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い申し上げます。

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ものですが、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て、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がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

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

SH7710 グループ
イーサネット受信設定例

要旨

この資料は、SH7710/7712/7713 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

SH7712

目次

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)

1.3 適用条件

・ マイコン: SH7712 (HD6417712)
・ 動作周波数: 内部クロック 198.00MHz
バスクロック 66.00MHz
周辺クロック 33.00MHz
・ 統合開発環境: ルネサステクノロジ製 High-performance Embedded Workshop Ver.4.03.00.001
・ 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 V.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=sh3dsp -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、SH7710/7712/7713 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。

「SH7710 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」
「SH7710 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)の 0 系、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)の 0 系を使用します。

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 は送受信制御および MAC 間転送を行います。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。なお、SH7713 の EtherC は MAC-0 のみで転送機能はありません。

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。EtherCはMAC層インタフェースを 2 系統(0 系、1 系)内蔵しており、それぞれ独立に送受信させることができます。また、EtherCは転送処理を制御するTSU(Transfer Switching Unit)を内蔵し、MACコントローラ間で相互にデータ転送を行うことが可能です。このTSUは、両MACコントローラに入力されたフレームの受信や転送を判定するために、32 エントリの CAM(Content Addressable Memory)エントリテーブルおよび2 本の外部CAMインタフェース入力端子を有しています。さらに、転送するフレームを保持するトータル 6kバイトの転送FIFOを内蔵しており、0 系→1 系および 1 系→0 系の各転送条件に対し転送FIFO容量の割り当てを自由に設定することができます。EtherCは 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。

なお、本初級編ではMAC間転送機能を使わないため、TSUの詳細説明は省略します。

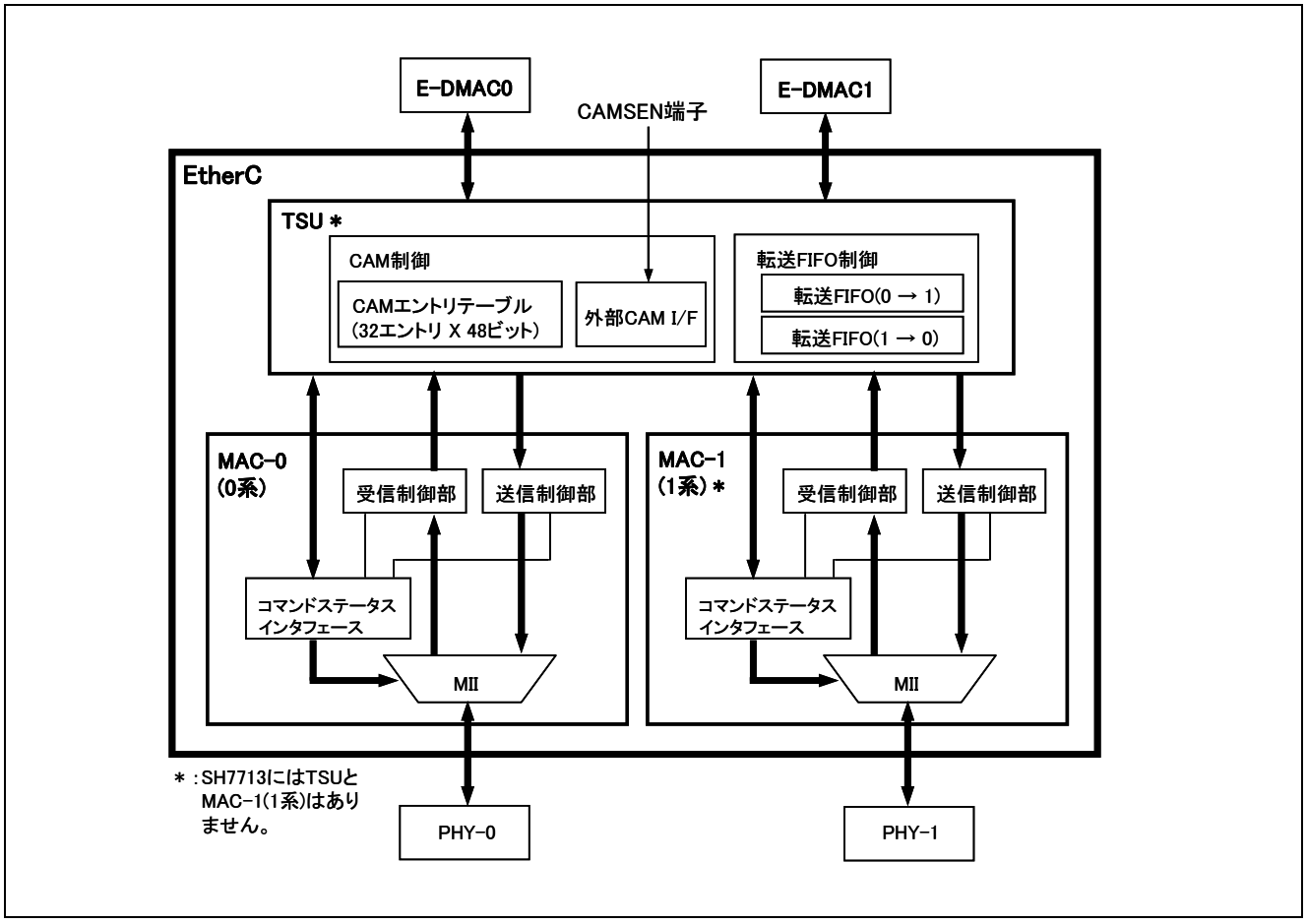


図1 EtherC の構成

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。この動作は、0 系および 1 系で共通です。また、受信時のフレーム処理ではCAM(Content Addressable Memory)の判定を参照することができます。なおCAMを使用しない場合は、EtherCの転送機能設定レジスタ(TSU_FWSLC)の設定値を変更する必要があります(デフォルトではCAMからの出力端子CAMSEN0 およびCAMSEN1 を参照する設定になっています)。初級編ではCAM機能を使用しませんのでCAM機能の説明は省略します。

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。

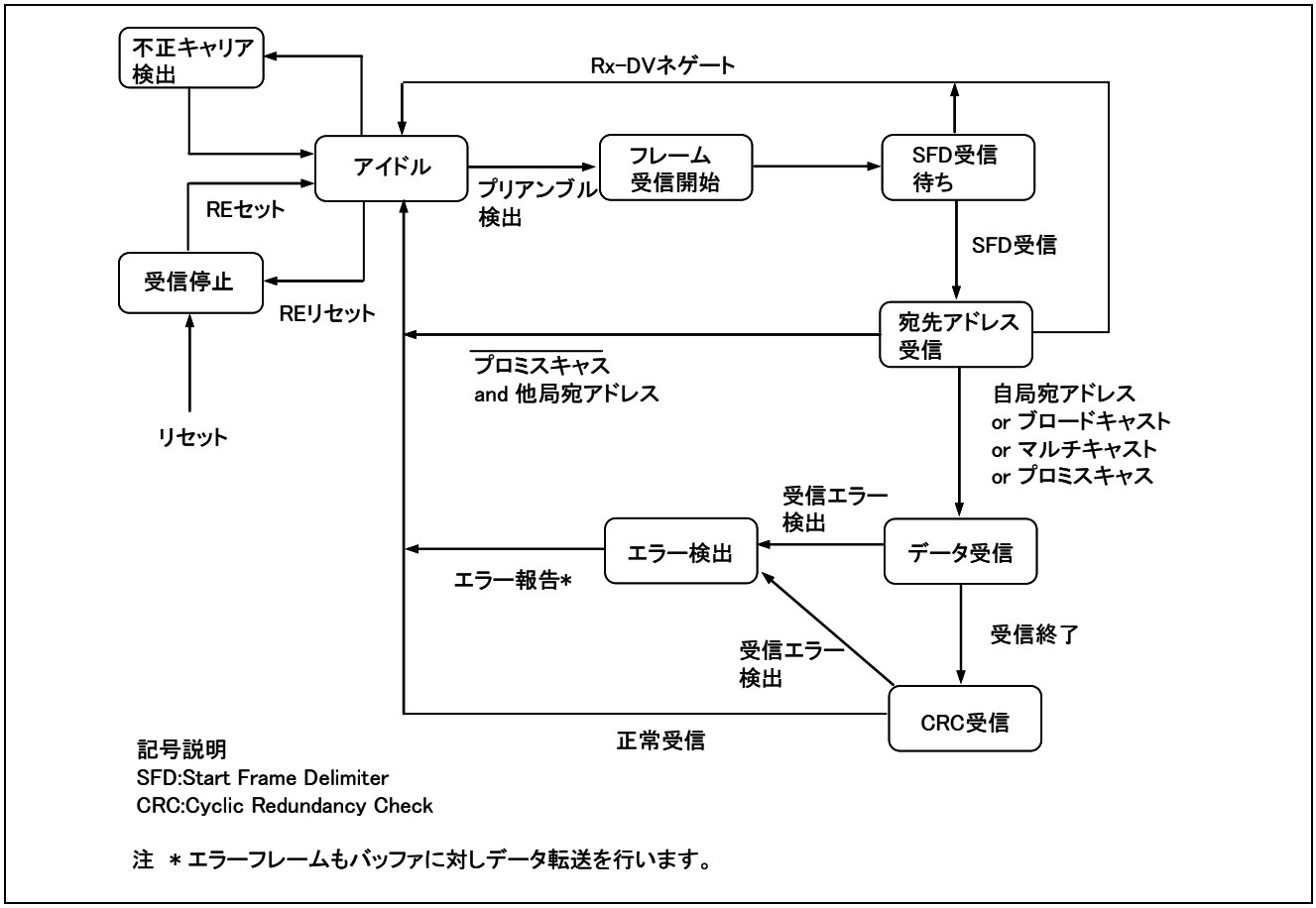


図2　EtherC 受信部状態遷移図

2.1.3 E-DMAC の概要

本LSIは、EtherCに直結した 2 系統のダイレクトメモリアクセスコントローラ(E-DMAC0/1)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。E-DMAC0 はEtherCのMAC-0 に対して、E-DMAC1 はEtherCのMAC-1 に対してデータ送受信を制御します。

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)

