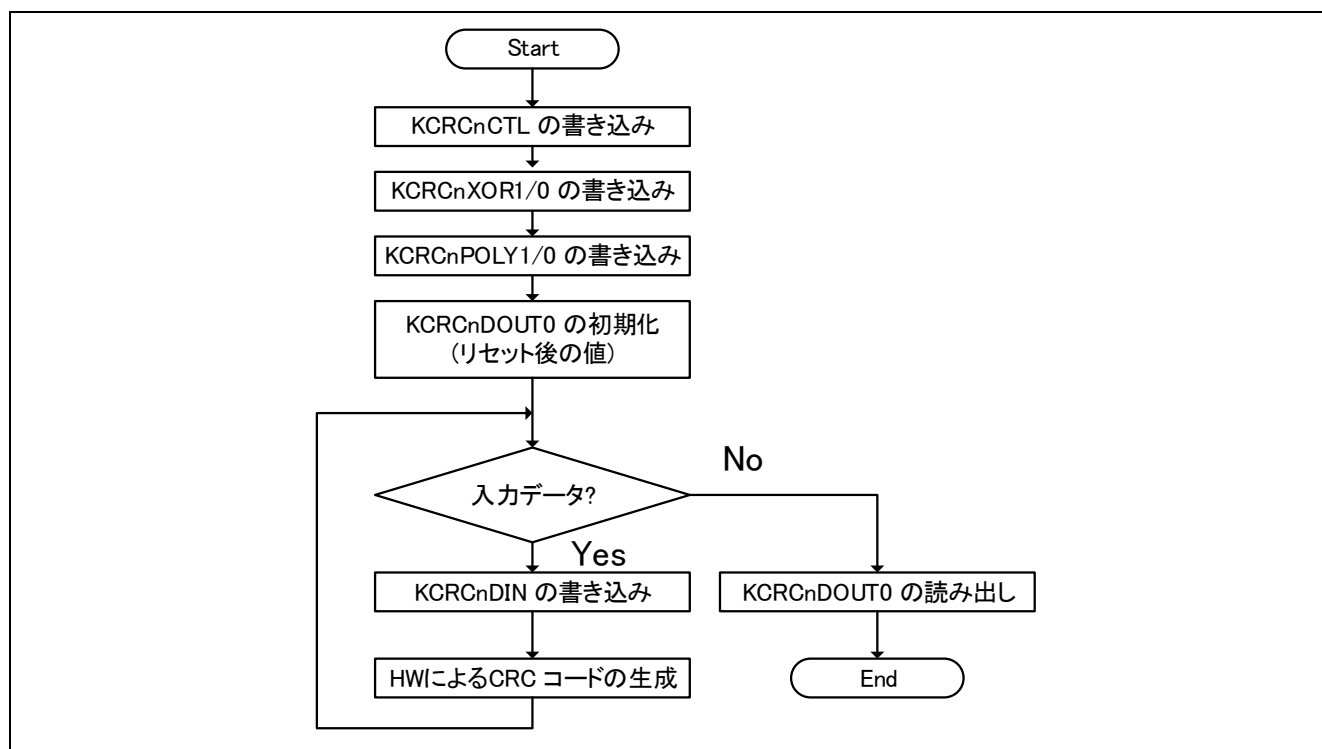


図 2-1 データ CRC 機能 K における CRC 生成のフローチャート

KCRC のすべてのレジスタは CPU から独立した DMA 経由のリード／ライトアクセスが可能です。本アプリケーションノートでは、sDMAC を併用し、CPU を介さない CRC 生成の方法を紹介しています。

3. 動作概要

3.1 動作例 1 SPI フレームに格納した CRC による送信データの検証

3.1.1 概要

SPI フレームに CRC を格納して送信することで、送信データの妥当性を検証することができます。本動作例における、初期設定以外の RAM 及びレジスタへのデータの読み書きは、sDMAC を使用して行い、CPU を介在させないようにしています。

CRC の生成多項式は、16 ビット CCITT CRC を使用しています。SPI 通信は、MSPI を使用して行い、16bit のデータを 8 回送受信し、8 個目のデータとして CRC を送信します。

図 3-1に本動作例の概略図を示します。図中の①～⑥までの処理は sDMAC によって実行され、⑦の処理は、CPU によって実行されます。

送信側

- ① RAM に格納した送信データを KCRC 入力レジスタに書き込み、CRC を生成します。
- ② 生成した CRC を、RAM に格納した送信データの末尾に格納します。
- ③ MSPI の送信データレジスタに送信データを書き込み、送信を開始します。

受信側

- ④ MSPI の受信データレジスタから受信データを読み出し、RAM に格納します
- ⑤ RAM に格納した受信データを KCRC 入力レジスタに書き込み、CRC' を生成します。
- ⑥ 生成した CRC' を、RAM に配置します。
- ⑦ 生成した CRC' と受信した CRC を CPU で比較して、送信データの妥当性を検証します。

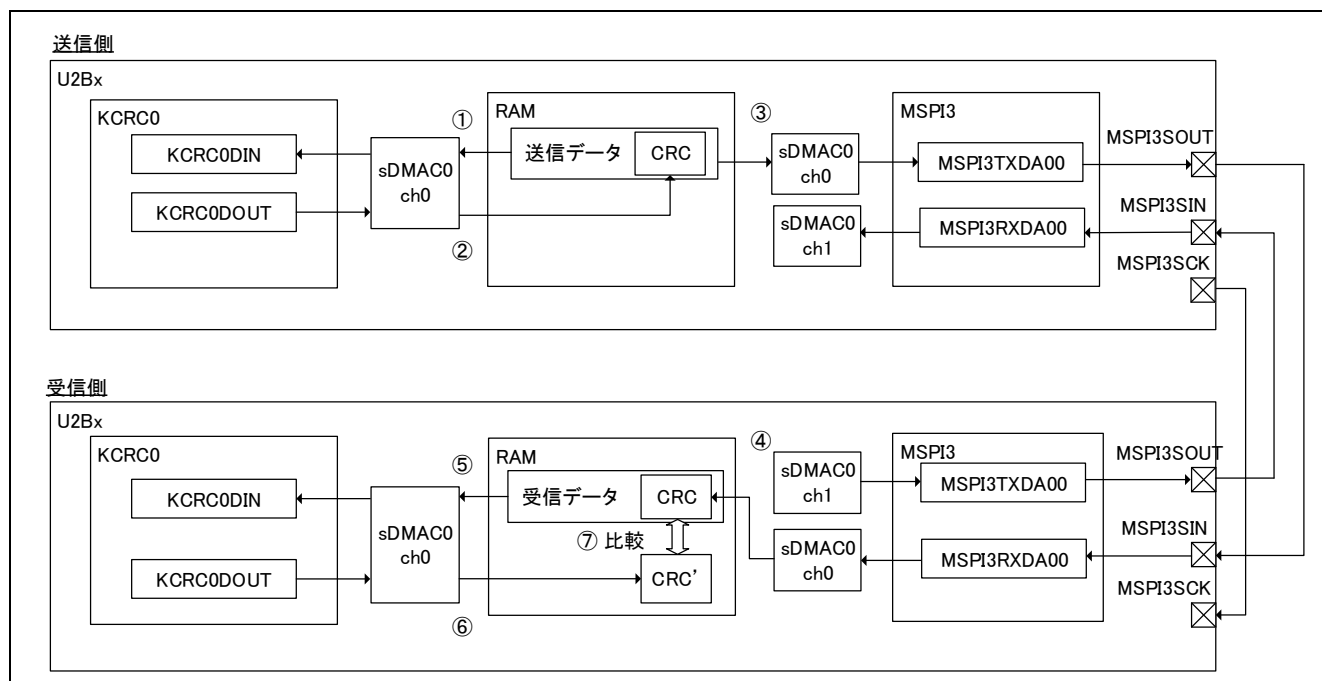


図 3-1 概略図

また、本動作例の sDMAC の動作は、ディスクリプタメモリに定義しており、順次実行されます。

ディスクリプタは、チャンネル上の 1 つの転送タスクを記述するのに必要なすべてのレジスタビットを含んでいます。チャンネルはディスクリプタメモリからレジスタ設定を再ロードすることによって複数のディスクリプタを連鎖で処理できます。現在のレジスタ設定で定義される DMA 転送が完了するとき、次のディスクリプタがディスクリプタメモリからロードされます。ロードするディスクリプタのアドレスは DMAjDPPTR_n.PTR ビットで定義され、ディスクリプタ動作の継続は、DMAjDPPTR_n.CF ビットで定義されます。

