

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

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、誤りが無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、海底中継機器、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。

SH7619 グループ

イーサネット受信設定例

要旨

この資料は、SH7619 のイーサネット受信設定例を示します。

動作確認デバイス

SH7619

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.....	34

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)
- イーサネットフィジカルレイヤトランシーバ(PHY)
- PHY インタフェース(PHYIF)
- 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: SH7619 (R4S76190)
- 動作周波数: 内部クロック 125.00MHz
バスクロック 62.5MHz
周辺クロック 31.25MHz
- 統合開発環境: ルネサステクノロジ製 High-performance Embedded Workshop Ver.4.03.00
- C コンパイラ: ルネサステクノロジ製
SuperH RISC engine ファミリ C/C++コンパイラパッケージ V9.01 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=sh2 -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

本資料の参考プログラムは、SH7619 初期設定例アプリケーションノートの設定条件で動作確認をしています。そちらも合わせてご参照ください。

また、以下のアプリケーションノートもご参照ください。

「SH7619 グループ アプリケーションノート 内蔵 PHY モジュールの初期設定例」

「SH7619 グループ アプリケーションノート 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 エラー! 参照元が見つかりません。に EtherC の構成を示します。

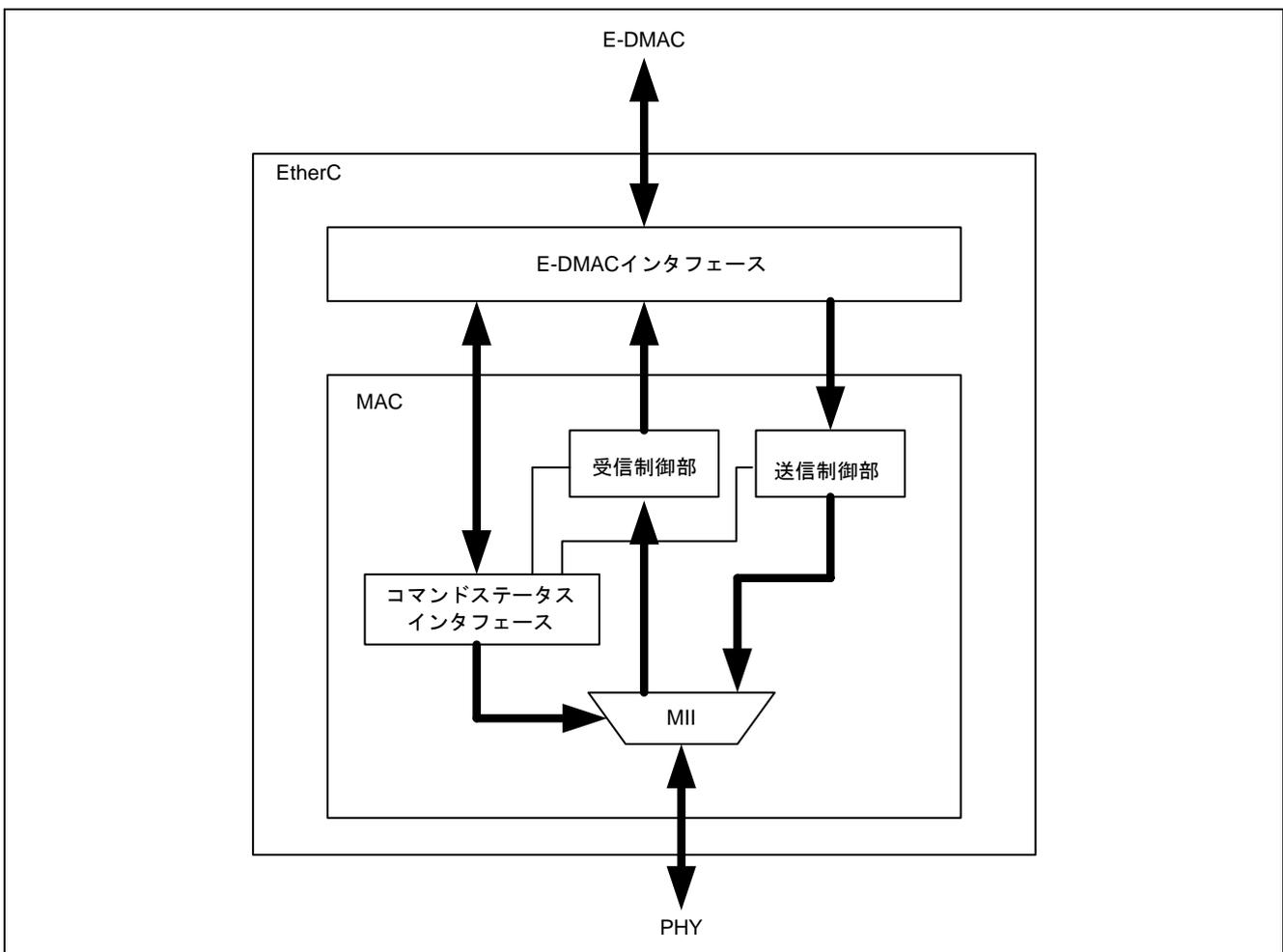


図1 EtherC の構成

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プル、SFD(Start Frame Delimiter)、データおよびCRC(Cyclic Redundancy Check)データに分解します。そしてプリアンプル、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プルに続く 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。

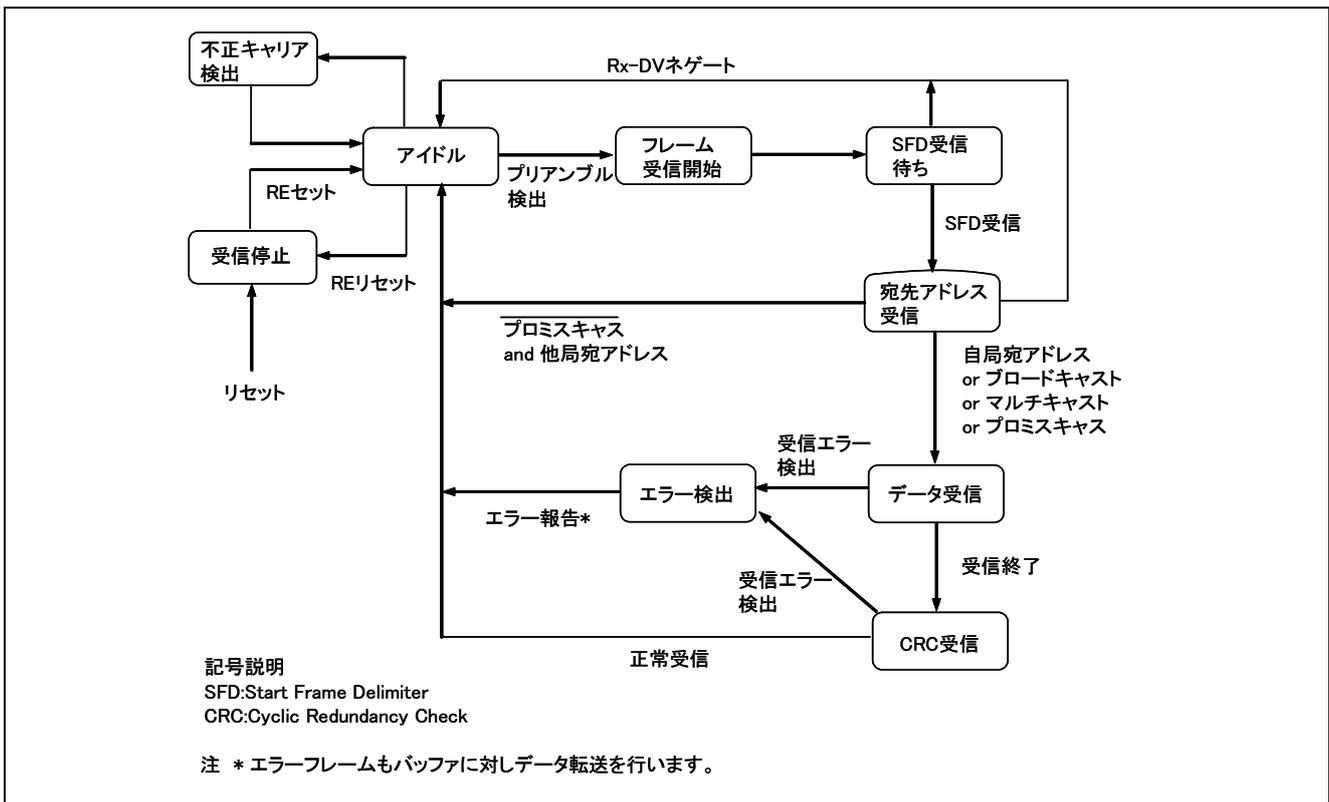


図2 EtherC 受信部状態遷移図

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)化することで、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。
エラー! 参照元が見つかりません。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)

