

SH7670グループ

イーサネット受信設定例

R01AN0303JJ0101
Rev. 1.01
2010.10.15

要旨

本アプリケーションノートでは、SH7670/SH7671/SH7672/SH7673 のイーサネット受信設定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.....	18
4. 参考ドキュメント.....	37

1. はじめに

1.1 仕様

- 本応用例ではイーサネットフレームを 10 フレーム受信します。受信のたびにフレーム受信割り込みを使用し、1 フレームずつユーザバッファにコピーします。

1.2 使用機能

- イーサネットコントローラ(EtherC)
- イーサネットコントローラ用ダイレクトメモリアクセスコントローラ(E-DMAC)
- 割り込みコントローラ(INTC)
- I²C バスインタフェース 3(IIC3)
- ピンファンクションコントローラ(PFC)

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 イーサネット PHY-LSI 自動交渉設定例
- SH7670 グループ イーサネット送信設定例

2. 応用例の説明

本応用例では、イーサネットコントローラ(EtherC)、およびイーサネットコントローラ用ダイレクトメモリアクセスコントローラ(E-DMAC)を使用します。

2.1 使用機能の動作概要

本 LSI では、イーサネット通信を行う場合必ず EtherC と E-DMAC を使用します。EtherC は送受信制御を行います。E-DMAC はその送信/受信 FIFO とユーザが指定するデータ格納先(バッファ)間の DMA 転送を専用に行います。

2.1.1 EtherC の概要

本 LSI は、イーサネットあるいは IEEE802.3 の MAC(Media Access Control)層規格に準拠したイーサネットコントローラ(EtherC)を内蔵しています。EtherC は、同規格に準拠した物理層 LSI(PHY-LSI)と接続することにより、イーサネット/IEEE802.3 フレームの送受信を行うことができます。本 LSI 内蔵の EtherC は MAC 層インタフェースを 1 系統内蔵しています。また EtherC は、本 LSI 内部で E-DMAC に接続されており、メモリとの高速アクセスが可能です。

図 1 に EtherC の構成を示します。

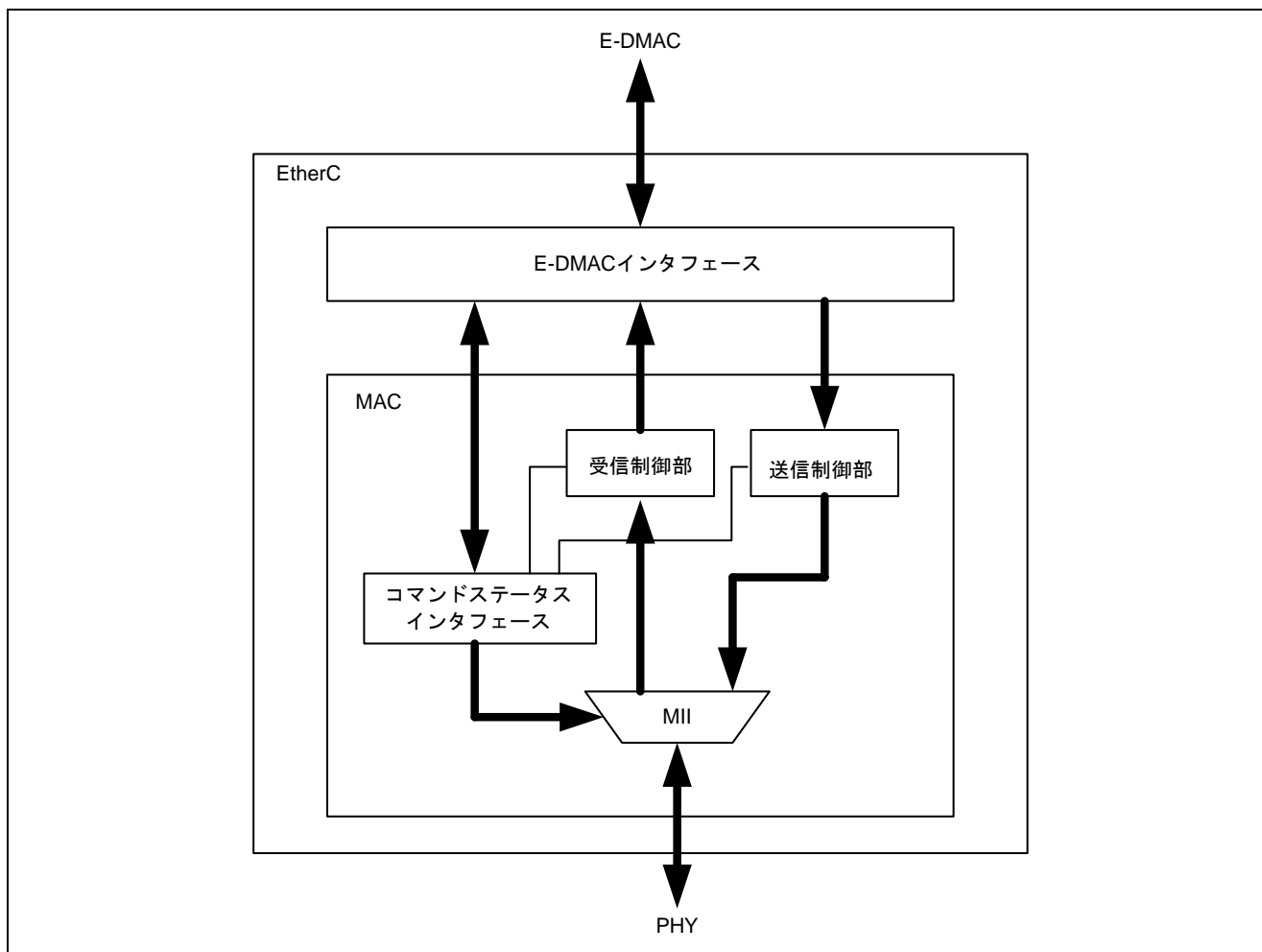


図1 EtherC の構成

2.1.2 EtherC 受信部の概要

EtherC受信部は、MII(Media Independent Interface)から入力されたフレームをプリアンブル、SFD(Start Frame Delimiter)、データおよびCRC(Cyclic Redundancy Check)データに分解します。そしてプリアンブル、SFD、CRCデータを除いた部分をE-DMAC受信部に出力します。図2にEtherC受信部の状態遷移図を示します。受信動作のフローは以下のようになります。

1. EtherC は EtherC モードレジスタ(ECMR)の受信許可(RE)ビットがセットされると、受信アイドル状態に遷移します。
2. 受信フレームのプリアンブルに続く SFD を検出すると受信処理を開始します。不当パターンの場合にはフレームを破棄します。
3. 通常モードでは、(i)宛先 MAC アドレスが本 LSI 宛の場合、(ii)ブロードキャストフレームの場合、または(iii)マルチキャストフレームの場合にデータ受信を開始します。プロミスキャスモードでは、フレームの種類にかかわらず受信を開始します。
4. MII からのフレームを受信後、フレームデータ部の CRC チェックを行います。結果はメモリ上にフレームデータをライトした後、ディスクリプタ内にステータスとして反映されます。異常時は、エラーステータスを EtherC/E-DMAC ステータスレジスタ(EESR)に設定します。
5. 1 フレームを受信後、アイドル状態に遷移し次のフレーム受信に備えます。

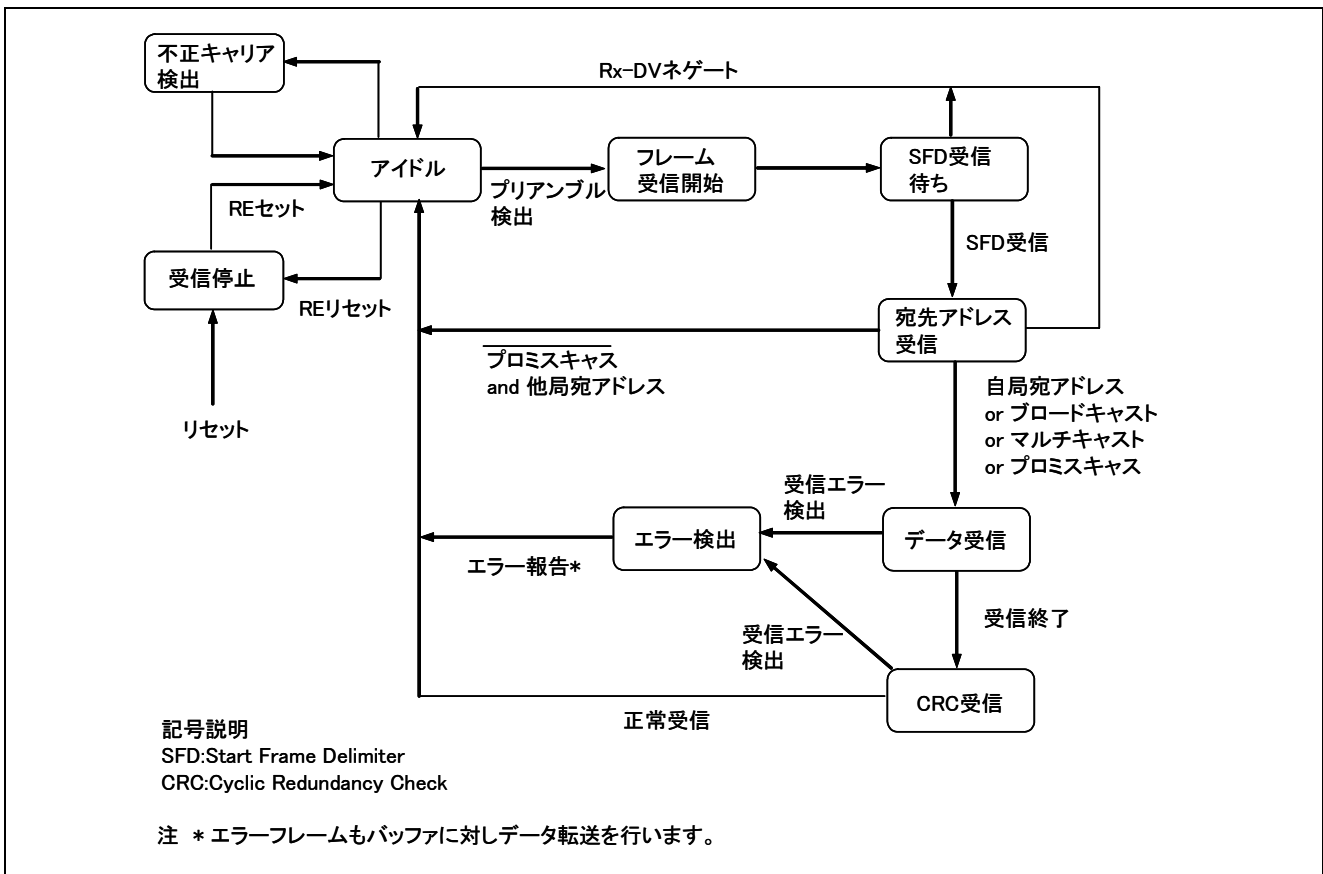


図2 EtherC 受信部状態遷移図

2.1.3 E-DMAC の概要

本LSIは、EtherCに直結したダイレクトメモリアクセスコントローラ(E-DMAC)を内蔵しています。E-DMACは、E-DMAC内蔵のDMACを使用し、E-DMAC内の送信/受信FIFOとユーザが指定するデータ格納先(送信/受信バッファ)との間で送受信データのDMA転送を行います。CPUにより直接送信/受信FIFOのデータを読み書きすることはできません。このDMA転送時に、E-DMACが参照する情報を送信/受信ディスクリプタ(次章で詳述)と呼び、ユーザがメモリ上に配置します。E-DMACは、イーサネットフレーム送受信に先立ちディスクリプタの情報を読み込み、その内容にしたがって送信データを送信バッファから読み込み、または受信データを受信バッファへ書き込みます。このディスクリプタを複数個並べディスクリプタ列(リスト)とすることで、複数のイーサネットフレームの送受信を連続的に行うことができます。

このE-DMACの機能によってCPUの負荷を軽減し、効率の良いデータ送受信制御を行うことができます。図3にE-DMACとディスクリプタおよびバッファの構成を示します。

E-DMACの特長は以下のようになります。

特長

- ・送信/受信2系統の独立したDMAC内蔵
- ・ディスクリプタ管理方式によるCPU負荷の軽減
- ・送受信フレームステータスのディスクリプタへの反映
- ・DMAブロック転送(16バイト単位)によるシステムバスの効率使用
- ・1フレーム/1ディスクリプタ、1フレーム/複数フレーム(マルチバッファ)方式対応可能(2.1.5 参照)

