

## V850E2/ML4

R01AN1018JJ0100

Rev.1.00

2012.06.14

## イーサネット送受信設定例

## 要旨

本アプリケーションノートでは、イーサネット・コントローラを使用した、イーサネット / IEEE802.3 フレームの送受信設定例を説明します。

## 動作確認デバイス

- V850E2/ML4 グループ (型名 uPD70F4022)
- Standard Microsystems Corporation 製 LAN8700i

## ターゲットボード

- V850E2/ML4 CPU ボード (型番 : R0K0F4022C000BR)

## 注意事項

- V850E2/ML4 は製品毎に内蔵 ROM および内蔵 RAM の容量が異なります。使用する製品に合わせて表 5.1 のセクション配置を修正ください。
- 参考プログラムは通信方式の選択に自動交渉機能を用いています。V850E2/ML4 と接続相手の自動交渉完了までの時間差が大きい場合、自動交渉が成功していても通信に失敗することがあります。「4.7.1 動作環境の注意点 1」に従い調整してください。

## 目次

1. はじめに .....	2
2. 初期設定の説明 .....	3
3. PHY-LSI 自動交渉設定の説明 .....	4
4. 送受信設定の説明 .....	15
5. 参考プログラムのセクション配置 .....	48
6. 参考ドキュメント .....	48

## 1. はじめに

### 1.1 仕様

- 参考プログラムは送受信エラー処理が実装されておりません。エラー処理が必要となる場合は別途お客様側でプログラムを作成してください。
- リセット解除後に、Hバスの設定、I/Oレジスタの設定、タイマの設定を行います。
- イーサネットPHY-LSIはStandard Microsystems CorporationのLAN8700iを使用しています。
- イーサネットPHY-LSIのリンクには自動交渉機能を用います。
- 参考プログラムでは次の2種類の処理を選択可能です。
  - イーサネットフレームを10フレーム送信
  - イーサネットフレームを10フレーム受信

### 1.2 使用機能

- Hバス
- I/Oポート
- タイマ
- イーサネット・コントローラ
- 割り込み

### 1.3 適用条件

- マイコン: V850E2/ML4
- 評価ボード: V850E2/ML4CPUボード(型番: R0K0F4022C000BR)
- 動作周波数:
  - 入力クロック: 10MHz
  - 内部システムクロック(fCLK): 200MHz
  - MII送信クロック(fMIITX): 25MHz
  - MII受信クロック(fMIIRX): 25MHz
  - Hバスクロック(fHCLK)33.3MHz
- 動作モード:
  - 通常動作モード
- 統合開発環境: (注1)
  - ルネサスエレクトロニクス製 CubeSuite+ V1.01.00 [20 Oct 2011]
    - Cコンパイラ: ルネサスエレクトロニクス製 CX V1.21
  - Green Hills社製 MULTI V.5.1.7D
- コンパイルオプション:
  - デフォルトでのコンパイルオプション
  - cpu=v850 -output=obj="\$(CONFIGDIR)¥\$(FILELEAF).obj" -nologo

【注】 1 本アプリケーションノートのサンプルコード「workspace」ディレクトリには、CubeSuite+に対応したプロジェクトと、MULTIに対応したプロジェクトを同梱しています。イーサネットドライバのソースコード及び以降の説明は共通です。

## 2. 初期設定の説明

参考プログラムのイーサネットドライバを使用するためには、パワーオンリセット後に、Hバスの初期化などハードウェアの初期化を行う初期設定プログラムが必要です。ここでは参考プログラムにおける設定例を説明します。

### 2.1 初期設定プログラムの説明

リセット解除後、メインプログラムの最初に、Hバスの設定、イーサネットで使用するポートの設定、タイマ (TAUA0) の設定、割り込み許可設定を行います。図 2.1 に初期化処理のフローを示します。