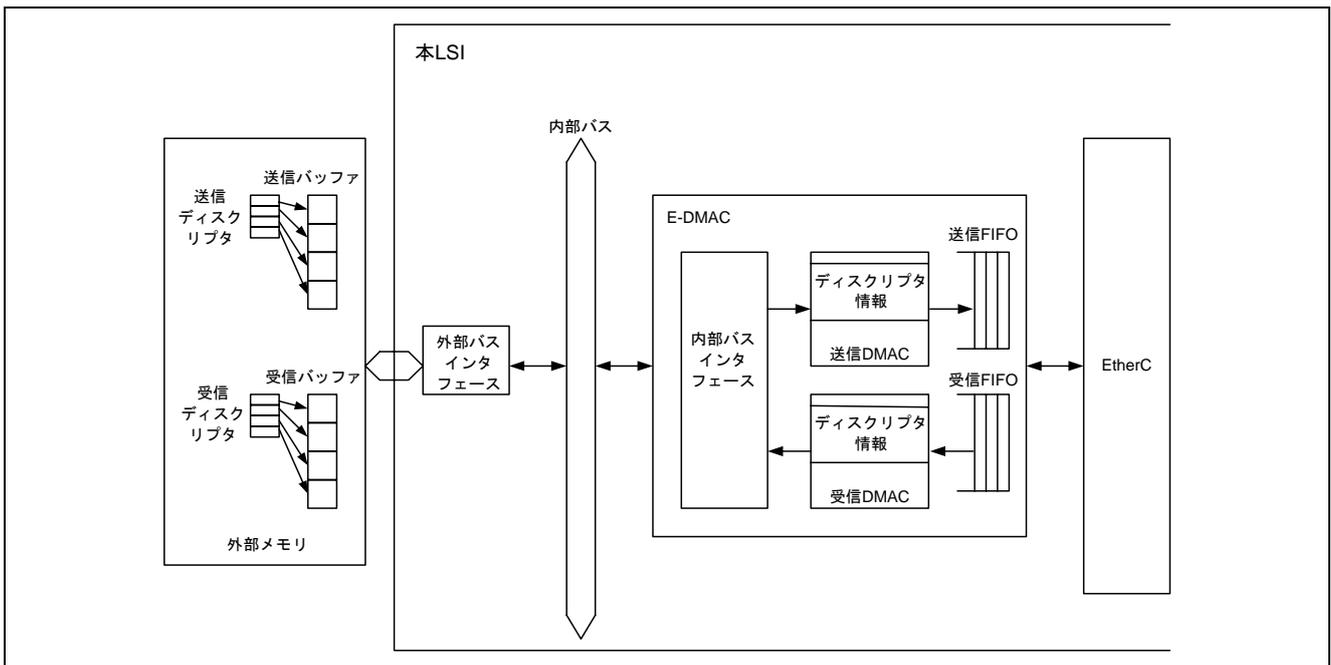


図3 E-DMAC とディスクリプタおよびバッファの構成

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。

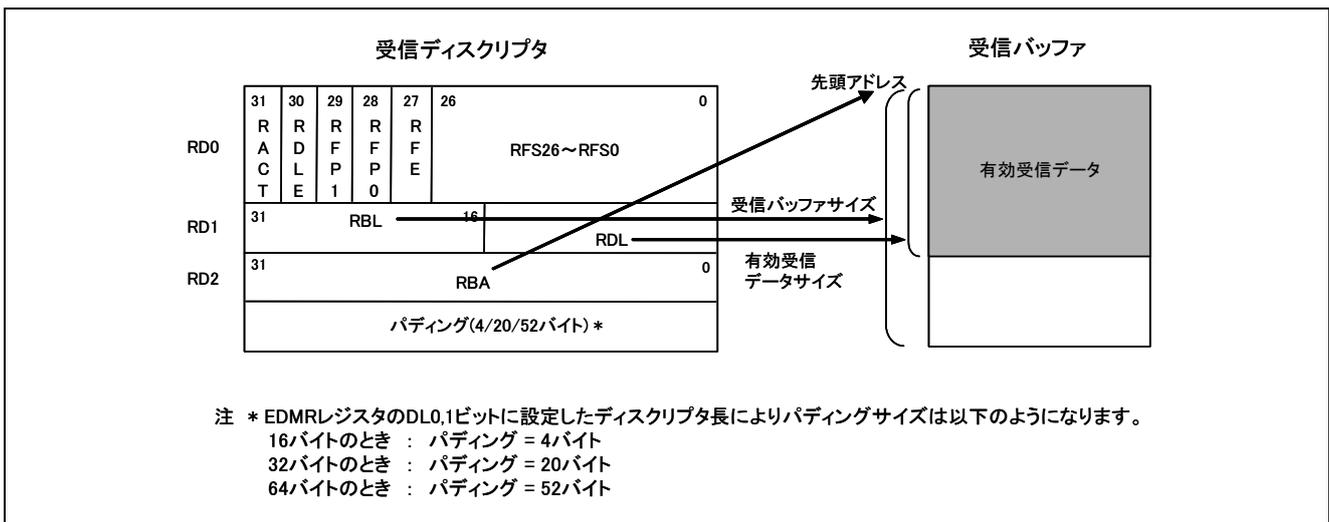


図4 受信ディスクリプタと受信バッファの関係

2.1.7 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。

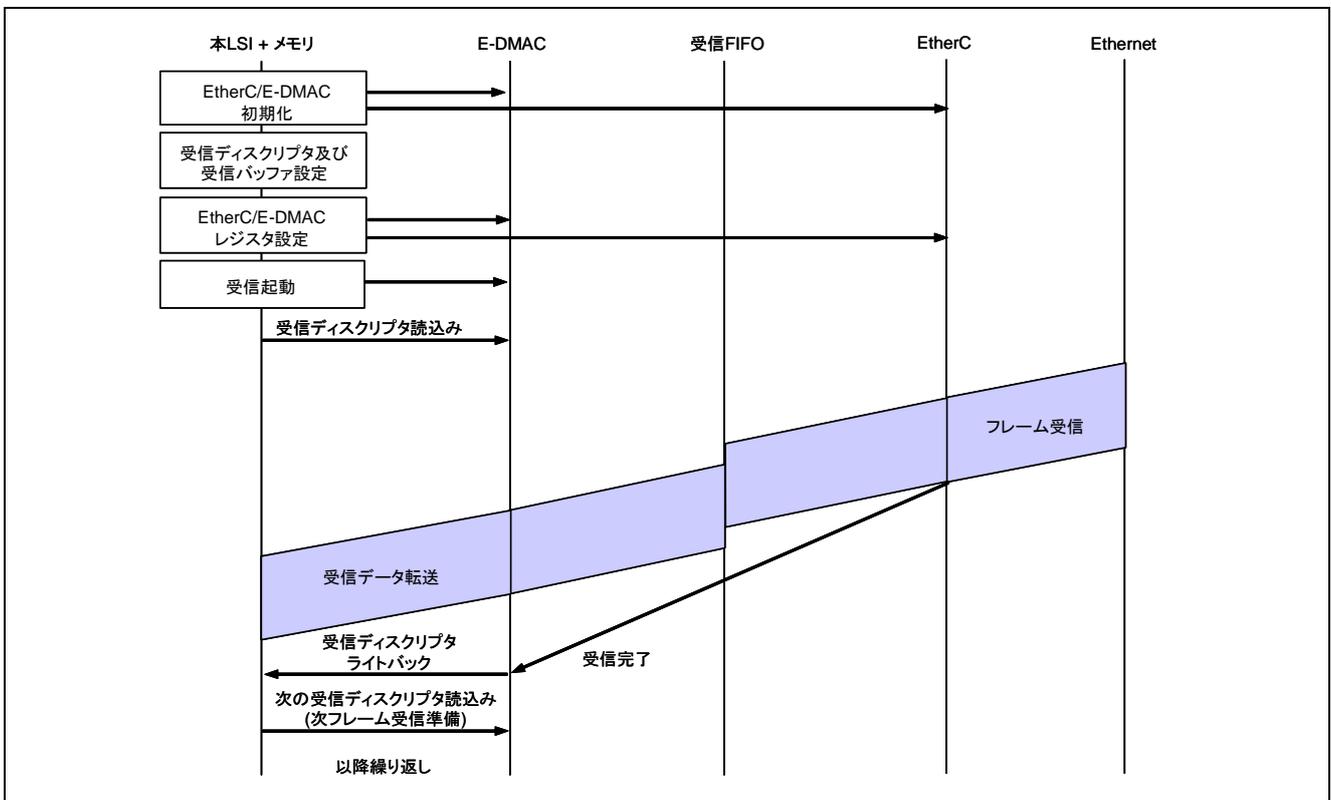


図6 受信フローの例(1フレーム/1ディスクリプタ)

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。

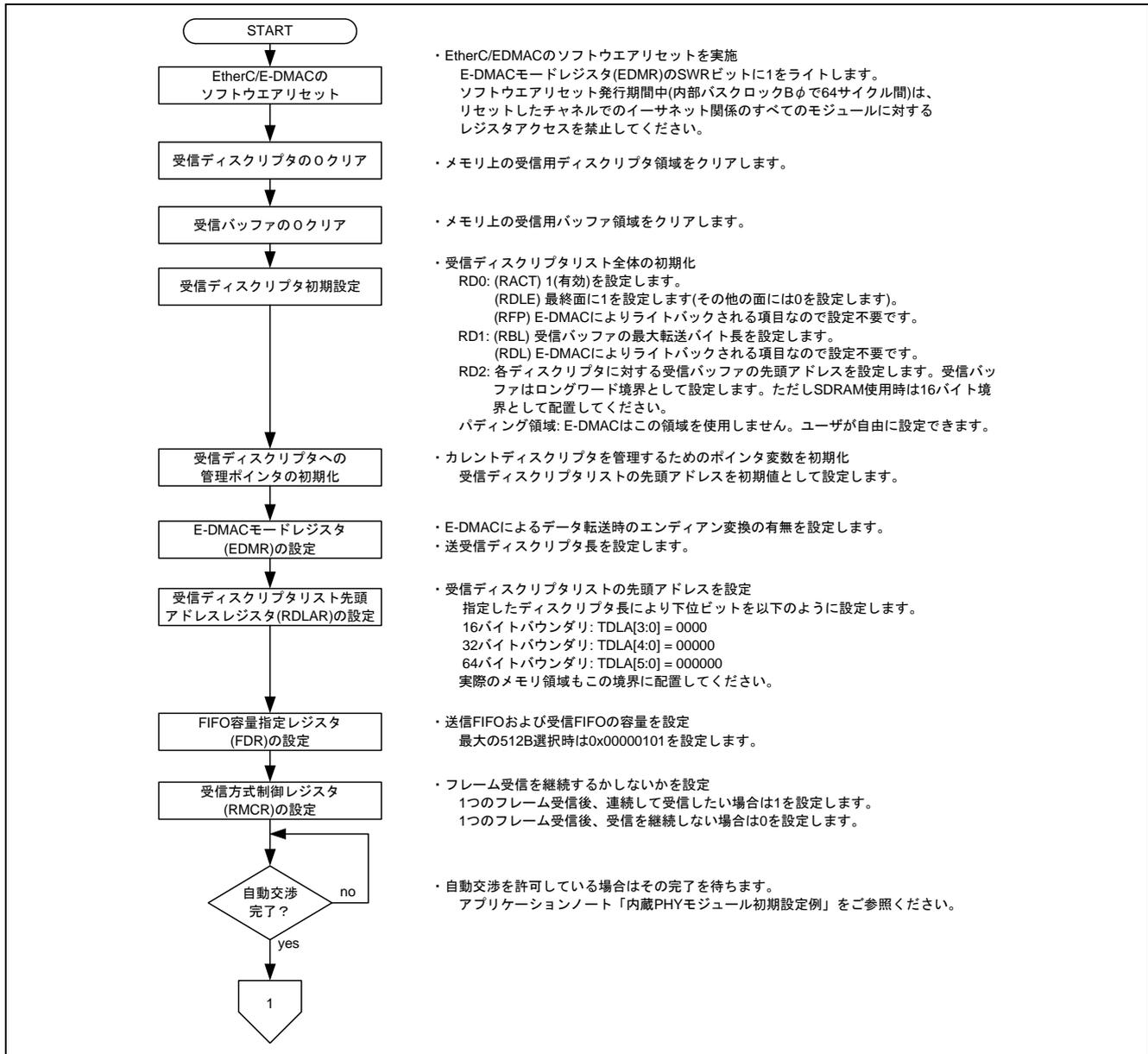


図7 イーサネット受信設定フロー例(1)

