
SH7670グループ

R01AN0305JJ0101

Rev. 1.01

DMAC メモリ間転送例

2010.10.15

要旨

本アプリケーションノートでは、SH7670/SH7671/SH7672/SH7673 のダイレクトメモリアクセスコントローラ (DMAC)のメモリ間転送例を説明しています。

動作確認デバイス

SH7670

目次

1. はじめに.....	2
2. 応用例の説明.....	3
3. 参考プログラムリスト.....	9
4. 参考ドキュメント.....	14

1. はじめに

1.1 仕様

- DMAC チャンネル 0 を使用して内蔵 RAM から外部メモリへのデータ転送を行います。データ転送はサイクルスチールモードを使用します。
- DMA 転送要求として、オートリクエスト (ソフトウェアによる転送要求) を使用します。

1.2 使用機能

- ダイレクトメモリアクセスコントローラ (DMAC チャンネル 0)

1.3 適用条件

マイコン	SH7670/SH7671/SH7672/SH7673 (R5S76700/R5S76710/R5S76720/R5S76730)
動作周波数	内部クロック : 200 MHz バスクロック : 66.67 MHz 周辺クロック : 33.33MHz
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Ver.4.03.00
Cコンパイラ	ルネサスエレクトロニクス製SuperH RISC engineファミリ C/C++コンパイラパッケージ Ver.9.01 Release01
コンパイルオプション	High-performance Embedded Workshopでのデフォルト設定 (-cpu=sh2afpu -fpu=single -object="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo)

1.4 関連アプリケーションノート

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。合わせて参照してください。

- SH7670 グループ 初期設定例
- SH7670 グループ DMAC デュアルアドレスモード
- SH7670 グループ DMAC 内蔵モジュールへの転送

2. 応用例の説明

本プログラム例ではダイレクトメモリアクセスコントローラ (DMAC) を使用し、内蔵 RAM から外部メモリへのデータ転送を行います。

2.1 使用機能の動作概要

DMAC は DMA 転送要求があると、決められたチャネルの優先順位にしたがって転送を開始し、転送終了条件が満たされると転送を終了します。転送要求にはオートリクエスト、外部リクエスト、内蔵周辺モジュールリクエストの3種類のモードがあります。バスモードはバーストモードとサイクルスチールモードを選択することができます。

表 1 に DMAC の概要を示します。図 1 に DMAC の概念図を示します。

表1 DMAC の概要

項目	概要
チャネル数	8 チャネル (CH0~CH7) CH0~CH1 の 2 チャネルのみ、外部リクエスト受け付けが可能
アドレス空間	4G バイト
転送データ長	バイト、ワード (2 バイト)、ロングワード (4 バイト)、 16 バイト (ロングワード×4)
最大転送回数	16,777,216 (24 ビット) 回
アドレスモード	シングルアドレスモード、デュアルアドレスモード
転送要求	オートリクエスト、外部リクエスト、内蔵周辺モジュールリクエスト (SCIF: 6 要因、IIC3: 2 要因、CMT: 2 要因、USB: 2 要因、SSI: 2 要因)
バスモード	サイクルスチールモード、バーストモード
優先順位	チャネル優先順位固定モード、ラウンドロビンモード
割り込み要求	データ転送 1/2 終了時またはデータ転送終了時に CPU へ割り込み要求発生
外部リクエスト検出	DREQ 入力のロー/ハイレベル検出、立ち上がり/立ち下がリエッジ検出
転送要求受付信号/ 転送終了信号	DACK/TEND 信号のアクティブレベル選択が可能