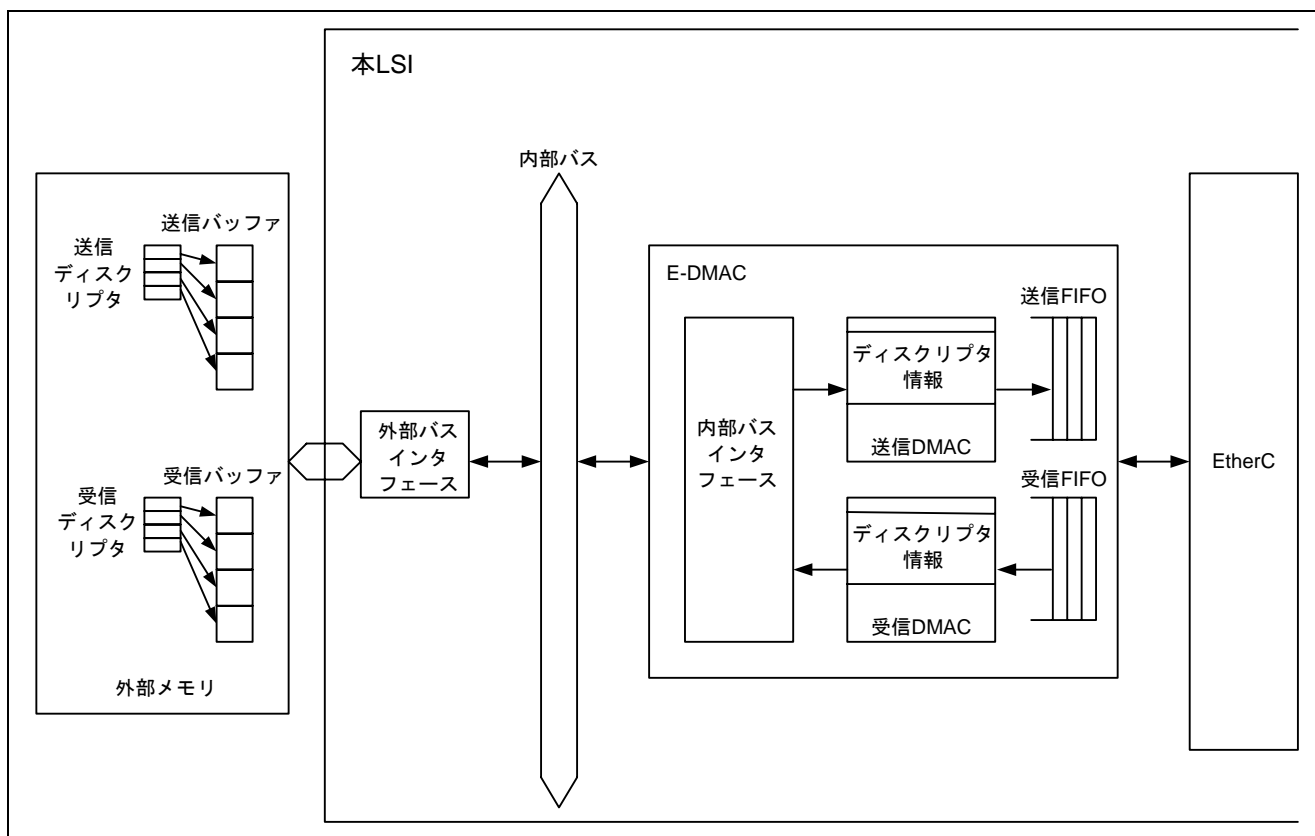


図3 E-DMACとディスクリプタおよびバッファの構成

2.1.4 ディスクリプタの概要

E-DMACがDMA転送を行うためには、ディスクリプタと呼ばれる送受信データの格納アドレス等が書かれた情報(データ)が必要になります。ディスクリプタには送信ディスクリプタと受信ディスクリプタの2種類があります。E-DMACは、E-DMAC送信要求レジスタ(EDTRR)のTRビットが1になると自動的に送信ディスクリプタの読み込みを、E-DMAC受信要求レジスタ(EDRRR)のRRビットが1になると自動的に受信ディスクリプタの読み込みを開始します。ユーザは送信/受信ディスクリプタにあらかじめ送信/受信データのDMA転送に関する情報を記述しておく必要があります。イーサネットフレームの送信/受信が完了した後は、E-DMACがディスクリプタの有効/無効ビット(送信時はTACTビット、受信時はRACTビット)を無効にし、送信/受信結果をステータスビット(送信時はTFS26~TFS0、受信時はRFS26~RFS0)に反映します。

ディスクリプタは、読み書き可能なメモリ空間に配置し、先頭ディスクリプタ(E-DMACが最初に読み込むディスクリプタ)のアドレスを送信ディスクリプタリスト先頭アドレスレジスタ(TDLAR)/受信ディスクリプタリスト先頭アドレスレジスタ(RDLAR)に設定します。複数のディスクリプタをディスクリプタ列(ディスクリプタリスト)として用意する場合には、E-DMACモードレジスタ(EDMR)のDL0,1ビットに設定したディスクリプタ長にしたがって連続したアドレスに配置します。

2.1.5 受信ディスクリプタの概要

図4に受信ディスクリプタと受信バッファの関係を示します。

受信ディスクリプタは、データの先頭から32ビット単位にRD0, RD1, RD2およびパディングで構成されます。RD0は、受信ディスクリプタの有効/無効、ディスクリプタの構成情報およびステータス情報を示します。RD1はそのディスクリプタが参照する受信バッファのサイズ(RBL)と受信したフレームのデータ長(RDL)を示します。RD2は受信バッファの先頭アドレスを示します。最後のパディングはEDMRレジスタのDL0,1ビットで指定するディスクリプタ長に従い長さが決まります。

受信ディスクリプタの設定内容により、ディスクリプタ1個で1フレームの受信データ全部を受信バッファに格納すること(1フレーム/1ディスクリプタ)も、ディスクリプタ複数個で1フレームの受信データを受信バッファに格納すること(1フレーム/マルチディスクリプタ)も可能です。1フレーム/マルチディスクリプタでは、あらかじめ複数のディスクリプタ(ディスクリプタリスト)を用意しておきます。E-DMACは、受信したフレームがディスクリプタのRBLを超える長さのフレームを受信した場合には、連続する次のディスクリプタを使用していくことによって受信バッファに転送していきます。たとえば各ディスクリプタのRBLを500バイトとしたときに1514バイトのイーサネットフレームを受信したとします。受信したイーサネットフレームは最初のディスクリプタから順に500バイトずつバッファに転送され、最後の14バイトだけが4つ目のバッファに転送されます。

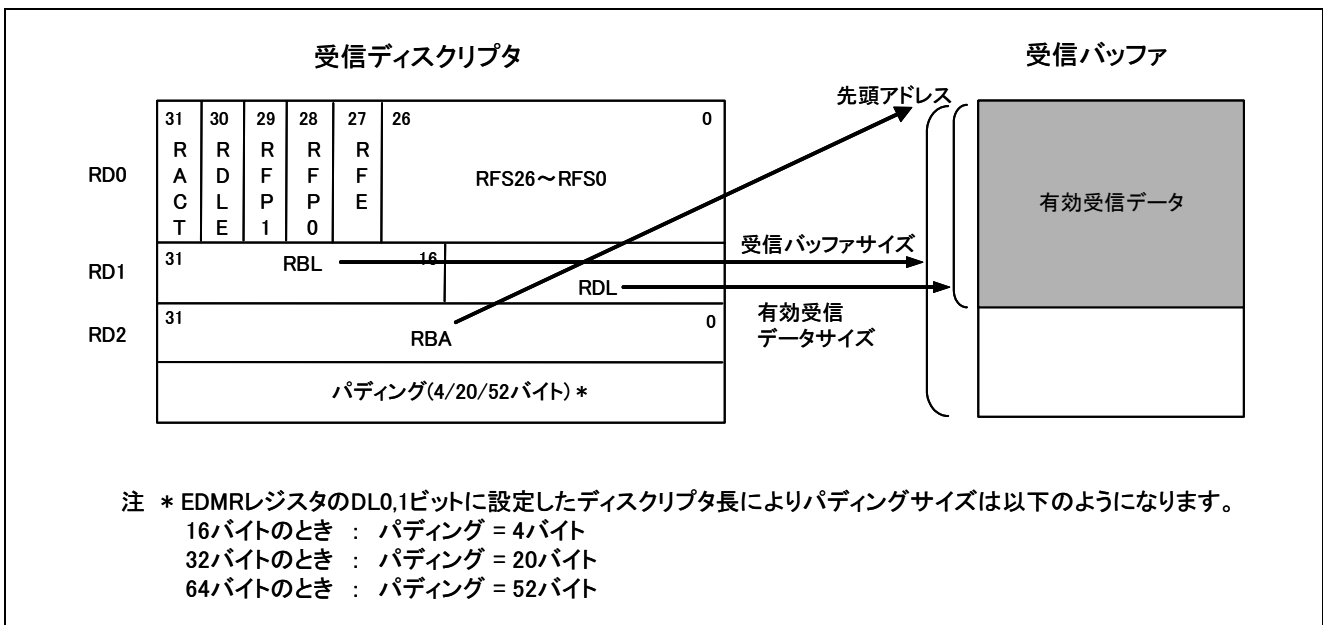


図4 受信ディスクリプタと受信バッファの関係

2.1.6 受信ディスクリプタの設定例

図5に受信ディスクリプタおよび受信バッファを各3面使用した場合の例を示します。各受信バッファのサイズを1520バイト確保し、1フレーム/1ディスクリプタになるようにします。図では各受信ディスクリプタをRD0部分のみに簡略化して記載しています。図中の番号①、②等は実行順を示します。

設定は以下のようになります。

1. 全ディスクリプタ面のRFP1,RFP0ビット、RFEビット、RFS26~RFS0ビットに0を設定します。
2. 第1面と第2面のディスクリプタのRDLEビットに0を設定します。第3面のディスクリプタのRDLEビットに1を設定することにより、第3面のディスクリプタの処理を終了すると第1面のディスクリプタを読み込みます。このような設定によりディスクリプタをリング構造にすることができます。
3. 図5では省略していますが、受信開始前に全ディスクリプタ面のRD1のRBLに受信バッファサイズ1520バイトを、RD2のRBAに対応する受信バッファの先頭アドレスを設定します。
4. 連続受信をさせるため、全ディスクリプタ面のRACTビットに1を設定します。受信手順の詳細は次章で説明します。

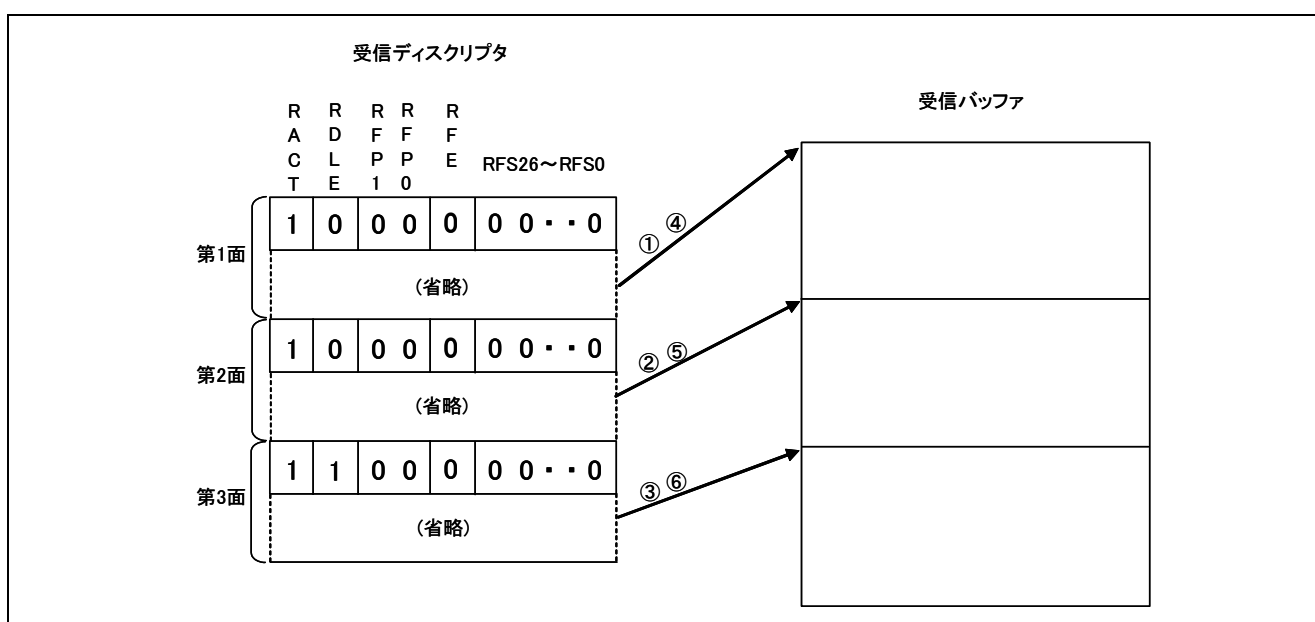


図5 受信ディスクリプタと受信バッファの関係

2.1.7 使用機能の動作手順(受信時)