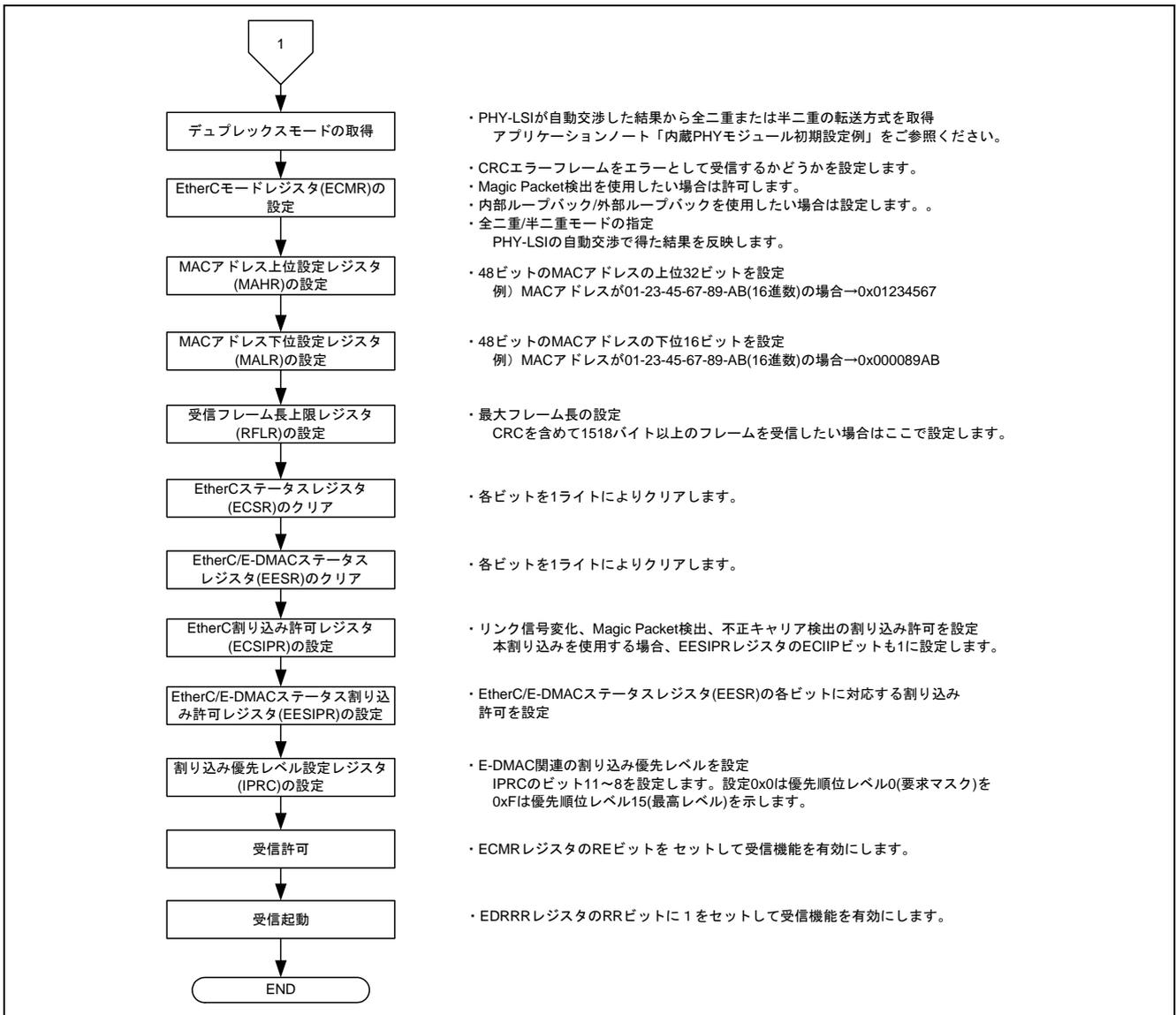


図8 イーサネット受信設定フロー例(2)

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(RD0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。

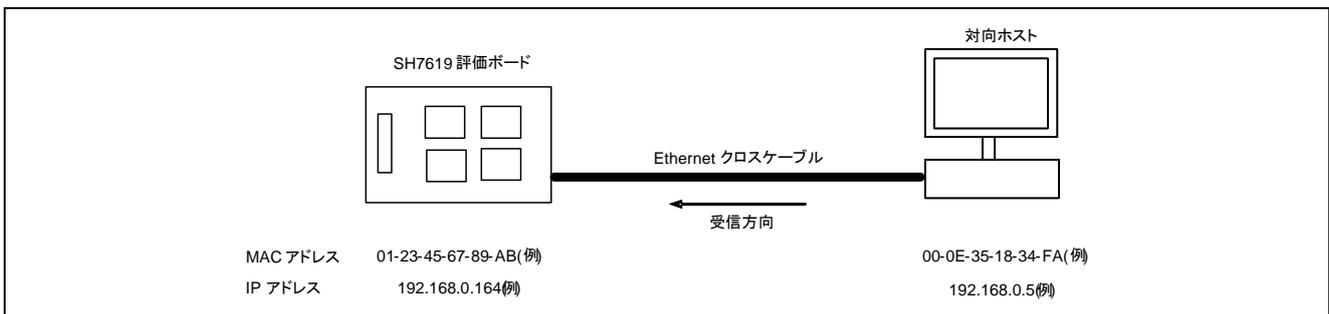


図9 参考プログラムの動作環境

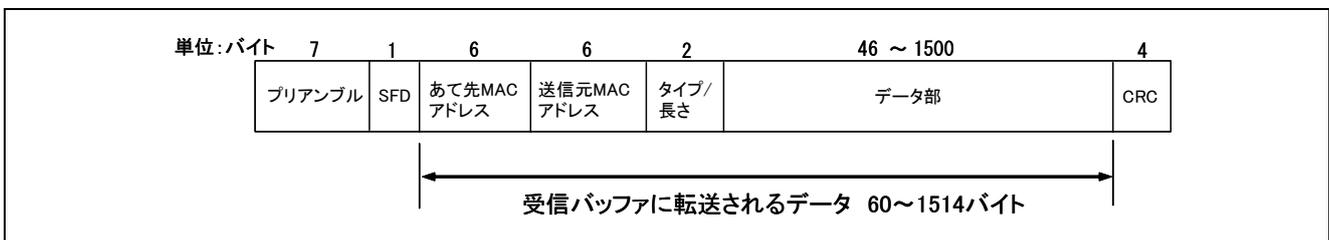


図10 イーサネットフレームフォーマット

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。

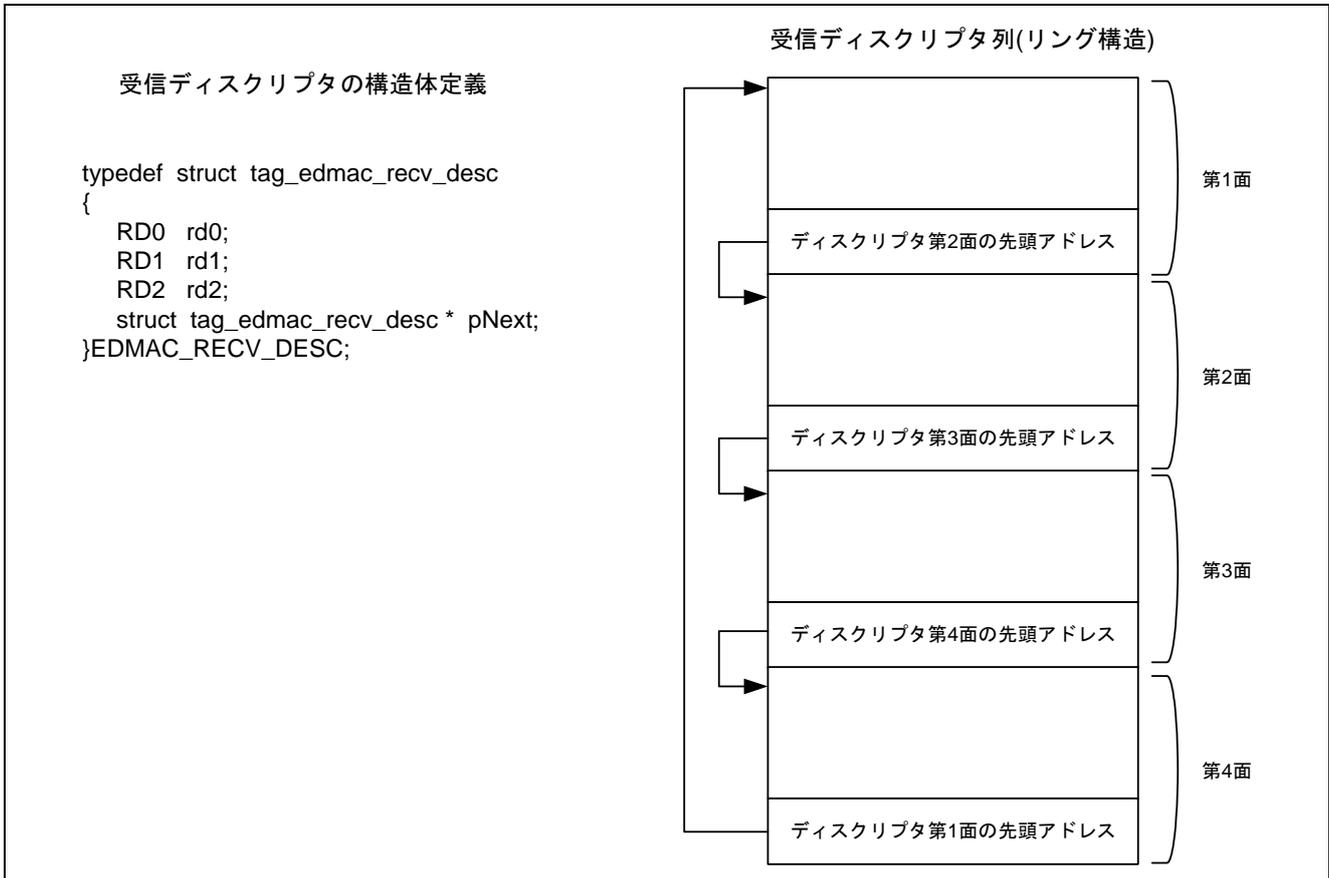


図11 受信ディスクリプタの構造体定義と受信ディスクリプタ列使用例

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「SH7619 グループ アプリケーションノート 内蔵PHYモジュール初期設定例」をご参照ください。