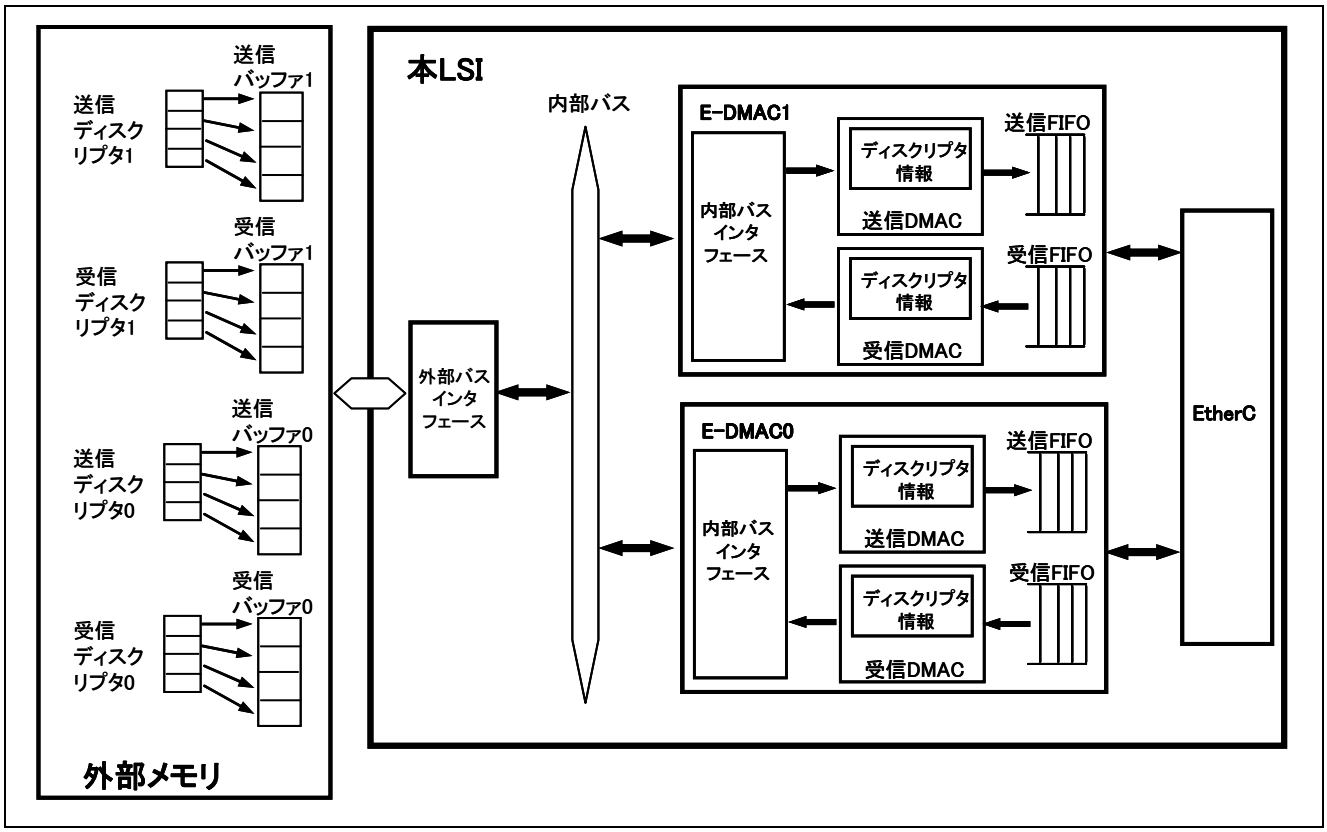


図3　E-DMAC とディスクリプタおよびバッファの構成

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。

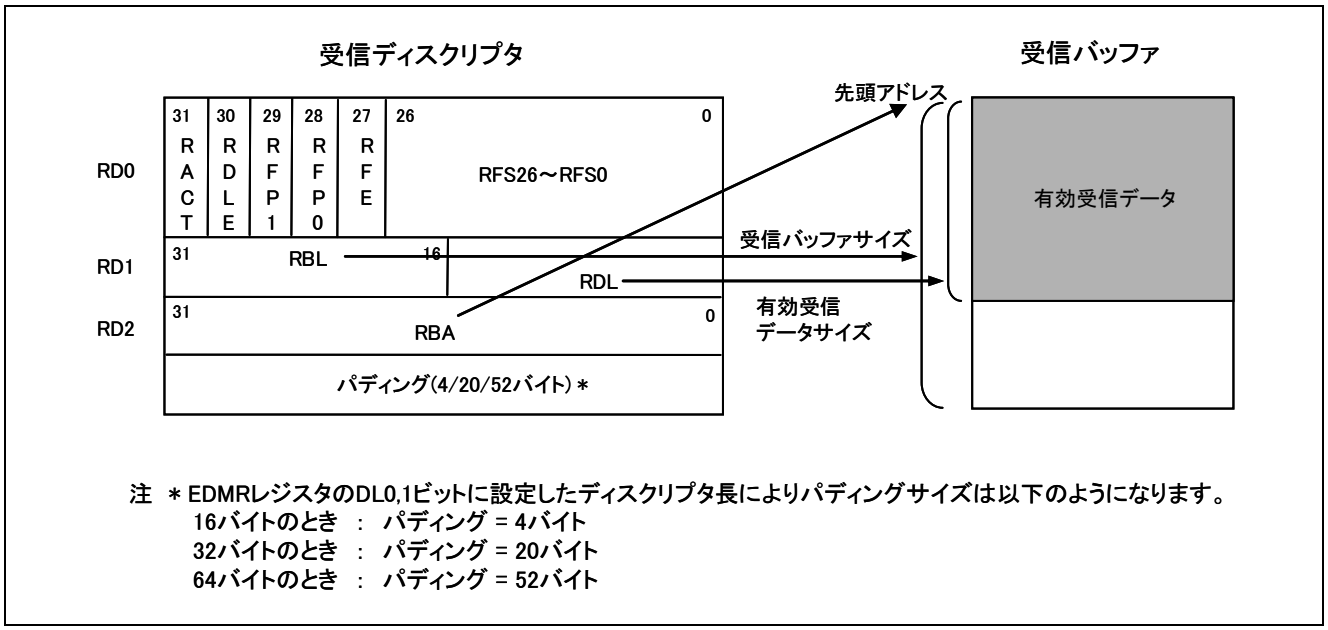


図4 受信ディスクリプタと受信バッファの関係

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。

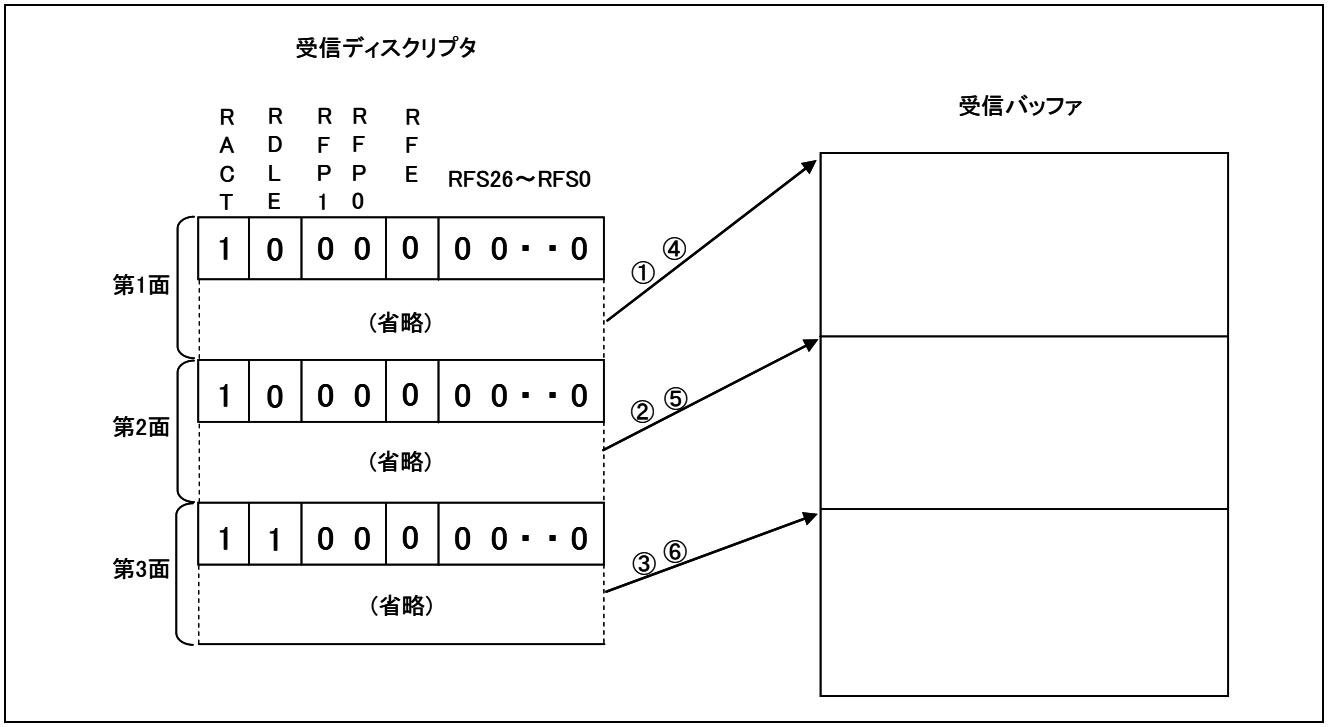


図5　受信ディスクリプタと受信バッファの関係

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のソフトウエ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。

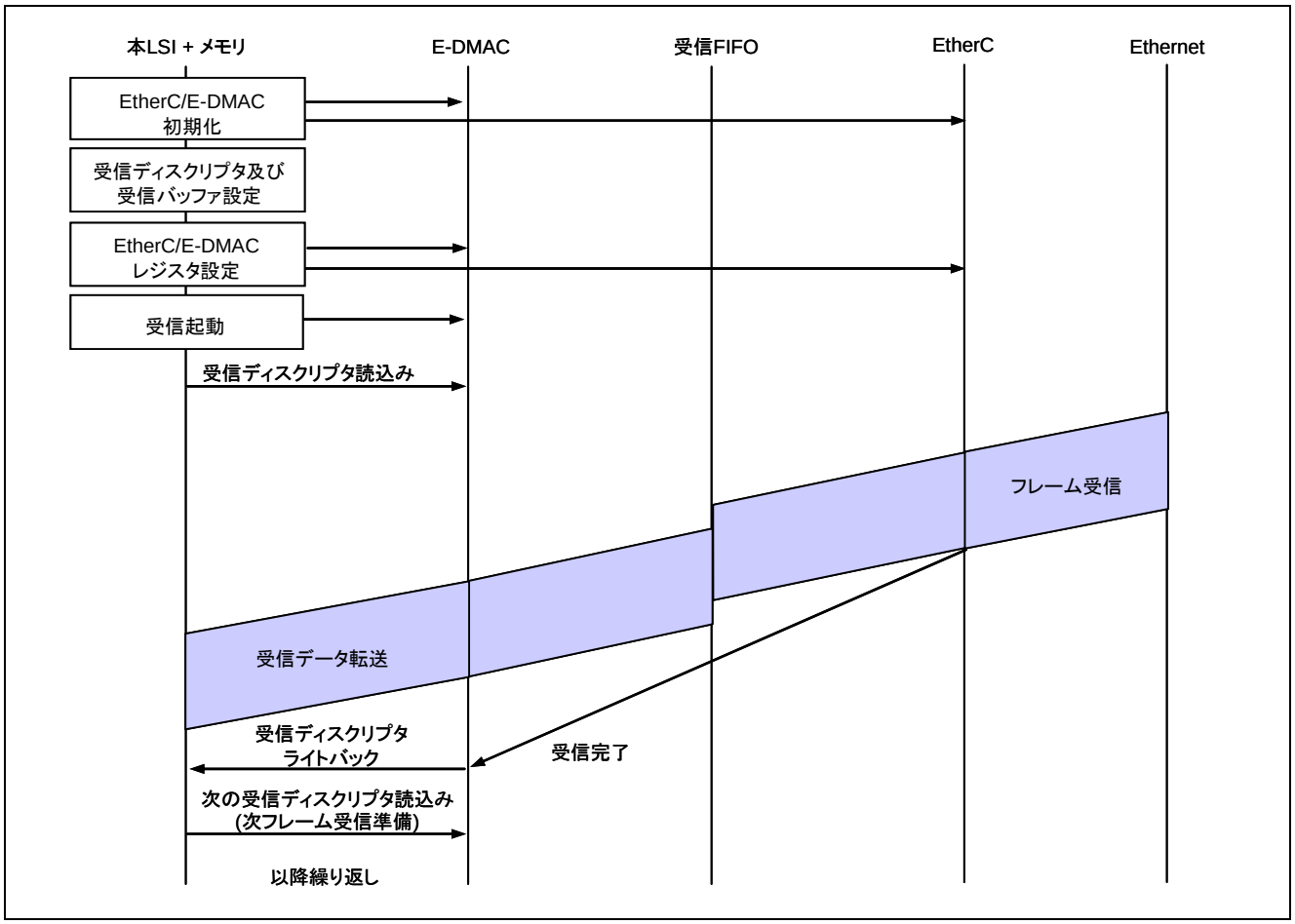
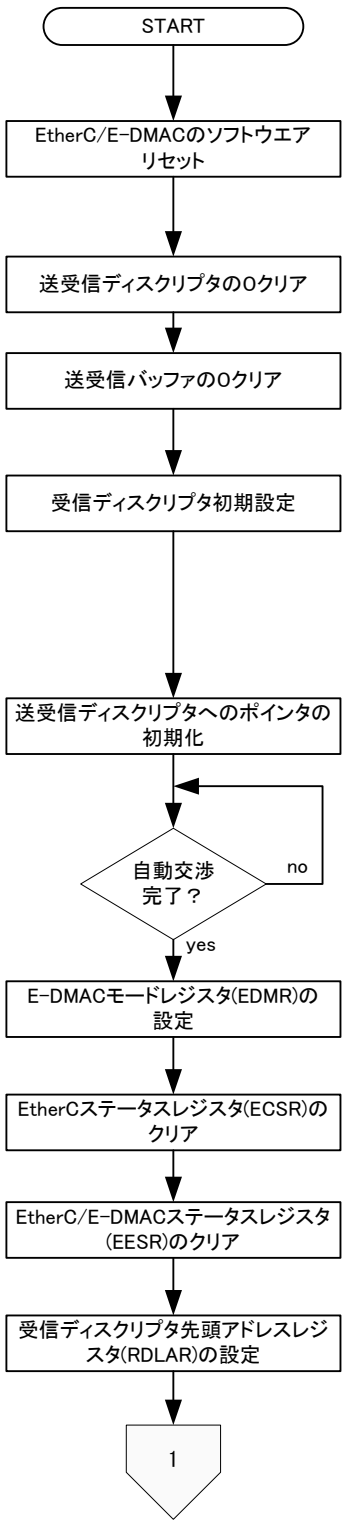