ECMRのREビットが1の状態ではE-DMAC受信要求レジスタ(EDRRR)の受信要求ビット(RR)に1を書き込むと、E-DMAC受信部が起動します。E-DMACは、EtherC/E-DMACのソフトウェアリセット後は受信ディスクリプタ先頭アドレスレジスタ(RDLAR)で示すディスクリプタを読み込み、RACTビットが1(有効)のときに受信待機状態になります。EtherCは自局宛(自局が受信を許可したアドレス)のフレームを受信すると、受信データを受信FIFOに格納します。受信ディスクリプタのRACTビットが1のときは、RD2で指定される受信バッファに転送します(RACTビットが0(無効)の場合は、RRビットをクリアしてE-DMACの受信動作を停止します。)。受信したフレームのデータ長がRD1で与えられるバッファ長よりも大きい場合は、E-DMACはバッファが満了となった時点でディスクリプタにライトバック(RFP=B'10 or B'00)を行い次のディスクリプタを読み込みます。フレームの受信が完了した場合、または何らかのエラーでフレーム受信を中断した場合は、当該ディスクリプタにライトバック(RFP=B'11 or B'01)を行います。その後、連続受信方式を選択している場合(受信方式制御レジスタ(RMCR)内の受信コントロールビット(RNC)が1の場合)、E-DMACは次のディスクリプタを読み込みRACTビットが1のときに受信待機状態になります。連続受信方式を選択していない場合(RMCRレジスタ内のRNCビットが0の場合)は、EDRRRレジスタのRRビットを0にしE-DMACは受信処理を終了します。そして再度RRビットを1に設定すると、E-DMACは最後に受信を行ったディスクリプタの次のディスクリプタを読み込み受信待機状態になります。

図6に受信フローの例(1フレーム/1ディスクリプタ、連続受信方式設定時の場合)を示します。

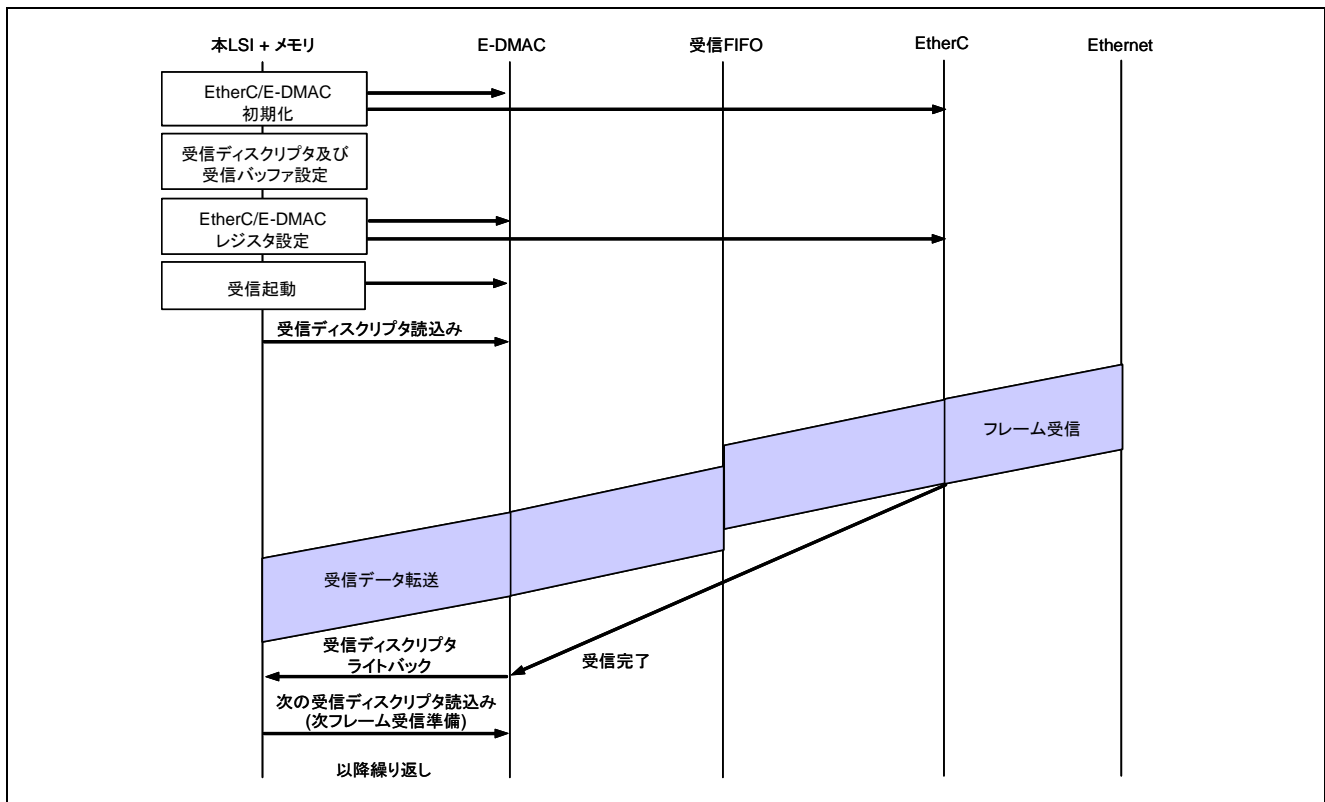


図6 受信フローの例(1フレーム/1ディスクリプタ)

2.1.8 使用機能の設定手順(受信時)

ここでは、イーサネット受信するための基本的な設定例について説明します。図7、図8にイーサネット受信設定フロー例を示します。

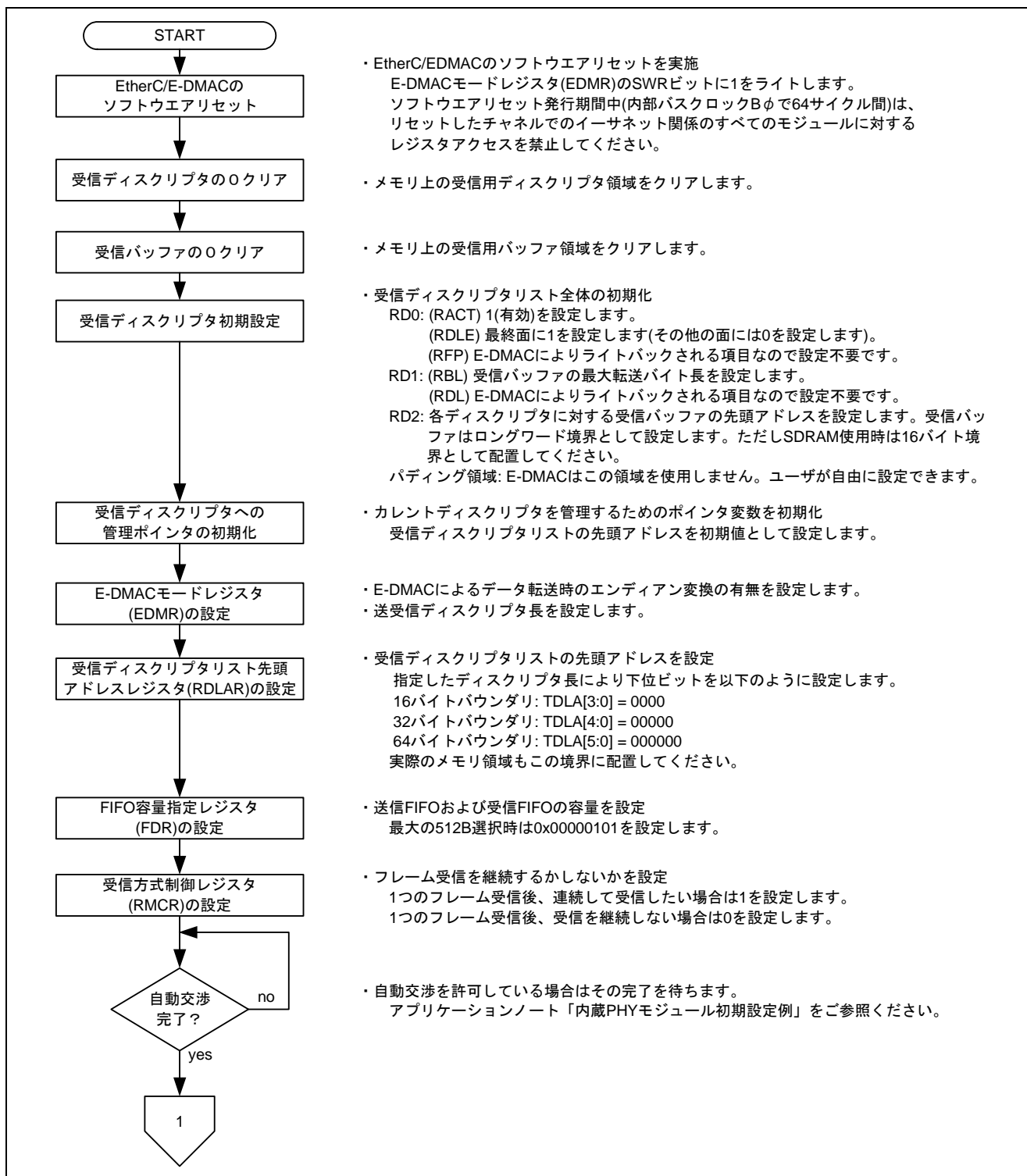


図7 イーサネット受信設定フロー例(1)

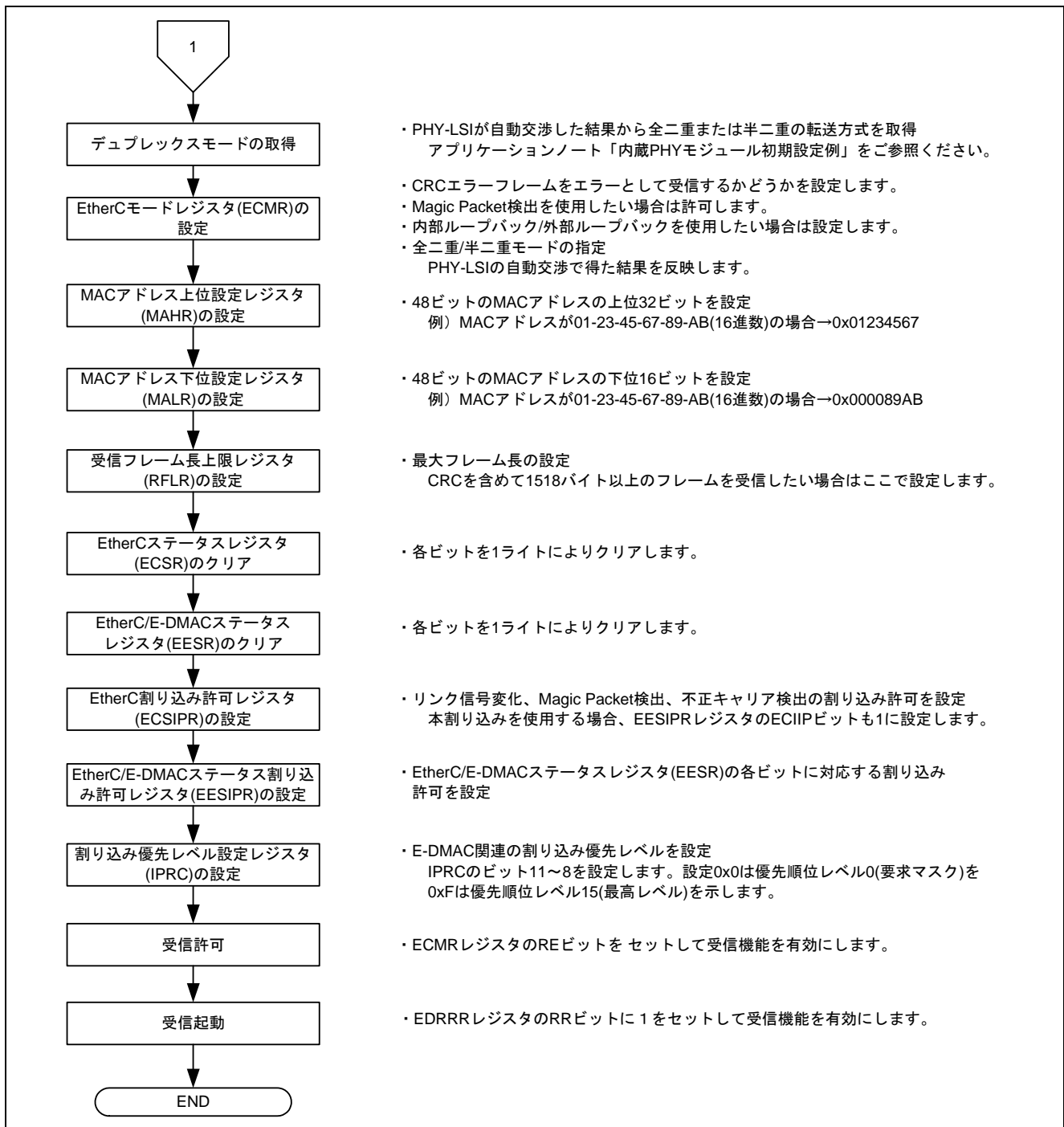


図8 イーサネット受信設定フロー例(2)