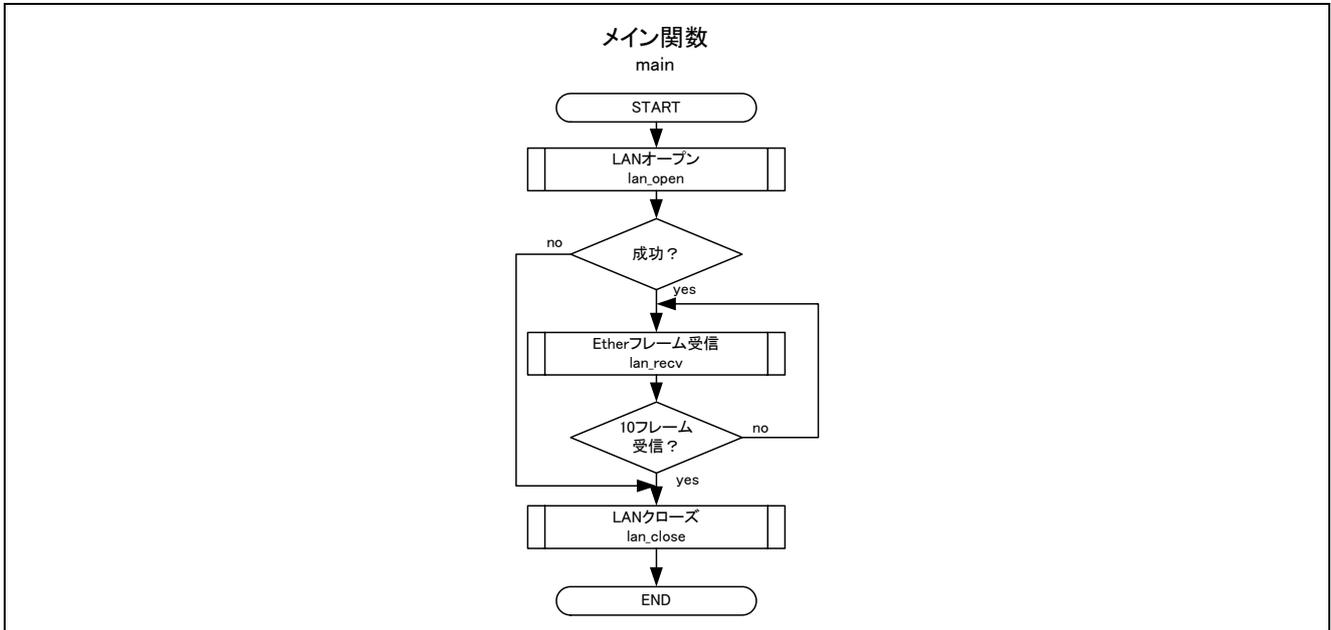


図12 メインプログラムの処理フロー例(1)

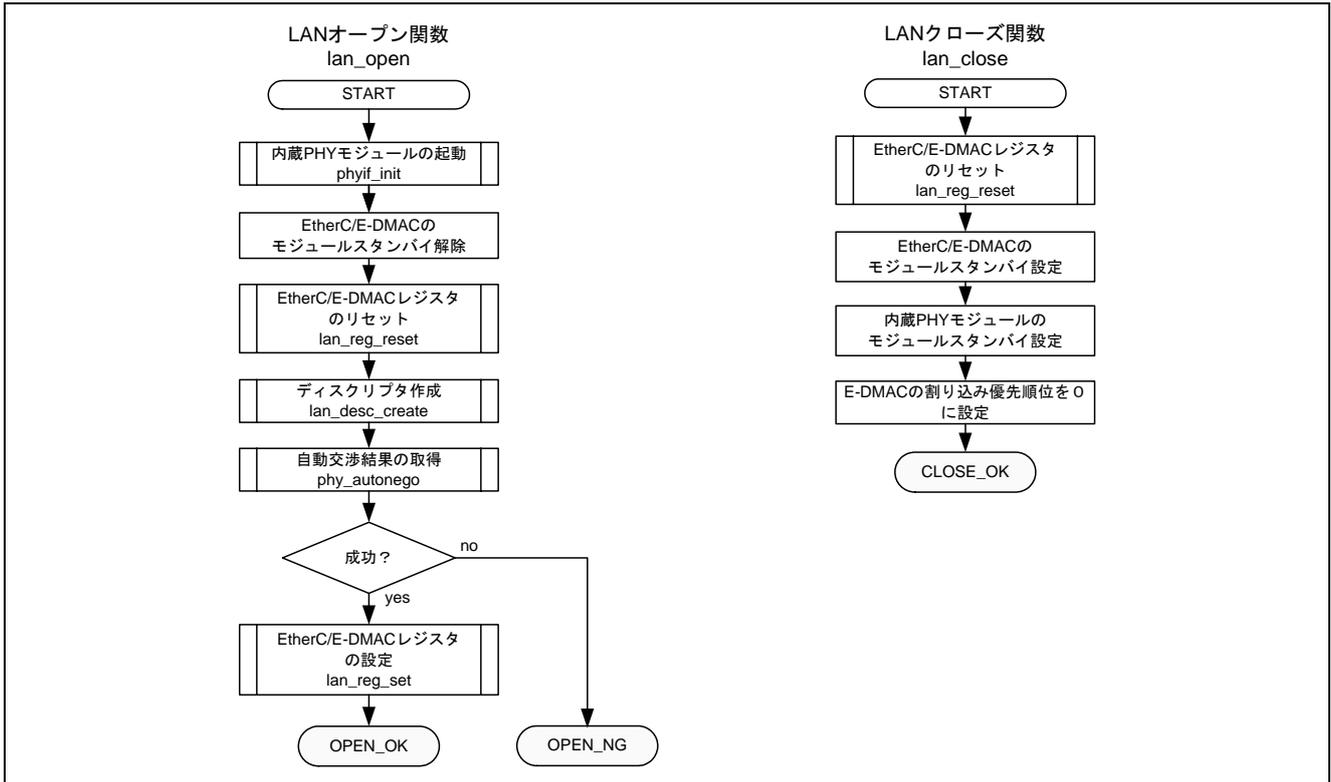


図13 参考プログラムの処理フロー例(2)

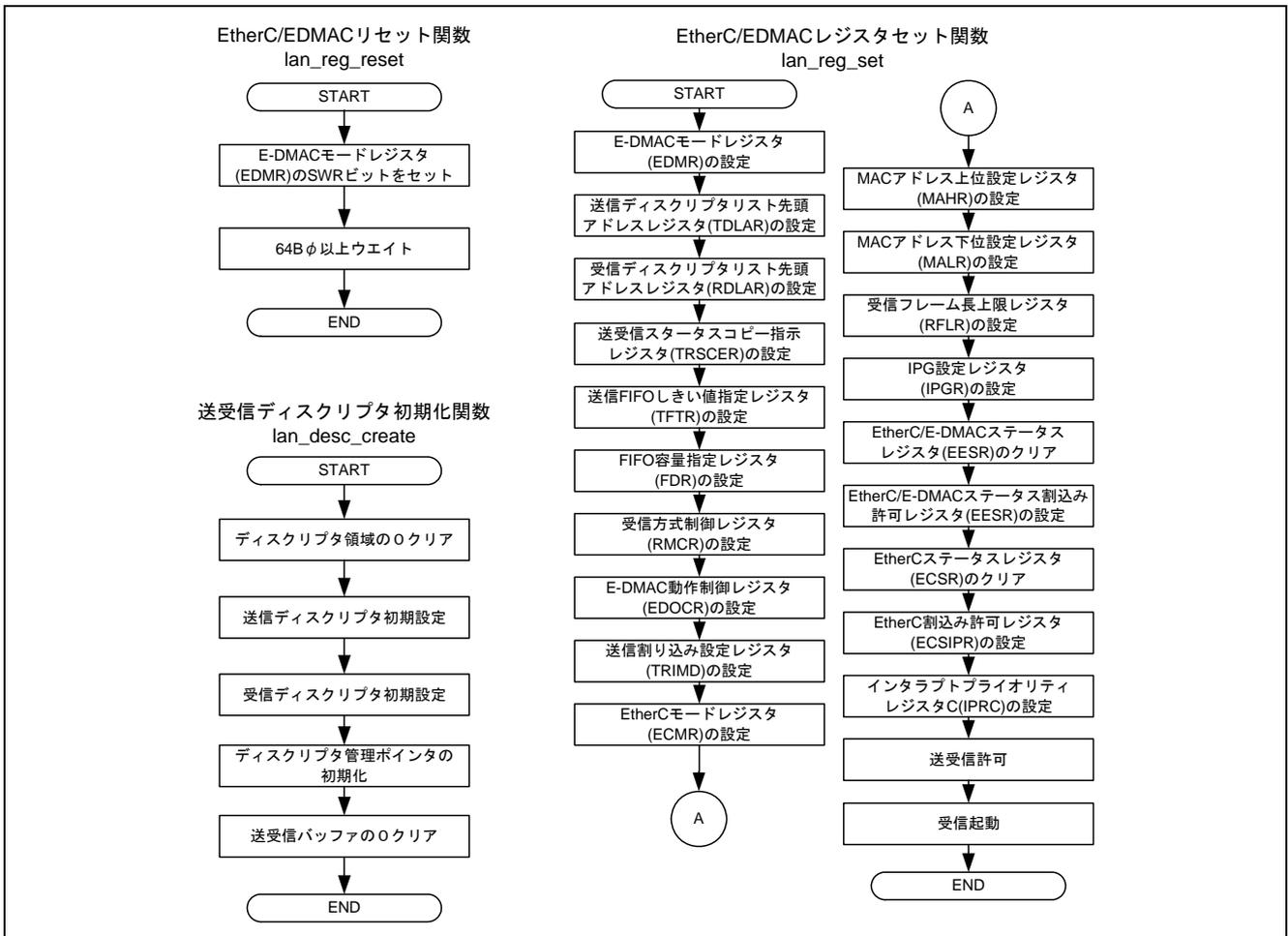


図14 参考プログラムの処理フロー例(3)