図 3-2に本動作例における、ディスクリプタメモリ設定例を示します。

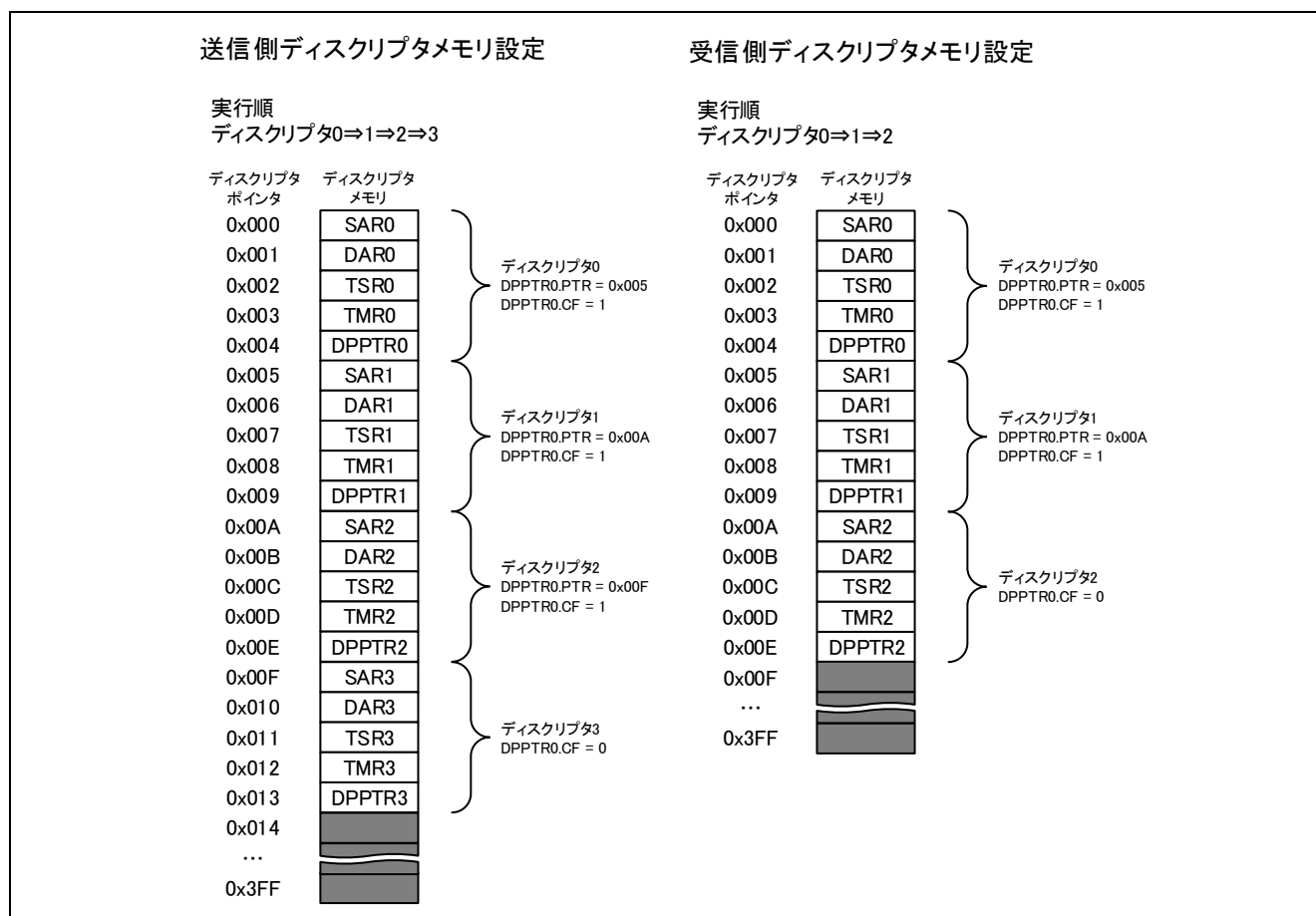


図 3-2 ディスクリプタメモリ設定例

3.1.2 使用機能の動作条件

本動作例で使用する機能の動作条件を以下に示します。

3.1.2.1 送信側プログラムの動作条件

送信側プログラムの動作条件を以下に示します。

表 3-1 KCRC0 の設定

項目	内容
入力ビット幅	16 ビット
CRC 生成方式	16 ビット CCITT CRC 多項式生成

表 3-2 MSPI3 の設定

項目	内容
MSPI3 動作クロック	80MHz (CLK_HSB_MSPI)
使用チャネル	ch3
動作モード	マスタモード
転送クロック周波数	10MHz
送信データ長	16 ビット
送信データ	0x1111, 0x2222, 0x3333, 0x4444, 0x5555, 0x6666, 0x7777, CRC(16bit)
割込み (DMA トリガ)	通信ステータス割込み

表 3-3 sDMAC0 の設定

項目	内容
sDMAC0 動作クロック	100MHz
使用チャネル	ch0,ch1
割込み	無し

表 3-4 ポート設定

項目	内容
使用端子	P01_5 MSPI3SI
	P01_6 MSPI3SO
	P01_7 MSPI3SC

3.1.2.2 受信側プログラムの動作条件

受信側プログラムの動作条件を以下に示します。

表 3-5 KCRC0 の設定

項目	内容
入力ビット幅	16 ビット
CRC 生成方式	16 ビット CCITT CRC 多項式生成

表 3-6 MSPI3 の設定

項目	内容
MSPI3 動作クロック	80MHz (CLK_HSB_MSPI)
使用チャネル	ch3
動作モード	スレーブモード
転送クロック周波数	10MHz
割込み (DMA トリガ)	受信ステータス割込み

表 3-7 sDMAC0 の設定

項目	内容
sDMAC0 動作クロック	100MHz
使用チャネル	ch0,ch1
割込み	転送完了割り込み

表 3-8 ポート設定

項目	内容
使用端子	P01_5 MSPI3SI
	P01_6 MSPI3SO
	P01_7 MSPI3SC

3.1.3 ソフトウェア説明

本動作例で使用する各レジスタの設定例を以下に示します。

3.1.3.1 送信側設定

送信側プログラムで使用する各レジスタの設定例を以下に示します。

表 3-9 KCRC0 レジスタ設定

レジスタ名	設定値	機能
KCRC0CTL	0x000F0001	CRC 入力ビット幅 : 16 ビット
		多項式サイズ : 16 ビット
KCRC0POLY1	0x00000000	CRC 生成方式 : CCITT CRC 多項式生成
KCRC0POLY0	0x00001021	
KCRC0DOUT0	0x0000FFFF	CRC コード生成の開始値

表 3-10 MSPI レジスタ設定

レジスタ名	設定値	機能
MSPI3CTL0	0x01	MSPI 機能許可
MSPI3CTL1	0x18000000	マスタモード
		次回エッジサンプリング
		MSPITCK のデフォルトレベルはハイレベル
		SOUT をローレベル
		CS アクティブロー
MSPI3CFG00	0x30700003	送信許可
		受信許可
		ダイレクトメモリモード
		チャンネル優先レベル : 7
		チャンネルロック操作禁止
		最終フレームで CHEN クリア & チャンネルオペレーション終了
		送信割り込み許可
		受信割り込み許可
MSPI3CFG01	0x03010000	クロック位相 : アイドル時は High
		データ位相 : 奇数エッジで変化
		MSB ファースト
		転送間のアイドルなし
		CS アサート : 非アクティブ
		パリティチェック無し
MSPI3CFG02	0x10	フレーム長 : 16 ビット
MSPI3CFG03	0x0004	ポーレート : MSPI3CLK/8(10MHz)
MSPI3SEUP0	0x0002	SCK ディレイタイム : 2
MSPI3HOLD0	0x0001	CS ネグートディレイタイム : 1
MSPI3IDLE0	0x0001	次フレームアイドル時間 : 1
MSPI3INDA0	0x0008	次フレーム入力データ時間 : 8
MSPI3CFSET0	0x0008	フレーム数 : 8
MSPI3SSEL0	0x0001	CS0 アクティブ
MSPI3CSTS0	0x0003	チャンネルアクティブ、チャンネル許可

表 3-11 PBG レジスタ設定

レジスタ名	設定値	機能
PBGERRSLV30 BGKCPROT	0xA5A5A501	PBG30 のレジスタアクセスを許可
PBG30 PBGPROT1_6	0x10000001	KCRC0 用 SPID : sDMAC、CPU0 を 1 に設定
PBG30 PBGPROT1_11	0x10000001	MSPI3 用 SPID : sDMAC、CPU0 を 1 に設定

表 3-12 DMA/DTS トリガレジスタ設定

レジスタ名	設定値	機能
MSPITGTGCTL3	0x0000000E	Trigger 1=INTMSPIInTX0、Trigger 2=INTMSPIInRX0

表 3-13 sDMAC ch0(送信用)レジスタ設定

レジスタ名	設定値	機能
DMA0CM_0	0x00001C00	チャンネルマスタ SPID : 0x1C スーパバイザモード
DMA0CHCR_0	0x0302	ディスクリプタ動作有効 ディスクリプタメモリの設定から DMA 転送開始 チャンネルアドレスエラー通知無効 チャンネルアドレスエラー割込み無効 ディスクリプタステップ終了割込み無効 転送完了割込み有効 DMA 転送無効
DMA0DPCR_0	0x0000000F	ディスクリプタのアップデートフラグ DMAjSAR_n レジスタアップデート有効 DMAjDAR_n レジスタアップデート有効 DMAjTSR_n レジスタアップデート有効 DMAjTMR_n レジスタアップデート有効 DMAjGIAI_n レジスタアップデート無効 DMAjGOAI_n レジスタアップデート無効 DMAjSIAI_n レジスタアップデート無効 DMAjSOAI_n レジスタアップデート無効 DMAjSGCR_n レジスタアップデート無効 DMAjRS_n レジスタアップデート無効 DMAjBUFCR_n レジスタアップデート無効
DMA0RS_0	0x00010084	ハードウェア要求あたりの転送回数 : 1 回 ハードウェア要求あたりの転送リミット : DMAjTMR_0.STS×DMAjRS_0.TC により指定された転送サイズ ブロード無効 DRQ 初期化無効 DMA 要求ソース : DMAMSPI24(group0-132)
DMATRGSSELDMACSEL0_8	0xFFFFFCFF	MSPI3(group0-132)
DMA0DPPTR_0	0x00000001	ディスクリプタのアドレス : 0x000 ディスクリプタ割り込み無効 ディスクリプタの継続有効
DMA0OR	0x0001	チャンネル間優先度 : CH0>CH1>...>CH14>CH15 すべてのチャンネルの DMA 転送有効