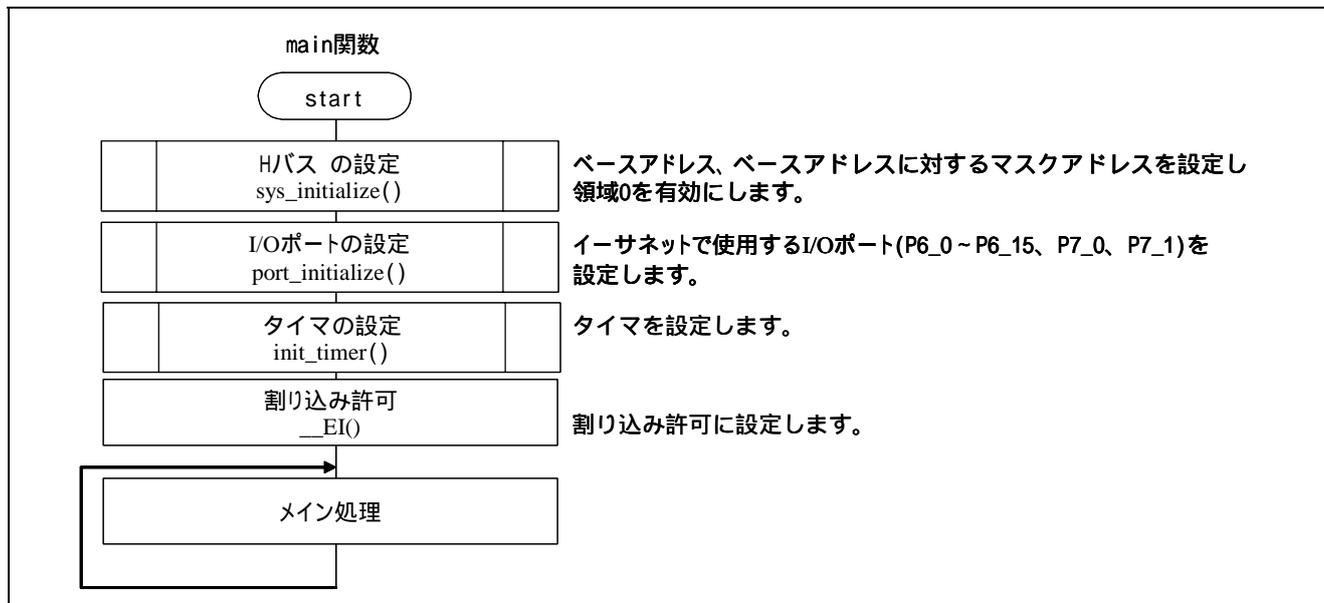


図 2.1 リセット後の初期設定フロー

### 2.2 初期設定の設定内容

表 2.1 に参考プログラムの設定を示します。

表 2.1 参考プログラムの設定

モジュール	設定内容
動作モード	通常動作モード
Hバスの設定	領域 0 を有効に設定
I/O ポート	イーサネットで使用する端子の設定 ( P6_0 ~ P6_15、 P7_0、 P7_1 )
タイマの設定	TAUA0 を 1ms 周期割り込みに設定

### 3. PHY-LSI 自動交渉設定の説明

参考プログラムでは、イーサネット PHY-LSI により自動交渉を行います。自動交渉結果は、ポート端子リード・レジスタ (PPR7) から読み出します。

#### 3.1 使用機能の動作概要

物理層のリンク処理はイーサネット PHY-LSI の機能で実行されます。そのため V850E2/ML4 内蔵のイーサネット・コントローラは、イーサネット PHY-LSI から結果を読み出すだけでリンク結果を取得できます。参考プログラムでは PHY-LSI の自動交渉機能を有効にしています。イーサネット PHY-LSI の詳細な機能については、イーサネット PHY-LSI のデータシートを参照ください。

イーサネット・コントローラとイーサネット PHY-LSI 間のインタフェースは、IEEE802.3 規格の MII (Media Independent Interface) および RMII (Reduced Media Independent Interface) で規格化されています。