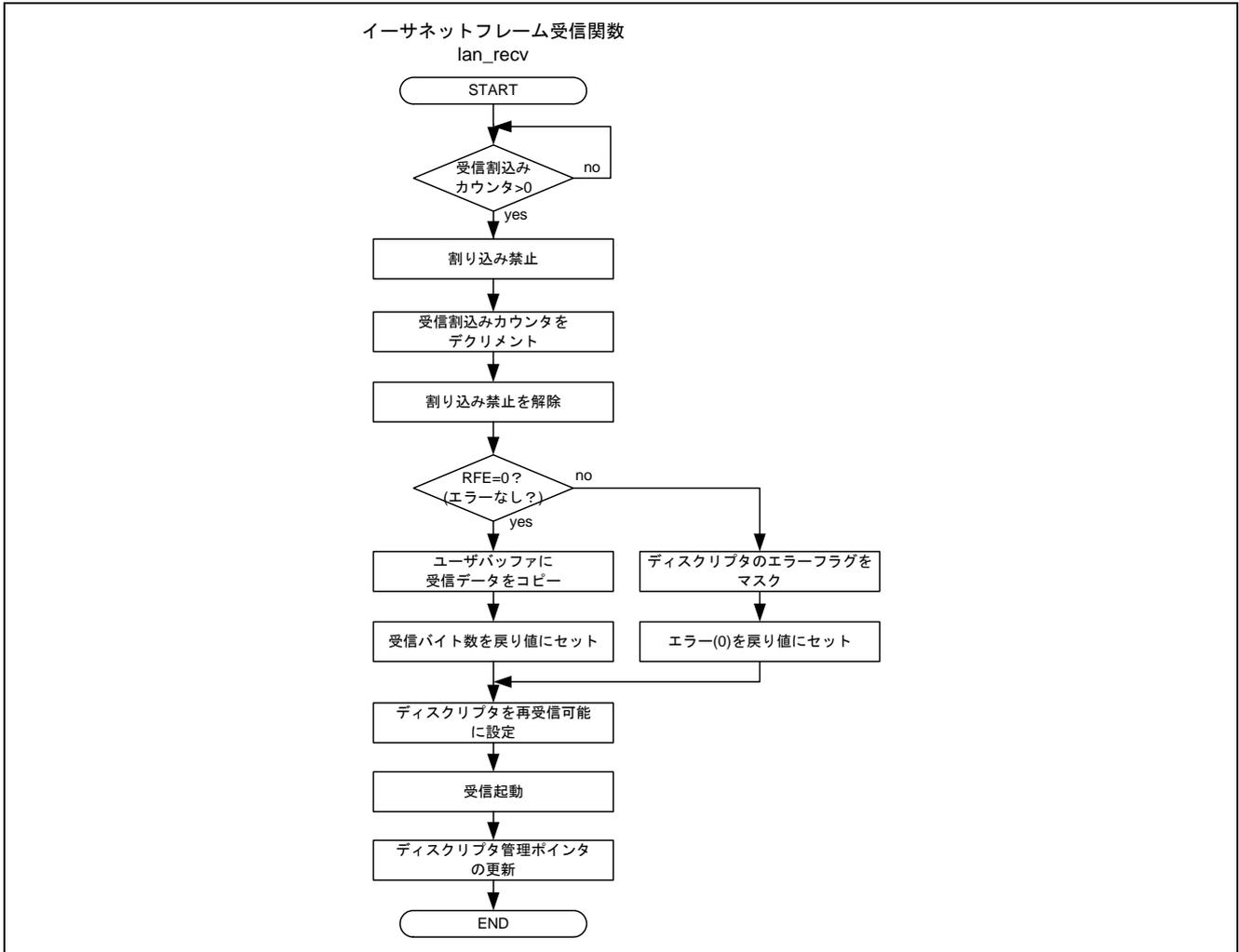


図15 参考プログラムの処理フロー例(4)

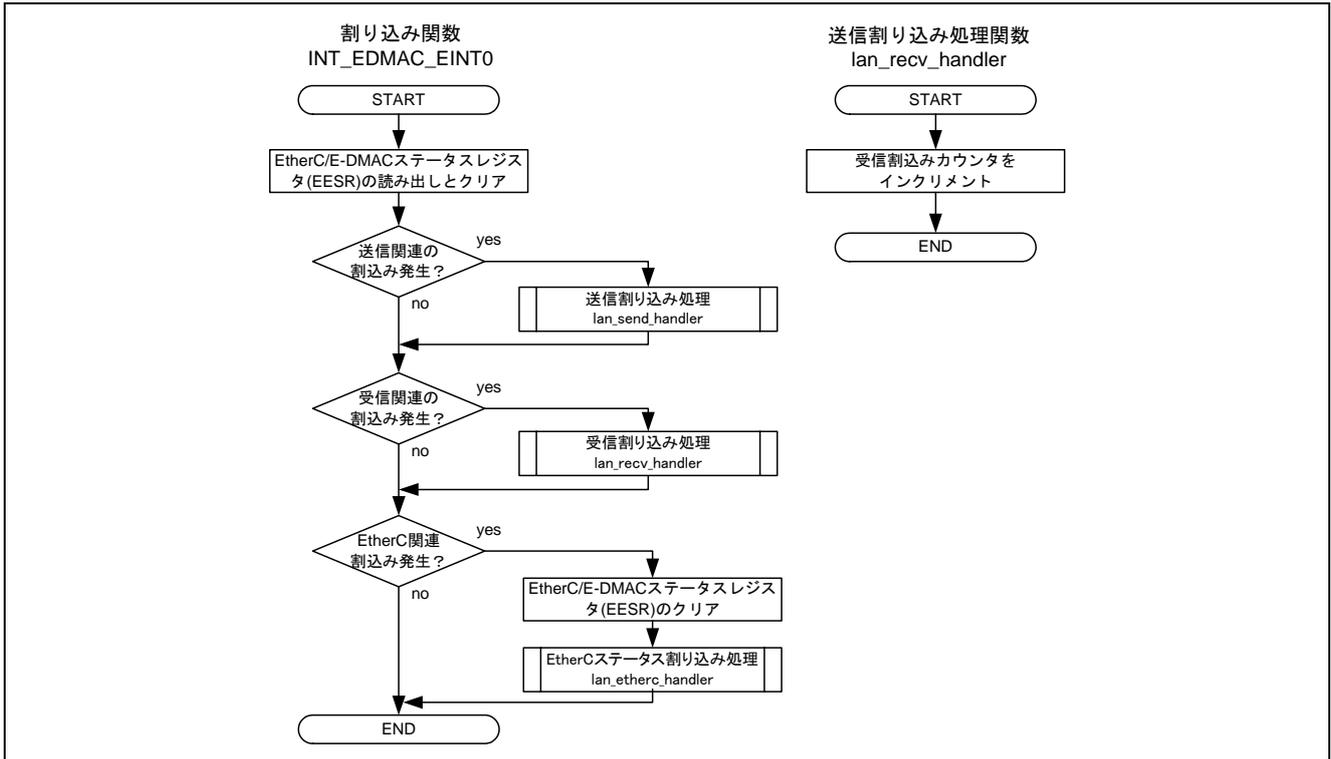


図16 参考プログラムの処理フロー例(4)

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  /*"FILE COMMENT"*****
2  *
3  * System Name : SH7619 Sample Program
4  * File Name   : main.c
5  * Contents    : イーサネット受信設定例
6  * Version     : 1.00.00
7  * Model       : M3A-HS19
8  * CPU         : SH7619
9  * Compiler    : SHC9.1.1.0
10 * note        :
11 *             <注意事項>
12 *             本サンプルプログラムはすべて参考資料であり、
13 *             その動作を保証するものではありません。
14 *             本サンプルプログラムはお客様のソフトウェア開発時の
15 *             技術参考資料としてご利用ください。
16 *
17 * Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
18 * and Renesas Solutions Corp. All Rights Reserved
19 *
20 * history     :2007.07.03 ver.1.00.00
21 *"FILE COMMENT END"*****/
22 #include "iodefine.h"
23 #include "defs.h"
24 #include "ether.h"
25
26 /* **** プロトタイプ宣言 **** */
27 void main(void);
28
29 /* **** 変数宣言 **** */
30 #pragma section ETH_BUFF /* 大容量のため SDRAM に配置 */
31 typedef struct{
32     unsigned char frame[SIZE_OF_BUFFER];
33     int len;
34     unsigned char wk[12];
35 }USER_BUFFER;
36 static USER_BUFFER recv[10];
37 #pragma section
    
```

3.2 サンプルプログラムリスト"main.c" (2)