表 3-14 sDMAC ch1(ダミーデータ受信用)レジスタ設定

レジスタ名	設定値	機能
DMA0CM_1	0x00001C00	チャンネルマスタ SPID : 0x1C スーパバイザモード
DMA0CHCR_1	0x0302	ディスクリプタ動作有効 ディスクリプタメモリの設定から DMA 転送開始 チャンネルアドレスエラー通知無効 チャンネルアドレスエラー割込み無効 ディスクリプタステップ終了割込み無効 転送完了割込み有効 DMA 転送無効
DMA0SAR_1	-	ソースアドレス : (unsigned long) &MSPI3.RXDA00.UINT32 (MSPI3 受信データレジスタ)
DMA0DAR_1	-	ディスティネーションアドレス : (unsigned long) &r_dat
DMA0TSR_1	0x00000020	転送サイズ : 32 バイト
DMA0TMR_1	0x00001022	転送要求要因 : ハードウェア要求 固定ディスティネーションアドレス 固定ソースアドレス ディスティネーション転送サイズ : 4 バイト転送 ソース転送サイズ : 4 バイト転送
DMA0RS_1	0x00010085	ハードウェア要求あたりの転送回数 : 1 回 ハードウェア要求あたりの転送リミット : DMAjTMR_0.STS×DMAjRS_0.TC により指定された転送サイズ プレロード無効 DRQ 初期化無効 DMA 要求ソース : DMAMSPI25(group0-133)
DMATRGSSELDMACSEL0_8	0xFFFFF3FF	MSPI3(group0-133)

表 3-15 ディスクリプタ 0 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) s_dat (送信データの先頭アドレス)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &KCRC0.DIN.UINT32 (KCRC0 インพุットレジスタ)
DMA0TSR_0	0x0000000E	転送サイズ : 14 バイト
DMA0TMR_0	0x00000111	転送要求要因 : 自動要求
		固定ディスティネーションアドレス
		インクリメントソースアドレス
		ディスティネーション転送サイズ : 2 バイト転送
		ソース転送サイズ : 2 バイト転送
DMA0DPPTR_0	0x00000015	ディスクリプタのアドレス : 0x014
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-16 ディスクリプタ 1 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) &KCRC0.DOUT.UINT32 (KCRC0 アウトพุットレジスタ)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &s_dat[7] (送信データの末尾のアドレス)
DMA0TSR_0	0x00000002	転送サイズ : 2 バイト
DMA0TMR_0	0x00000012	転送要求要因 : 自動要求
		固定ディスティネーションアドレス
		固定ソースアドレス
		ディスティネーション転送サイズ : 2 バイト転送
		ソース転送サイズ : 4 バイト転送
DMA0DPPTR_0	0x00000029	ディスクリプタのアドレス : 0x028
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-17 ディスクリプタ 2 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) s_dat (送信データの先頭アドレス)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &MSPI3.TXDA00.UINT32 (MSPI3 送信データレジスタ)
DMA0TSR_0	0x00000002	転送サイズ : 2 バイト
DMA0TMR_0	0x00000111	転送要求要因 : 自動要求
		固定ディスティネーションアドレス
		インクリメントソースアドレス
		ディスティネーション転送サイズ : 2 バイト転送
		ソース転送サイズ : 2 バイト転送
DMA0DPPTR_0	0x0000003D	ディスクリプタのアドレス : 0x03C
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-18 ディスクリプタ 3 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) &s_dat[1] (2 個目の送信データのアドレス)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &MSPI3.TXDA00.UINT32 (MSPI3 送信データレジスタ)
DMA0TSR_0	0x0000000E	転送サイズ : 14 バイト
DMA0TMR_0	0x00001111	転送要求要因 : ハードウェア要求
		固定ディスティネーションアドレス
		インクリメントソースアドレス
		ディスティネーション転送サイズ : 2 バイト転送
		ソース転送サイズ : 2 バイト転送
DMA0DPPTR_0	0x00000000	ディスクリプタ割り込み無効
		ディスクリプタの継続無効

表 3-19に本動作例で使用する関数一覧を示します。

表 3-19 関数一覧

関数名	概要
main_pe0	各関数の呼び出しを行います。
mspi_init	MSPI の初期化を行います。
sdmac_snd_init	sDMAC ch0(送信用)の初期化を行います。
sdmac_rcv_init	sDMAC ch1(ダミーデータ受信用)の初期化を行います。
descriptor_init	ディスクリプタ RAM の初期化を行います。
kcrc_init	KCRC の初期設定を行います。
port_init	I/O ポートの初期設定を行います。
intc_init	割り込みの初期設定を行います。

3.1.3.2 受信側設定

受信側プログラムで使用する各レジスタの設定例を以下に示します。

表 3-20 KCRC0 レジスタ設定

レジスタ名	設定値	機能
KCRC0CTL	0x000F0001	CRC 入力ビット幅 : 16 ビット
		多項式サイズ : 16 ビット
KCRC0XOR1	0x00000000	CRC XOR マスク上位 : 0x00000000
KCRC0XOR0	0x00000000	CRC XOR マスク下位 : 0x00000000
KCRC0POLY1	0x00000000	CRC 生成方式 : CCITT CRC 多項式生成
KCRC0POLY0	0x00001021	
KCRC0DOUT0	0x0000FFFF	CRC コード生成の開始値

表 3-21 MSPI3 レジスタ設定

レジスタ名	設定値	機能
MSPI3CTL1	0x18000000	スレーブモード
		MSPITSCK のデフォルトレベルはハイレベル
		SOUT をローレベル
		CS アクティブラー
MSPI3CTL0	0x01	MSPI 機能許可
MSPI3CFG00	0x30700003	送信許可
		受信許可
		ダイレクトメモリモード
		チャンネル優先レベル : 7
		チャンネルロック操作禁止
		最終フレームで CHEN クリア & チャンネルオペレーション終了
		送信割り込み許可
MSPI3CFG01	0x03010000	受信割り込み許可
		クロック位相とデータ位相 : デフォルト値
		MSB ファースト
		転送間のアイドルなし
		CS アサート : 非アクティブ
MSPI3CFG02	0x0010	パリティチェック無し
		フレーム長 : 16 ビット
		ボーレート : MSPI3CLK/2 (デフォルト値)
		SCK ディレイタイム : 1
		CS ネゲートディレイタイム : 1
MSPI3HOLD0	0x0001	次フレームアイドル時間 : 1
MSPI3IDLE0	0x0001	次フレーム入力データ時間 : 8
MSPI3INDA0	0x0008	フレーム数 : 8
MSPI3CFSET0	0x0008	チャンネルアクティブ、チャンネル許可
MSPI3CSTS0	0x0003	

表 3-22 PBG レジスタ設定

レジスタ名	設定値	機能
PBGERRSLV30 BGKCPROT	0xA5A5A501	PBG30 のレジスタアクセスを許可
PBG30 PBGPROT1_6	0x10000001	KCRC0 用 SPID : sDMAC、CPU0 を 1 に設定
PBG30 PBGPROT1_11	0x10000001	MSPI3 用 SPID : sDMAC、CPU0 を 1 に設定

表 3-23 DMA/DTS トリガレジスタ設定

レジスタ名	設定値	機能
MSPITGTGCTL3	0x0000000E	Trigger 1=INTMSPInTX0、Trigger 2=INTMSPInRX0

表 3-24 sDMAC ch0(受信用)レジスタ設定

レジスタ名	設定値	機能
DMA0CM_0	0x00001C00	チャンネルマスタ SPID : 0x1C スーパバイザモード
DMA0CHCR_0	0x0302	ディスクリプタ動作有効 ディスクリプタメモリの設定から DMA 転送開始 チャンネルアドレスエラー通知無効 チャンネルアドレスエラー割込み無効 ディスクリプタステップ終了割込み無効 転送完了割込み有効 DMA 転送無効
DMA0DPCR_0	0x0000000F	ディスクリプタのアップデートフラグ DMAjSAR_n レジスタアップデート有効 DMAjDAR_n レジスタアップデート有効 DMAjTSR_n レジスタアップデート有効 DMAjTMR_n レジスタアップデート有効 DMAjGIAI_n レジスタアップデート無効 DMAjGOAI_n レジスタアップデート無効 DMAjSIAI_n レジスタアップデート無効 DMAjSOAI_n レジスタアップデート無効 DMAjSGCR_n レジスタアップデート無効 DMAjRS_n レジスタアップデート無効 DMAjBUFCR_n レジスタアップデート無効
DMA0RS_0	0x00010085	ハードウェア要求あたりの転送回数 : 1 回 ハードウェア要求あたりの転送リミット : DMAjTMR_0.STS×DMAjRS_0.TC により指定された転送サイズ ブロード無効 DRQ 初期化無効 DMA 要求ソース : DMAMSPI25(group0-133)
DMA0DPPTR_0	0x00000001	ディスクリプタのアドレス : 0x000 ディスクリプタ割り込み無効 ディスクリプタの継続有効
DMATRGSSELDMACSEL0_8	0xFFFFF3FF	MSPI3(group0-133)
DMA0OR	0x0001	チャンネル間優先度 : CH0>CH1>...>CH14>CH15 すべてのチャンネルの DMA 転送有効