2.2 参考プログラムの動作

参考プログラムでは、EtherCおよびE-DMACを使用し、対向ホストからイーサネットフレームを10フレーム受信します。受信ディスクリプタと1520バイトの受信バッファを4面用意しています。受信方式制御レジスタ(RMCR)内の受信コントロールビット(RNC)に1を設定し、連続受信方式にしています。フレーム受信割り込み(FR)等受信に関連する割り込みが発生するたびに受信ディスクリプタのRFEビット(RDOのビット27)をチェックし、エラーがなければ(RFE=0の場合)受信バッファにある1フレーム分のデータをユーザバッファにコピーします。その後当該ディスクリプタを初期化し次の受信に備えます。エラーがあれば(RFE=1の場合)、ユーザバッファへのコピーは行わず当該ディスクリプタを初期化するだけにします。

なお、受信バッファにはイーサネットフレームのうちプリアンプル、SFD、およびCRCを除いた部分が転送されます。

図9に参考プログラムの動作環境を、図10にイーサネットフレームフォーマットを示します。

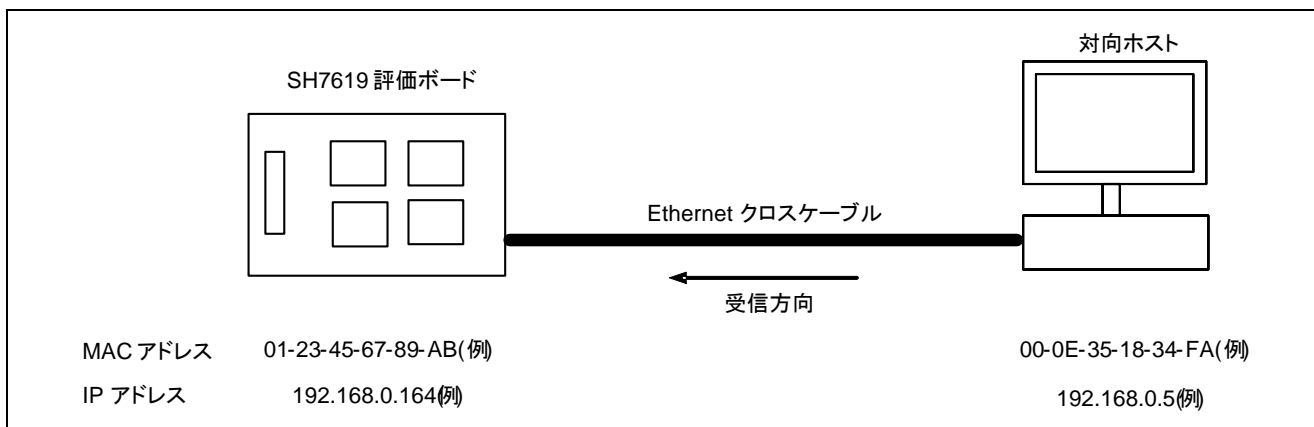


図9 参考プログラムの動作環境

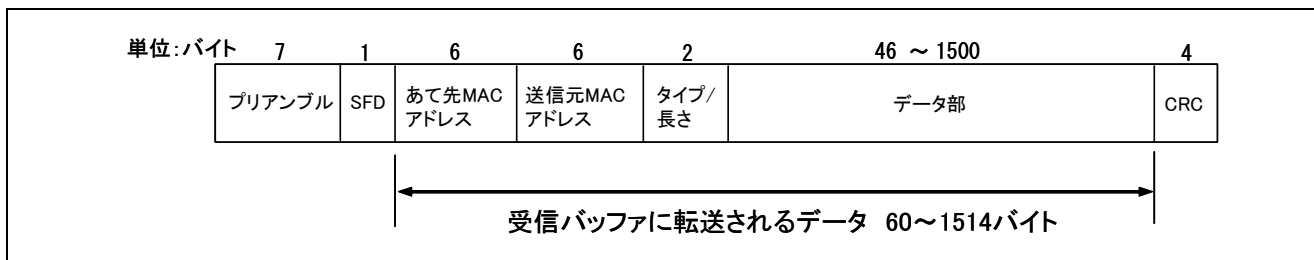


図10 イーサネットフレームフォーマット

2.3 参考プログラムのディスクリプタ定義

E-DMACではディスクリプタのパディング領域を使用しません。ユーザが自由に使用できます。本プログラムではこの領域に次のディスクリプタの先頭アドレスを設定し、ソフトウェアにてもリング構造を実現しています。図 11に参考プログラムでの受信ディスクリプタ構造体の定義と受信ディスクリプタ列の使用例を示します。

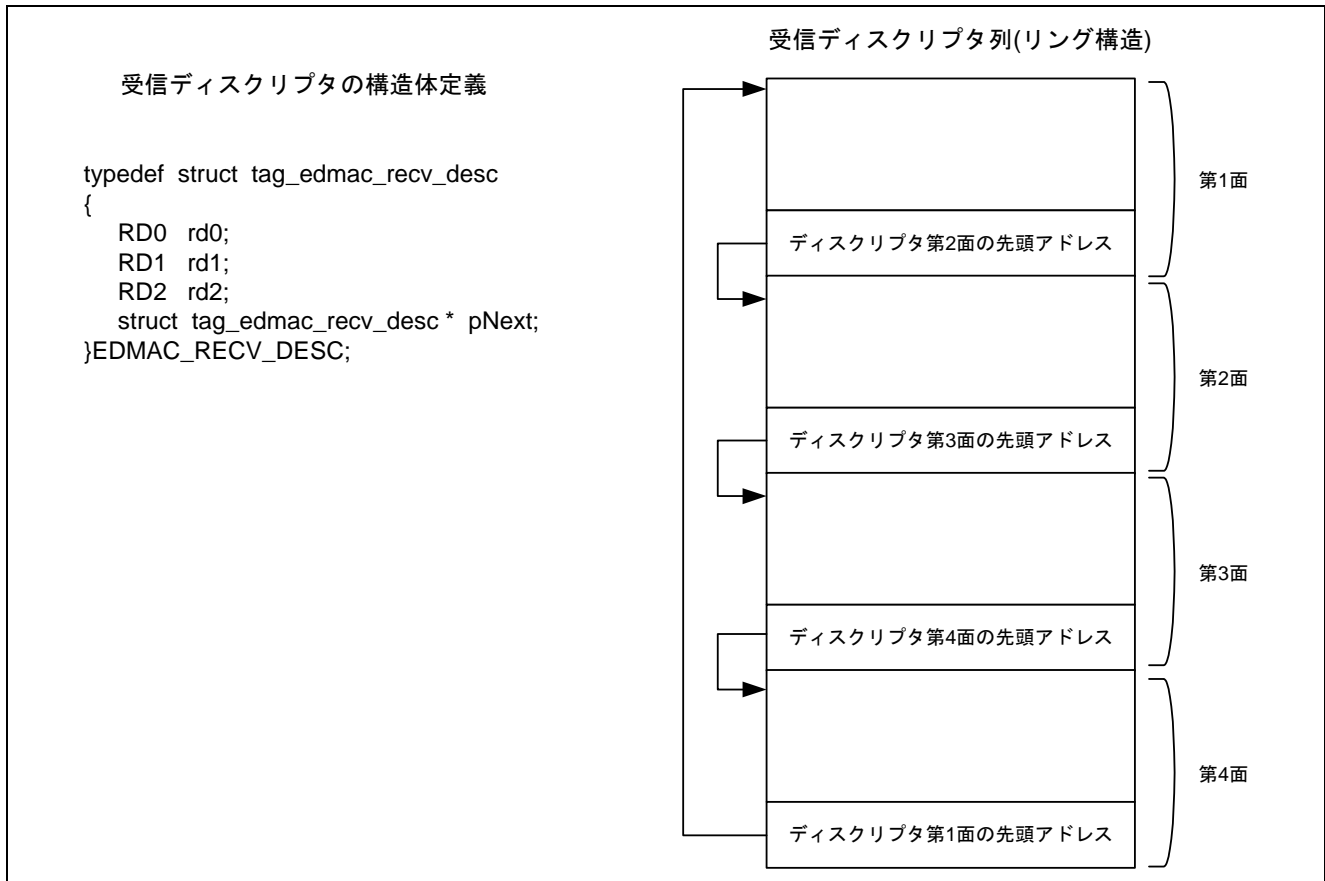


図11 受信ディスクリプタの構造体定義と受信ディスクリプタ列使用例

2.4 参考プログラムの処理手順

図12～図15に参考プログラムの処理フローを示します。PHY自動交渉関数phy_autonegoの詳細は「SH7670グループ アプリケーションノート イーサネットPHY-LSI自動交渉設定例」をご参照ください。

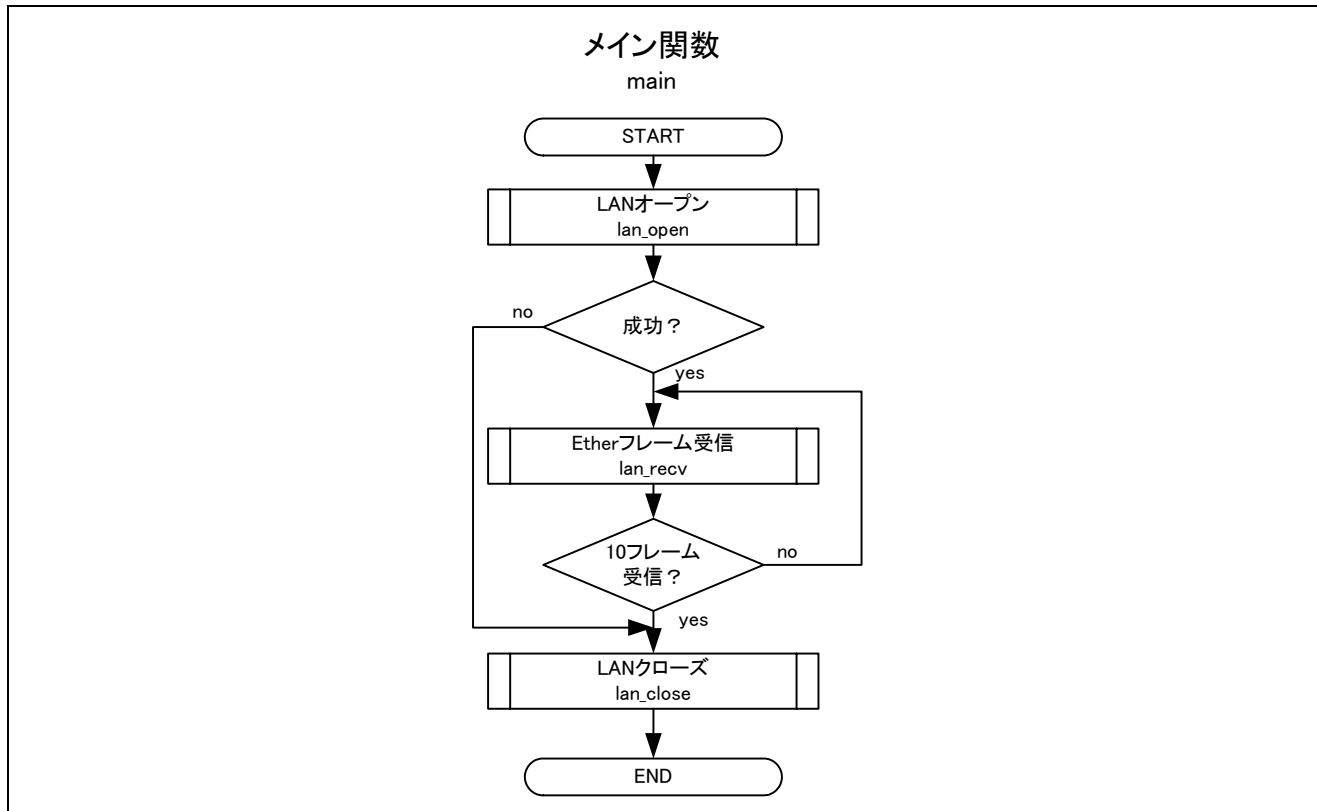


図12 メインプログラムの処理フロー例(1)