図 3.1 V850E2/ML4 と LAN8700i の接続例を示します。

自動交渉結果はイーサネット PHY-LSI 内部のレジスタに格納され、MDC 端子と MDIO 端子によるシリアルインタフェース (Serial Management Interface) を使用して読み出します。V850E2/ML4 では、ポート制御により両端子のリード/ライトが可能です。PHY-LSI 内部のレジスタアクセス手順は「3.2 MII/RMII レジスタのアクセス手順」を参照ください。

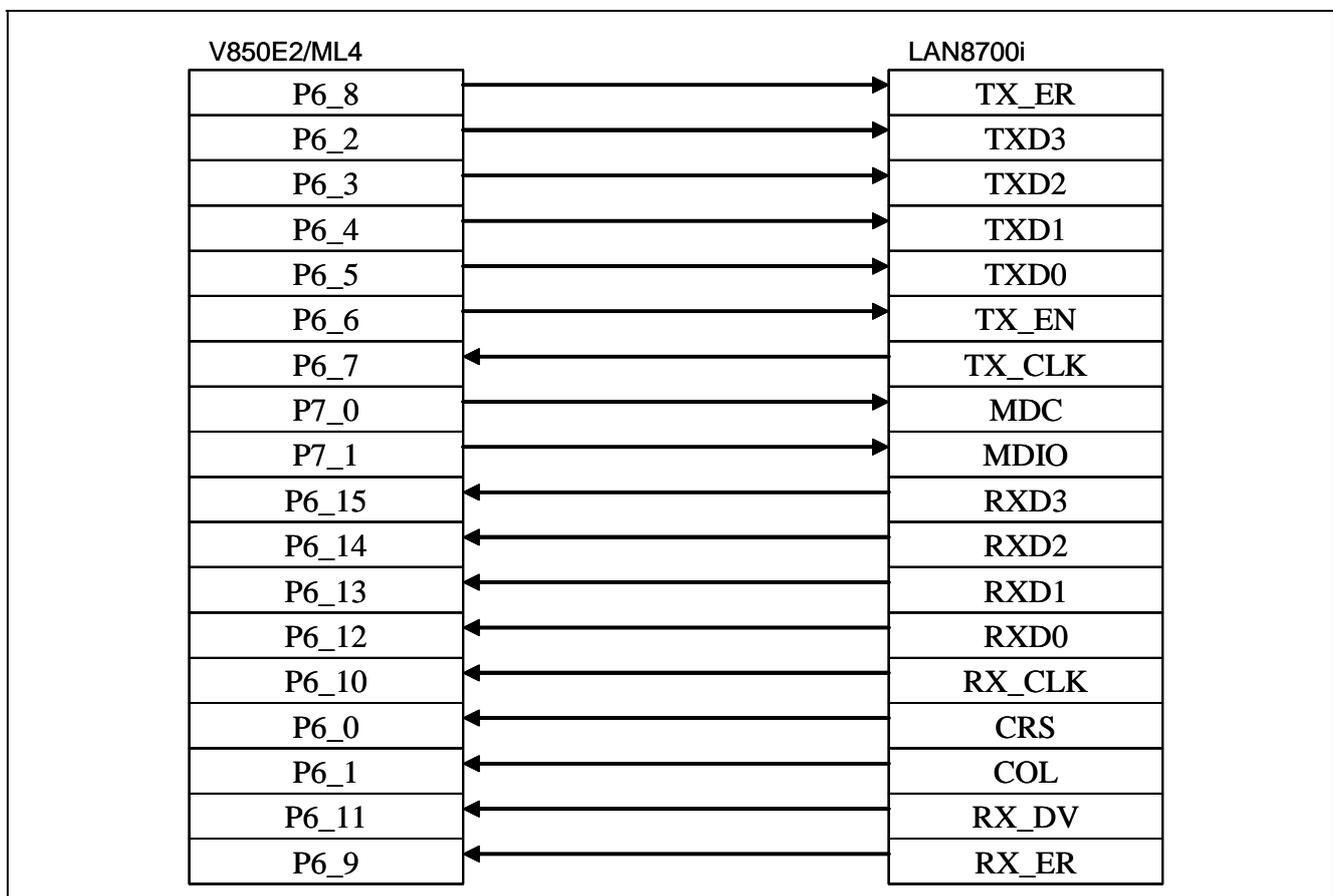


図 3.1 LAN8700i との接続例 (MII)

### 3.2 MII/RMII レジスタのアクセス手順

イーサネット PHY-LSI の内部レジスタである MII/RMII レジスタをアクセスする手順を説明します。

MII/RMII レジスタのアクセスに使用するシリアルインタフェース (Serial Management Interface) は、MDC と MDIO (いずれもイーサネット・コントローラ側の端子名) の 2 本からなり、MDC が同期用クロック端子、MDIO がデータ入出力端子です。各端子の状態はポート制御により参照または変更できます。MII では必ず規定されたフォーマット (MII/RMII 管理フレーム) 通りにデータを出力しなければなりません。MII 管理フレームを図 3.2 に示します。参考プログラムでは IDLE 状態で Z0 出力を 1 ビット実行します。IEEE802.3 規格ではクロック入力について言及されていませんが、PHY-LSI によっては正しく接続できない場合があり、安全のため実行しています。