【注】 DMAC についての詳細は、「SH7670グループ ハードウェアマニュアル ダイレクトメモリアクセスコントローラ」の章をご参照ください。

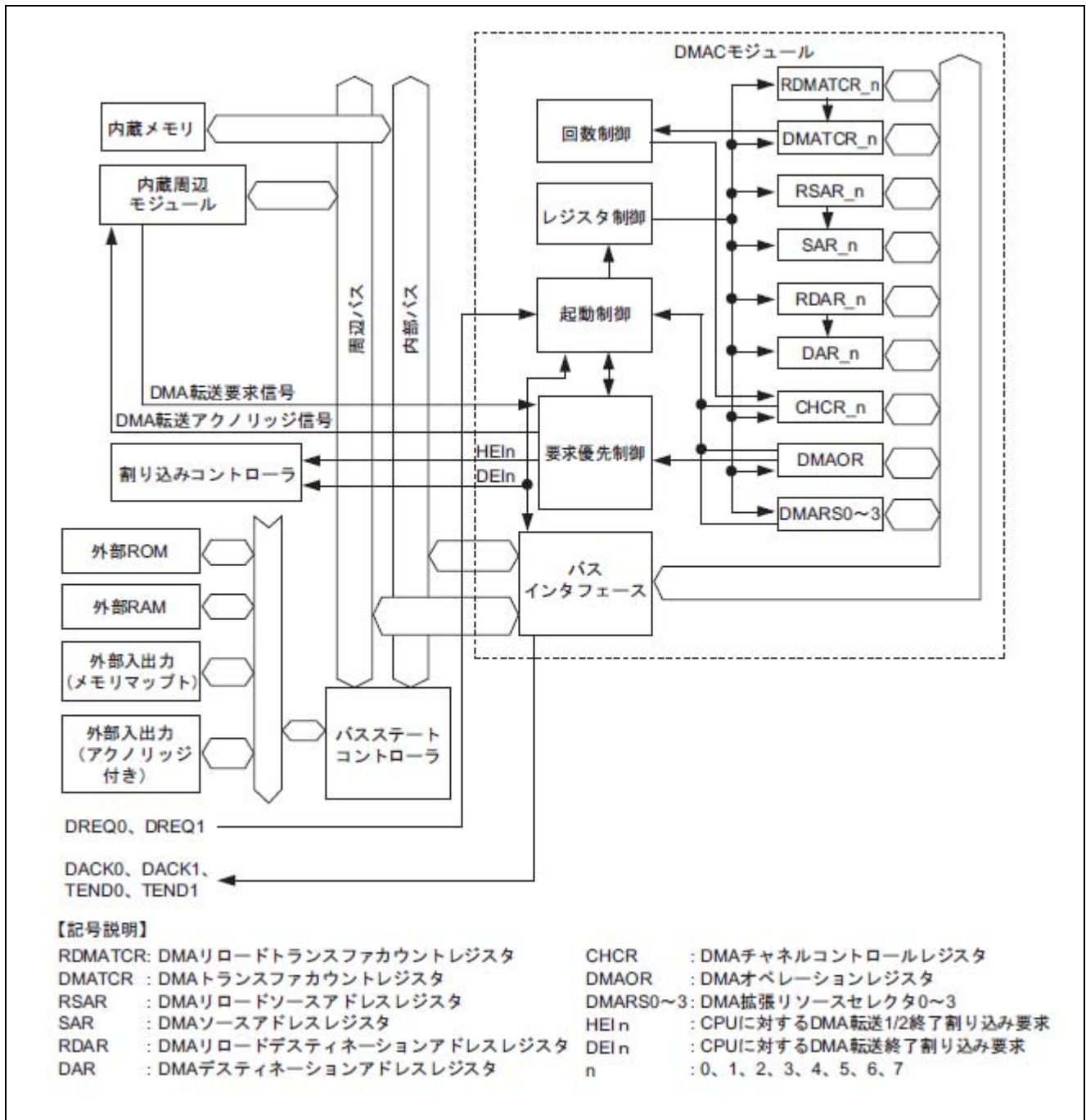


図1 DMAC の概念図

2.2 使用機能の設定手順

ここでは、DMACによるメモリ間転送を行う場合の初期設定手順について説明します。転送要求はオートリクエストを使用します。図2にDMACの初期設定フロー例を示します。なお、各レジスタ設定の詳細は、「SH7670グループ ハードウェアマニュアル」を参照ください。