表 3-25 sDMAC ch1(ダミーデータ送信用)レジスタ設定

レジスタ名	設定値	機能
DMA0CM_1	0x00001C00	チャンネルマスタ SPID : 0x1C スーパバイザモード
DMA0CHCR_1	0x0302	ディスクリプタ動作有効 ディスクリプタメモリの設定から DMA 転送開始 チャンネルアドレスエラー通知無効 チャンネルアドレスエラー割込み無効 ディスクリプタステップ終了割込み無効 転送完了割込み有効 DMA 転送無効
DMA0SAR_1	-	ソースアドレス : (unsigned long) &dummys32 (ダミーデータ)
DMA0DAR_1	-	ディスティネーションアドレス : (unsigned long) &MSPI3.TXDA00.UINT32 (MSPI 送信データレジスタ)
DMA0TSR_1	0x0000001C	転送サイズ : 28 バイト
DMA0TMR_1	0x00001022	転送要求要因 : ハードウェア要求 固定ディスティネーションアドレス 固定ソースアドレス ディスティネーション転送サイズ : 4 バイト転送 ソース転送サイズ : 4 バイト転送
DMA0RS_1	0x00010084	ハードウェア要求あたりの転送回数 : 1 回 ハードウェア要求あたりの転送リミット : DMAjTMR_0.STS×DMAjRS_0.TC により指定された転送サイズ プレロード無効 DRQ 初期化無効 DMA 要求ソース : DMAMSPI24 (group0-132)
DMATRGSSELDMACSEL0_8	0xFFFFFCFF	MSPI3(group0-132)

表 3-26 ディスクリプタ 0 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) &MSPI3.RXDA00.UINT32 (MSPI3 受信データレジスタ)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) r_dat (受信データの先頭アドレス)
DMA0TSR_0	0x00000010	転送サイズ : 16 バイト
DMA0TMR_0	0x00001411	転送要求要因 : ハードウェア要求
		インクリメントディスティネーションアドレス
		固定ソースアドレス
		ディスティネーション転送サイズ : 2 バイト転送
DMA0DPPTR_0	0x00000015	ソース転送サイズ : 2 バイト転送
		ディスクリプタのアドレス : 0x014
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-27 ディスクリプタ 1 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) r_dat (受信データの先頭アドレス)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &KCRC0.DIN.UINT32 (KCRC0 インプットレジスタ)
DMA0TSR_0	0x0000000E	転送サイズ : 14 バイト
DMA0TMR_0	0x00000111	転送要求要因 : 自動要求
		固定ディスティネーションアドレス
		インクリメントソースアドレス
		ディスティネーション転送サイズ : 2 バイト転送
DMA0DPPTR_0	0x00000029	ソース転送サイズ : 2 バイト転送
		ディスクリプタのアドレス : 0x028
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-28 ディスクリプタ 2 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス: (unsigned long) &KCRC0.DOUT.UINT32 (KCRC0 アウトプットレジスタ)
DMA0DAR_0	-	ディスティネーションアドレス: (unsigned long) &kcrc_out (CRC 結果格納変数)
DMA0TSR_0	0x00000002	転送サイズ: 2 バイト
DMA0TMR_0	0x00000021	転送要求要因: 自動要求
		固定ディスティネーションアドレス
		固定ソースアドレス
		ディスティネーション転送サイズ: 2 バイト転送
DMA0DPPTR_0	0x00000000	ソース転送サイズ: 4 バイト転送
		ディスクリプタ割り込み無効
		ディスクリプタの継続無効

表 3-29 割り込みレジスタ設定例

レジスタ名	設定値	機能
EIBD70	0x00000000	PE0 (CPU0) に割り込みをバインド
EIC70	0x0040	テーブル参照/優先レベル 0

表 3-30に本動作例で使用する関数一覧を示します。

表 3-30 関数一覧

関数名	概要
main_pe0	各関数の呼び出しを行います。
mspi_init	MSPI の初期化を行います。
sdmac_snd_init	sDMAC ch0(受信用)の初期化を行います。
sdmac_rcv_init	sDMAC ch1(ダミーデータ送信用)の初期化を行います。
descriptor_init	ディスクリプタ RAM の初期化を行います。
kcrc_init	KCRC の初期設定を行います。
port_init	I/O ポートの初期設定を行います。
intc_init	割り込みの初期設定を行います。

3.1.4 動作シーケンス

図 3-3に送信動作の動作シーケンスを示します。()内の数字は、対応するディスクリプタの番号です。各ディスクリプタの設定による DMA 転送は、以下のように行われます。

- (0) KCRC 入力レジスタに、送信データを書き込み、CRC を生成します。
- (1) (0)で生成した CRC を読み出して、送信データの末尾に書き込みます。
- (2) MSPI の送信データレジスタに、1 データ目を書き込み、送信を開始します。
- (3) MSPI の通信ステータス割り込みをトリガとして、残りのデータを送信データレジスタに書き込みます。

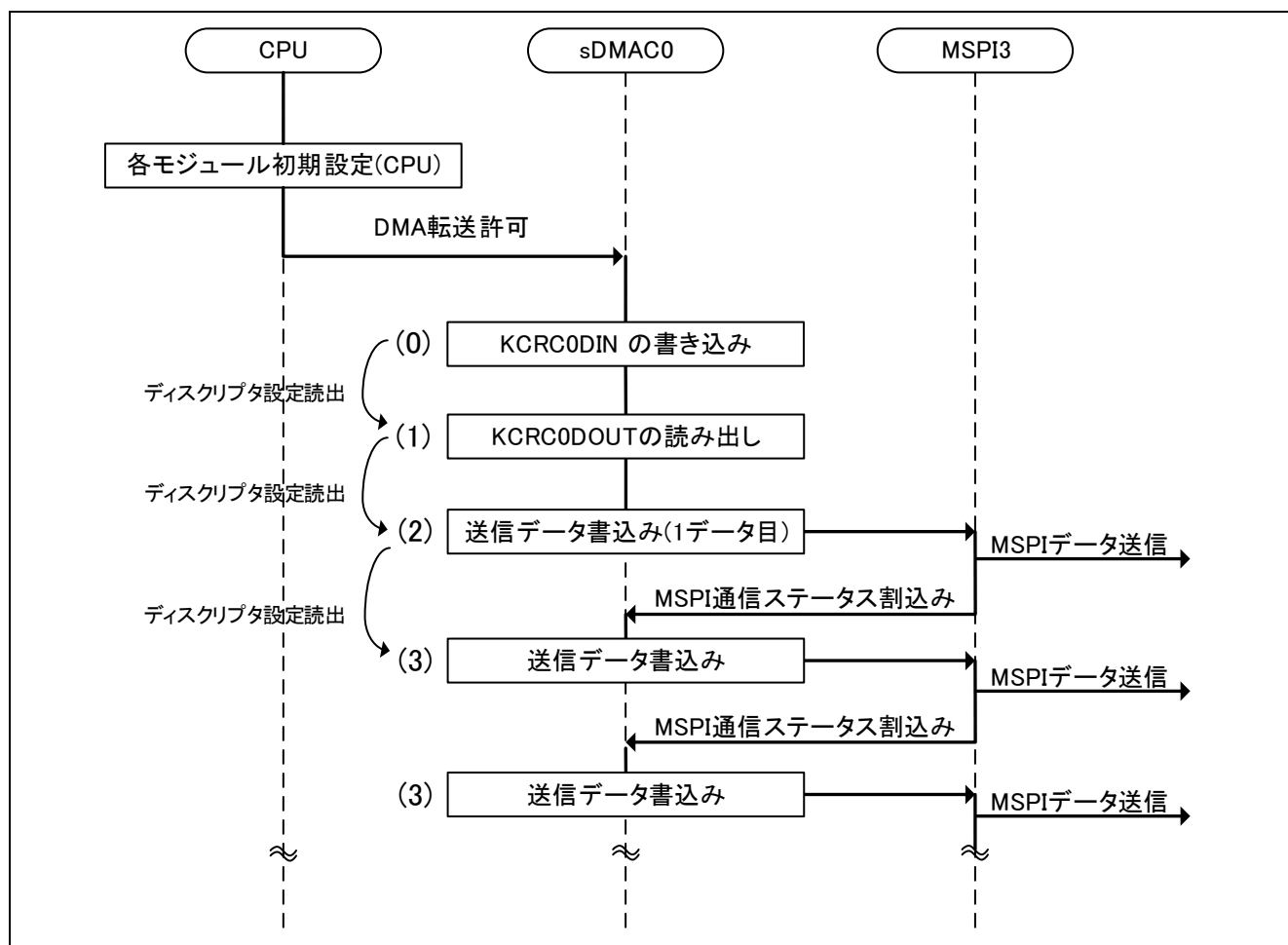


図 3-3 送信動作の動作シーケンス

図 3-4に受信動作の動作シーケンスを示します。()内の数字は、対応するディスクリプタの番号です。各ディスクリプタの設定による DMA 転送は、以下のように行われます。