```

38  /*"FUNC COMMENT"*****
39  * ID      :
40  * Outline  : イーサネット受信サンプルプログラム メイン関数
41  *-----
42  * Include  : #include "iodefine.h"
43  *-----
44  * Declaration : void main(void)
45  *-----
46  * Function  : 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
47  *            : 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
48  *            : 使用して、イーサネットフレームを受信します。
49  *            : また PHY モジュールには、SH7619 内蔵 PHY モジュールを使用します。
50  *            : 受信ディスクリプタを複数面用意して、連続受信します。
51  *-----
52  * Argument  : void
53  *-----
54  * ReturnValue : void
55  *-----
56  * Notice    :
57  *"FUNC COMMENT END"*****/
58 void main(void)
59 {
60     int i,j;
61     int ret;
62
63     /* ==== イーサネット初期設定 ==== */
64     ret = lan_open();
65     if( ret == OPEN_OK ){
66         /* ==== 10 フレーム受信開始 ==== */
67         for(i=0; i<10; i++){
68             /* ---- 受信 ---- */
69             recv[i].len = lan_recv( recv[i].frame );
70             if( recv[i].len == 0 ){
71                 i--;
72             }
73         }
74     }
75     /* ==== イーサネット送受信停止 ==== */
76     lan_close();
77 }
78 /* End of file */

```

3.3 サンプルプログラムリスト” ether.c”(1)

```

1  /*"FILE COMMENT"*****
2  *
3  *   System Name   : SH7619 Sample Program
4  *   File Name    : ether.c
5  *   Contents     : イーサネット受信設定例
6  *   Version      : 1.01.01
7  *   Model        : M3A-HS19
8  *   CPU          : SH7619
9  *   Compiler     : SHC9.1.1.0
10 *   note         : イーサネット受信に必要な初期設定および受信処理を記述します。
11 *
12 *               <注意事項>
13 *               本サンプルプログラムはすべて参考資料であり、
14 *               その動作を保証するものではありません。
15 *               本サンプルプログラムはお客様のソフトウェア開発時の
16 *               技術参考資料としてご利用ください。
17 *
18 *   Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
19 *   and Renesas Solutions Corp. All Rights Reserved
20 *
21 *   history      :2007.07.03 ver.1.00.00
22 *                 :2007.09.28 ver.1.01.00 lan_reg_set() レジスタ設定順序変更
23 *                 lan_close() 割込みマスク追加
24 *                 lan_desc_create() メモリクリア追加
25 *                 :2007.10.22 ver.1.01.01 main()プロトタイプ宣言削除
26 *"FILE COMMENT END"*****/
27 #include "machine.h"
28 #include "string.h"
29 #include "iodefine.h"
30 #include "defs.h"
31 #include "phy.h"
32 #include "ether.h"
33
34 /* **** プロトタイプ宣言 **** */
35 void lan_send_handler( unsigned long status );
36 static void lan_desc_create( void );
37 static void lan_reg_reset( void );
38 static void lan_reg_set( int link );
39
40 /* **** 変数宣言 **** */
41 /* ---- ディスクリプタ ---- */
42 #pragma section ETH_DESC          /* 16 バイト境界へ配置 */
43 static volatile TXRX_DESCRIPTOR_SET desc; /* ディスクリプタ領域 */
44 #pragma section
45 /* ---- バッファ ---- */
46 #pragma section ETH_BUFFER        /* 16 バイト境界へ配置 */
47 static volatile TXRX_BUFFER_SET buf; /* 送受信バッファ領域 */
48 #pragma section
49 /* ---- MAC アドレス ---- */
50 static const unsigned long my_macaddr_h = 0x00010203;
51 static const unsigned long my_macaddr_l = 0x0405; /* 00:01:02:03:04:05 */
52 /* ---- その他 ---- */
53 static volatile int c_recv = 0; /* 受信フレームカウンタ */
    
```

3.4 サンプルプログラムリスト” ether.c”(2)

```

54
55 /*"FUNC COMMENT"*****
56 * ID      :
57 * Outline : イーサネットオープン関数
58 *-----
59 * Include : #include "iodefine.h"
60 *         : #include "phy.h"
61 *         : #include "ether.h"
62 *-----
63 * Declaration : int lan_open(void)
64 *-----
65 * Function   : E-DMAC, EtherC, PHY, バッファメモリの初期化を行う。
66 *           : 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。
67 *           : 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。
68 *-----
69 * Argument   : void
70 *-----
71 * ReturnValue : OPEN_OK(0) :オープン成功
72 *            : OPEN_NG(-1):オープン失敗
73 *-----
74 * Notice     :
75 *"FUNC COMMENT END"*****/
76 int lan_open(void)
77 {
78     int link;
79
80     /* ==== 内蔵PHYモジュール起動 ==== */
81     phyif_init();
82     /* ==== EtherC/EDMACのモジュールスタンバイ解除 ==== */
83     SBY.CR4.BIT._EtherC = 0;
84     /* ==== EtherC,E-DMACの停止 === */
85     lan_reg_reset();
86     /* ==== バッファメモリの初期化 ==== */
87     lan_desc_create();
88     /* ==== EtherC,E-DMACの設定 ==== */
89     link = phy_autonego();                /* デュプレックスモード確認 */
90     if( link == NEGO_FAIL ){
91         return OPEN_NG;                    /* OPEN失敗 */
92     }
93     else{
94         lan_reg_set(link);
95     }
96     return OPEN_OK;
97 }
    
```

3.5 サンプルプログラムリスト” ether.c”(3)

```

98  /*"FUNC COMMENT"*****
99  * ID      :
100 * Outline  : イーサネットクローズ関数
101 *-----
102 * Include  : #include "iodefine.h"
103 *          : #include "ether.h"
104 *-----
105 * Declaration : int lan_close(void)
106 *-----
107 * Function   : EDMAC/EtherC, PHY の動作を停止する。
108 *             : EDMAC/EtherC, PHY はクロックの供給も停止する。
109 *-----
110 * Argument   : void
111 *-----
112 * ReturnValue : int  CLOSE_OK( 0):クローズ成功
113 *             :      CLOSE_NG(-1):クローズ失敗
114 *-----
115 * Notice     :
116 *"FUNC COMMENT END"*****/
117 int lan_close( void )
118 {
119     int i;
120
121     /* ==== EtherC,E-DMAC のリセット ==== */
122     lan_reg_reset();
123     /* ==== EtherC,E-DMAC の停止 ==== */
124     SBY.CR4.BIT._EtherC = 1;
125     /* ==== 内蔵 PHY モジュールの停止 ==== */
126     SBY.CR4.BIT._PHYIF = 1;
127     /* ==== E-DMAC 関連の割り込みを禁止 ==== */
128     INTC.IPRC.BIT._EDMAC = 0;
129
130     return CLOSE_OK;
131 }
132
133 /*"FUNC COMMENT"*****
134 * ID      :
135 * Outline  : フレーム受信関数
136 *-----
137 * Include  : #include "ether.h"
138 *          : #include "iodefine.h"
139 *-----
140 * Declaration : int lan_recv( unsigned char *addr )
141 *-----
142 * 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。
143 *             : 受信フレームが存在しなければ本関数内で待ちます。
144 *             : 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
145 *             : 行ってください。
146 *-----
147 * Argument   : なし
148 *-----
149 * ReturnValue : int : 受信したフレームのバイト数 (受信エラー:0 )
150 *-----
151 * Notice     :
152 *"FUNC COMMENT END"*****/
    
```

3.6 サンプルプログラムリスト” ether.c”(4)

```

153 int lan_recv( unsigned char *addr )
154 {
155     int i;
156     int pri;
157     int ret = 0;
158     EDMAC_RECV_DESC *p;
159
160     /* ==== 受信待ち ==== */
161     while( c_recv <= 0 ){
162         /* wait */
163     }
164     /* ==== 割込みカウンタのデクリメント ==== */
165     pri = INTC.IPRC.BIT._EDMAC;                /* 排他制御 (割込み禁止) */
166     INTC.IPRC.BIT._EDMAC = 0;
167     --c_recv;
168     INTC.IPRC.BIT._EDMAC = pri;
169
170     /* ==== 受信フレームをコピー ==== */
171     p = desc.pRecv_end;
172     if( p->rd0.BIT.RFE == 0 ){
173         memcpy(addr, p->rd2.RBA, p->rd1.RDL);
174         ret = p->rd1.RDL;
175     }
176     /* ---- 受信エラー ---- */
177     else{
178         p->rd0.LONG &= 0x70000000;            /* エラーフラグを処理 */
179         ret = 0;                            /* 受信エラー : 0 */
180     }
181     /* ==== ディスクリプタを受信できる状態に戻す ==== */
182     p->rd0.BIT.RACT = 1;
183     /* ---- 受信起動 ---- */
184     if( EDMAC.EDRRR.BIT.RR == 0 ){          /* 1 ライト前に必ず 0 をリード */
185         EDMAC.EDRRR.BIT.RR = 1;
186     }
187     /* ==== カレントポインタの更新 ==== */
188     desc.pRecv_end = p->pNext;
189
190     return ret;
191 }
192

```

3.7 サンプルプログラムリスト” ether.c”(5)

```

193  /*"FUNC COMMENT"*****
194  * ID      :
195  * Outline  : ディスクリプタ構築関数
196  *-----
197  * Include   : #include "ether.h"
198  *-----
199  * Declaration : static void lan_desc_create( void )
200  *-----
201  * 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
202  *             : 行います。1 フレーム/1 バッファを前提としています。
203  *-----
204  * Argument   : void
205  *-----
206  * ReturnValue : void
207  *-----
208  * Notice    :
209  *"FUNC COMMENT END"*****/
210 static void lan_desc_create( void )
211 {
212     int i;
213     /* ==== ディスクリプタ領域の構築 ==== */
214     /* ---- メモリクリア ---- */
215     memset(&desc, 0, sizeof(desc) );
216     /* ---- 送信ディスクリプタ ---- */
217     for(i=0; i<NUM_OF_TX_DESCRIPTOR; i++){
218         desc.send[i].td2.TBA = buf.send[i];          /* TD2 */
219         desc.send[i].td1.TDL = 0;                   /* TD1 */
220         desc.send[i].td0.LONG= 0x30000000;          /* TD0:1frame/1buf、送信無効*/
221         if( i != (NUM_OF_TX_DESCRIPTOR-1) ){ /* pNext */
222             desc.send[i].pNext = &desc.send[i+1];
223         }
224     }
225     desc.send[i-1].td0.BIT.TDLE = 1;
226     desc.send[i-1].pNext = &desc.send[0];
227     /* ==== 受信ディスクリプタ ==== */
228     for(i=0; i<NUM_OF_RX_DESCRIPTOR; i++){
229         desc.recv[i].rd2.RBA = buf.recv[i];          /* RD2 */
230         desc.recv[i].rd1.RBL = SIZE_OF_BUFFER;       /* RD1 */
231         desc.recv[i].rd0.LONG= 0xb0000000;          /* RD0:1frame/1buf、受信有効*/
232         if( i != (NUM_OF_RX_DESCRIPTOR-1) ){ /* pNext */
233             desc.recv[i].pNext = &desc.recv[i+1];
234         }
235     }
236     desc.recv[i-1].rd0.BIT.RDLE = 1;                  /* 最終ディスクリプタの設定 */
237     desc.recv[i-1].pNext = &desc.recv[0];
238
239     /* ---- ディスクリプタ管理情報の初期化 ---- */
240     desc.pSend_top = &desc.send[0];
241     desc.pRecv_end = &desc.recv[0];
242
243     /* ==== バッファ領域の構築 ==== */
244     /* ---- 領域クリア ---- */
245     memset(&buf, 0, sizeof(buf) );
246 }
    