図6　受信フローの例(1 フレーム/1 ディスクリプタ)

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。



・EtherC/EDMACのソフトウエアリセットを実施
SWRビットに1をライトします。
チャネル(0または1)ごとに設定が必要です。
ソフトウエアリセット発行期間中(内部バスクロックBφで64サイクル間)は、リセットしたチャネルでのイーサネット関係のすべてのモジュールに対するレジスタアクセスを禁止してください。

・メモリ上の送受信用ディスクリプタ領域をクリアします。

・メモリ上の送受信用バッファ領域をクリアします。

・受信ディスクリプタリスト全体の初期化
RD0: RACT:1(有効)を設定します。
RDLE:最終面に1を設定します。
RD1: RBL: 受信バッファのデータ長を設定します。
RD2: RBA: 各ディスクリプタに対する受信バッファの先頭アドレスを設定します。
受信バッファは16バイト境界に配置します。
パディング領域: E-DMACはこの領域を使用しません。ユーザが自由に設定できます。

・送信ディスクリプタリストの先頭アドレスを初期値として設定します。
・受信ディスクリプタリストの先頭アドレスを初期値として設定します。

・PHY-LSIが自動交渉した結果から全二重または半二重の転送方式を取得します。
・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。

・送受信ディスクリプタ長を設定します。

・各ビットを1ライトによりクリアします。

・各ビットを1ライトによりクリアします。

・受信ディスクリプタリストの先頭アドレスを設定
指定したディスクリプタ長により下位ビットを以下のように設定します。
16バイトバウンダリ: RDLA[3:0] = 0000
32バイトバウンダリ: RDLA[4:0] = 00000
64バイトバウンダリ: RDLA[5:0] = 000000
・実際のメモリ領域もこの境界に配置してください。

図7　イーサネット受信設定フロー例(1)

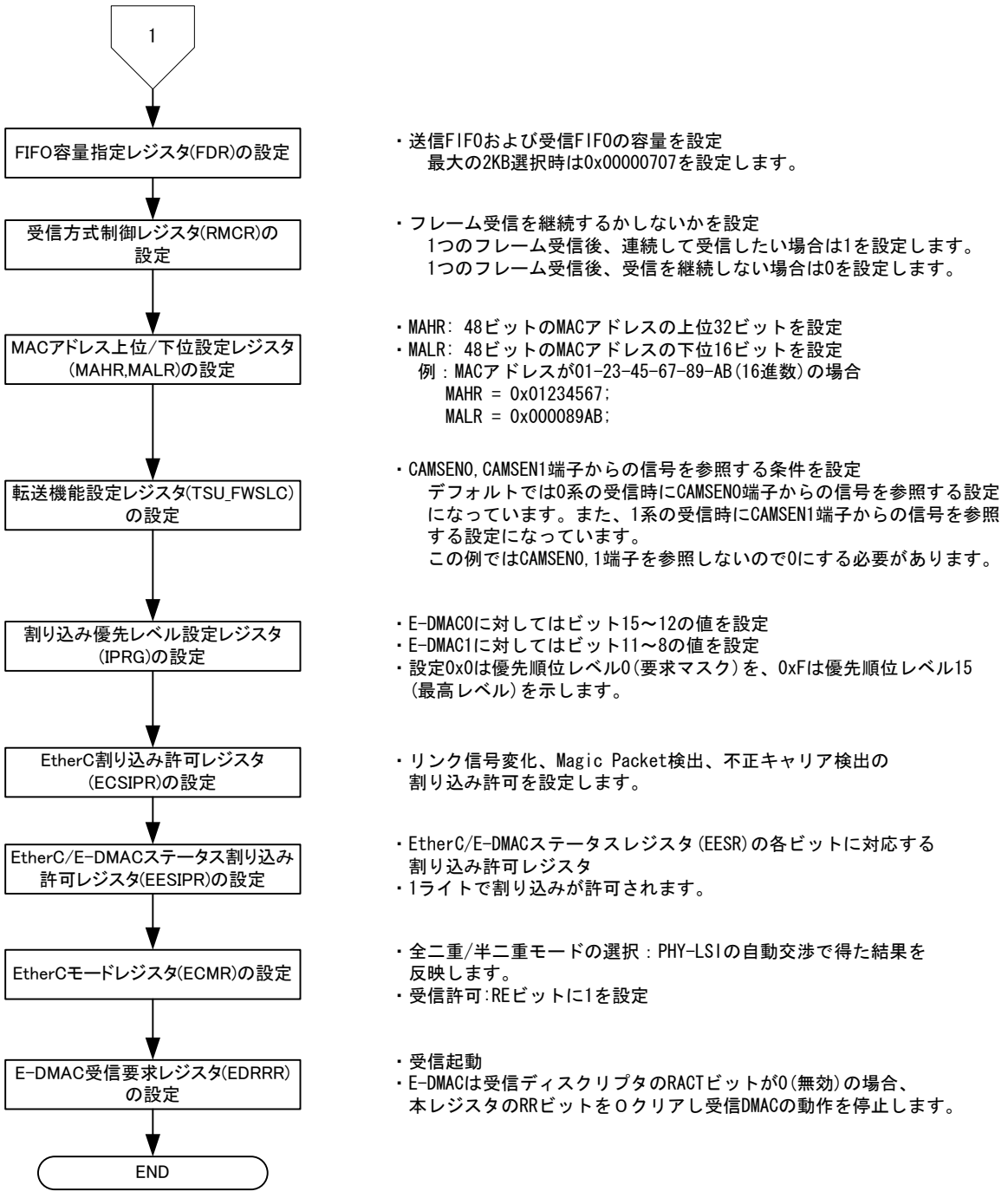


図8　イーサネット受信設定フロー例(2)

2.2 参考プログラムの動作

参考プログラムでは、EtherCの 0 系(MAC-0)およびE-DMACの 0 系(E-DMAC0)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回の受信に備えます。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ブル、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。

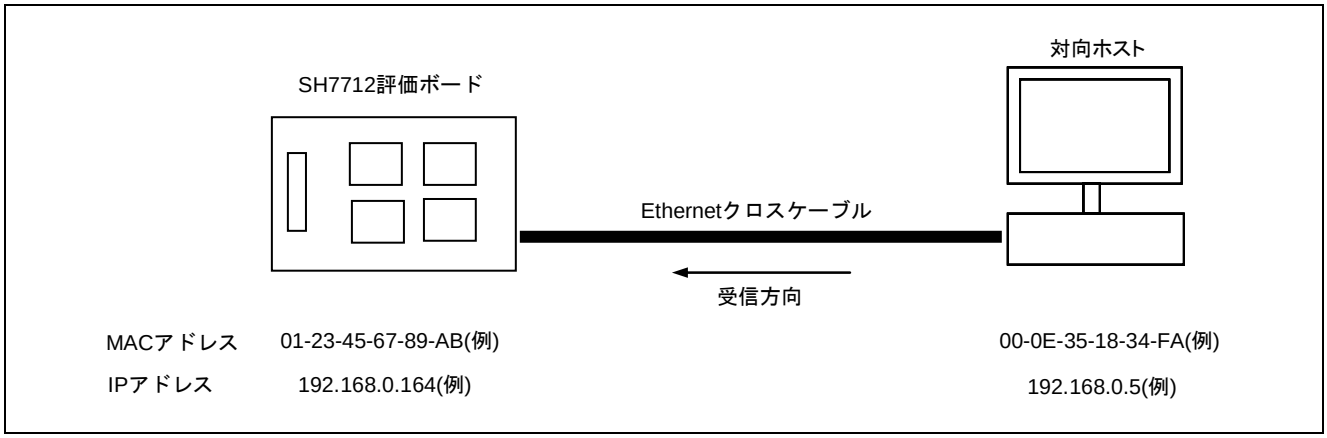


図9　参考プログラムの動作環境

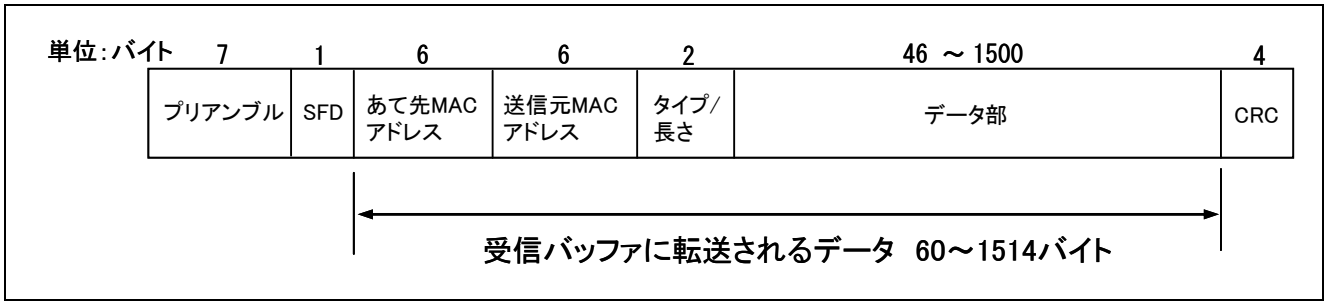


図10　イーサネットフレームフォーマット

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、ソフトウエ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。