MII/RMII 管理フレームの入出力は、PRE から順に 1 ビット単位で行います。1 ビット単位の入出力フローを図 3.3 ~ 図 3.6 に示します。MDC および MDIO の入出力タイミングは IEEE802.3 規格に準拠してください。表 3.1 および図 3.7 に IEEE802.3 規格の入出力タイミングを示します。

アクセス種別	MII/RMII 管理フレーム							
	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
ビット数	32	2	2	5	5	2	16	—
リード	1..1	01	10	00001	RRRRR	Z0	D..D	—
ライト	1..1	01	01	00001	RRRRR	10	D..D	X

#### 【記号説明】

- PRE : 32個の連続した1  
 ST : フレームの先頭を示す01のライト  
 OP : アクセス種別を示すコードのライト  
 PHYAD : PHY-LSIのアドレスが1番の場合、00001をライト (MSBから順次ライト)。このビットは、PHY-LSIアドレスによって可変となる。  
 REGAD : レジスタアドレスが1番の場合、00001をライト (MSBから順次ライト)。このビットは、PHY-LSIのレジスタアドレスによって可変となる。  
 TA : MII/RMIIインタフェース上でデータの送信元を切り替える時間  
 (a) ライト時は、10をライト  
 (b) リード時は、「バス解放」 (Z0と表記) を行う  
 DATA : 16ビットのデータ。MSBから順次ライトあるいはリード  
 (a) ライト時は、16ビットデータのライト  
 (b) リード時は、16ビットデータのリード  
 IDLE : 次のMII管理フォーマット入力までの待機時間  
 (a) ライト時は、「単独バス解放」 (Xと表記) を行う  
 (b) リード時は、すでにTA時にバス解放済みであり制御不要