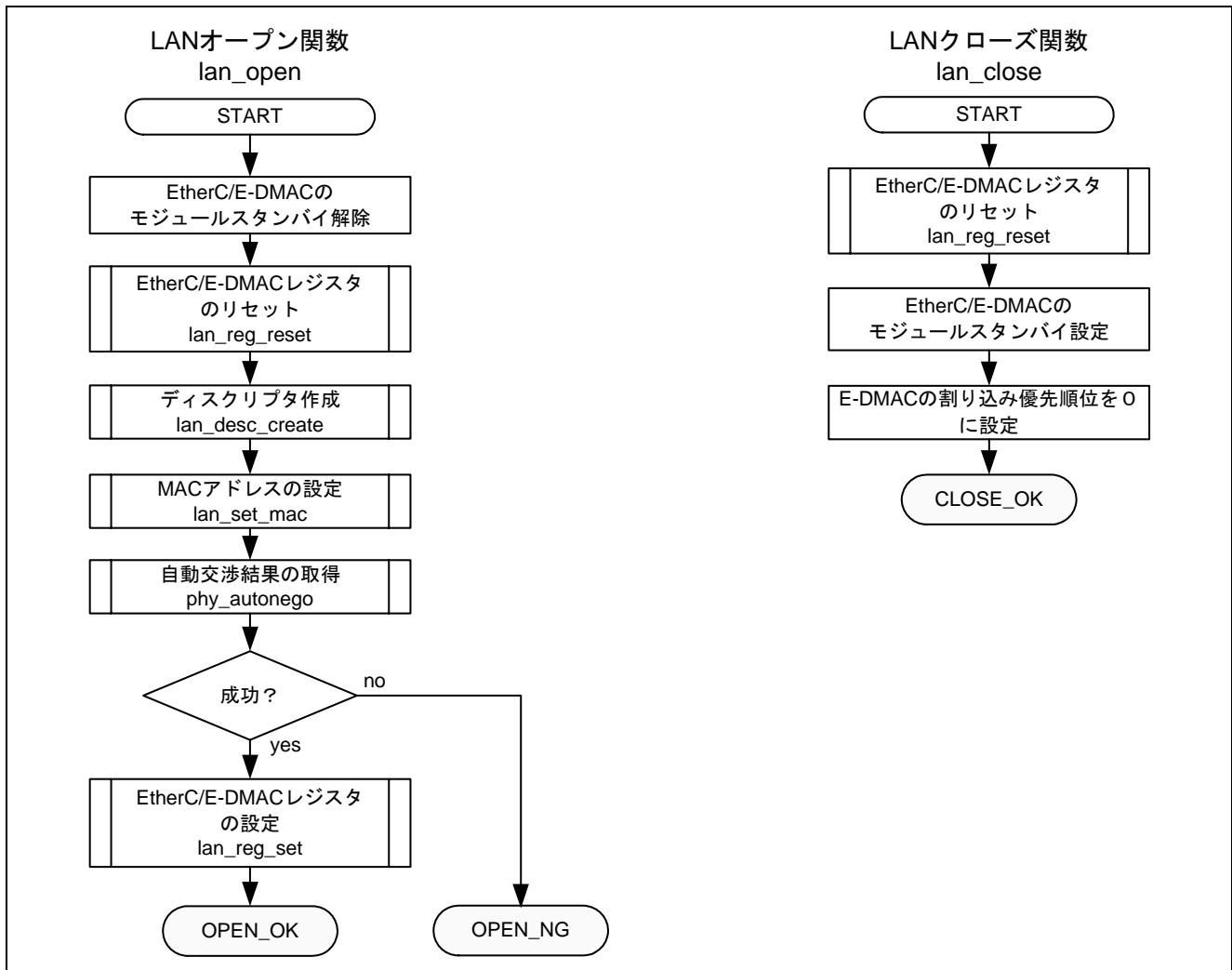


図13 参考プログラムの処理フロー例(2)

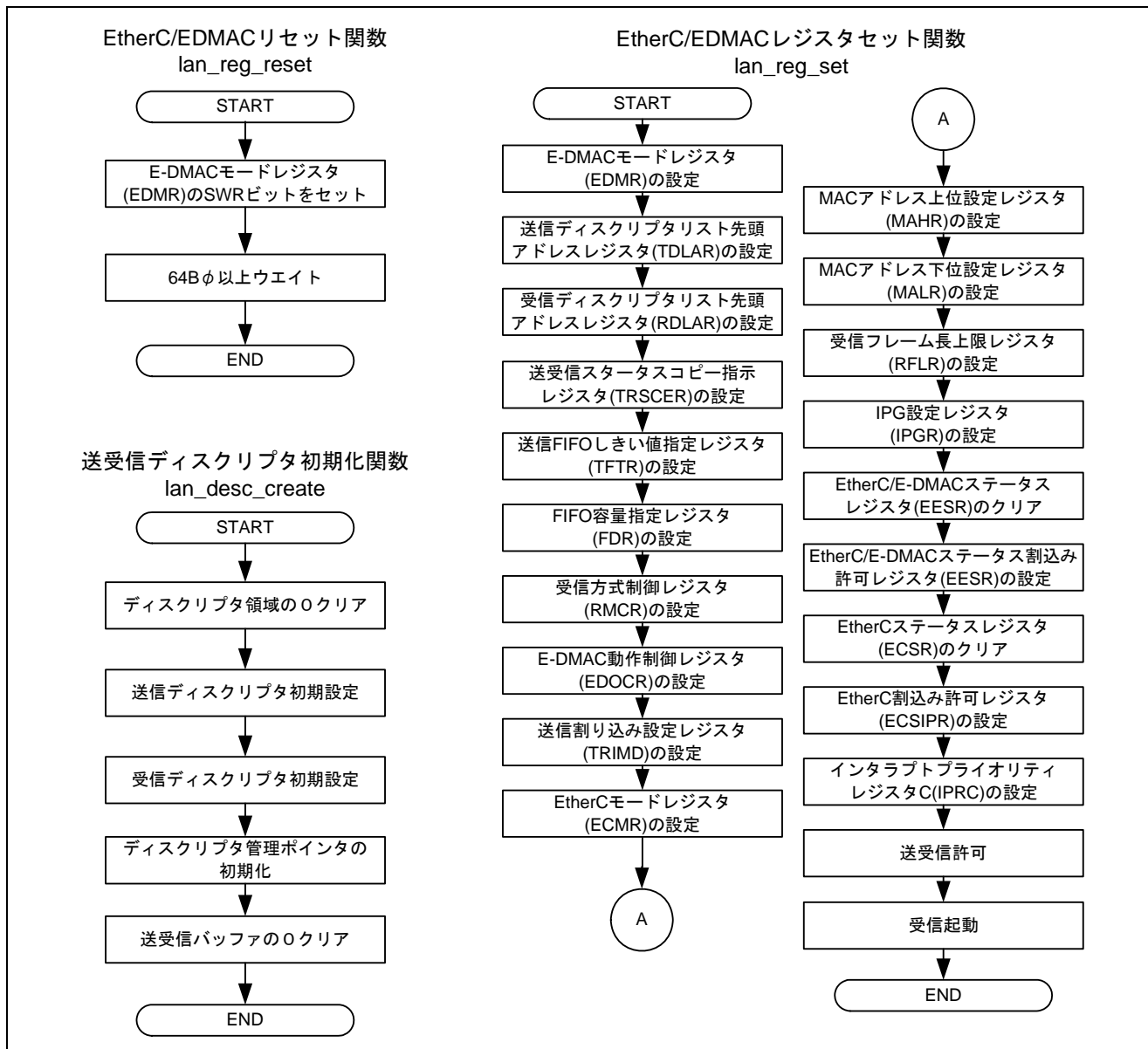


図14 参考プログラムの処理フロー例(3)

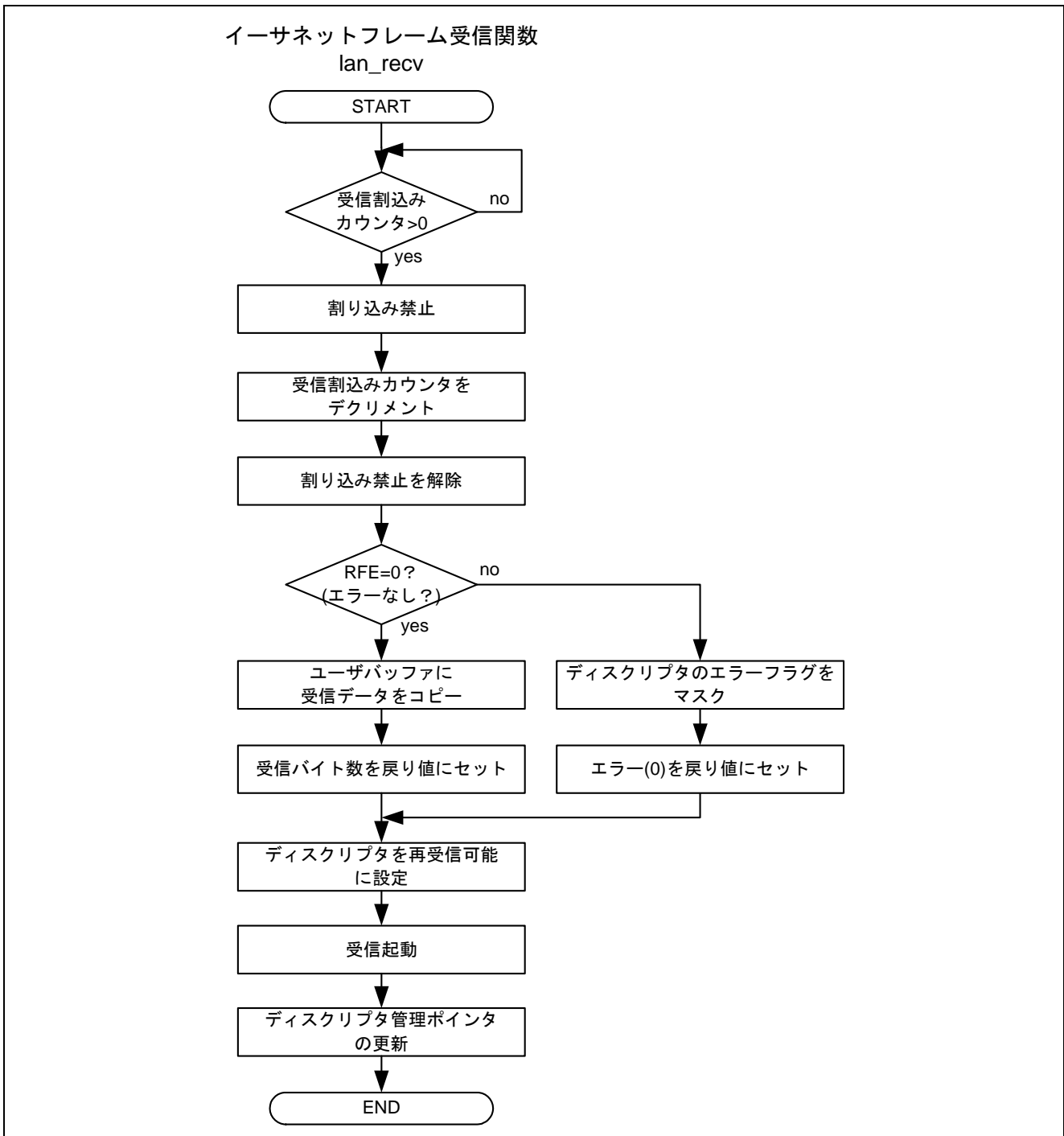


図15 参考プログラムの処理フロー例(4)