```

3.8 サンプルプログラムリスト” ether.c”(6)

```

247  /*"FUNC COMMENT"*****
248  * ID      :
249  * Outline  : EtherC,E-DMAC レジスタ初期化関数
250  *-----
251  * Include  : #include "iodefine.h"
252  *-----
253  * Declaration : static void lan_reg_reset( void )
254  *-----
255  * Function   : EtherC と E-DMAC のレジスタをリセットします。
256  *             : 関数内でリセット期間を Bφ64 サイクル以上確保しています。
257  *-----
258  * Argument   : void
259  *-----
260  * ReturnValue : void
261  *-----
262  * Notice     :
263  *"FUNC COMMENT END"*****/
264  static void lan_reg_reset( void )
265  {
266     volatile int j = 100;          /* Bφ64 サイクルウェイト */
267
268     /* ---- ソフトウェアリセット ---- */
269     EDMAC.EDMR.BIT.SWR = 1;
270
271     /* ---- リセット期間の確保 ---- */
272     while(j--){
273         /* Bφ 64 サイクル ウェイト */
274     }
275 }
276 /*"FUNC COMMENT"*****
277 * ID      :
278 * Outline  : EhterC,E-DMAC レジスタ設定
279 *-----
280 * Include  : #include "iodefine.h"
281 *           : #include "phy.h"
282 *           : #include "ether.h"
283 *-----
284 * Declaration : void lan_reg_set(int link)
285 *-----
286 * Function   : E-DMAC, EtherC の初期化を行います。
287 *             : 送受信ともに有効に設定しています。
288 *-----
289 * Argument   : int link : I :EhterC に設定するデュプレックスモード。
290 *             :           :phy_autonego 関数の戻り値を利用。
291 *-----
292 * ReturnValue : void
293 *-----
294 * 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。
295  *"FUNC COMMENT END"*****/

```

3.9 サンプルプログラムリスト” ether.c”(7)

```

296 static void lan_reg_set( int link )
297 {
298     /* ==== EDMAC ==== */
299     EDMAC.EDMR.LONG = 0x00000000; /* エンディアン変換なし(ビッグエンディアン) */
300                                     /* ディスクリプタ長 16 バイト */
301     EDMAC.TDLAR = &desc.send[0]; /* 送信ディスクリプタ先頭 */
302     EDMAC.RDLAR = &desc.recv[0]; /* 受信ディスクリプタ先頭 */
303     EDMAC.TRSCER.LONG = 0x00000000; /* 全ステータスをディスクリプタにコピー */
304     EDMAC.TFTR = 0x00; /* 送信 FIFO しきい値(ストア&フォワード) */
305     EDMAC.FDR.BIT.TFD = 1; /* 送信 FIFO 容量(512 バイト) */
306     EDMAC.FDR.BIT.RFD = 1; /* 受信 FIFO 容量(512 バイト) */
307     EDMAC.RCR.BIT.RNC = 1; /* 連続受信可能 */
308     EDMAC.EDOCR.LONG = 0x00000000; /* FIFO エラー時も動作継続 */
309     EDMAC.FCFTR.LONG = 0x00070000; /* フロー制御しきい値設定 ※EtherCで禁止*/
310     EDMAC.TRIMD.BIT.TIS = 0; /* ライトバック完了通知をしない */
311     /* ==== EtherC ==== */
312     EtherC.ECMR.LONG = 0x00000000; /* フロー制御無効 */
313                                     /* CRC フレームを「エラーあり」と認識 */
314                                     /* Magic Packet 検出を許可しない */
315                                     /* 受信禁止 */
316                                     /* 送信禁止 */
317                                     /* 内部ループバックを行わない */
318                                     /* 外部ループバックを行わない */
319                                     /* デュプレックスモード (半二重) */
320                                     /* プロミスキャスモードを行わない */
321     if( link == FULL_TX || link == FULL_10M ){
322         EtherC.ECMR.BIT.DM = 1; /* 全二重モードに設定 */
323     }
324     EtherC.MAHR = my_macaddr_h; /* MAC アドレスの設定 */
325     EtherC.MALR = my_macaddr_l;
326     EtherC.RFLR = 0x000; /* 最大受信フレーム長(1518 バイト) */
327     EtherC.IPGR = 0x14; /* パケット間ギャップ(96 ビット時間) */
328     /* ==== 割り込み関連 ==== */
329     EDMAC.EESR.LONG = 0x47FF0F9F; /* 全ステータスクリア(1 ライトでクリア) */
330     EDMAC.EESIPR.LONG = EDMAC_EESIPR_INI_SEND | EDMAC_EESIPR_INI_RECV |
EDMAC_EESIPR_INI_EtherC;
331                                     /* 送受信+EtherCの割り込みを許可 */
332     EtherC.ECSR.LONG = 0x00000017; /* 全ステータスクリア(1 ライトでクリア) */
333     EtherC.ECSIPR.LONG = EtherC_ECSIPR_INI; /* 割り込み許可 */
334     INTC.IPRC.BIT._EDMAC= 5; /* E-DMAC(EINT0)割り込み優先順位 5 */
335     /* ==== 送受信の許可設定 ==== */
336     /* ---- EtherC ---- */
337     EtherC.ECMR.BIT.RE = 1; /* 受信機能有効 */
338     EtherC.ECMR.BIT.TE = 1; /* 送信機能有効 */
339     /* ---- E-DMAC ---- */
340     if(EDMAC.EDRRR.BIT.RR == 0){
341         EDMAC.EDRRR.BIT.RR = 1; /* 受信可能状態 */
342     }
343 }
    
```

3.10 サンプルプログラムリスト” ether.c”(8)

```

344  /*"FUNC COMMENT"*****
345  * ID      :
346  * Outline : 送信割込み関数
347  *-----
348  * Include : #include "iodefine.h"
349  *         : #include "ether.h"
350  *-----
351  * Declaration : void lan_send_handler( unsigned long status )
352  *-----
353  * Function    : EDMAC(EESR)に関する送信関連の割込みハンドラ。
354  *             :
355  *-----
356  * Argument    : unsigned long status : I : EESR の状態 (割込み許可したビットのみ)
357  *-----
358  * ReturnValue : なし
359  *-----
360  * Notice     : 本サンプルでは何も行いません。
361  *"FUNC COMMENT END"*****/
362  void lan_send_handler( unsigned long status )
363  {
364  }
365  /*"FUNC COMMENT"*****
366  * ID      :
367  * Outline : 受信割込み関数
368  *-----
369  * Include : #include "iodefine.h"
370  *         : #include "ether.h"
371  *-----
372  * Declaration : void lan_recv_handler( unsigned long status )
373  *-----
374  * Function    : EDMAC(EESR)に関する受信関連の割込みハンドラ。
375  *             :
376  *-----
377  * Argument    : unsigned long status : I : EESR の状態 (割込み許可したビットのみ)
378  *-----
379  * ReturnValue : なし
380  *-----
381  * Notice     :
382  *"FUNC COMMENT END"*****/
383  void lan_recv_handler( unsigned long status )
384  {
385      c_recv++;          /* 受信割込み回数をカウント */
386  }
    
```

3.11 サンプルプログラムリスト” ether.c”(9)

```

387  /*"FUNC COMMENT"*****
388  * ID      :
389  * Outline : EtherC 割込み関数
390  *-----
391  * Include : #include "iodefine.h"
392  *         : #include "ether.h"
393  *-----
394  * Declaration : void lan_etherc_handler( unsigned long status )
395  *-----
396  * Function   : EtherC(ECSR)に関する割込みハンドラ。
397  *           :
398  *-----
399  * Argument   : unsigned long status : I : ECSR の状態 (割込み許可したビットのみ)
400  *-----
401  * ReturnValue : なし
402  *-----
403  * Notice     : 本サンプルでは何も行いません。
404  *"FUNC COMMENT END"*****/
405  void lan_etherc_handler( unsigned long status )
406  {
407  }
408
409  /* End of file */
    
```

3.12 サンプルプログラムリスト” ether.h”(1)