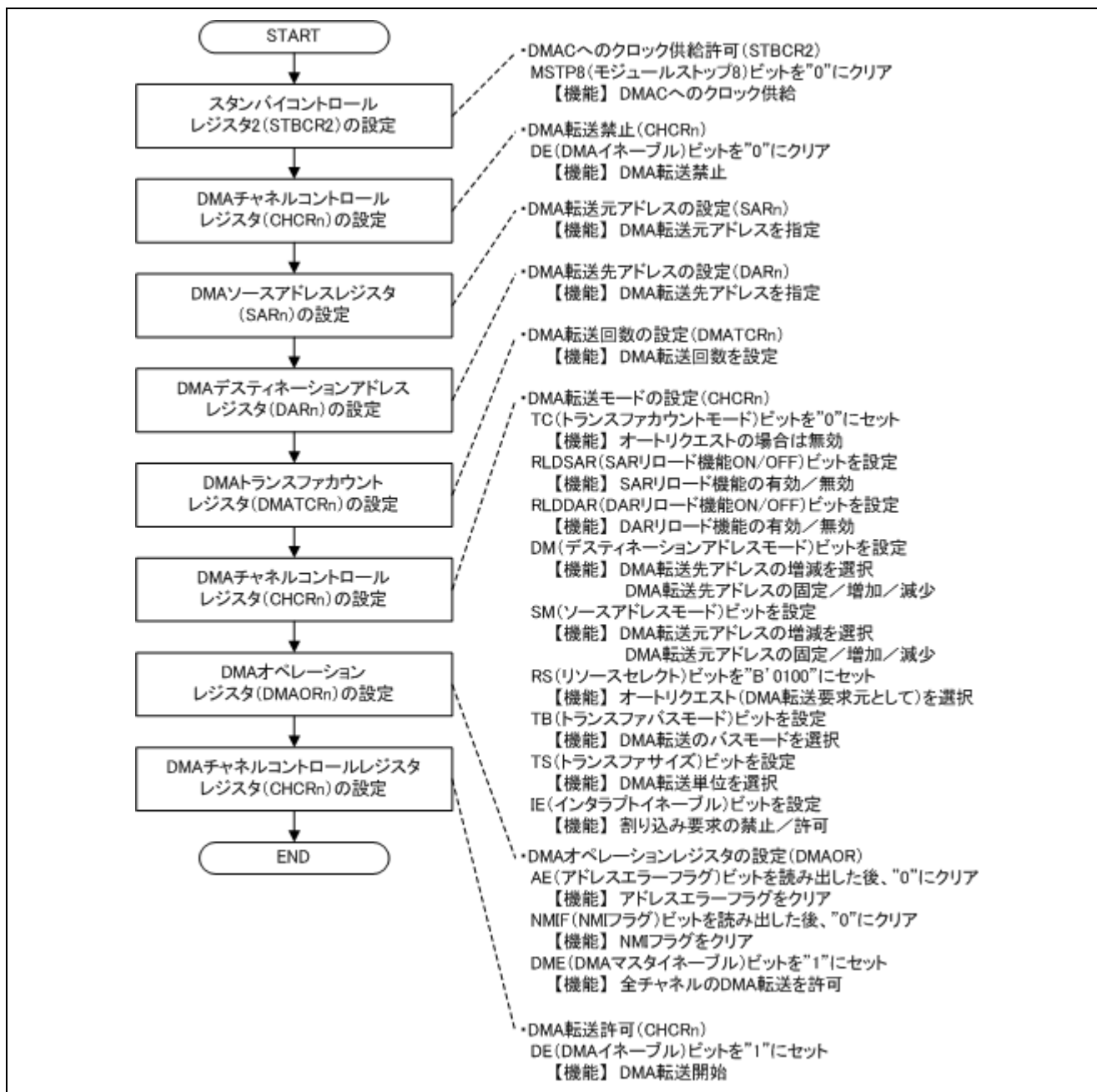


図2 DMACの初期設定フロー例

2.3 参考プログラムの動作

参考プログラムでは、オートリクエストにより DMAC チャンネル0 を起動し、内蔵 RAM から外部メモリへのデータ転送をサイクルスチールモードで行います。サイクルスチール転送のため、1 データ転送ごとに、DMAC はバス権を CPU に解放します。図 3 に参考プログラムの動作タイミングを示します。