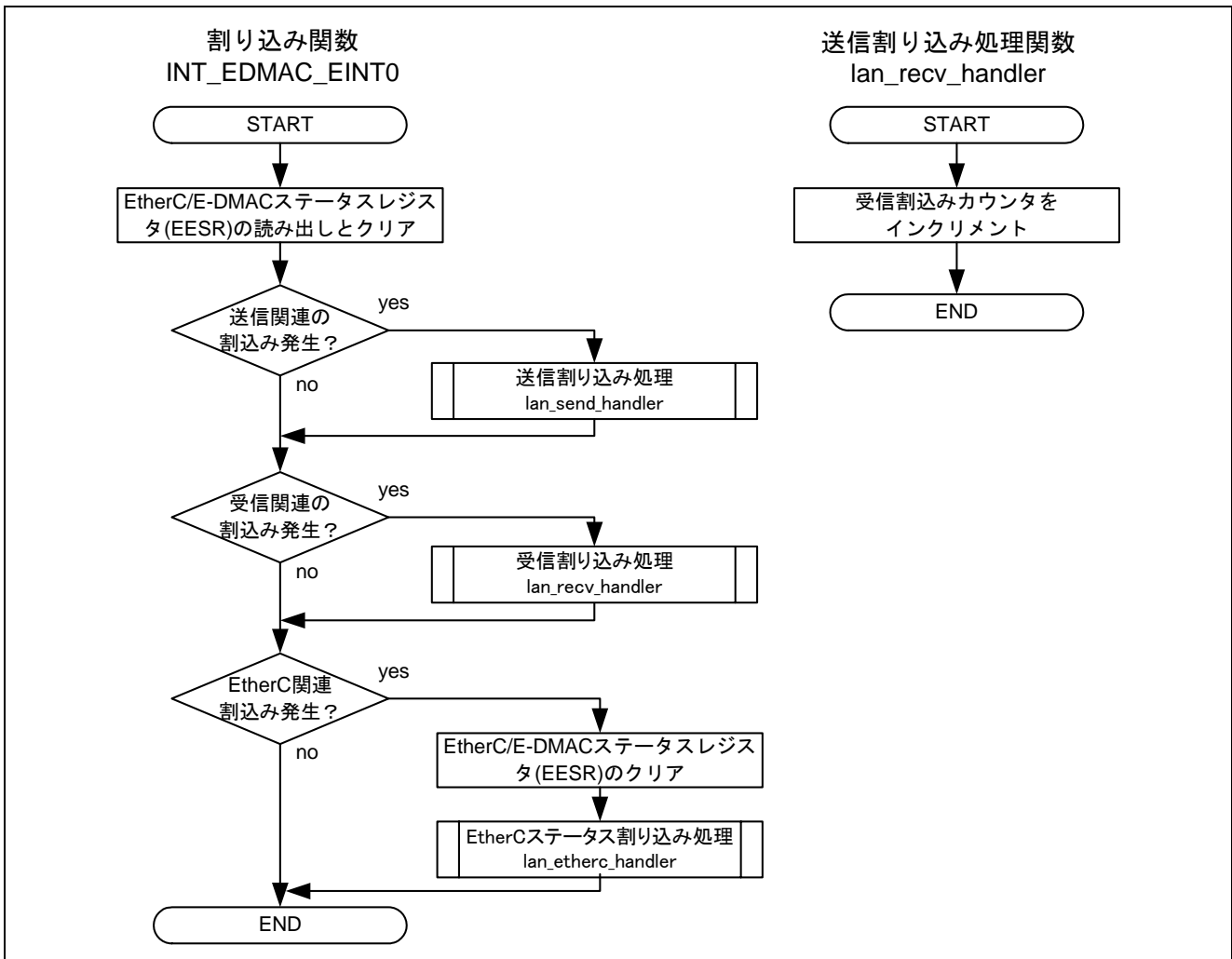


図16 参考プログラムの処理フロー例(4)

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) 2007(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    : イーサネット受信設定例
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     : Jul.04,2007 ver.1.00.00
43 *              : Oct.08,2010 ver.1.02.01 Changed the company name and device name
44 * "FILE COMMENT END"*****/
45 #include "iodefine.h"
46 #include "defs.h"
47 #include "ether.h"
48
```

3.2 サンプルプログラムリスト"main.c" (2)

```

49  /* **** プロトタイプ宣言 **** */
50  void main(void);
51
52  /* **** 変数宣言 **** */
53  #pragma section ETH_BUFF                      /* 大容量のため SDRAM に配置 */
54  typedef struct{
55      unsigned char frame[SIZE_OF_BUFFER];
56      int len;
57      unsigned char wk[12];
58  }USER_BUFFER;
59  static USER_BUFFER recv[10];
60  #pragma section
61
62  /*"FUNC COMMENT"*****
63  * ID          :
64  * Outline     : イーサネット受信サンプルプログラム メイン関数
65  *-----
66  * Include     : "iodefine.h"
67  *-----
68  * Declaration : void main(void);
69  *-----
70  * Description : 内蔵イーサネットコントローラ (EtherC) とイーサネットコント
71  *              : ローラ用ダイナミックメモリアクセスコントローラ (E-DMAC) を
72  *              : 使用して、イーサネットフレームを受信します。
73  *              : また PHY モジュールには、REALTEK 社製 RTL8201CP を使用します。
74  *              : 受信ディスクリプタを複数面用意して、連続受信します。
75  *-----
76  * Argument    : void
77  *-----
78  * Return Value : void
79  *-----
80  * Note        :
81  *"FUNC COMMENT END"*****/
82  void main(void)
83  {
84      int i,j;
85      int ret;
86
87      /* ==== イーサネット初期設定 ==== */
88      ret = lan_open();
89      if( ret == OPEN_OK ){
90          /* ==== 10 フレーム受信開始 ==== */
91          for(i=0; i<10; i++){
92              /* ---- 受信 ---- */
93              recv[i].len = lan_recv( recv[i].frame );
94              if( recv[i].len == 0 ){
95                  i--;
96              }
97          }
98      }
99      /* ==== イーサネット送受信停止 ==== */
100     lan_close();
101 }
102 /* End of file */

```

3.3 サンプルプログラムリスト” ether.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) 2007(2010) Renesas Electronics Corporation. All rights reserved.
29 * "FILE COMMENT"***** Technical reference data *****
30 * System Name : SH7671 Sample Program
31 * File Name   : ether.c
32 * Abstract    : イーサネット受信設定例
33 * Version     : 1.02.01
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     : Jul.04,2007 ver.1.00.00
43 *              : Sep.20,2007 ver.1.01.00 MACをEEPROMから取得するコードを追加
44 *              : Oct.12,2007 ver.1.01.01 lan_reg_set() レジスタ設定順序変更
45 *              :                               lan_close() 割込みマスク追加
46 *              :                               lan_desc_create() メモリクリア追加
47 *              : Jan.17,2008 ver.1.01.02 コメント修正
48 *              : Mar.07,2008 ver.1.02.00 lan_set_mac 関数変更
49 *              : Oct.08,2010 ver.1.02.01 Changed the company name and device name
50 * "FILE COMMENT END"*****/
51 #include "machine.h"
52 #include "string.h"
53 #include "iodefine.h"
```

3.4 サンプルプログラムリスト"ether.c"(2)

```

54 #include "defs.h"
55 #include "phy.h"
56 #include "ether.h"
57 #include "siic.h"
58
59 /* **** マクロ宣言 **** */
60 #define DEVADDR_EEPROM 0 /* EEPROM の PIN 設定に依存 */
61 #define ROMADDR_MAC 0 /* EEPROM 内の MAC アドレス格納場所 */
62 #define DEFAULT_MAC_H 0x00010203 /* デバッグ用 */
63 #define DEFAULT_MAC_L 0x00000405
64 #define MACSET_OK 0
65 #define MACSET_NG -1
66
67 /* **** プロトタイプ宣言 **** */
68 void main(void);
69 void lan_send_handler( unsigned long status );
70 static void lan_desc_create( void );
71 static void lan_reg_reset( void );
72 static void lan_reg_set( int link );
73 static int lan_set_mac( void );
74
75 /* **** 変数宣言 **** */
76 /* ---- ディスクリプタ ---- */
77 #pragma section ETH_DESC /* 16 バイト境界へ配置 */
78 static volatile TXRX_DESCRIPTOR_SET desc; /* ディスクリプタ領域 */
79 #pragma section
80 /* ---- バッファ ---- */
81 #pragma section ETH_BUFFER /* 16 バイト境界へ配置 */
82 static volatile TXRX_BUFFER_SET buf; /* 送受信バッファ領域 */
83 #pragma section
84 /* ---- MAC アドレス ---- */
85 static unsigned long my_macaddr_h;
86 static unsigned long my_macaddr_l;
87 /* ---- その他 ---- */
88 static volatile int c_rcv = 0; /* 受信フレームカウンタ */
89
90 /*"FUNC COMMENT"*****
91 * ID :
92 * Outline : イーサネットオープン関数
93 *-----
94 * Include : "iodefine.h", "phy.h", "ether.h"
95 *-----
96 * Declaration : int lan_open(void);
97 *-----
98 * Description : E-DMAC, EtherC, PHY, バッファメモリの初期化を行う。
99 * : 関数内でイーサネットに必要な初期化を行い、送受信可能な状態にする。
100 * : 送受信可能な状態に設定することができないときはエラーを返す。
101 *-----
102 * Argument : void
103 *-----
104 * Return Value : OPEN_OK(0) :オープン成功
105 * : OPEN_NG(-1):オープン失敗
106 *-----
107 * Note :
108 *"FUNC COMMENT END"*****

```

3.5 サンプルプログラムリスト"ether.c"(3)

```

109 int lan_open(void)
110 {
111     int link;
112
113     /* ==== PFCの設定 ==== */
114     //PORT.PBCRL1.BIT.PB6MD = 1;      /* DK30686 基板使用時の設定 */
115     PORT.PCCRHL1.WORD = 0x0155;     /* EtherC 機能 */
116     PORT.PCCRL1.WORD = 0x5555;
117     PORT.PCCRL2.WORD = 0x5555;
118     /* ==== EtherC/EDMAC のモジュールスタンバイ解除 ==== */
119     CPG.STBCR4.BIT.MSTP40 = 0;
120     /* ==== EtherC,E-DMAC の停止 === */
121     lan_reg_reset();
122     /* ==== バッファメモリの初期化 ==== */
123     lan_desc_create();
124     /* ==== MAC アドレス取得 ==== */
125     lan_set_mac();
126     /* ==== EtherC,E-DMAC の設定 ==== */
127     link = phy_autonego();           /* デュプレックスモード確認 */
128     if( link == NEGO_FAIL ){
129         return OPEN_NG;             /* OPEN 失敗 */
130     }
131     else{
132         lan_reg_set(link);
133     }
134     return OPEN_OK;
135 }
136 /*"FUNC COMMENT"*****
137 * ID      :
138 * Outline : イーサネットクローズ関数
139 *-----
140 * Include : "iodefine.h", "ether.h"
141 *-----
142 * Declaration : int lan_close(void)
143 *-----
144 * Function  : EDMAC/EtherC を停止する。
145 *           : EDMAC/EtherC はクロックの供給も停止する。
146 *-----
147 * Argument  : void
148 *-----
149 * ReturnValue : int  CLOSE_OK( 0):クローズ成功
150 *           :      CLOSE_NG(-1):クローズ失敗
151 *-----
152 * Notice    :
153 *"FUNC COMMENT END"*****