- (0) MSPI の受信ステータス割り込みをトリガとして、MSPI の受信データレジスタを読み出します。
- (1) (0)で読み出した受信データを、KCRC 入力レジスタに書き込み、CRC を生成します。
- (2) (1)で生成した CRC を読み出して、RAM に保存します。

全ての DMA 転送完了後、sDMAC 転送完了割り込みが発生し、割り込み処理内で受信した CRC と、生成した CRC を CPU で比較して、送信データの妥当性を検証します。

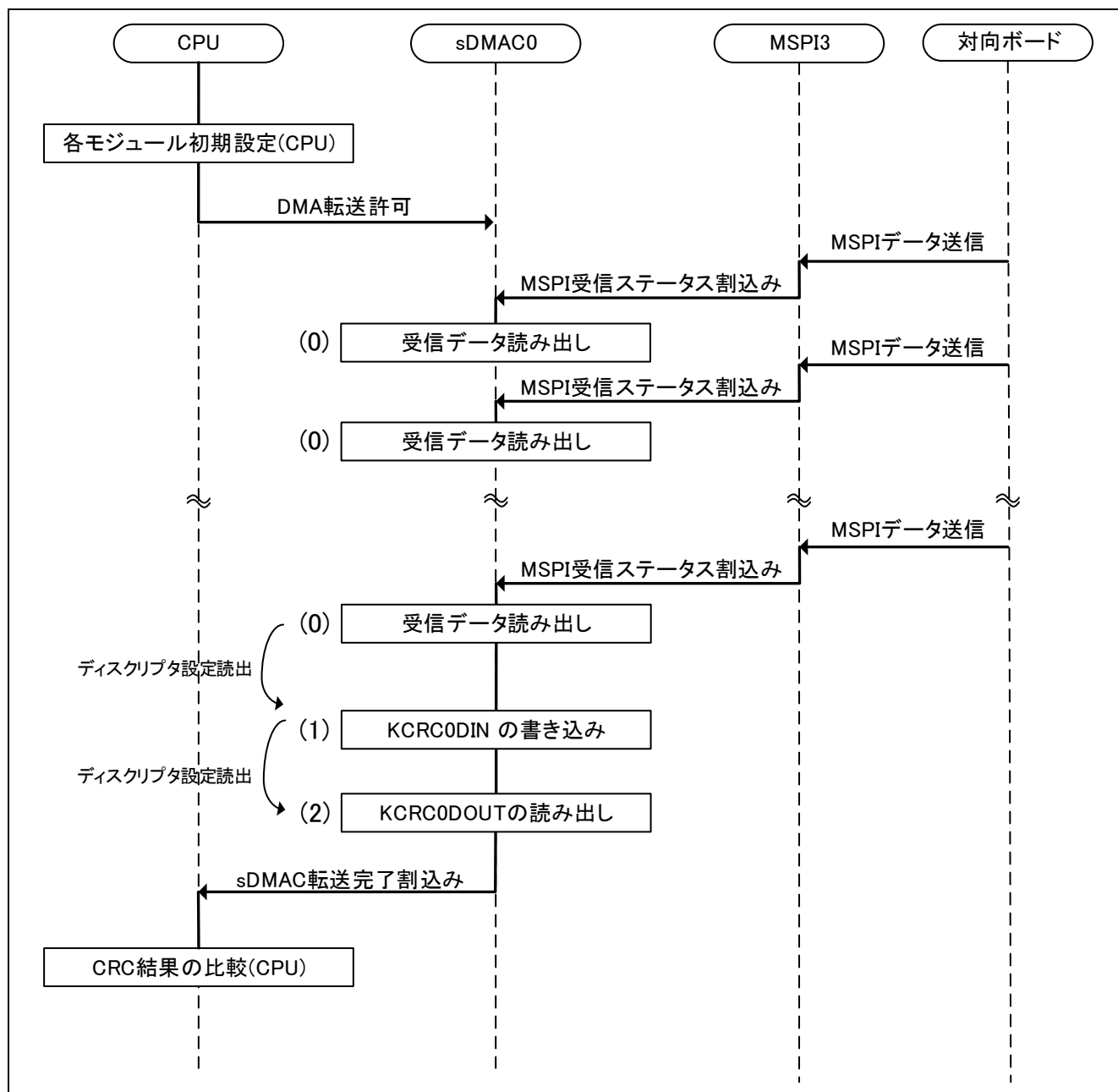


図 3-4 受信動作の動作シーケンス

3.2 動作例 2 CAN FD フレームに格納した CRC による送信データの検証

3.2.1 概要

CANFD フレームに CRC を格納して送信することで、送信データの妥当性を検証することができます。本動作例における、初期設定以外の RAM 及びレジスタへのデータの読み書きは、sDMAC を使用して行い、CPU を介在させないようにしています。

CRC の生成多項式は、32 ビットイーサネット CRC を使用しています。CANFD 通信は、RSCANFD を使用して行い、64byte のデータを 1 回送受信し、末尾の 4byte に CRC を格納して送信します。図 3-5に本動作例の概略図を示します。

送信側

- ① RAM に配置した送信データを KCRC 入力レジスタに書き込み、CRC を生成します。
- ② 生成した CRC を、RAM に配置した送信データの末尾に格納します。
- ③ RSCANFD の送受信 FIFO に送信データを書き込み、送信を開始します。

受信側

- ④ RSCANFD の受信 FIFO から受信データを読み出し、RAM に書き込みます
- ⑤ RAM に配置した受信データを KCRC 入力レジスタに書き込み、CRC' を生成します。
- ⑥ 生成した CRC' を、RAM に配置します。
- ⑦ 生成した CRC' と受信した CRC を CPU で比較して、送信データの妥当性を検証します。

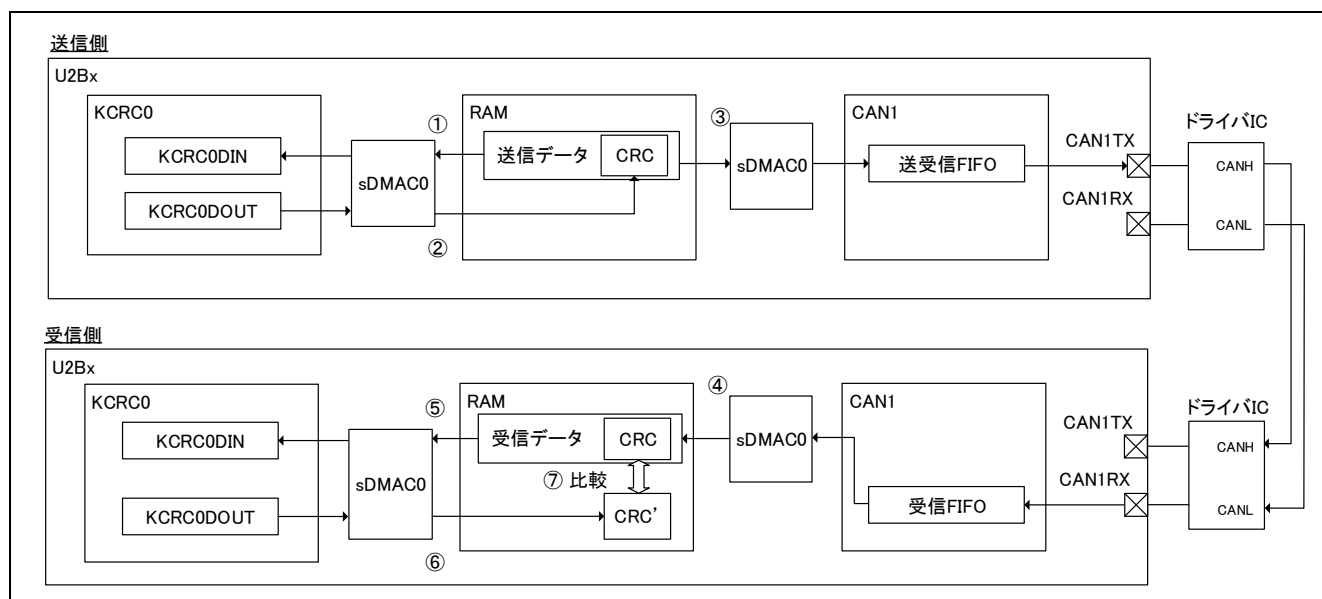


図 3-5 概略図

また、本動作例の sDMAC の動作は、ディスクリプタメモリに定義しており、順次実行されます。ディスクリプタの概要については、3.1.1 概要を参照してください。