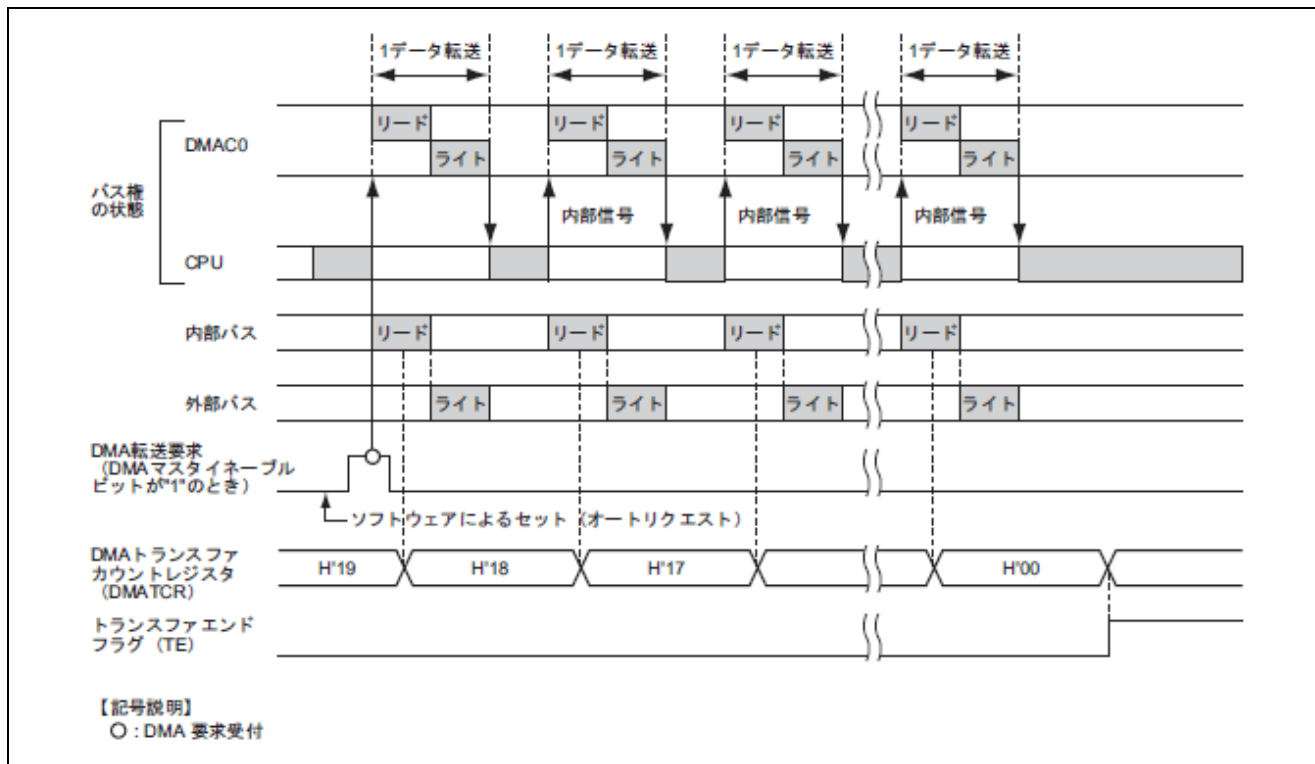


図3 参考プログラムの動作タイミング

2.4 参考プログラム使用時の注意事項

参考プログラムでは、データの転送元の開始アドレスと転送先の開始アドレスを明確にするため、絶対番地で開始アドレスを割り付けています。開始アドレスを割り付けた絶対番地が、ユーザプログラムで使用しているセクションと重ならないようにしてください。