受信ディスクリプタの構造体定義

```
typedef struct tag_edmac_recv_desc
{
    RD0    rd0;
    RD1    rd1;
    RD2    rd2;
    struct tag_edmac_recv_desc   *pNext;
}EDMAC_RECV_DESC;
```

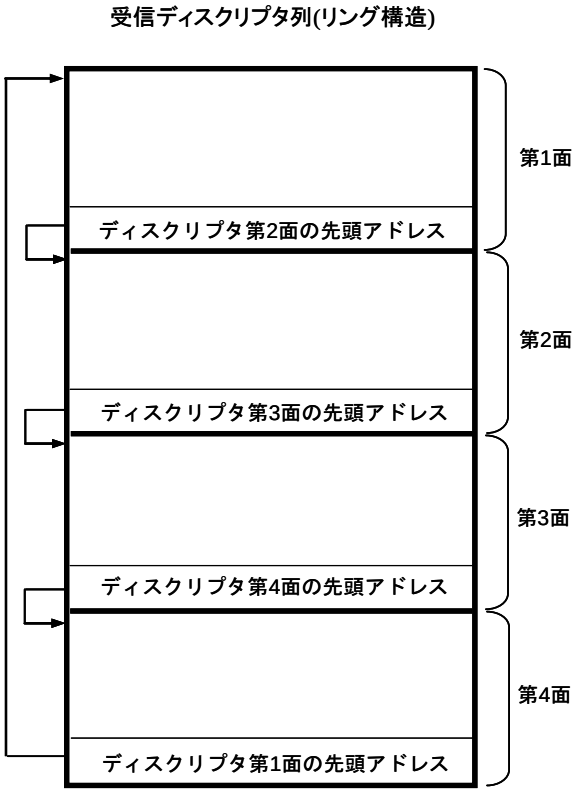


図11　受信ディスクリプタの構造体定義と受信ディスクリプタ列使用例

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「SH7710 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。

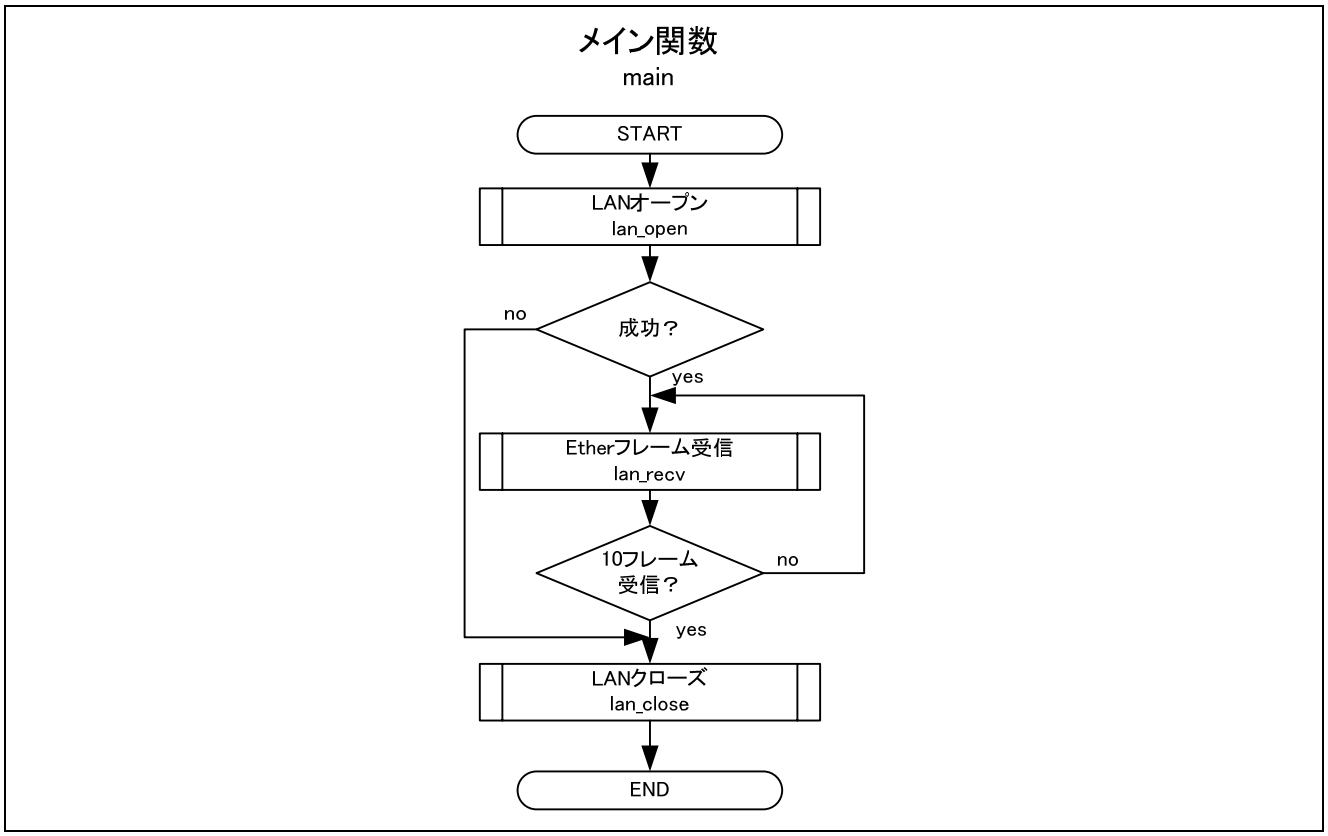


図12　メインプログラムの処理フロー例(1)

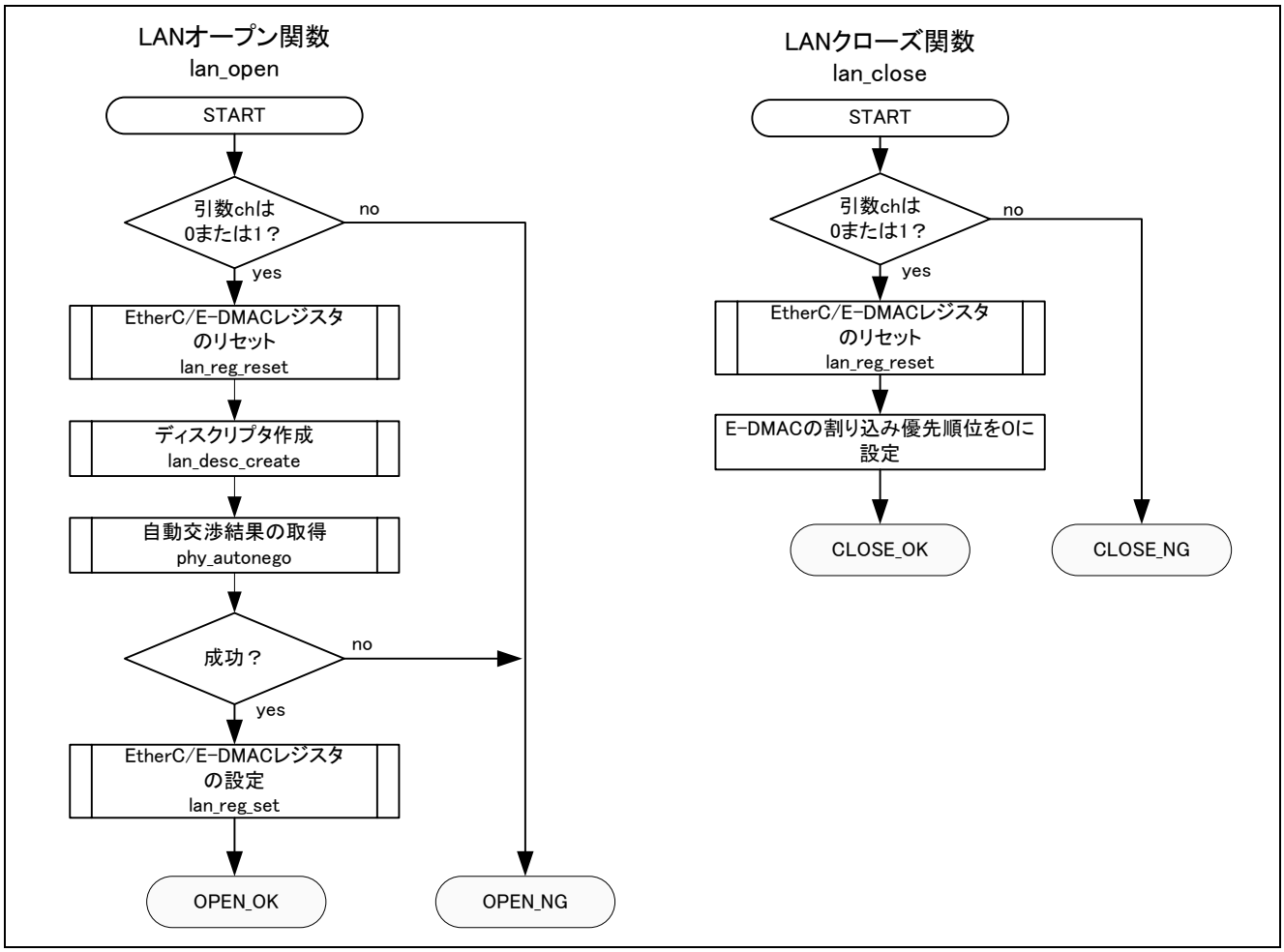


図13　参考プログラムの処理フロー例(2)