3.2.2 使用機能の動作条件

本動作例で使用する機能の動作条件を以下に示します。

3.2.2.1 送信側プログラムの動作条件

送信側プログラムの動作条件を以下に示します。

表 3-31 KCRC0 の設定

項目	内容
入力ビット幅	32 ビット
CRC 生成方式	32 ビットイーサネット CRC 多項式生成

表 3-32 RSCANFD1 の設定

項目	内容
SPI0 動作クロック	40MHz
使用チャネル	ch1
動作モード	CANFD モード
通常ビットレート	1Mbps
データビットレート	2Mbps
送信データ長	64 バイト
送信データ	0x12,0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0, 0x23,0x45,0x67,0x89,0xab,0xcd,0xef,0x01, 0x34,0x56,0x78,0x9a,0x12,0x34,0x56,0x78, 0x9a,0xbc,0xde,0xf0,0x23,0x45,0x67,0x89, 0xab,0xcd,0xef,0x01,0x34,0x56,0x78,0x9a, 0x12,0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0, 0x23,0x45,0x67,0x89,0xab,0xcd,0xef,0x01, 0x34,0x56,0x78,0x9a,CRC(32bit)
送信用バッファ	送受信 FIFO

表 3-33 sDMAC0 の設定

項目	内容
sDMAC0 動作クロック	100MHz
使用チャネル	ch0

表 3-34 ポート設定

項目	内容
使用端子	P2_10 ALT_IN7:CRX1
	P2_7 ALT_OUT7:CTX1

3.2.2.2 受信側プログラムの動作条件

受信側プログラムの動作条件を以下に示します。

表 3-35 KCRC0 の設定

項目	内容
入力ビット幅	32 ビット
CRC 生成方式	32 ビットイーサネット CRC 多項式生成

表 3-36 RSCANFD1 の設定

項目	内容
SPI0 動作クロック	40MHz
使用チャネル	ch1
動作モード	CANFD モード
通常ビットレート	1Mbps
データビットレート	2Mbps
受信用バッファ	受信 FIFO
DMA トリガ	受信 FIFO DMA 要求 0

表 3-37 sDMAC0 の設定

項目	内容
sDMAC0 動作クロック	100MHz
使用チャネル	ch0
割り込み	転送完了割り込み

表 3-38 ポート設定

項目	内容
使用端子	P2_10 ALT_IN7:CRX1
	P2_7 ALT_OUT7:CTX1

3.2.3 ソフトウェア説明

本動作例で使用する各レジスタの設定例を以下に示します。

3.2.3.1 送信側設定

送信側プログラムで使用する各レジスタの設定例を以下に示します。

表 3-39 KCRC0 レジスタ設定

レジスタ名	設定値	機能
KCRC0CTL	0x001F0120	CRC 入力ビット幅 : 32 ビット
		多項式サイズ : 32 ビット
KCRC0XOR1	0x00000000	CRC XOR マスク上位 : 0x00000000
KCRC0XOR0	0xFFFFFFFF	CRC XOR マスク下位 : 0xFFFFFFFF
KCRC0POLY1	0x00000000	CRC 生成方式 : CCITT CRC 多項式生成
KCRC0POLY0	0x04C11DB7	
KCRC0DOUT0	0xFFFFFFFF	CRC コード生成の開始値

表 3-40 RSCANFD1 レジスタ設定

レジスタ名	設定値	機能
RSCFD0CFDGRMCFG	0x00000001	インタフェースモード : CAN FD モード
RSCFD0CFDC0NCFG	0x0D181000	通常ビットレート : 1Mbps
RSCFD0CFDC0DCFG	0x026B0000	データビットレート : 2Mbps
RSCFD0CFDCFCC3	0x00411175	送信バッファリンク : 送信バッファ 3
		送受信 FIFO モード選択 : 送信モード
		送受信 FIFO 割り込み要因選択 : 1 メッセージ送信が完了するごとに FIFO 送信割り込み要求発生
		送受信 FIFO バッファ段数 : 8 メッセージ
		送受信 FIFO バッファペイロード格納サイズ : 64 バイト
		送受信 FIFO 送信割り込み許可
		送受信 FIFO バッファを使用する
RSCFD0CFDCFID3	0x000007F0	標準 ID
		データフレーム
		送信履歴データをバッファに格納しない
		送受信 FIFO バッファ ID : 0x7F0
RSCFD0CFDCFPTR3	0xF0120000	送受信 FIFO バッファ DLC 設定 : 64 データバイト
		送受信 FIFO バッファラベルデータ : 0x12
RSCFD0CFDCFFDCSTS3	0x00000006	CAN FD フレーム
		データ領域のビットレートは変わる

表 3-41 PBG レジスタ設定

レジスタ名	設定値	機能
PBGERRSLV30 BGKCPROT	0xA5A5A501	PBG30 のレジスタアクセスを許可
PBGERRSLV100 BGKCPROT	0xA5A5A501	PBG100 のレジスタアクセスを許可
PBG30 PBGPROT1_6	0x10000001	KCRC0 用 SPID : sDMAC、CPU0 を 1 に設定
PBG100 PBGPROT1_4	0x10000001	RSCAN0-CAN0 用 SPID : sDMAC、CPU0 を 1 に設定

表 3-42 sDMAC レジスタ設定

レジスタ名	設定値	機能
DMA0CM_0	0x00001C00	チャンネルマスタ SPID : 0x1C スーパバイザモード
DMA0CHCR_0	0x0302	ディスクリプタ動作有効 ディスクリプタメモリの設定から DMA 転送開始 チャンネルアドレスエラー通知無効 チャンネルアドレスエラー割込み無効 ディスクリプタステップ終了割込み無効 転送完了割込み有効 DMA 転送無効
DMA0DPCR_0	0x0000000F	ディスクリプタのアップデートフラグ DMAjSAR_n レジスタアップデート有効 DMAjDAR_n レジスタアップデート有効 DMAjTSR_n レジスタアップデート有効 DMAjTMR_n レジスタアップデート有効 DMAjGIAI_n レジスタアップデート無効 DMAjGOAI_n レジスタアップデート無効 DMAjSIAI_n レジスタアップデート無効 DMAjSOAI_n レジスタアップデート無効 DMAjSGCR_n レジスタアップデート無効 DMAjRS_n レジスタアップデート無効 DMAjBUFCR_n レジスタアップデート無効
DMA0DPPTR_0	0x00000001	ディスクリプタのアドレス : 0x000 ディスクリプタ割り込み無効 ディスクリプタの継続有効
DMA0OR	0x0001	チャンネル間優先度 : CH0>CH1>...>CH14>CH15 すべてのチャンネルの DMA 転送有効

表 3-43 ディスクリプタ 0 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) tx_buf_table[0].DB (送信データの先頭アドレス)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &KCRC0.DIN.UINT32 (KCRC0 インพุットレジスタ)
DMA0TSR_0	0x0000003C	転送サイズ : 60 バイト
DMA0TMR_0	0x00000122	転送要求要因 : 自動要求
		固定ディスティネーションアドレス
		インクリメントソースアドレス
		ディスティネーション転送サイズ : 4 バイト転送
		ソース転送サイズ : 4 バイト転送
DMA0DPPTR_0	0x00000015	ディスクリプタのアドレス : 0x014
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-44 ディスクリプタ 1 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) &KCRC0.DOUT.UINT32 (KCRC0 アウトพุットレジスタ)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &tx_buf_table[0].DB[60] (送信データの末尾のアドレス)
DMA0TSR_0	0x00000004	転送サイズ : 4 バイト
DMA0TMR_0	0x00000022	転送要求要因 : 自動要求
		固定ディスティネーションアドレス
		固定ソースアドレス
		ディスティネーション転送サイズ : 4 バイト転送
		ソース転送サイズ : 4 バイト転送
DMA0DPPTR_0	0x00000029	ディスクリプタのアドレス : 0x028
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-45 ディスクリプタ 2 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) tx_buf_table (送信データの先頭アドレス)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &RSCFD0.CFDCFD3.UINT32 (送受信 FIFO バッファアクセス ID レジスタ)
DMA0TSR_0	0x0000004C	転送サイズ : 76 バイト
DMA0TMR_0	0x00000522	転送要求要因 : 自動要求
		インクリメントディスティネーションアドレス
		インクリメントソースアドレス
		ディスティネーション転送サイズ : 4 バイト転送
		ソース転送サイズ : 4 バイト転送
DMA0DPPTR_0	0x0000003D	ディスクリプタのアドレス : 0x03C
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-46 ディスクリプタ 3 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) &can_fifo_ptr_inc (送受信 FIFO バッファポインタ制御レジスタ書き込みデータのアドレス)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &RSCFD0.CFDCFPCTR3.UINT32 (送受信 FIFO バッファポインタ制御レジスタ)
DMA0TSR_0	0x00000004	転送サイズ : 4 バイト
DMA0TMR_0	0x00000022	転送要求要因 : 自動要求
		固定ディスティネーションアドレス
		固定ソースアドレス
		ディスティネーション転送サイズ : 4 バイト転送
		ソース転送サイズ : 4 バイト転送
DMA0DPPTR_0	0x00000000	ディスクリプタ割り込み無効
		ディスクリプタの継続無効