2.5 参考プログラムの処理手順

参考プログラムでは、内蔵 RAM に格納された 100 バイトのデータを外部メモリに DMA 転送します。DMA 転送終了検出は、トランスファエンドフラグ (TE ビット) で確認します。

表 2 に参考プログラムのレジスタ設定を示します。また、表 3 に参考プログラムで使用しているマクロ定義を示します。図 4 に参考プログラムの処理フローを示します。

表2 参考プログラムのレジスタ設定

レジスタ名	アドレス	設定値	機能
スタンバイコントロール レジスタ 2 (STBCR2)	H'FFFE 0018	H'00	MSTP8="0": DMAC は動作
DMA チャンネルコントロール レジスタ_0 (CHCR0)	H'FFFE 100C	H'0000 0000	DE="0": DMA 転送禁止
		H'0000 5410	TC = "0" オートリクエストの場合は無効 RLDSAR = "0": SAR リロード機能無効 RLDDAR = "0": DAR リロード機能無効 DM = "B'01": デスティネーションアドレス増加 SM = "B'01": ソースアドレス増加 RS = "B'0100": オートリクエスト TB = "0": サイクルスチールモード TS = "B'10": ロングワード転送 IE = "0": 割り込み要求禁止
		H'0000 5411	DE = "1": DMA 転送許可
DMA ソースアドレスレジスタ_0 (SAR0)	H'FFFE 1000	H'FFF8 4000	転送元の開始アドレス: 内蔵 RAM 領域に設定
DMA デスティネーション アドレスレジスタ_0 (DAR0)	H'FFFE 1004	H'2C00 1000	転送先の開始アドレス: 外部メモリ領域に設定*
DMA トランスファカウンタ レジスタ_0 (DMATCR0)	H'FFFE 1008	H'64	転送回数: 100 回 (H'64)
DMA オペレーションレジスタ (DMAOR)	H'FFFE 1200	H'0001	DME = "1": 全チャンネル DMA 転送許可
DMA 拡張リソースセクタ_0 (DMARS0)	H'FFFE 1300	H'0000	オートリクエストでは 使用していません

【注】 *外部メモリ領域のアドレスは、使用するターゲットボードによって異なります。

表3 参考プログラムで使用しているマクロ定義

マクロ定義	設定値	機能
SDRAM_DST_ADR	H'2C00 1000	• SDRAM のアドレス
SRAM_SRC_ADR	H'FFF8 4000	• 内蔵 RAM の先頭アドレス
SIZE	H'64	• 転送回数
DMA_SIZE_BYTE	H'0000	• バイト転送
DMA_SIZE_WORD	H'0001	• ワード転送
DMA_SIZE_LONG	H'0002	• ロングワード転送
DMA_SIZE_LONGx4	H'0003	• 16 バイト転送
DMA_INT_DISABLE	H'0000	• DMA 転送完了割り込み未使用
DMA_INT_ENABLE	H'0010	• DMA 転送完了割り込み使用