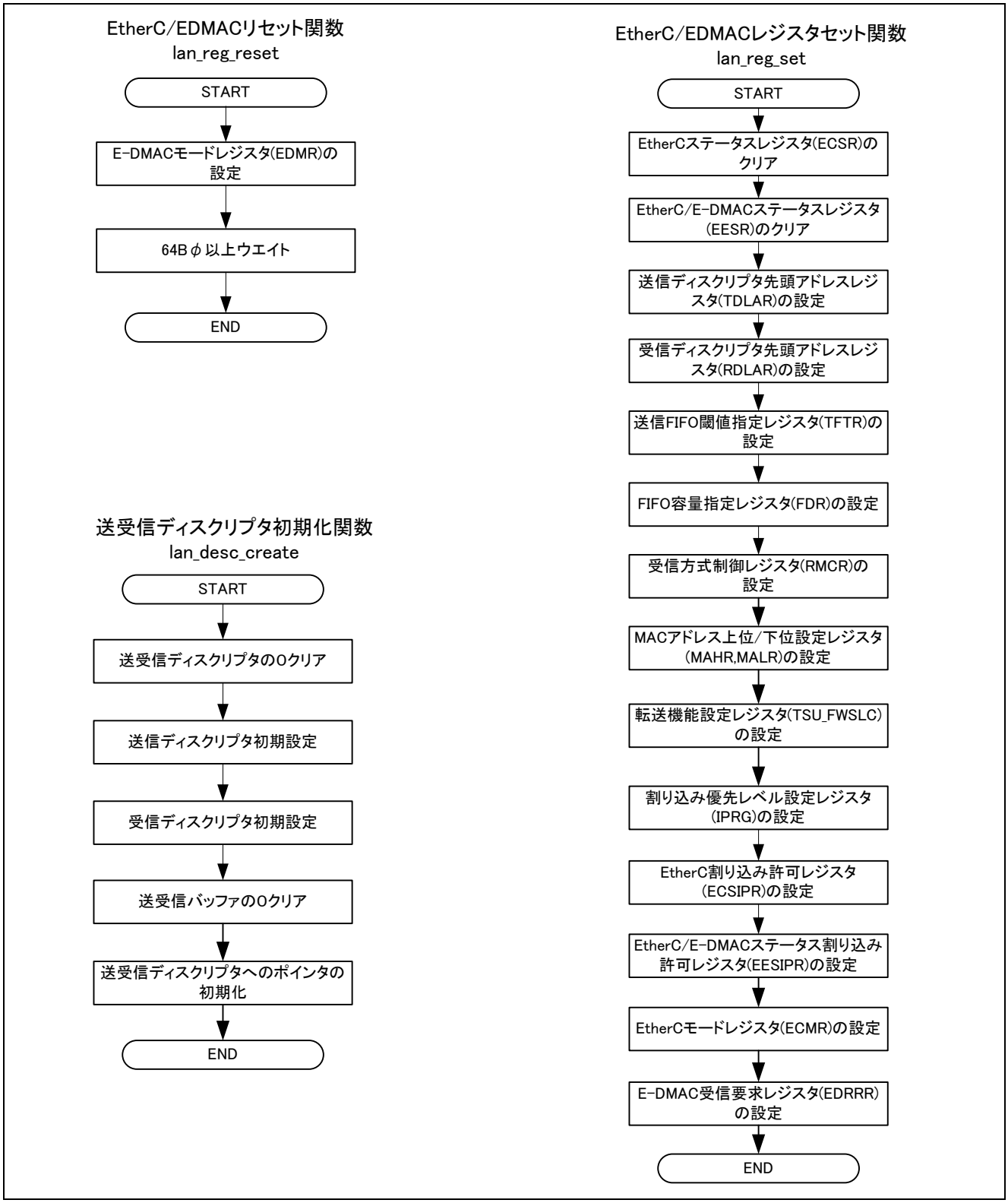


図14　参考プログラムの処理フロー例(3)

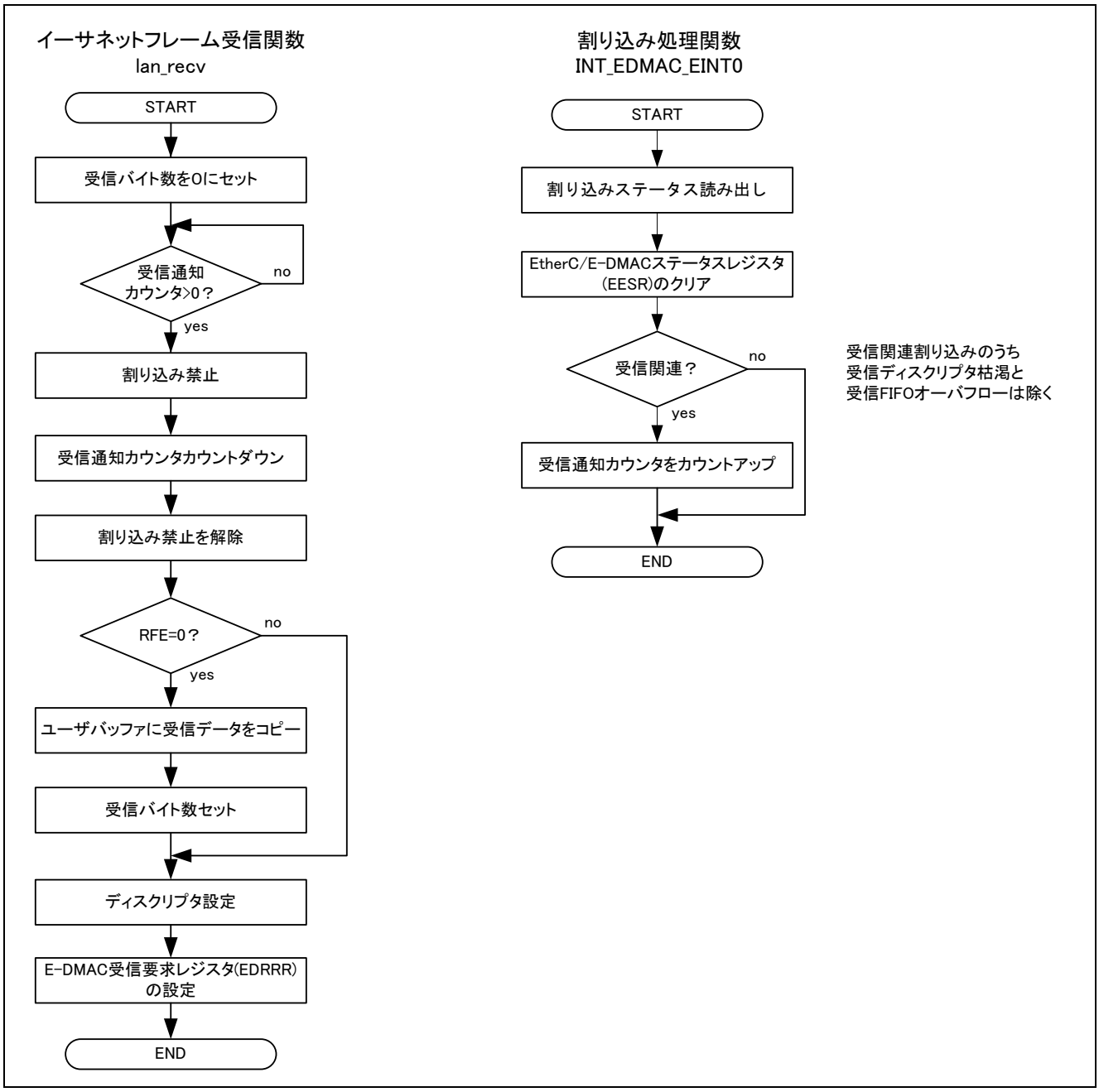


図15　参考プログラムの処理フロー例(4)

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  : SH7712 Sample Program
4   *   FILE Name    : main.c
5   *   Version      : 1.00.00
6   *   Contents     : Ethernet 受信処理
7   *   Model        : MS7712SE01
8   *   CPU          : SH7712
9   *   Compiler     : SHC9.1.1.0
10  *   OS           : None
11  *
12  *   note         : Ethernet 受信の参考プログラムです。
13  *                  <注意事項>
14  *                  本サンプルプログラムはすべて参考資料であり
15  *                  その動作を保証するものではありません。
16  *                  本サンプルプログラムはお客様のソフトウエア開発時の
17  *、                 技術参考資料としてご利用ください。
18  *
19  *
20  *   Copyright  (C)  2007 Renesas Technology Corp. All Rights Reserved
21  *      AND Renesas Solutions Corp. All Rights Reserved
22  *
23  *   history      : 2007.11.07 ver 1.00.00
24  *""FILE COMMENT END""*************************************************************/
25  #include <machine.h>
26  #include "iodefine.h"
27  #include "ether.h"
28
29  /* ==== プロトタイプ宣言 ==== */
30  void main(void);
31
32  /* ==== グローバル変数宣言 ==== */
33  unsigned char   user_buffer[10][1520];
```

3.2 サンプルプログラムリスト”main.c” (2)

```
34  /*""FUNC COMMENT""*****************************************************************
35  ID            :
36  * モジュール概要  : サンプルプログラムメイン
37  *-------------------------------------------------------------------------------
38  * Include       : #include  "iodefine.h"
39  *-------------------------------------------------------------------------------
40  * 宣言          : void main(void)
41  *-------------------------------------------------------------------------------
42  * 機能          : Ethernet から 10 フレームの受信を行います。
43  *               :
44  *------------------------------------------------------------------------ -------
45  * 引数          : なし
46  *-------------------------------------------------------------------------------
47  * 戻り値        : なし
48  *-------------------------------------------------------------------------------
49  * 注意事項      :
50  *               :
51  *               :
52  *               :
53  *""FUNC COMMENT END""*************************************************************/
54  void main(void)
55  {
56      OPEN_STATUS    opensts;
57      CLOSE_STATUS   closests;
58
59      int i;
60
61      /* ==== EtherC/E-DMAC、PHY、バッファメモリの初期化  ==== */
62      opensts = lan_open(0);     /*  ch0 を選択 */
63
64      /* ==== オープン時に受信  ==== */
65      if(opensts == OPEN_OK){
66
67          /* ==== パケット受信  ==== */
68          for(i=0;i<10;i++){
69              lan_recv(0,&user_buffer[i][0]);
70          }
71      }
72
73
74      /* ==== LAN クローズ ==== */
75      closests = lan_close(0);
76      while(closests == CLOSE_NG){
77          ;   /* waiting */
78      }
79  }
80  /* End of File */
81
```

3.3 サンプルプログラムリスト” ether.c”(1)

```
1   /*""FILE COMMENT""*****************************************************************
2   *
3   *   System Name  : SH7712 Sample Program
4   *   FILE Name    : ether.c
5   *   Version      : 1.00.00
6   *   Contents     : Ethernet 送信処理
7   *   Model        : MS7712SE01
8   *   CPU          : SH7712
9   *   Compiler     : SHC9.1.1.0
10  *   OS           :  None
11  *
12  *   note         : Ethernet 送信の参考プログラムです。
13  *                  <注意事項>
14  *                  本サンプルプログラムはすべて参考資料であり
15  *                  その動作を保証するものではありません。
16  *                  本サンプルプログラムはお客様のソフトウエア開発時の
17  *、                 技術参考資料としてご利用ください。
18  *
19  *   Copyright  (C)  2007 Renesas Technology Corp. All Rights Reserved
20  *      AND Renesas Solutions Corp. All Rights Reserved
21  *
22  *   history      : 2007.11.07 ver 1.00.00
23  *                :
24  *""FILE COMMENT END""*************************************************************/
25  #include <machine.h>
26  #include <string.h>
27  #include "iodefine.h"
28  #include "ether.h"
29  #include "phy.h"
30
31  /* ==== プロトタイプ宣言 ==== */
32  int lan_open(int ch);
33  int lan_recv(int ch, unsigned char *addr);
34  int lan_close(int ch);
35  static void lan_reg_reset(int ch);
36  static void lan_reg_set(int ch, int link);
37  static void lan_desc_create(int ch);
38
39  /* ==== 変数宣言 ==== */
40  static int  tx_flag0;
41  static int  tx_flag1;
42  static int  rx_flag0;
43  static int  rx_flag1;
44  static  EDMAC_SEND_DESC  *psenddesc0;
45  static  EDMAC_SEND_DESC  *psenddesc1;
46  static  EDMAC_RECV_DESC  *precvdesc0;
47  static  EDMAC_RECV_DESC  *precvdesc1;
48
49  /* ディスクリプタを 16 バイト境界に配置する必要があるため、専用のセクションとします。 */
50  #pragma section TXRXBUFFDESC
51  static  TXRX_BUFFER_SET  buffer0, buffer1;
52  static  TXRX_DESCRIPTOR_SET  descriptor0, descriptor1;
53  #pragma section
```

3.4 サンプルプログラムリスト” ether.c”(2)