表 3-47に本動作例で使用する関数一覧を示します。

表 3-47 関数一覧

関数名	概要
main_pe0	各関数の呼び出しを行います。
PORT_Init	I/O ポートの初期設定を行います。
R_CAN_Init	RSCANFD の初期化を行います。
R_CAN_GlobalStart	RSCANFD のグローバル動作を開始します。
R_CAN_ChStart	RSCANFD のチャネル動作を開始します。
sdmac_snd_init	sDMAC の初期化を行います。
descriptor_init	ディスクリプタ RAM の初期化を行います。
kcrc_init	KCRC の初期設定を行います。
intc_init	割り込みの初期設定を行います。

3.2.3.2 受信側設定

受信側プログラムで使用する各レジスタの設定例を以下に示します。

表 3-48 KCRC0 レジスタ設定

レジスタ名	設定値	機能
KCRC0CTL	0x001F0120	CRC 入力ビット幅 : 32 ビット
		多項式サイズ : 32 ビット
KCRC0XOR1	0x00000000	CRC XOR マスク上位 : 0x00000000
KCRC0XOR0	0xFFFFFFFF	CRC XOR マスク下位 : 0xFFFFFFFF
KCRC0POLY1	0x00000000	CRC 生成方式 : CCITT CRC 多項式生成
KCRC0POLY0	0x04C11DB7	
KCRC0DOUT0	0xFFFFFFFF	CRC コード生成の開始値

表 3-49 RSCANFD1 レジスタ設定

レジスタ名	設定値	機能
RSCFD0CFDGRMCFG	0x00000001	インタフェースモード : CAN FD モード
RSCFD0CFDC0NCFG	0x0D181000	通常ビットレート : 1Mbps
RSCFD0CFDC0DCFG	0x026B0000	データビットレート : 2Mbps
RSCFD0CFDRFCC1	0x00000271	受信 FIFO 割り込み要因選択 : 1 メッセージ受信完了ごとに発生
		受信 FIFO バッファ段数 : 8 メッセージ
		受信 FIFO バッファペイロード格納サイズ : 64 バイト
		受信 FIFO 割り込み禁止
		受信 FIFO バッファを使用する
RSCFD0CFDGAFLECTR	0x00000100	受信ルールテーブル書き込み許可
RSCFD0CFDGAFLCFG0	0x00000001	チャンネル 1 受信ルール数 : 1
RSCFD0CFDGAFLID0	0x000007F0	標準 ID
		データフレーム
		受信ルール対象メッセージ : 他の CAN ノードが送信したメッセージを受信時
		受信ルールの ID : 0x7F0
RSCFD0CFDGAFLM0	0xC00007FF	IDE ビットを比較する
		RTR ビットを比較する
		ID ビットを比較する
RSCFD0CFDGAFLP0_0	0xF1230000	受信ルール DLC 設定 : 64 データバイト
		受信バッファを使用しない
RSCFD0CFDGAFLP1_0	0x00000002	受信 FIFO バッファ 1 選択
RSCFD0CFDCDTCT	0x00000002	受信 FIFO バッファ 1DMA 転送要求許可

表 3-50 PBG レジスタ設定

レジスタ名	設定値	機能
PBGERRSLV30 BGKCPROT	0xA5A5A501	PBG30 のレジスタアクセスを許可
PBG30 PBGPROT1_6	0x10000001	KCRC0 用 SPID : sDMAC、CPU0 を 1 に設定

表 3-51 sDMAC レジスタ設定

レジスタ名	設定値	機能
DMA0CM_0	0x00001C00	チャンネルマスタ SPID : 0x1C
		スーパバイザモード
DMA0CHCR_0	0x0302	ディスクリプタ動作有効
		ディスクリプタメモリの設定から DMA 転送開始
		チャンネルアドレスエラー通知無効
		チャンネルアドレスエラー割込み無効
		ディスクリプタステップ終了割込み無効
		転送完了割込み有効
		DMA 転送無効
DMA0DPCR_0	0x0000000F	ディスクリプタのアップデートフラグ
		DMAjSAR_n レジスタアップデート有効
		DMAjDAR_n レジスタアップデート有効
		DMAjTSR_n レジスタアップデート有効
		DMAjTMR_n レジスタアップデート有効
		DMAjGIAI_n レジスタアップデート無効
		DMAjGOAI_n レジスタアップデート無効
		DMAjSIAI_n レジスタアップデート無効
		DMAjSOAI_n レジスタアップデート無効
		DMAjSGCR_n レジスタアップデート無効
		DMAjRS_n レジスタアップデート無効
		DMAjBUFCR_n レジスタアップデート無効
DMA0RS_0	0x001000C3	ハードウェア要求あたりの転送回数 : 16 回
		ハードウェア要求あたりの転送リミット : DMAjTMR_0.STS×DMAjRS_0.TC により指定された転送サイズ
		プレロード無効
		DRQ 初期化無効
		DMA 要求ソース : INTRCANRFDREQ1(group0-195)
DMA0DPPTR_0	0x00000001	ディスクリプタのアドレス : 0x000
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効
DMATRGSELDMACSEL0_12	0xFFFFFFFF3F	CAN0(group0-195)

表 3-52 ディスクリプタ 0 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) &RSCFD0.CFDRFDF0_0.UINT32 (受信 FIFO バッファ アクセスデータフィールド)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) r_dat (受信データの先頭アドレス)
DMA0TSR_0	0x00000004	転送サイズ : 4 バイト
DMA0TMR_0	0x00001522	転送要求要因 : ハードウェア要求
		インクリメントディスティネーションアドレス
		インクリメントソースアドレス
		ディスティネーション転送サイズ : 4 バイト転送
DMA0DPPTR_0	0x00000015	ソース転送サイズ : 4 バイト転送
		ディスクリプタのアドレス : 0x014
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-53 ディスクリプタ 1 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) &RSCFD0.CFDRFDF1_0.UINT32 (受信 FIFO バッファ アクセスデータフィールド)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) r_dat [1](受信データの先頭アドレス+1)
DMA0TSR_0	0x0000003C	転送サイズ : 60 バイト
DMA0TMR_0	0x00000522	転送要求要因 : 自動要求
		インクリメントディスティネーションアドレス
		インクリメントソースアドレス
		ディスティネーション転送サイズ : 4 バイト転送
DMA0DPPTR_0	0x00000029	ソース転送サイズ : 4 バイト転送
		ディスクリプタのアドレス : 0x028
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-54 ディスクリプタ 2 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) r_dat (受信データの先頭アドレス)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &KCRC0.DIN.UINT32 (KCRC インพุットレジスタ)
DMA0TSR_0	0x0000003C	転送サイズ : 60 バイト
DMA0TMR_0	0x00000122	転送要求要因 : 自動要求
		固定ディスティネーションアドレス
		インクリメントソースアドレス
		ディスティネーション転送サイズ : 4 バイト転送
		ソース転送サイズ : 4 バイト転送
DMA0DPPTR_0	0x0000003D	ディスクリプタのアドレス : 0x03C
		ディスクリプタ割り込み無効
		ディスクリプタの継続有効

表 3-55 ディスクリプタ 3 設定

レジスタ名	設定値	機能
DMA0SAR_0	-	ソースアドレス : (unsigned long) &KCRC0.DOUT.UINT32 (KCRC0 アウトพุットレジスタ)
DMA0DAR_0	-	ディスティネーションアドレス : (unsigned long) &kcrc_out (CRC 結果格納変数のアドレス)
DMA0TSR_0	0x00000004	転送サイズ : 4 バイト
DMA0TMR_0	0x00000022	転送要求要因 : 自動要求
		固定ディスティネーションアドレス
		固定ソースアドレス
		ディスティネーション転送サイズ : 4 バイト転送
		ソース転送サイズ : 4 バイト転送
DMA0DPPTR_0	0x00000000	ディスクリプタ割り込み無効
		ディスクリプタの継続無効

表 3-56 割り込みレジスタ設定例

レジスタ名	設定値	機能
EIBD70	0x00000000	PE0 (CPU0) に割り込みをバインド
EIC70	0x0040	テーブル参照/優先レベル 0

表 3-57に本動作例で使用する関数一覧を示します。

表 3-57 関数一覧

関数名	概要
main_pe0	各関数の呼び出しを行います。
PORT_Init	I/O ポートの初期設定を行います。
R_CAN_Init	RSCANFD の初期化を行います。
R_CAN_GlobalStart	RSCANFD のグローバル動作を開始します。
R_CAN_ChStart	RSCANFD のチャネル動作を開始します。
sdmac_rcv_init	sDMAC の初期化を行います。
descriptor_init	ディスクリプタ RAM の初期化を行います。
kcrc_init	KCRC の初期設定を行います。
intc_init	割り込みの初期設定を行います。

3.2.4 動作シーケンス

図 3-6に送信動作の動作シーケンスを示します。()内の数字は、対応するディスクリプタの番号です。各ディスクリプタの設定による DMA 転送は、以下のように行われます。