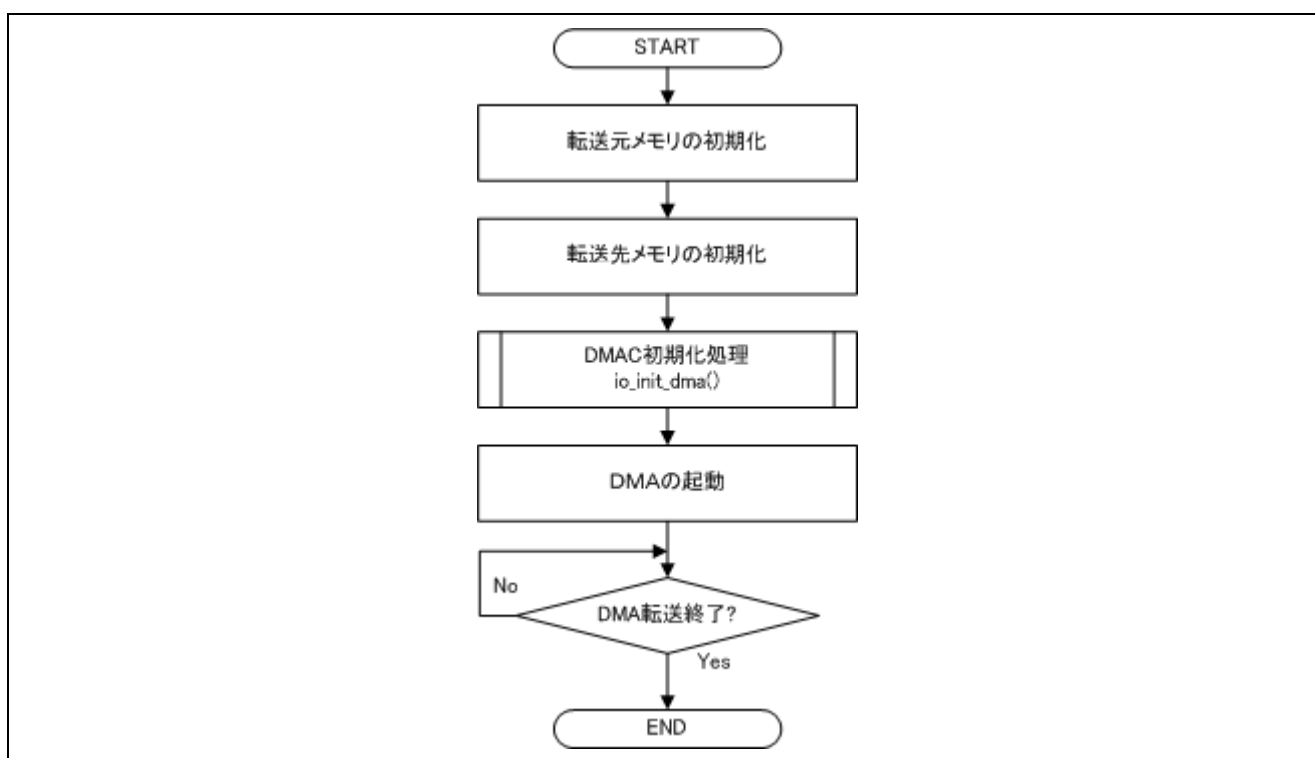


図4 参考プログラムの処理フロー

3. 参考プログラムリスト

3.1 サンプルプログラムリスト"main.c" (1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 * Copyright (C) 2008(2010) Renesas Electronics Corporation All rights reserved.
29 * "FILE COMMENT" ***** Technical reference data *****
30 * System Name : SH7671 Sample Program
31 * File Name   : main.c
32 * Abstract    : DMAC サンプルプログラム
33 * Version     : 1.00.01
34 * Device      : SH7671
35 * Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              : (Ver.9.01 Release01).
38 * OS          : None
39 * H/W Platform: M3A-HS71(CPU board)
40 * Description :
41 *****/
42 * History     : Apr.24,2008 ver.1.00.00
43 *              : Oct.08,2010 ver.1.00.01 Changed the company name and device name
44 * "FILE COMMENT END" *****/
45 #include <machine.h>
46 #include <stdio.h>
47 #include "iodefine.h"          /* SH7670 iodefine */
48
```

3.2 サンプルプログラムリスト"main.c" (2)