```
54    /*""FUNC COMMENT""*****************************************************************
55    * ID            :
56    * モジュール概要  : LAN オープン関数
57    *-------------------------------------------------------------------------------
58    * Include        : #include  "iodefine.h"
59    *                 : #include  "ether.h"
60    *                 : #include  "phy.h"
61    *-------------------------------------------------------------------------------
62    * 宣言           : int lan_open(int ch)
63    *-------------------------------------------------------------------------------
64    * 機能           : E-DMAC,EtherC,送受信バッファおよび送受信ディスクリプタの初期化を行います。
65    *                 : 引数 ch をチェックし、0ch または 1ch 以外のときにエラーを返します。
66    *                 : PHY-LSI の自動交渉結果を取得し、自動交渉が失敗の場合はエラーを返します。
67    *-------------------------------------------------------------------------------
68    * 引数           : int ch: I : チャネル番号(0 または 1)
69    *-------------------------------------------------------------------------------
70    * 戻り値         :  0(OPEN_OK) : オープン成功
71    *                 : -1(OPEN_NG) : オープン失敗
72    *-------------------------------------------------------------------------------
73    * 注意事項       :
74    *
75    *""FUNC COMMENT END""*************************************************************/
76    int lan_open(int ch)
77    {
78        unsigned int physts;
79
80        /* ==== 引数 ch の確認 ==== */
81        if(ch == 0 || ch == 1){
82
83            /* ==== EtherC/E-DMAC のリセット ==== */
84            lan_reg_reset(ch);
85
86            /* ==== 送受信ディスクリプタの初期化 ==== */
87            lan_desc_create(ch);
88
89            /* ==== PHY の自動交渉結果の取得 ==== */
90            physts = phy_autonego(ch);
91
92            /* ==== 自動交渉成功時 ==== */
93            if(physts != NEGO_FAIL){
94
95                /* ==== EtherC/E-DMAC レジスタ設定 ==== */
96                lan_reg_set(ch,physts);
97
98                return OPEN_OK;
99            }
100       }
101       return OPEN_NG;
102   }
```

3.5 サンプルプログラムリスト” ether.c”(3)

```
/*""FUNC COMMENT""*************************************************************
* ID            :
* モジュール概要  : LAN クローズ関数
*-----------------------------------------------------------------------------
* Include       : #include  "iodefine.h"
*               : #include  "ether.h"
*-----------------------------------------------------------------------------
* 宣言          : int lan_close(int ch)
*-----------------------------------------------------------------------------
* 機能          : Ether 機能を停止して送受信を禁止します。
*               : 引数 ch をチェックし、0ch または 1ch 以外のときにエラーを返します。
*-----------------------------------------------------------------------------
* 引数          : int ch: I : チャネル番号(0 または 1)
*-----------------------------------------------------------------------------
* 戻り値        : 0(CLOSE_OK) : クローズ成功
*               : -1(CLOSE_NG) : クローズ失敗
*-----------------------------------------------------------------------------
* 注意事項      :
*               :
*               :
*               :
*""FUNC COMMENT END""*************************************************************/
int lan_close(int ch)
{
    /* ==== 引数 ch の確認 ==== */
    if(ch == 0 || ch == 1){

        /* ==== EtherC,E-DMAC 関連レジスタのリセット ==== */
        lan_reg_reset(ch);

        /* ==== 割り込み優先レベル設定レジスタ(IPRG)の設定 ==== */
        if(ch == 0){
            INTX.IPRG.BIT._EDMAC1 = 0x0;   /* E-DMAC0 の優先レベルを 0 に設定 */
        }
        else{
            INTX.IPRG.BIT._EDMAC2 = 0x0;   /* E-DMAC1 の優先レベルを 0 に設定 */
        }
        return CLOSE_OK;
    }
    return CLOSE_NG;
}
```

3.6 サンプルプログラムリスト” ether.c”(4)

```
144  /*""FUNC COMMENT""*****************************************************************
145  * ID            :
146  * モジュール概要  : EtherC/E-DMAC リセット関数
147  *-------------------------------------------------------------------------------
148  * Include       :#include  "iodefine.h"
149  *-------------------------------------------------------------------------------
150  * 宣言          : static void lan_reg_reset(int ch)
151  *-------------------------------------------------------------------------------
152  * 機能          : E-DMAC,EtherC のソフトウエアリセットを行います。
153  *               :
154  *-------------------------------------------------------------------------------
155  * 引数          : int ch: I : チャネル番号(0 または 1)
156  *-------------------------------------------------------------------------------
157  * 戻り値        : なし
158  *               :
159  *-------------------------------------------------------------------------------
160  * 注意事項      :
161  *               :
162  *               :
163  *               :
164  *""FUNC COMMENT END""*************************************************************/
165  static void lan_reg_reset(int ch)
166  {
167      volatile int t = 200;   /* 約 1us ウエイトカウント @198MHz */
168
169      /* ==== E-DMAC モードレジスタ(EDMR)の設定 ==== */
170      if(ch ==0){
171          EDMAC0.EDMR.BIT.SWR =1;
172      }
173      else if(ch ==1){
174          EDMAC1.EDMR.BIT.SWR =1;
175      }
176      else{
177          /* DO NOTHING */
178      }
179
180      /* ==== Bφ で 64 サイクル(約 970ns@Bφ=66MHz)待つ必要あり。マニュアル 19-4 参照 ==== */
181      while(--t){
182          /* wait */
183      }
184  }
185
```

3.7 サンプルプログラムリスト” ether.c”(5)

```
/*""FUNC COMMENT""*****************************************************************
* ID            :
* モジュール概要  : EtherC/E-DMAC レジスタの初期設定
*-------------------------------------------------------------------------------
* Include       : #include  "iodefine.h"
*               : #include  "ether.h"
*               : #include  "phy.h"
*-------------------------------------------------------------------------------
* 宣言          : static void lan_reg_set(int ch, int link)
*-------------------------------------------------------------------------------
* 機能          : E-DMAC,EtherC レジスタの設定を行います。
*               : 受信時に CAMSEN0,1 端子からの信号を参照しないように初期値を変更しています。
*-------------------------------------------------------------------------------
* 引数          : int ch:  I : チャネル番号(0 または 1)
*               : int link: I : PHY 自動交渉結果
*               :             HALF_10M(1), FULL_10M(2), HALF_TX(3), FULL_TX(4)
*               :
*-------------------------------------------------------------------------------
* 戻り値        :  なし
*               :
*-------------------------------------------------------------------------------
* 注意事項      : 外部 CAM を使用しないことを前提としています。
*               :
*               :
*               :
*""FUNC COMMENT END""*************************************************************/
static void lan_reg_set(int ch,int link)
{
    if(ch ==0){
        /* ==== EtherC ステータスレジスタ(ECSR)のクリア ==== */
        MAC0.ECSR.LONG = 0x00000007; /* 1 ライトクリア */

        /* ==== EtherC/E-DMAC ステータスレジスタ(EESR)のクリア ==== */
        EDMAC0.EESR.LONG = 0x47FF0F9F;   /* 1 ライトクリア */

        /* ==== 送信ディスクリプタ先頭アドレスレジスタ(TDLAR)の設定 ==== */
        EDMAC0.TDLAR = descriptor0.send_desc;

        /* ==== 受ディスクリプタ先頭アドレスレジスタ(RDLAR)の設定 ==== */
        EDMAC0.RDLAR = descriptor0.recv_desc;

        /* ==== 送受信ステータスコピー指示レジスタ(TRSCER)の設定 ==== */
        EDMAC0.TRSCER.LONG = 0x00000000;

        /* ==== 送信 FIFO しきい値指定レジスタ(TFTR)の設定 ==== */
        EDMAC0.TFTR = 0x00000000;   /* ストア&フォワードモード */

        /* ==== FIFO 容量指定レジスタ(FDR)の設定 ==== */
        EDMAC0.FDR.LONG = 0x00000707;   /* 送受信 FIFO 容量を 2KB に設定 */

        /* ==== 受信方式制御レジスタ(RMCR)の設定 ==== */
        EDMAC0.RCR.BIT.RNC = 0x1;   /* 連続受信 */
```

3.8 サンプルプログラムリスト” ether.c”(6)

```
238         /* ==== MAC アドレス上位/下位設定レジスタ(MAHR,MALR)の設定 ==== */
239         MAC0.MAHR = MAC_ADDRESS_HIGH0;
240         MAC0.MALR = MAC_ADDRESS_LOW0;
241
242         /* ==== TSU_FWSLC の初期設定  ==== */
243         TSU.TSU_FWSLC.LONG = 0x00000000;    /* CAMSEN0,1 端子を使わないときは 0 に設定してください */
244
245         /* ==== 割り込み優先レベル設定レジスタ(IPRG)の設定 ==== */
246         INTX.IPRG.BIT._EDMAC1 = EDMAC0_PRIORITY;    /* E-DMAC0 の優先レベルを設定 */
247
248         /* ==== EtherC 割り込み許可レジスタ(ECSIPR)の設定 ==== */
249         MAC0.ECSIPR.LONG = 0x0000;          /* 全て不許可とします(リンク信号変化
250                                                Magic Packet 検出
251                                                不正キャリア検出) */
252
253         /* ==== EtherC/E-DMAC ステータス割り込み許可レジスタ(EESIPR)の設定 ==== */
254         EDMAC0.EESIPR.LONG = 0x0304009f;    /* 受信関連要因のみ許可 */
255                                             /* 但し受信ディスクリプタ枯渇と
256                                                受信 FIFO オーバフローを除きます */
257
258         /* ==== EtherC モードレジスタ(ECMR)の設定 ==== */
259         if(link == FULL_TX || link == FULL_10M){
260             MAC0.ECMR.BIT.DM =1;            /* 全二重方式 */
261         }
262         else {
263             MAC0.ECMR.BIT.DM =0;            /* 半二重方式 */
264         }
265         MAC0.ECMR.BIT.RE = 1;               /* 受信許可 */
266         MAC0.ECMR.BIT.TE = 1;               /* 送信許可 */
267
268         /* ==== E-DMAC 受信要求レジスタ(EDRRR)の設定 ==== */
269         EDMAC0.EDRRR.LONG = 0x00000001;     /* 受信可能状態 */
270     }
271     else if(ch ==1){
272         /* ==== EtherC ステータスレジスタ(ECSR)のクリア ==== */
273         MAC1.ECSR.LONG = 0x00000007;        /* 1 ライトクリア */
274
275         /* ==== EtherC/E-DMAC ステータスレジスタ(EESR)のクリア ==== */
276         EDMAC1.EESR.LONG = 0x47FF0F9F;      /* 1 ライトクリア */
277
278         /* ==== 送信ディスクリプタ先頭アドレスレジスタ(TDLAR)の設定 ==== */
279         EDMAC1.TDLAR = descriptor1.send_desc;
280
281         /* ==== 受ディスクリプタ先頭アドレスレジスタ(RDLAR)の設定 ==== */
282         EDMAC1.RDLAR = descriptor1.recv_desc;
283
284         /* ==== 送受信ステータスコピー指示レジスタ(TRSCER)の設定 ==== */
285         EDMAC1.TRSCER.LONG = 0x00000000;
286
```

3.9 サンプルプログラムリスト” ether.c”(7)