```

3.6 サンプルプログラムリスト” ether.c”(4)

```

154 int lan_close( void )
155 {
156     int i;
157
158     /* ==== EtherC,E-DMACのリセット ==== */
159     lan_reg_reset();
160     /* ==== EtherC,E-DMACの停止 ==== */
161     CPG.STBCR4.BIT.MSTP40 = 1;
162     /* ==== E-DMAC関連の割り込みを禁止 ==== */
163     INTC.IPR12.BIT._ETC = 0;
164
165     return CLOSE_OK;
166 }
167
168 /*"FUNC COMMENT"*****
169 * ID      :
170 * Outline : フレーム受信関数
171 *-----
172 * Include : "iodefine.h", "ether.h"
173 *-----
174 * Declaration : int lan_recv( unsigned char *addr )
175 *-----
176 * Function    : 受信したフレームを指定されたバッファにコピーします。
177 *              : 受信フレームが存在しなければ本関数内で待ちます。
178 *              : ディスクリプタ面数分のフレームを受信するまでに本関数で処理を
179 *              : 行ってください。
180 *-----
181 * Argument    : unsigned char addr: I : 受信フレームを格納するバッファの先頭アドレス
182 *-----
183 * ReturnValue : int : 受信したフレームのバイト数 (受信エラー:0 )
184 *-----
185 * Notice      :
186 *"FUNC COMMENT END"*****
187 int lan_recv( unsigned char *addr )
188 {
189     int i;
190     int pri;
191     int ret = 0;
192     EDMAC_RECV_DESC *p;
193
194     /* ==== 受信待ち ==== */
195     while( c_recv <= 0 ){
196         /* wait */
197     }
198     /* ==== 割り込みカウンタのデクリメント ==== */
199     pri = INTC.IPR12.BIT._ETC;                /* 排他制御 (割り込み禁止) */
200     INTC.IPR12.BIT._ETC = 0;
201     --c_recv;
202     INTC.IPR12.BIT._ETC = pri;
203

```

3.7 サンプルプログラムリスト” ether.c”(5)

```

204     /* ==== 受信フレームをコピー ==== */
205     p = desc.pRecv_end;
206     if( p->rd0.BIT.RFE == 0 ){
207         memcpy(addr, p->rd2.RBA, p->rd1.RDL);
208         ret = p->rd1.RDL;
209     }
210     /* ---- 受信エラー ---- */
211     else{
212         p->rd0.LONG &= 0x70000000;          /* エラーフラグを処理 */
213         ret = 0;                          /* 受信エラー : 0 */
214     }
215     /* ==== ディスクリプタを受信できる状態に戻す ==== */
216     p->rd0.BIT.RACT = 1;
217     /* ---- 受信起動 ---- */
218     if( EDMAC.EDRRR.BIT.RR == 0 ){        /* 1 ライト前に必ず 0 をリード*/
219         EDMAC.EDRRR.BIT.RR = 1;
220     }
221     /* ==== カレントポインタの更新 ==== */
222     desc.pRecv_end = p->pNext;
223
224     return ret;
225 }
226
227 /*"FUNC COMMENT"*****
228 * ID      :
229 * Outline : ディスクリプタ構築関数
230 *-----
231 * Include : "ether.h"
232 *-----
233 * Declaration : static void lan_desc_create( void )
234 *-----
235 * Function   : イーサネットで必要な送受信バッファとディスクリプタの初期化を
236 *             : 行います。1 フレーム/1 バッファを前提としています。
237 *-----
238 * Argument   : void
239 *-----
240 * ReturnValue : void
241 *-----
242 * Notice     :
243 *"FUNC COMMENT END"*****/
244 static void lan_desc_create( void )
245 {
246     int i;
247     /* ==== ディスクリプタ領域の構築 ==== */
248     /* ---- メモリクリア ---- */
249     memset(&desc, 0, sizeof(desc) );
250     /* ---- 送信ディスクリプタ ---- */
251     for(i=0; i<NUM_OF_TX_DESCRIPTOR; i++){
252         desc.send[i].td2.TBA = buf.send[i]; /* TD2 */
253         desc.send[i].td1.TDL = 0;          /* TD1 */
254         desc.send[i].td0.LONG= 0x30000000; /* TD0:1frame/1buf、送信無効*/
255         if( i != (NUM_OF_TX_DESCRIPTOR-1) ){ /* pNext */
256             desc.send[i].pNext = &desc.send[i+1];
257         }
258     }

```


3.8 サンプルプログラムリスト” ether.c”(6)

```

259     desc.send[i-1].td0.BIT.TDLE = 1;
260     desc.send[i-1].pNext      = &desc.send[0];
261     /* ---- 受信ディスクリプタ ---- */
262     for(i=0; i<NUM_OF_RX_DESCRIPTOR; i++){
263         desc.recv[i].rd2.RBA = buf.recv[i];          /* RD2 */
264         desc.recv[i].rd1.RBL = SIZE_OF_BUFFER;      /* RD1 */
265         desc.recv[i].rd0.LONG= 0xb0000000;          /* RD0:1frame/1buf、受信有効*/
266         if( i != (NUM_OF_RX_DESCRIPTOR-1) ){        /* pNext */
267             desc.recv[i].pNext = &desc.recv[i+1];
268         }
269     }
270     desc.recv[i-1].rd0.BIT.RDLE = 1;                  /* 最終ディスクリプタの設定 */
271     desc.recv[i-1].pNext      = &desc.recv[0];
272
273     /* ---- ディスクリプタ管理情報の初期化 ---- */
274     desc.pSend_top = &desc.send[0];
275     desc.pRecv_end = &desc.recv[0];
276
277     /* ==== バッファ領域の構築 ==== */
278     /* ---- 領域クリア ---- */
279     memset(&buf, 0, sizeof(buf) );
280 }
281 /*"FUNC COMMENT"*****
282 * ID      :
283 * Outline : EtherC,E-DMAC レジスタ初期化関数
284 *-----
285 * Include : "iodefine.h"
286 *-----
287 * Declaration : static void lan_reg_reset( void )
288 *-----
289 * Function   : EtherC と E-DMAC のレジスタをリセットします。
290 *             : 関数内でリセット期間を Bφ 64 サイクル以上確保しています。
291 *-----
292 * Argument   : void
293 *-----
294 * ReturnValue : void
295 *-----
296 * Notice     :
297 *"FUNC COMMENT END"*****/
298 static void lan_reg_reset( void )
299 {
300     volatile int j = 100;          /* Bφ 64 サイクルウェイト */
301
302     /* ---- ソフトウェアリセット ---- */
303     EDMAC.EDMR.BIT.SWR = 1;
304
305     /* ---- リセット期間の確保 ---- */
306     while(j--){
307         /* Bφ 64 サイクル ウェイト */
308     }
309 }

```

3.9 サンプルプログラムリスト” ether.c”(7)

```

310 /*"FUNC COMMENT"*****
311 * ID      :
312 * Outline : EhterC,E-DMAC レジスタ設定
313 *-----
314 * Include : "iodefine.h", "phy.h", "ether.h"
315 *-----
316 * Declaration : void lan_reg_set(int link)
317 *-----
318 * Function    : E-DMAC, EtherC の初期化を行います。
319 *              : 送受信ともに有効に設定しています。
320 *-----
321 * Argument    : int link : I :EhterC に設定するデュプレックスモード。
322 *              :              :phy_autonego 関数の戻り値を利用。
323 *-----
324 * ReturnValue : void
325 *-----
326 * Notice      : 本関数は EDMAC ソフトウェアリセット後の送受信停止状態で実行ください。
327 *"FUNC COMMENT END"*****/
328 static void lan_reg_set( int link )
329 {
330     /* ==== EDMAC ==== */
331     EDMAC.EDMR.LONG = 0x00000000; /* エンディアン変換なし(ビッグエンディアン) */
332                                 /* ディスクリプタ長 16 バイト */
333     EDMAC.TDLAR     = &desc.send[0];/* 送信ディスクリプタ先頭 */
334     EDMAC.RDLAR     = &desc.recv[0];/* 受信ディスクリプタ先頭 */
335     EDMAC.TRSCER.LONG = 0x00000000; /* 全ステータスをディスクリプタにコピー */
336     EDMAC.TFTR      = 0x00;         /* 送信 FIFO しきい値(ストア&フォワード) */
337     EDMAC.FDR.BIT.TFD = 1;          /* 送信 FIFO 容量(512 バイト) */
338     EDMAC.FDR.BIT.RFD = 1;          /* 受信 FIFO 容量(512 バイト) */
339     EDMAC.RMCR.BIT.RNC = 1;         /* 連続受信可能 */
340     EDMAC.EDOCR.LONG = 0x00000000; /* FIFO エラー時も動作継続 */
341     EDMAC.FCFTR.LONG = 0x00070000; /* フロー制御しきい値設定 ※EtherC で禁止*/
342     EDMAC.TRIMD.BIT.TIS = 0;        /* ライトバック完了通知をしない */
343     /* ==== EtherC ==== */
344     EtherC.ECMR.LONG = 0x00000000; /* フロー制御無効 */
345                                 /* CRC フレームを「エラーあり」と認識 */
346                                 /* Magic Packet 検出を許可しない */
347                                 /* 受信禁止 */
348                                 /* 送信禁止 */
349                                 /* 内部ループバックを行わない */
350                                 /* 外部ループバックを行わない */
351                                 /* デュプレックスモード (半二重) */
352                                 /* プロミスキャスモードを行わない */
353     if( link == FULL_TX || link == FULL_10M ){
354         EtherC.ECMR.BIT.DM = 1;      /* 全二重モードに設定 */
355     }
356     EtherC.MAHR     = my_macaddr_h; /* MAC アドレスの設定 */
357     EtherC.MALR     = my_macaddr_l;
358     EtherC.RFLR     = 0x000;        /* 最大受信フレーム長(1518 バイト) */
359     EtherC.IPGR     = 0x14;         /* パケット間ギャップ(96 ビット時間) */

```

3.10 サンプルプログラムリスト” ether.c”(8)

```

360  /* ==== 割り込み関連 ==== */
361  EDMAC.EESR.LONG = 0x47FF0F9F; /* 全ステータスクリア(1ライトでクリア) */
362  EDMAC.EESIPR.LONG = EDMAC_EESIPR_INI_SEND | EDMAC_EESIPR_INI_RECV |
                                EDMAC_EESIPR_INI_EtherC;
363                                /* 送受信+EtherCの割り込みを許可 */
364  EtherC.ECSR.LONG = 0x00000017; /* 全ステータスクリア(1ライトでクリア) */
365  EtherC.ECSIPR.LONG = EtherC_ECSIPR_INI; /* 割り込み許可 */
366  INTC.IPR12.BIT_ETC = 5; /* E-DMAC(EINT0)割り込み優先順位 5 */
367  /* ==== 送受信の許可設定 ==== */
368  /* ---- EtherC ---- */
369  EtherC.ECMR.BIT.RE = 1; /* 受信機能有効 */
370  EtherC.ECMR.BIT.TE = 1; /* 送信機能有効 */
371  /* ---- E-DMAC ---- */
372  if(EDMAC.EDRRR.BIT.RR == 0){
373      EDMAC.EDRRR.BIT.RR = 1; /* 受信起動 */
374  }
375 }
376 /*"FUNC COMMENT"*****
377 * ID :
378 * Outline : 送信割り込み関数
379 *-----
380 * Include : "iodefine.h", "ether.h"
381 *-----
382 * Declaration : void lan_send_handler( unsigned long status )
383 *-----
384 * Function : EDMAC(EESR)に関する送信関連の割り込みハンドラ。
385 *-----
386 * Argument : unsigned long status : I : EESRの状態(割り込み許可したビットのみ)
387 *-----
388 * ReturnValue : なし
389 *-----
390 * Notice : 本サンプルでは何も行いません。
391 *"FUNC COMMENT END"*****/
392 void lan_send_handler( unsigned long status )
393 {
394 }

```

3.11 サンプルプログラムリスト” ether.c”(9)

```
395 /*"FUNC COMMENT"*****
396 * ID      :
397 * Outline : 受信割込み関数
398 *-----
399 * Include : "iodefine.h", "ether.h"
400 *-----
401 * Declaration : void lan_recv_handler( unsigned long status )
402 *-----
403 * Function   : EDMAC(EESR)に関する受信関連の割込みハンドラ。
404 *-----
405 * Argument   : unsigned long status : I : EESR の状態 (割込み許可したビットのみ)
406 *-----
407 * ReturnValue : なし
408 *-----
409 * Notice     :
410 *"FUNC COMMENT END"*****/
411 void lan_recv_handler( unsigned long status )
412 {
413     c_recv++;          /* 受信割込み回数をカウント */
414 }
415 /*"FUNC COMMENT"*****
416 * ID      :
417 * Outline : EtherC 割込み関数
418 *-----
419 * Include : "iodefine.h", "ether.h"
420 *-----
421 * Declaration : void lan_etherc_handler( unsigned long status )
422 *-----
423 * Function   : EtherC(ECSR)に関する割込みハンドラ。
424 *-----
425 * Argument   : unsigned long status : I : ECSR の状態 (割込み許可したビットのみ)
426 *-----
427 * ReturnValue : なし
428 *-----
429 * Notice     : 本サンプルでは何も行いません。
430 *"FUNC COMMENT END"*****/
431 void lan_etherc_handler( unsigned long status )
432 {
433 }
434
```