```
49  /* ==== symbol definition ==== */
50  #define SDRAM_DST_ADR ((void *)0x2c001000) /* External SDRAM address */
51  #define SRAM_SRC_ADR ((void *)0xffff84000) /* Internal SRAM address */
52  #define SIZE 100 /* Transmission bytes */
53  #define DMA_SIZE_BYTE 0x0000u
54  #define DMA_SIZE_WORD 0x0001u
55  #define DMA_SIZE_LONG 0x0002u
56  #define DMA_SIZE_LONGx4 0x0003u
57  #define DMA_INT_DISABLE 0x0000u
58  #define DMA_INT_ENABLE 0x0010u
59  #define DMA_INT (DMA_INT_ENABLE >> 4u)
60
61  /* ==== prototype declaration ==== */
62  void main(void);
63  void io_init_dma(void *src, void *dst, size_t size, unsigned int mode);
64  void io_dma_enable(void);
65  void io_dma_stop(void);
66
67  /*"FUNC COMMENT"*****
68  * ID :
69  * Outline : メイン関数
70  *-----
71  * Include : "iodefine.h"
72  *-----
73  * Declaration : void main(void);
74  *-----
75  * Description : サンプルプログラムメイン
76  *-----
77  * Argument : void
78  *-----
79  * Return Value : void
80  *-----
81  * Note :
82  *"FUNC COMMENT END"*****/
83  void main(void)
84  {
85      int i;
86      volatile unsigned char *ptr;
87
88      /* ==== Initialize source memory ==== */
89      ptr = SRAM_SRC_ADR;
90      for(i=0; i < SIZE; i++){
91          *ptr++ = 0x55;
92      }
93
94      /* ==== Initialize destination memory ==== */
95      ptr = SDRAM_DST_ADR;
96      for(i=0; i < SIZE; i++){
97          *ptr++ = 0;
98      }
99
```

3.3 サンプルプログラムリスト"main.c" (3)

```

100     /* ==== Setting of DMAC ==== */
101     io_init_dma(SRAM_SRC_ADR, SDRAM_DST_ADR, SIZE , DMA_SIZE_LONG | DMA_INT_DISABLE);
102
103     /* ---- DMA start ---- */
104     io_dma_enable();
105
106     /* ---- DMA stop ---- */
107     io_dma_stop();
108
109     while(1){
110         /* Program end */
111     }
112 }
113
114 /*"FUNC COMMENT"*****
115 * ID          :
116 * Outline     : DMAC によるメモリ転送初期設定
117 *-----
118 * Include     : "iodefine.h"
119 *-----
120 * Declaration : void io_init_dma(void *src, void *dst, size_t size, unsigned int mode);
121 *-----
122 * Description : ソースアドレス src からディスティネーションアドレス dst へ
123 *              : size バイト分 DMAC による転送を行います。
124 *              : 転送はオートリクエストにより実行します。
125 *              : mode には、転送サイズ、割り込みの使用/未使用を指定します。
126 *-----
127 * Argument    : void *src ; I : ソースアドレス
128 *              : void *dst ; O : ディスティネーションアドレス
129 *              : size_t size ; I : 転送サイズ (バイト)
130 *              : unsigned int mode ; I : 転送モード、以下のモードを論理和で指定します。
131 *              :     DMA_SIZE_BYTE (0x0000) バイト転送
132 *              :     DMA_SIZE_WORD (0x0001) ワード転送
133 *              :     DMA_SIZE_LONG (0x0002) ロング転送
134 *              :     DMA_SIZE_LONGx4(0x0003) 16 バイト転送
135 *              :     DMA_INT_DISABLE(0x0000) DMA 転送完了割り込み未使用
136 *              :     DMA_INT_ENABLE (0x0010) DMA 転送完了割り込み使用
137 *-----
138 * Return Value : void
139 *-----
140 * Note         : 転送サイズとソース/ディスティネーションアドレスの
141 *              : アライメントが一致しない場合の動作は保証しません。
142 *              : 割り込み使用時は、割り込みルーチンを登録する必要があります
143 *"FUNC COMMENT END"*****
144 void io_init_dma(void *src, void *dst, size_t size, unsigned int mode)
145 {
146     unsigned int ts;
147     unsigned long ie;
148
149     ts = mode & 0x3u;
150     ie = (mode & 0x00f0u ) >> 4u;
151

```

3.4 サンプルプログラムリスト"main.c" (4)