```
287          /* ==== 送信 FIFO しきい値指定レジスタ(TFTR)の設定 ==== */
288          EDMAC1.TFTR = 0x00000000;          /* ストア＆フォワードモード */
289
290          /* ==== FIFO 容量指定レジスタ(FDR)の設定 ==== */
291          EDMAC1.FDR.LONG = 0x00000707;      /* 送受信 FIFO 容量を 2KB に設定 */
292
293          /* ==== 受信方式制御レジスタ(RMCR)の設定 ==== */
294          EDMAC1.RCR.BIT.RNC = 0x1;          /* 連続受信 */
295
296          /* ==== MAC アドレス上位/下位設定レジスタ(MAHR,MALR)の設定 ==== */
297          MAC1.MAHR = MAC_ADDRESS_HIGH1;
298          MAC1.MALR = MAC_ADDRESS_LOW1;
299
300          /* ==== TSU_FWSLC の初期設定  ==== */
301          TSU.TSU_FWSLC.LONG = 0x00000000;  /* CAMSEN0,1 端子を使わないときは 0 に設定してください */
302
303          /* ==== 割り込み優先レベル設定レジスタ(IPRG)の設定 ==== */
304          INTX.IPRG.BIT._EDMAC2 = EDMAC1_PRIORITY;
305                                             /* E-DMAC1 の優先レベルを設定 */
306
307          /* ==== EtherC 割り込み許可レジスタ(ECSIPR)の設定 ==== */
308          MAC1.ECSIPR.LONG = 0x0000;         /* 全て不許可とします(リンク信号変化
309                                                Magic Packet 検出
310                                                不正キャリア検出) */
311
312          /* ==== EtherC/E-DMAC ステータス割り込み許可レジスタ(EESIPR)の設定 ==== */
313          EDMAC1.EESIPR.LONG = 0x0304009f;  /* 受信関連要因のみ許可 */
314                                             /* 但し受信ディスクリプタ枯渇と
315                                                受信 FIFO オーバフローを除く */
316
317          /* ==== EtherC モードレジスタ(ECMR)の設定 ==== */
318          if(link == FULL_TX || link == FULL_10M){
319              MAC1.ECMR.BIT.DM =1;           /* 全二重方式 */
320          }
321          else {
322              MAC1.ECMR.BIT.DM =0;           /* 半二重方式 */
323          }
324          MAC1.ECMR.BIT.RE = 1;              /* 受信許可 */
325          MAC1.ECMR.BIT.TE = 1;              /* 送信許可 */
326
327          /* ==== E-DMAC 受信要求レジスタ(EDRRR)の設定 ==== */
328          EDMAC1.EDRRR.LONG = 0x00000001;   /* 受信可能状態 */
329      }
330      else{
331          /* DO NOTHING */
332      }
333  }
```

3.10 サンプルプログラムリスト” ether.c”(8)

```
334  /*""FUNC COMMENT""************************************************************
335  * ID            :
336  * モジュール概要  : 送受信ディスクリプタとバッファの初期化
337  *-------------------------------------------------------------------------------
338  * Include       : #include  " ether.h"
339  *-------------------------------------------------------------------------------
340  * 宣言          : static void lan_desc_create(int ch)
341  *-------------------------------------------------------------------------------
342  * 機能          : 送受信ディスクリプタ及びバッファを初期化します。
343  *               : ディスクリプタをリング構造にします。
344  *-------------------------------------------------------------------------------
345  * 引数          : int ch: I :チャネル番号(0 または 1)
346  *               :
347  *-------------------------------------------------------------------------------
348  * 戻り値        :  なし
349  *               :
350  *-------------------------------------------------------------------------------
351  * 注意事項      :
352  *               :
353  *               :
354  *               :
355  *               :
356  *""FUNC COMMENT END""*********************************************************/
357  static void lan_desc_create(int ch)
358  {
359      int i;
360      EDMAC_SEND_DESC *psnd;
361      EDMAC_RECV_DESC *prcv;
362
363      if(ch ==0){
364          /* ==== 送受信ディスクリプタの 0 クリア ==== */
365          memset(&descriptor0,0x0,sizeof(descriptor0));
366
367          /* ==== 送信ディスクリプタ初期設定 ==== */
368              /* 省略 */
369
370          /* ==== 受信ディスクリプタ初期設定 ==== */
371          prcv = descriptor0.recv_desc;
372          for(i = 0; i<NUM_OF_RX_DESCRIPTOR ;i++){
373              prcv->rd0.BIT.RACT =0x1;    /* 受信ディスクリプタ有効 */
374              prcv->rd1.RBL = 0x05f0;
375              prcv->rd2.RBA = &buffer0.recv_buf[i][0];
376              prcv->pNext = prcv +1;
377              prcv++;
378          }
379          prcv--;
380          prcv->rd0.BIT.RDLE = 1;
381          prcv->pNext = descriptor0.recv_desc;
382
383          /* ==== 送受信バッファの 0 クリア ==== */
384          memset(&buffer0,0x0,sizeof(buffer0));
385
```

3.11 サンプルプログラムリスト” ether.c”(9)

```
386          /* ==== 送受信ディスクリプタへのポインタ初期設定 ==== */
387          psenddesc0 = descriptor0.send_desc;
388          precvdesc0 = descriptor0.recv_desc;
389      }
390      else if(ch ==1){
391          /* ==== 送受信ディスクリプタの０クリア ==== */
392          memset(&descriptor1,0x0,sizeof(descriptor1));
393
394          /* ==== 送信ディスクリプタ初期設定 ==== */
395              /* 省略 */
396
397          /* ==== 受信ディスクリプタ初期設定 ==== */
398          prcv = descriptor1.recv_desc;
399          for(i = 0; i<NUM_OF_RX_DESCRIPTOR ;i++){
400              prcv->rd0.BIT.RACT =0x1;    /* 受信ディスクリプタ有効 */
401              prcv->rd1.RBL = 0x05f0;
402              prcv->rd2.RBA = &buffer1.recv_buf[i][0];
403              prcv->pNext = prcv +1;
404              prcv++;
405          }
406          prcv--;
407          prcv->rd0.BIT.RDLE = 1;
408          prcv->pNext = descriptor1.recv_desc;
409
410          /* ==== 送受信バッファの０クリア ==== */
411          memset(&buffer1,0x0,sizeof(buffer1));
412
413          /* ==== 送受信ディスクリプタへのポインタ初期設定 ==== */
414          psenddesc1 = descriptor1.send_desc;
415          precvdesc1 = descriptor1.recv_desc;
416      }
417      else{
418          /* DO NOTHING */
419      }
420  }
```

3.12 サンプルプログラムリスト” ether.c”(10)

```
421  /*""FUNC COMMENT""*****************************************************************
422  * ID            :
423  * モジュール概要  : イーサネットフレーム受信関数
424  *-------------------------------------------------------------------------------
425  * Include       : #include  "iodefine.h"
426  *               : #include  "ether.h"
427  *-------------------------------------------------------------------------------
428  * 宣言           : int lan_recv(int ch, unsigned char *addr)
429  *-------------------------------------------------------------------------------
430  * 機能           : イーサネットフレームを 1 フレーム分だけ受信します。
431  *               : 受信するたびに”受信通知カウンタ”の値を 1 デクリメントします。
432  *               : 受信フレームが存在しなければ本関数内で待ちます。
433  *               : 受信フレームにエラーがないときは引数で指定したユーザバッファにコピーします。
434  *-------------------------------------------------------------------------------
435  * 引数           : int ch: I : チャネル番号(0 または 1)
436  *               : unsigned char *addr : O : 受信した Ether フレームのコピー先先頭アドレス
437  *               :
438  *-------------------------------------------------------------------------------
439  * 戻り値         : 受信したフレームのバイト数 : 受信成功時
440  *               :
441  *-------------------------------------------------------------------------------
442  * 注意事項       :
443  *               :
444  *""FUNC COMMENT END""*************************************************************/
445  int lan_recv(int ch, unsigned char *addr)
446  {
447      int i;
448      int dsize = 0;  /* 受信バイト数 */
449
450      if(ch == 0){
451
452          while(rx_flag0 == 0){   /* 受信通知カウンタが 0 */
453              ;/* wait */
454          }
455
456          /* ==== 排他制御のため割り込み禁止 ==== */
457          INTX.IPRG.BIT._EDMAC1 = 0x0;
458          /* ==== 受信通知カウンタのカウントダウン ==== */
459          rx_flag0--;
460
461          /* ==== 割り込み禁止の解除 ==== */
462          INTX.IPRG.BIT._EDMAC1 = EDMAC0_PRIORITY;
463
464          /* ==== 受信フレームエラーの確認 ==== */
465          if(precvdesc0->rd0.BIT.RFE == 0){   /* 受信フレームエラー未発生時 */
466              memcpy(addr,precvdesc0->rd2.RBA,precvdesc0->rd1.RDL);
467              dsize = precvdesc0->rd1.RDL;
468          }
469
```

3.13 サンプルプログラムリスト” ether.c”(11)

```
470          /* ==== 受信ディスクリプタの初期化 ==== */
471          precvdesc0->rd0.LONG &= 0x40000000; /* RDLE 以外は 0 クリア */
472          precvdesc0->rd0.BIT.RACT = 1;
473          precvdesc0->rd1.RDL = 0x0000;
474          precvdesc0 = precvdesc0->pNext;      /* ディスクリプタ管理用ポインタの更新 */;
475
476          /* ==== 受信起動 ==== */
477          /* マニュアル 19-36  19.4.1 E-DMAC 受信要求レジスタ (EDRRR) 使用上の注意事項 */
478          if(EDMAC0.EDRRR.BIT.RR == 0){
479              EDMAC0.EDRRR.BIT.RR = 1;
480          }
481          return dsize;
482      }
483      else if(ch == 1){
484          while(rx_flag1 == 0){               /* 受信通知カウンタが 0 */
485              ;/* wait */
486          }
487          /* ==== 排他制御のため割り込み禁止 ==== */
488          INTX.IPRG.BIT._EDMAC2 = 0x0;
489          /* ==== 受信通知カウンタのカウントダウン ==== */
490          rx_flag1--;
491
492          /* ==== 割り込み禁止の解除 ==== */
493          INTX.IPRG.BIT._EDMAC2 = EDMAC1_PRIORITY;
494
495          /* ==== 受信フレームエラーの確認 ==== */
496          if(precvdesc1->rd0.BIT.RFE == 0){   /* 受信フレームエラー未発生時 */
497
498              /* ==== ユーザバッファに受信データをコピー ==== */
499              memcpy(addr,precvdesc1->rd2.RBA,precvdesc1->rd1.RDL);
500
501              /* ==== 受信サイズ数をセット ==== */
502              dsize = precvdesc1->rd1.RDL;
503          }
504
505          /* ==== 受信ディスクリプタの初期化 ==== */
506          precvdesc1->rd0.LONG &= 0x40000000; /* RDLE 以外は 0 クリア */
507          precvdesc1->rd0.BIT.RACT = 1;
508          precvdesc1->rd1.RDL = 0x0000;
509          precvdesc1 = precvdesc1->pNext;      /* ディスクリプタ管理用ポインタの更新 */;
510
511          /* ==== 受信起動 ==== */
512          /* マニュアル 19-36  19.4.1 E-DMAC 受信要求レジスタ (EDRRR) 使用上の注意事項 */
513          if(EDMAC1.EDRRR.BIT.RR == 0){
514              EDMAC1.EDRRR.BIT.RR = 1;
515          }
516
517          return dsize;
518      }
519      else{
520          /* DO NOTHING */
521      }
522  }
```

3.14 サンプルプログラムリスト” ether.c”(12)