```

1  /*"FILE COMMENT"*****
2  *
3  *   System Name   : SH7619 Sample Program
4  *   File Name    : ether.h
5  *   Contents     : イーサネット受信設定例
6  *   Version      : 1.00.00
7  *   Model        : M3A-HS19
8  *   CPU          : SH7619
9  *   Compiler     : SHC9.1.1.0
10 *   note         :
11 *               <注意事項>
12 *               本サンプルプログラムはすべて参考資料であり、
13 *               その動作を保証するものではありません。
14 *               本サンプルプログラムはお客様のソフトウェア開発時の
15 *               技術参考資料としてご利用ください。
16 *
17 *   Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
18 *   and Renesas Solutions Corp. All Rights Reserved
19 *
20 *   history      :2007.07.03 ver.1.00.00
21 *"FILE COMMENT END"*****/
22 #ifndef _ETHER_H
23 #define _ETHER_H
24
25 /* **** マクロ定義 **** */
26 #define NUM_OF_TX_DESCRIPTOR 4
27 #define NUM_OF_RX_DESCRIPTOR 4
28 #define NUM_OF_TX_BUFFER 4
29 #define NUM_OF_RX_BUFFER 4
30 #define SIZE_OF_BUFFER 1520 /* 16 バイトの整数倍にすること */
31
32 #define OPEN_OK 0
33 #define OPEN_NG -1
34 #define SEND_OK 0
35 #define SEND_NG -1
36 #define CLOSE_OK 0
37 #define CLOSE_NG -1
38 #define MIN_FRAME_SIZE 60
39 #define MAX_FRAME_SIZE 1514
40
41 #define EDMAC_EESIPR_INI_SEND 0x04280F00 /* 0x04000000 : 送信中断検出 *
42 * 0x00200000 : フレーム送信完了 *
43 * 0x00080000 : 送信 FIFO アンダフロー*
44 * 0x00000800 : キャリア未検出 *
45 * 0x00000400 : キャリア消失検出 *
46 * 0x00000200 : 遅延衝突検出 *
47 * 0x00000100 : 送信リトライオーバ */
48 #define EDMAC_EESIPR_INI_RECV 0x0205001F /* 0x02000000 : 受信中断検出 *
49 * 0x00040000 : フレーム受信 *
50 * 0x00010000 : 受信 FIFO オーバフロー *
51 * 0x00000010 : 端数ビットフレーム受信 *
52 * 0x00000008 : ロングフレーム受信 *
53 * 0x00000004 : ショートフレーム受信 *
54 * 0x00000002 : PHY-LSI 受信エラー *
55 * 0x00000001 : 受信フレーム CRC エラー */
    
```

3.13 サンプルプログラムリスト” ether.h”(2)

```

56 #define EDMAC_EESIPR_INI_EtherC 0x00400000 /* 0x00400000 : EtherC ステータスレジスタ*/
57 #define EtherC_ECSIPR_INI 0x00000004 /* 0x00000004 : リンク信号変化 */
58
59 /* **** 型定義 **** */
60
61 /* ==== 送信ディスクリプタ ==== */
62 typedef union{
63     unsigned long LONG;
64     struct{
65         unsigned int TACT:1; /* 送信ディスクリプタ有効 */
66         unsigned int TDLE:1; /* 送信ディスクリプタ最終 */
67         unsigned int TFP :2; /* 送信フレーム内位置 1,0 */
68         unsigned int TFE :1; /* 送信フレームエラー */
69         unsigned int reserved :23; /* TFS26~4 : 予約 */
70         unsigned int TFS3:1; /* ノーキャリア検出 (EESR-CND ビット) */
71         unsigned int TFS2:1; /* キャリア消失検出 (EESR-DLC ビット) */
72         unsigned int TFS1:1; /* 送信中の遅延衝突検出 (EESR-CD ビット) */
73         unsigned int TFS0:1; /* 送信リトライオーバー (EESR-TRO ビット) */
74     }BIT;
75 }TD0;
76 typedef struct{
77     unsigned short TDL; /* 送信バッファデータ長 */
78     unsigned short reserved;
79 }TD1;
80 typedef struct{
81     unsigned char *TBA; /* 送信バッファのアドレス */
82 }TD2;
83 typedef struct tag_edmac_send_desc{
84     TD0 td0;
85     TD1 td1;
86     TD2 td2;
87     struct tag_edmac_send_desc *pNext;
88 }EDMAC_SEND_DESC;
89
90 /* ==== 受信ディスクリプタ ==== */
91 typedef union{
92     unsigned long LONG;
93     struct{
94         unsigned int RACT:1; /* 受信ディスクリプタ有効 */
95         unsigned int RDLE:1; /* 受信ディスクリプタ最終 */
96         unsigned int RFP :2; /* 受信フレーム内位置 1,0 */
97         unsigned int RFE :1; /* 受信フレームエラー */
98         unsigned int reserved1:17; /* TFS26~10 : 予約 */
99         unsigned int RFS9:1; /* 受信 FIFO オーバフロー (EESR-RFOF ビット) */
100        unsigned int reserved2:1; /* : 予約 */
101        unsigned int RFS7:1; /* マルチキャストフレームを受信 (EESR-RMAF ビット) */
102        unsigned int reserved3:1; /* : 予約 */
103        unsigned int reserved4:1; /* : 予約 */
104        unsigned int RFS4:1; /* 端数ビットフレーム受信エラー (EESR-RRF ビット) */
105        unsigned int RFS3:1; /* ロングフレーム受信エラー (EESR-RTLE ビット) */
106        unsigned int RFS2:1; /* ショートフレーム受信エラー (EESR-RTSF ビット) */
107        unsigned int RFS1:1; /* PHY-LSI 受信エラー (EESR-PRE ビット) */
108        unsigned int RFS0:1; /* 受信フレーム CRC エラー検出 (EESR-CERF ビット) */
109    }BIT;
110 }RD0;
    
```

3.14 サンプルプログラムリスト” ether.h”(3)

```

111 typedef struct{
112     unsigned short RBL;                /* 受信バッファ長                */
113     unsigned short RDL;                /* 受信データ長                */
114 }RD1;
115 typedef struct{
116     unsigned char *RBA;                /* 受信バッファのアドレス        */
117 }RD2;
118 typedef struct tag_edmac_recv_desc{
119     RD0 rd0;
120     RD1 rd1;
121     RD2 rd2;
122     struct tag_edmac_recv_desc *pNext;
123 }EDMAC_RECV_DESC;
124
125 /* ===== 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) ===== */
126 typedef struct{
127     EDMAC_SEND_DESC send[NUM_OF_TX_DESCRIPTOR];
128     EDMAC_RECV_DESC recv[NUM_OF_RX_DESCRIPTOR];
129     EDMAC_SEND_DESC *pSend_top;        /* 送信ディスクリプタの登録位置 */
130     EDMAC_RECV_DESC *pRecv_end;       /* 受信ディスクリプタの受信完了/登録位置 */
131 }TXRX_DESCRIPTOR_SET;
132
133 /* ===== 送受信バッファ(16バイト境界に配置すること) ===== */
134 /* ----- 全送受信バッファ領域の定義 ----- */
135 typedef struct{
136     unsigned char send[NUM_OF_TX_BUFFER][SIZE_OF_BUFFER];
137     unsigned char recv[NUM_OF_RX_BUFFER][SIZE_OF_BUFFER];
138 }TXRX_BUFFER_SET;
139
140 /* **** プロトタイプ宣言 **** */
141 int lan_open( void );
142 int lan_close( void );
143 int lan_recv( unsigned char *addr );
144
145 #endif
146
147 /* End of File */
148
    
```

3.15 サンプルプログラムリスト”intprg_eth.c”(1)

```

1  /*"FILE COMMENT"*****
2  *
3  *   System Name : SH7619 Sample Program
4  *   File Name   : intprg_eth.c
5  *   Version    : 1.01.00
6  *   Contents   : EDMAC/EtherC 送受信設定例 割り込みプログラム
7  *   Model      : M3A-HS19
8  *   CPU        : SH7619
9  *   Compiler   : SHC9.1.1.0
10 *   OS         : none
11 *
12 *   note       : 本ファイルはRenesas Project Generator (Ver.4.06) により
13 *               生成されたファイルをアプリケーションノート用に変更しています。
14 *
15 *               <注意事項>
16 *               本サンプルプログラムはすべて参考資料であり、
17 *               その動作を保証するものではありません。
18 *               本サンプルプログラムはお客様のソフトウェア開発時の
19 *               技術参考資料としてご利用ください。
20 *
21 *   Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
22 *   AND Renesas Solutions Corp. All Rights Reserved
23 *
24 *   history    : 2007.07.03 ver.1.00.00
25 *               : 2007.09.28 ver.1.01.00 ベクタ番号 4-31 を RAM 転送対象外とする
26 * "FILE COMMENT END"*****/
27 #include <machine.h>
28 #include "vect.h"
29 #include <stdio.h>
30 #include "iodefine.h"
31 #include "defs.h"
32 #include "ether.h"
    
```

(途中省略)

3.16 サンプルプログラムリスト”intprg_eth.h”(2)

```

200 // 85 EDMAC EINT0
201 extern void lan_send_handler( unsigned long status);
202 extern void lan_recv_handler( unsigned long status);
203 extern void lan_etherc_handler( unsigned long status);
204
205 void INT_EDMAC_EINT0(void){
206     unsigned long stat_edmac;
207     unsigned long stat_etherc;
208
209     /* ---- 割り込み要求フラグのクリア ---- */
210     stat_edmac = EDMAC.EESR.LONG & EDMAC.EESIPR.LONG; /* 許可した割り込みのみ対象 */
211     EDMAC.EESR.LONG = stat_edmac;
212     /* ==== 送信関連 ==== */
213     if( stat_edmac & EDMAC.EESIPR.INI_SEND ){
214         lan_send_handler( stat_edmac & EDMAC.EESIPR.INI_SEND );
215     }
216     /* ==== 受信関連 ==== */
217     if( stat_edmac & EDMAC.EESIPR.INI_RECV ){
218         lan_recv_handler( stat_edmac & EDMAC.EESIPR.INI_RECV );
219     }
220     /* ==== EtherC 関連 ==== */
221     if( stat_edmac & EDMAC.EESIPR.INI_EtherC ){
222         /* ---- 割り込み要求フラグのクリア ---- */
223         stat_etherc = EtherC.ECSR.LONG & EtherC.ECSIPLR.LONG; /* 許可した割り込みのみ対象 */
224         EtherC.ECSR.LONG = stat_etherc;
225         lan_etherc_handler( stat_etherc );
226     }
227 }
    
```

(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-1/SH-2/ SH-DSP ソフトウェアマニュアル Rev7.00
(最新版をルネサス テクノロジ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
SH7619 グループハードウェアマニュアル Rev.5.00
(最新版をルネサス テクノロジ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

- ・ ルネサステクノロジホームページ
<http://japan.renesas.com/>
- ・ お問い合わせ先
<http://japan.renesas.com/inquiry>
csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2007.12.28	—	新規作成

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事用途の目的で使用しないでください。また、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、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりますは、事前に弊社営業窓口で最新の情報をご確認頂きますとともに、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意下さい。
5. 本資料に記載した情報は、正確を期すため慎重に制作したものです。万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断して下さい。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会下さい。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないで下さい。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行なうもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質及および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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致します。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願い致します。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断り致します。
13. 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弊社営業窓口までご照会下さい。