3.12 サンプルプログラムリスト” ether.c”(10)

```
435  /*"FUNC COMMENT"*****
436  * ID      :
437  * Outline : MAC アドレス設定
438  *-----
439  * Include : "siic.h"
440  *-----
441  * Declaration : static int lan_set_mac( void )
442  *-----
443  * Function   : EEPROM から MAC アドレスを取得する関数
444  *-----
445  * Argument   : なし
446  *-----
447  * ReturnValue : MACSET_OK(0) :取得成功
448  *             : MACSET_NG(-1):取得失敗
449  *-----
450  * Notice     :
451  *"FUNC COMMENT END"*****/
452  static int lan_set_mac( void )
453  {
454      volatile int ret, i;
455      unsigned char buf[10];
456
457      /* ==== EEPROM driver 初期設定 ==== */
458      siic_Init_Driver();
459
460      /* ==== EEPROM をリード ==== */
461      ret = siic_EepRomRW(DEVADDR_EEPROM, ROMADDR_MAC, 6, buf, SIIC_MODE_EEP_READ);
462      if (ret < SIIC_OK) {
463          /* ---- リード失敗 ---- */
464          my_macaddr_h = DEFAULT_MAC_H;
465          my_macaddr_l = DEFAULT_MAC_L;
466          return MACSET_NG;
467      }
468      do{
469          ret = siic_Chk_Eep();
470          if( ret < SIIC_OK ){
471              /* ---- リード失敗 ---- */
472              my_macaddr_h = DEFAULT_MAC_H;
473              my_macaddr_l = DEFAULT_MAC_L;
474              return MACSET_NG;
475          }
476      }while( ret != SIIC_OK);
477      /* ---- リード成功 ---- */
478      for(i=0; i<6; i++){
479          if( buf[i] != 0xff ){
480              break;
481          }
482      }
```

3.13 サンプルプログラムリスト” ether.c”(11)

```
483     if( i == 6 ){
484         /* ---- Eeprom が未設定の場合はデフォルト値を設定 ---- */
485         my_macaddr_h = DEFAULT_MAC_H;
486         my_macaddr_l = DEFAULT_MAC_L;
487     }
488     else{
489         /* ---- リードしたアドレスを設定 ---- */
490         my_macaddr_h = buf[0];
491         my_macaddr_h <<= 8;
492         my_macaddr_h |= buf[1];
493         my_macaddr_h <<= 8;
494         my_macaddr_h |= buf[2];
495         my_macaddr_h <<= 8;
496         my_macaddr_h |= buf[3];
497         my_macaddr_l = buf[4];
498         my_macaddr_l <<= 8;
499         my_macaddr_l |= buf[5];
500     }
501     return MACSET_OK;
502 }
503
504 /* End of file */
```

3.14 サンプルプログラムリスト” ether.h”(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) 2007(2010) Renesas Electronics Corporation. All rights reserved.
29 * "FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7671 Sample Program
31 * File Name   : ether.h
32 * Abstract    : イーサネット受信設定例
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     : Jul.04,2007 ver.1.00.00
43 *              : Oct.08,2010 ver.1.02.01 Changed the company name and device name
44 * "FILE COMMENT END"*****/
```

3.15 サンプルプログラムリスト” ether.h”(2)

```

45  #ifndef _ETHER_H
46  #define _ETHER_H
47
48  /* **** マクロ定義 **** */
49  #define NUM_OF_TX_DESCRIPTOR 4
50  #define NUM_OF_RX_DESCRIPTOR 4
51  #define NUM_OF_TX_BUFFER 4
52  #define NUM_OF_RX_BUFFER 4
53  #define SIZE_OF_BUFFER 1520 /* 16 バイトの整数倍にすること */
54
55  #define OPEN_OK 0
56  #define OPEN_NG -1
57  #define SEND_OK 0
58  #define SEND_NG -1
59  #define CLOSE_OK 0
60  #define CLOSE_NG -1
61  #define MIN_FRAME_SIZE 60
62  #define MAX_FRAME_SIZE 1514
63
64  #define EDMAC_EESIPR_INI_SEND 0x04280F00 /* 0x04000000 : 送信中断検出 *
65                                          * 0x00200000 : フレーム送信完了 *
66                                          * 0x00080000 : 送信 FIFO アンダフロー*
67                                          * 0x00000800 : キャリア未検出 *
68                                          * 0x00000400 : キャリア消失検出 *
69                                          * 0x00000200 : 遅延衝突検出 *
70                                          * 0x00000100 : 送信リトライオーバ */
71  #define EDMAC_EESIPR_INI_RECV 0x0205001F /* 0x02000000 : 受信中断検出 *
72                                          * 0x00040000 : フレーム受信 *
73                                          * 0x00010000 : 受信 FIFO オーバフロー *
74                                          * 0x00000010 : 端数ビットフレーム受信 *
75                                          * 0x00000008 : ロングフレーム受信 *
76                                          * 0x00000004 : ショートフレーム受信 *
77                                          * 0x00000002 : PHY-LSI 受信エラー *
78                                          * 0x00000001 : 受信フレーム CRC エラー */
79  #define EDMAC_EESIPR_INI_EtherC 0x00400000 /* 0x00400000 : EtherC ステータスレジスタ*/
80  #define EtherC_ECSIPR_INI 0x00000004 /* 0x00000004 : リンク信号変化 */
81
82  /* **** 型定義 **** */
83
84  /* ==== 送信ディスクリプタ ==== */
85  typedef union{
86      unsigned long LONG;
87      struct{
88          unsigned int TACT:1; /* 送信ディスクリプタ有効 */
89          unsigned int TDLE:1; /* 送信ディスクリプタ最終 */
90          unsigned int TFP :2; /* 送信フレーム内位置 1,0 */
91          unsigned int TFE :1; /* 送信フレームエラー */
92          unsigned int reserved :23; /* TFS26~4 : 予約 */
93          unsigned int TFS3:1; /* ノーキャリア検出 (EESR-CND ビット) */
94          unsigned int TFS2:1; /* キャリア消失検出 (EESR-DLC ビット) */
95          unsigned int TFS1:1; /* 送信中の遅延衝突検出 (EESR-CD ビット) */
96          unsigned int TFS0:1; /* 送信リトライオーバ (EESR-TRO ビット) */
97      }BIT;
98  }TD0;

```


3.16 サンプルプログラムリスト” ether.h”(3)

```

99  typedef struct{
100     unsigned short TDL;                /* 送信バッファデータ長          */
101     unsigned short reserved;
102 }TD1;
103  typedef struct{
104     unsigned char *TBA;                /* 送信バッファのアドレス        */
105 }TD2;
106  typedef struct tag_edmac_send_desc{
107     TD0 td0;
108     TD1 td1;
109     TD2 td2;
110     struct tag_edmac_send_desc *pNext;
111 }EDMAC_SEND_DESC;
112
113  /* ==== 受信ディスクリプタ ==== */
114  typedef union{
115     unsigned long LONG;
116     struct{
117         unsigned int RACT:1;          /* 受信ディスクリプタ有効          */
118         unsigned int RDLE:1;          /* 受信ディスクリプタ最終          */
119         unsigned int RFP :2;          /* 受信フレーム内位置 1,0          */
120         unsigned int RFE :1;          /* 受信フレームエラー              */
121         unsigned int reserved1:17;    /* TFS26~10 : 予約                  */
122         unsigned int RFS9:1;          /* 受信 FIFO オーバフロー(EESR-RFOF ビット) */
123         unsigned int reserved2:1;     /*          : 予約                    */
124         unsigned int RFS7:1;          /* マルチキャストフレームを受信(EESR-RMAF ビット) */
125         unsigned int reserved3:1;     /*          : 予約                    */
126         unsigned int reserved4:1;     /*          : 予約                    */
127         unsigned int RFS4:1;          /* 端数ビットフレーム受信エラー(EESR-RRF ビット) */
128         unsigned int RFS3:1;          /* ロングフレーム受信エラー(EESR-RTLE ビット) */
129         unsigned int RFS2:1;          /* ショートフレーム受信エラー(EESR-RTSF ビット) */
130         unsigned int RFS1:1;          /* PHY-LSI 受信エラー (EESR-PRE ビット) */
131         unsigned int RFS0:1;          /* 受信フレーム CRC エラー検出(EESR-CERF ビット) */
132     }BIT;
133 }RD0;
134  typedef struct{
135     unsigned short RBL;                /* 受信バッファ長                  */
136     unsigned short RDL;                /* 受信データ長                    */
137 }RD1;
138  typedef struct{
139     unsigned char *RBA;                /* 受信バッファのアドレス          */
140 }RD2;
141  typedef struct tag_edmac_recv_desc{
142     RD0 rd0;
143     RD1 rd1;
144     RD2 rd2;
145     struct tag_edmac_recv_desc *pNext;
146 }EDMAC_RECV_DESC;

```

3.17 サンプルプログラムリスト” ether.h”(4)

```
147
148 /* ==== 送受信ディスクリプタ全体(16バイト境界に配置すること) ==== */
149 typedef struct{
150     EDMAC_SEND_DESC send[NUM_OF_TX_DESCRIPTOR];
151     EDMAC_RECV_DESC recv[NUM_OF_RX_DESCRIPTOR];
152     EDMAC_SEND_DESC *pSend_top;          /* 送信ディスクリプタの登録位置 */
153     EDMAC_RECV_DESC *pRecv_end;        /* 受信ディスクリプタの受信完了/登録位置 */
154 }TXRX_DESCRIPTOR_SET;
155
156 /* ==== 送受信バッファ(16バイト境界に配置すること) ==== */
157 /* ---- 全送受信バッファ領域の定義 ---- */
158 typedef struct{
159     unsigned char send[NUM_OF_TX_BUFFER][SIZE_OF_BUFFER];
160     unsigned char recv[NUM_OF_RX_BUFFER][SIZE_OF_BUFFER];
161 }TXRX_BUFFER_SET;
162
163 /* **** プロトタイプ宣言 **** */
164 int lan_open( void );
165 int lan_close( void );
166 int lan_send( unsigned char *addr, int flen );
167
168
169 #endif
170
171 /* End of File */
```

3.18 サンプルプログラムリスト”intprg_eth.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) 2007(2010) Renesas Electronics Corporation. All rights reserved.
29 * "FILE COMMENT"***** Technical reference data *****
30 * System Name : SH7671 Sample Program
31 * File Name   : intprg_eth.c
32 * Abstract    : interrupt entry function
33 * Version     : 1.01.01
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     : Jul.04,2007 ver.1.00.00
43 *              : Sep.18,2007 ver.1.01.00 セクション変更 P->PIntPRG
44 *              : Oct.08,2010 ver.1.01.01 Changed the company name and device name
45 * "FILE COMMENT END"*****/
```

(途中省略)

3.19 サンプルプログラムリスト”intprg_eth.h”(2)

```
654 // 171 ETC EINT0
655 void INT_ETC_EINT0(void)
656 {
657     unsigned long stat_edmac;
658     unsigned long stat_EtherC;
659
660     /* ---- 割込み要求フラグのクリア ---- */
661     stat_edmac = EDMAC.EESR.LONG & EDMAC.EESIPR.LONG; /* 許可した割込みのみ対象 */
662     EDMAC.EESR.LONG = stat_edmac;
663     /* ==== 送信関連 ==== */
664     if( stat_edmac & EDMAC_EESIPR_INI_SEND ){
665         lan_send_handler( stat_edmac & EDMAC_EESIPR_INI_SEND );
666     }
667     /* ==== 受信関連 ==== */
668     if( stat_edmac & EDMAC_EESIPR_INI_RECV ){
669         lan_recv_handler( stat_edmac & EDMAC_EESIPR_INI_RECV );
670     }
671     /* ==== EtherC 関連 ==== */
672     if( stat_edmac & EDMAC_EESIPR_INI_EtherC ){
673         /* ---- 割込み要求フラグのクリア ---- */
674         stat_EtherC = EtherC.ECSR.LONG & EtherC.ECSIPR.LONG; /* 許可した割込みのみ対象 */
675         EtherC.ECSR.LONG = stat_EtherC;
676         lan_etherc_handler( stat_EtherC );
677     }
678 }
679
680
681
```

(以下省略)

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/>

お問合せ先

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

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2008.02.28	—	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>