```
152     /* ==== Setting of DMAC ==== */
153     /* ==== Setting of power down mode ==== */
154     CPG.STBCR2.BIT.MSTP8 = 0x0u;      /* Clear the DMAC module standby mode */
155
156     /* ---- DMA Channel Control Registers(CHCR) ---- */
157     DMAC.CHCR0.BIT.DE = 0u;          /* DMA disable */
158
159     /* ---- DMA Source Address Registers(SAR) ---- */
160     DMAC.SAR0 = (unsigned long)src;
161
162     /* ---- DMA Destination Address Registers(DAR) ---- */
163     DMAC.DAR0 = (unsigned long)dst;
164
165     /* ---- DMA Transfer Count Registers(DMATCR) ---- */
166     switch(ts){
167     case DMA_SIZE_BYTE:
168         DMAC.DMATCR0 = size;          /* 転送回数(1/1)を設定 */
169         DMAC.RDMATCR0 = size;
170         break;
171     case DMA_SIZE_WORD:
172         DMAC.DMATCR0 = size >> 1u;    /* 転送回数(1/2)を設定 */
173         DMAC.RDMATCR0 = size >> 1u;
174         break;
175     case DMA_SIZE_LONG:
176         DMAC.DMATCR0 = size >> 2u;    /* 転送回数(1/4)を設定 */
177         DMAC.RDMATCR0 = size >> 2u;
178         break;
179     case DMA_SIZE_LONGx4:
180         DMAC.DMATCR0 = size >> 4u;    /* 転送回数(1/16)を設定 */
181         DMAC.RDMATCR0 = size >> 4u;
182         break;
183     default:
184         break;
185     }
186
187     /* ---- DMA Channel Control Registers(CHCR) ---- */
188     DMAC.CHCR0.LONG = 0x00005400ul | (ts << 3u) | (ie << 2u) ;
189         /* Destination address is incremented */
190         /* Source address is incremented */
191         /* Auto request */
192         /* Cycle steal mode */
193         /* Transfer Size : Longword unit */
194
195     /* ---- DMA Operation Register(DMAOR) ---- */
196     DMAC.DMAOR.WORD &= 0xfff9u;      /* AE,NMIF clear */
197
198     if(DMAC.DMAOR.BIT.DME == 0u){ /* DMA Master Enable */
199         DMAC.DMAOR.BIT.DME = 1u;
200     }
201 }
202
```

3.5 サンプルプログラムリスト"main.c" (5)

```
203  /*"FUNC COMMENT"*****
204  * ID          :
205  * Outline     : DMAC 起動
206  *-----
207  * Include     : "iodefine.h"
208  *-----
209  * Declaration : void io_dma_enable(void);
210  *-----
211  * Description : DMA 転送を実行します。
212  *-----
213  * Argument    : void
214  *-----
215  * Return Value : void
216  *-----
217  * Note        :
218  *"FUNC COMMENT END"*****/
219 void io_dma_enable(void)
220 {
221     /* ---- DMA start ---- */
222     DMAC.CHCR0.BIT.DE = 1ul; /* DMA enable */
223 }
224
225 /*"FUNC COMMENT"*****
226 * ID          :
227 * Outline     : DMAC 停止
228 *-----
229 * Include     : "iodefine.h"
230 *-----
231 * Declaration : void io_dma_stop(void);
232 *-----
233 * Description : 転送終了検出確認を行い、DMA 転送を停止します。
234 *-----
235 * Argument    : void
236 *-----
237 * Return Value : void
238 *-----
239 * Note        :
240 *"FUNC COMMENT END"*****/
241 void io_dma_stop(void)
242 {
243     /* Transmission end detection */;
244     while(DMAC.CHCR0.BIT.TE == 0ul){
245         /* wait TE bit set */
246     }
247     /* ---- DMA stop ---- */
248     DMAC.CHCR0.BIT.DE = 0ul; /* DMA disable */
249 }
250
251 /* End of File */
```

4. 参考ドキュメント

- ソフトウェアマニュアル
SH-2A/SH-2A-FPU ソフトウェアマニュアル Rev3.00
(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。)
- ハードウェアマニュアル
SH7670 グループ ハードウェアマニュアル Rev.2.00
(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。)

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/>

お問い合わせ先

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

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2008.04.23	—	初版発行
1.01	2010.10.15	—	フォーマット変更 参考プログラムの修正（AC 特性切り替え処理を削除）

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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連して発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続きを行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更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 エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/inquiry>