```
523    /*""FUNC COMMENT""*****************************************************************
524    * ID            :
525    * モジュール概要  : ch0 用イーサネット受信割り込み処理関数
526    *-------------------------------------------------------------------------------
527    * Include      : #include  "iodefine.h"
528    *-------------------------------------------------------------------------------
529    * 宣言           : void INT_EDMAC_EINT0(void)
530    *-------------------------------------------------------------------------------
531    * 機能           : ch0 でイーサネットフレーム受信(エラーフレーム含む)が発生したとき、
532    *                : ”受信通知カウンタ”の値を 1 インクリメントします。
533    *-------------------------------------------------------------------------------
534    * 引数           : なし
535    *                :
536    *-------------------------------------------------------------------------------
537    * 戻り値         : なし
538    *                :
539    *-------------------------------------------------------------------------------
540    * 注意事項       :
541    *                :
542    *                :
543    *                :
544    *""FUNC COMMENT END""*************************************************************/
545    void INT_EDMAC_EINT0(void)
546    {
547        unsigned int status;
548
549        /* ==== 割り込みステータス保持 ==== */
550        status = EDMAC0.EESR.LONG & EDMAC0.EESIPR.LONG;
551
552        /* ==== 割り込み要因クリア ==== */
553        EDMAC0.EESR.LONG = status;  /* 1 ライトクリア */
554
555        /* ==== 受信関連割り込み時 ==== */
556        if(status & 0x0304009f){
557            rx_flag0++;
558        }
559                /* 上記 0x0304009f の内容 */
560                /* Bit25: 受信中断検出 */
561                /* Bit24: 受信フレームカウンタオーバフロー */
562                /* Bit18: フレーム受信 */
563                /* Bit7:  マルチキャストアドレスフレーム受信 */
564                /* Bit4:  端数ビットフレーム受信 */
565                /* Bit3:  ロングフレーム受信 */
566                /* Bit2:  ショートフレーム受信 */
567                /* Bit1:  PHY-LSI 受信エラー */
568                /* Bit0:  受信フレーム CRC エラー */
569
570    }
571    /*INT_EDMAC_EINT1(void)は省略 */
572    /* End of File */
573
```

3.15 サンプルプログラムリスト” ether.h”(1)

```
/*""FILE COMMENT""*************************************************************
*
*   System Name  : SH7712 Sample Program
*   FILE Name    : ether.h
*   Version      : 1.00.00
*   Contents     : ヘッダファイル
*   Model        : MS7712SE01
*   CPU          : SH7712
*   Compiler     : SHC9.1.1.0
*   OS           : None
*
*   note         : 送受信ディスクリプタ、送受信バッファ設定の参考プログラムです。
*                  <注意事項>
*                  本サンプルプログラムはすべて参考資料であり
*                  その動作を保証するものではありません。
*                  本サンプルプログラムはお客様のソフトウエア開発時の
*                  技術参考資料としてご利用ください。
*
*
*   Copyright  (C)  2007 Renesas Technology Corp. All Rights Reserved
*      AND Renesas Solutions Corp. All Rights Reserved
*
*   history      : 2007.11.07 ver 1.00.00
*""FILE COMMENT END""*********************************************************/
#ifndef _ETHER_H
#define _ETHER_H

/* ==== 外部参照プロトタイプ宣言 ==== */
int lan_open(int ch);
int lan_recv(int ch, unsigned char *addr);
int lan_close(int ch);

/* ==== マクロ定義 ==== */
#define NUM_OF_TX_DESCRIPTOR        4   /* 送信用ディスクリプタの個数 */
#define NUM_OF_RX_DESCRIPTOR        4   /* 受信用ディスクリプタの個数 */
#define SIZE_OF_TX_BUFFER        1520   /* 送信バッファサイズ 16 バイトの整数倍にすること */
#define SIZE_OF_RX_BUFFER        1520   /* 受信バッファサイズ 16 バイトの整数倍にすること */
#define MAC_ADDRESS_HIGH0   0x01234567   /* MAC アドレスが 01-23-45-67-89-AB(16 進数)の場合 */
#define MAC_ADDRESS_LOW0    0x000089AB
#define MAC_ADDRESS_HIGH1   0x01234567   /* MAC アドレスが 01-23-45-67-89-AA(16 進数)の場合 */
#define MAC_ADDRESS_LOW1    0x000089AA
#define EDMAC0_PRIORITY           0xf   /* E-DMAC0 の優先レベル */
#define EDMAC1_PRIORITY           0xf   /* E-DMAC1 の優先レベル */
#define TX_FLAG_ON  1
#define TX_FLAG_OFF 0

/* ==== lan_open()用戻り値の列挙定数定義 ==== */
typedef enum{OPEN_OK= 0,OPEN_NG= -1}OPEN_STATUS;

/* ==== lan_send()用戻り値の列挙定数定義 ==== */
typedef enum{SEND_OK = 0,SEND_NG =-1}SEND_STATUS;
```

3.16 サンプルプログラムリスト” ether.h”(2)

```
52    /* ==== lan_close()用戻り値の列挙定数定義 ==== */
53    typedef enum{CLOSE_OK= 0,CLOSE_NG= -1}CLOSE_STATUS;
54
55    /* ==== 送信ディスクリプタ構造体定義 ==== */
56    typedef union
57    {
58        unsigned long LONG;
59        struct{
60            unsigned int    TACT:1;         /* 送信ディスクリプタ有効ビット */
61            unsigned int    TDLE:1;         /* 送信最終ディスクリプタリスト最終 */
62            unsigned int    TFP:2;          /* 送信フレーム内位置 */
63            unsigned int    TFE:1;          /* 送信フレームエラー発生(エラーの要因は TFSx を参照) */
64            unsigned int    reserved1:11;   /* 未使用 */
65            unsigned int    reserved2:7;    /* 未使用 */
66            unsigned int    TFS8:1;         /* 送信アボート */
67            unsigned int    reserved3:4;    /* 未使用 */
68            unsigned int    TFS3:1;         /* 送信開始時キャリア未検出 */
69            unsigned int    TFS2:1;         /* 送信中キャリア消失 */
70            unsigned int    TFS1:1;         /* 遅延衝突発生 */
71            unsigned int    TFS0:1;         /* 送信リトライオーバ */
72        }BIT;
73    }TD0;
74    typedef struct
75    {
76        unsigned short  TDL;                /* 送信バッファのサイズ(バイト数) */
77        unsigned short  reserved;
78    }TD1;
79    typedef struct
80    {
81        unsigned char   *TBA;               /* 送信バッファのアドレス */
82    }TD2;
83
84    typedef struct tag_edmac_send_desc
85    {
86        TD0    td0;
87        TD1    td1;
88        TD2    td2;
89        struct  tag_edmac_send_desc    *pNext;
90    }EDMAC_SEND_DESC;
```

3.17 サンプルプログラムリスト” ether.h”(3)

```
91    /* ==== 受信ディスクリプタ構造体定義 ==== */
92    typedef union
93    {
94        unsigned long LONG;
95        struct{
96            unsigned int    RACT:1;         /* 受信ディスクリプタ有効 */
97            unsigned int    RDLE:1;         /* 最終ディスクリプタリスト最終 */
98            unsigned int    RFP:2;          /* 受診フレーム内位置 */
99            unsigned int    RFE:1;          /* 受信フレームエラー発生(エラーの要因は TFSx を参照) */
100           unsigned int    reserved1:3;     /* 未使用 */
101           unsigned int    reserved2:8;     /* 未使用 */
102           unsigned int    reserved3:6;     /* 未使用 */
103           unsigned int    RFS9:1;         /* 受信 FIFO オーバフロー */
104           unsigned int    RFS8:1;         /* 受信中アボート検出 */
105           unsigned int    RFS7:1;         /* マルチキャストアドレスフレーム受診 */
106           unsigned int    reserved4:2;     /* 未使用 */
107           unsigned int    RFS4:1;         /* 端数ビットフレームエラー */
108           unsigned int    RFS3:1;         /* ロングフレーム受信エラー */
109           unsigned int    RFS2:1;         /* ショートフレーム受信エラー */
110           unsigned int    RFS1:1;         /* PHY-LSI 受信エラー */
111           unsigned int    RFS0:1;         /* 受信フレーム CRC エラー */
112       }BIT;
113   }RD0;
114   typedef struct
115   {
116       unsigned short  RBL;                /* 受信バッファデータ長(単位：バイト、16 バイト境界で指定) */
117       unsigned short  RDL;                /* 受信データ長(フレームの最後を受信したときに設定) */
118   }RD1;
119   typedef struct
120   {
121       unsigned char   *RBA;               /* 受信バッファの開始アドレス、SDRAM 時 16 バイト境界 */
122   }RD2;
123
124   typedef struct tag_edmac_recv_desc
125   {
126       RD0    rd0;
127       RD1    rd1;
128       RD2    rd2;
129       struct  tag_edmac_recv_desc  *pNext;
130   }EDMAC_RECV_DESC;
```

3.18 サンプルプログラムリスト” ether.h”(4)

```
131   /* ==== 送受信バッファ構造体定義 ==== */
132   typedef struct
133   {
134       /* 送信バッファ領域(16 バイト境界でアラインされていること) */
135       unsigned char   send_buf[NUM_OF_TX_DESCRIPTOR][SIZE_OF_TX_BUFFER];
136
137       /* 受信バッファ領域(16 バイト境界でアラインされていること) */
138       unsigned char   recv_buf[NUM_OF_RX_DESCRIPTOR][SIZE_OF_RX_BUFFER];
139   }TXRX_BUFFER_SET;
140
141   /* ==== 送受信ディスクリプタ構造体定義 ==== */
142   typedef struct
143   {
144       /* 送信ディスクリプタ(16 バイト境界でアラインされていること) */
145       EDMAC_SEND_DESC    send_desc[NUM_OF_TX_DESCRIPTOR];
146
147       /* 受信ディスクリプタ(16 バイト境界でアラインされていること) */
148       EDMAC_RECV_DESC    recv_desc[NUM_OF_RX_DESCRIPTOR];
149   }TXRX_DESCRIPTOR_SET;
150
151   #endif
152
153   /* End of File */
```

4. 参考ドキュメント

・ ソフトウェアマニュアル
SH3、SH3E、SH3-DSP ソフトウェアマニュアル　Rev5.00
(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
SH7710 グループハードウェアマニュアル　Rev.2.00
SH7712 ハードウェアマニュアル　Rev.1.00
SH7713 ハードウェアマニュアル　Rev.1.00
(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)。

5. 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/

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2007.11.07	—	初版発行

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しては、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が、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は負いません。
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（含むハードウエアおよびソフトウエア）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エアは、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。