- (0) KCRC 入力レジスタに、送信データを書き込み、CRC を生成します。
- (1) (0)で生成した CRC を読み出して、送信データの末尾に書き込みます。
- (2) RSCANFD の送受信 FIFO に、送信データを書き込みます。
- (3) 送受信 FIFO バッファポインタ制御レジスタに FFh を書き込み、送信を開始します。

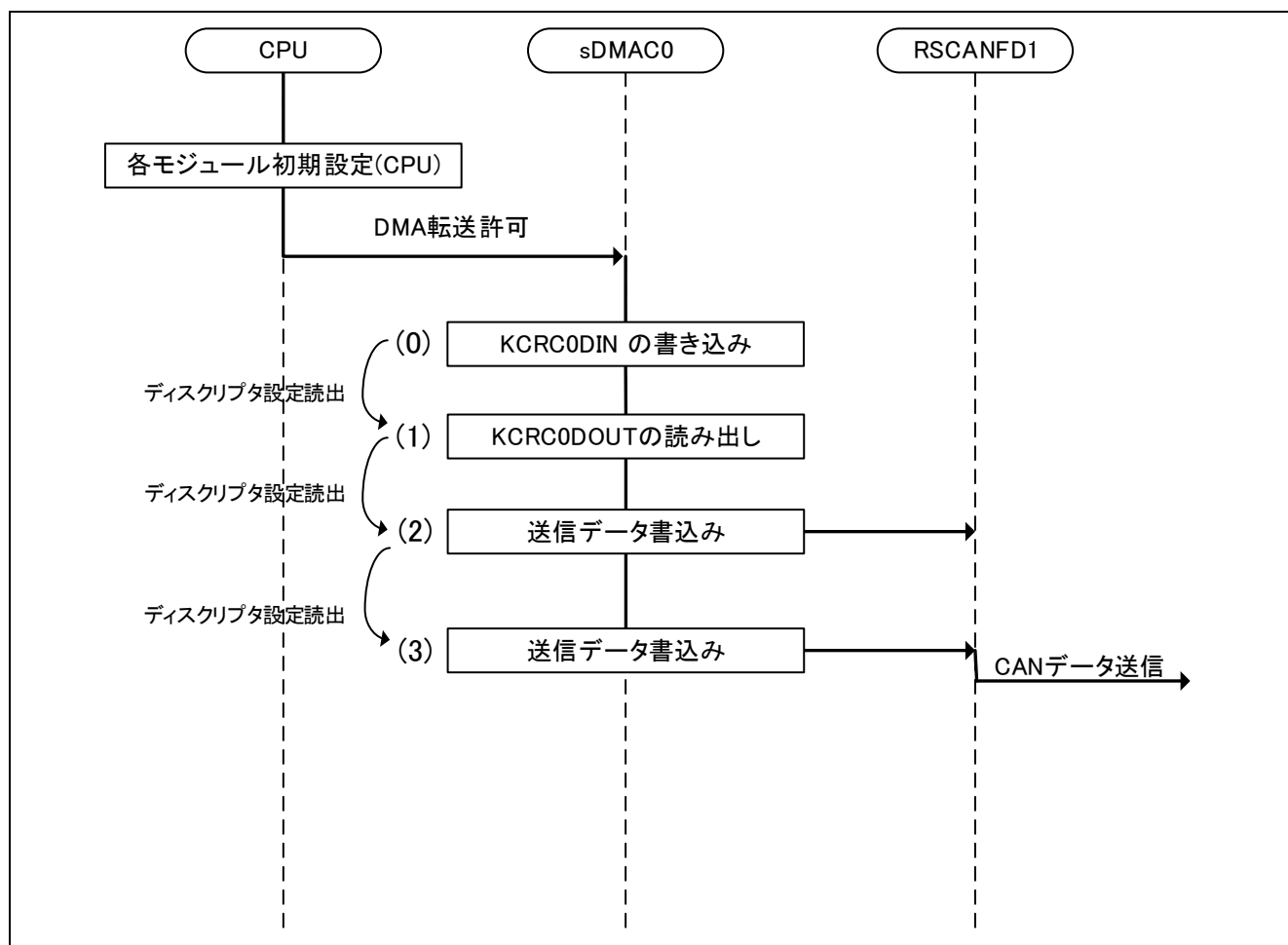


図 3-6 送信動作の動作シーケンス

図 3-7に受信動作の動作シーケンスを示します。()内の数字は、対応するディスクリプタの番号です。各ディスクリプタの設定による DMA 転送は、以下のように行われます。

- (0) RSCANFD の受信 FIFO DMA 要求をトリガとして、受信 FIFO から受信データを読み出します。
- (1) (4)で読み出した受信データを、KCRC 入力レジスタに書き込み、CRC を生成します。
- (2) (5)で生成した CRC を読み出して、RAM に保存します。

全ての DMA 転送完了後、sDMAC 転送完了割り込みが発生し、割り込み処理内で受信した CRC と、生成した CRC を比較して、送信データの妥当性を検証します。

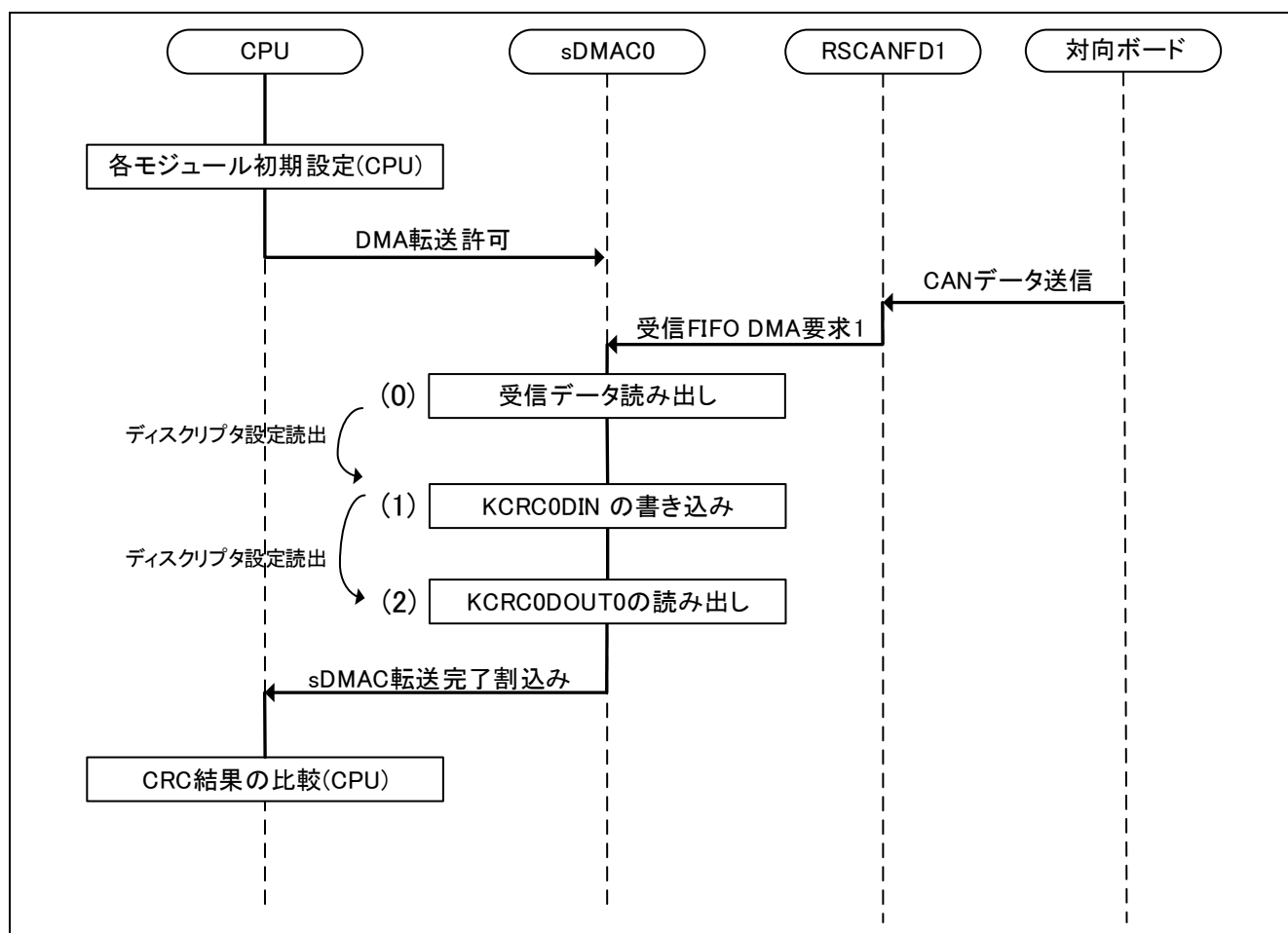


図 3-7 受信動作の動作シーケンス

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2023.8.31	－	初版発行

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. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力ブルアップ電源を入れないでください。入力信号や入出力ブルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「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。

5. クロックについて

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。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

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、将来の拡張機能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。

8. 製品間の相違について

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、フラッシュメモリ、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、ノイズ幅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まれます。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、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、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想定していません。たとえ、当社が想定してい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も、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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、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、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。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. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
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. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

- 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直接的、間接的に支配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.4.0-1 2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。