図 3.2 MII/RMII 管理フレーム・フォーマット

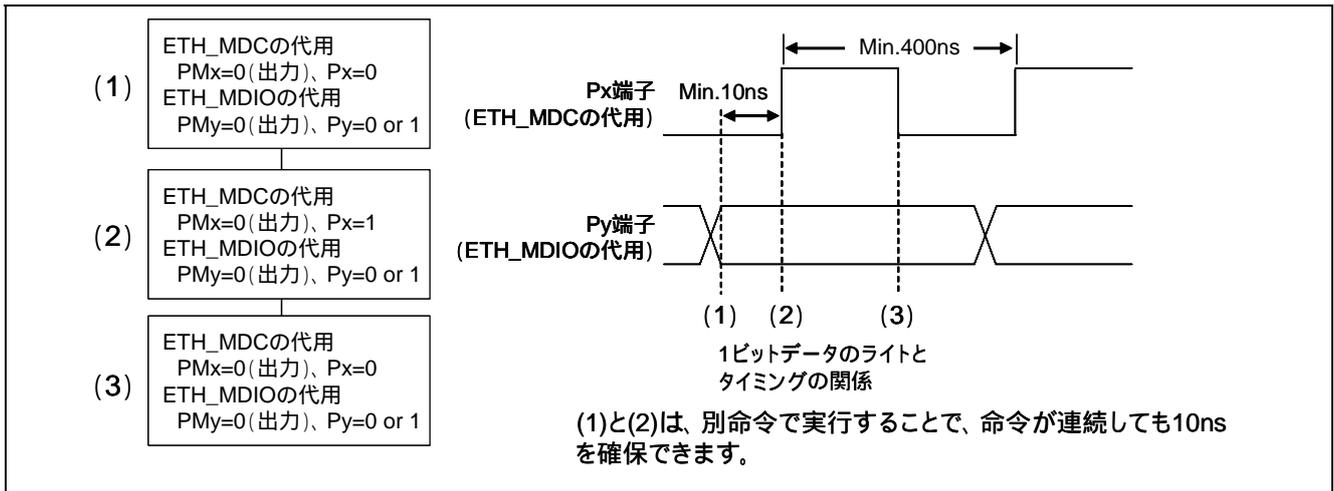


図 3.3 1 ビットデータのライトフロー

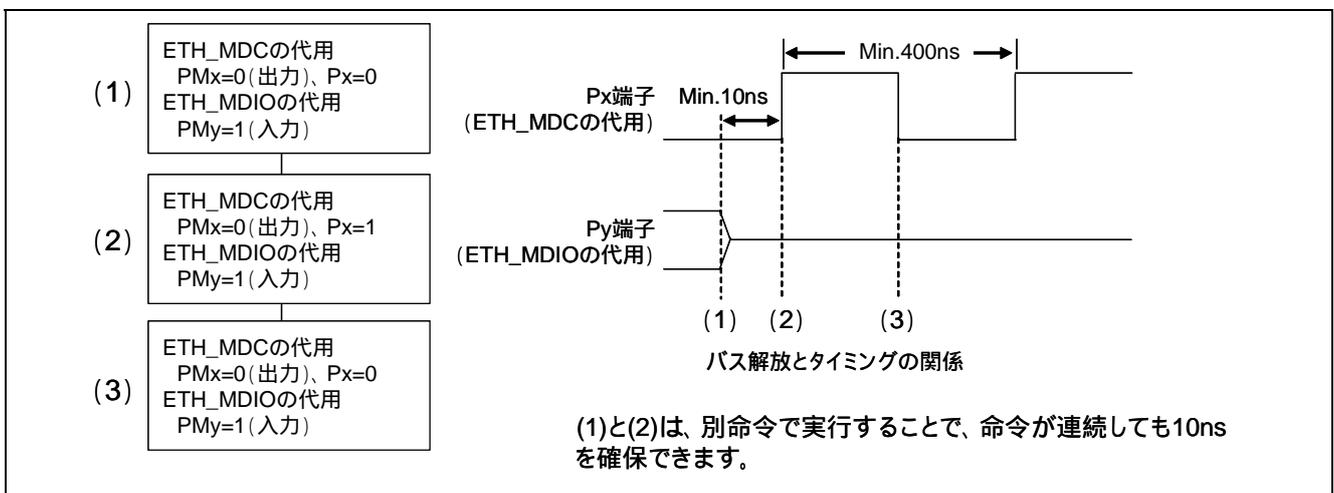


図 3.4 バス解放フロー（リード時の TA）

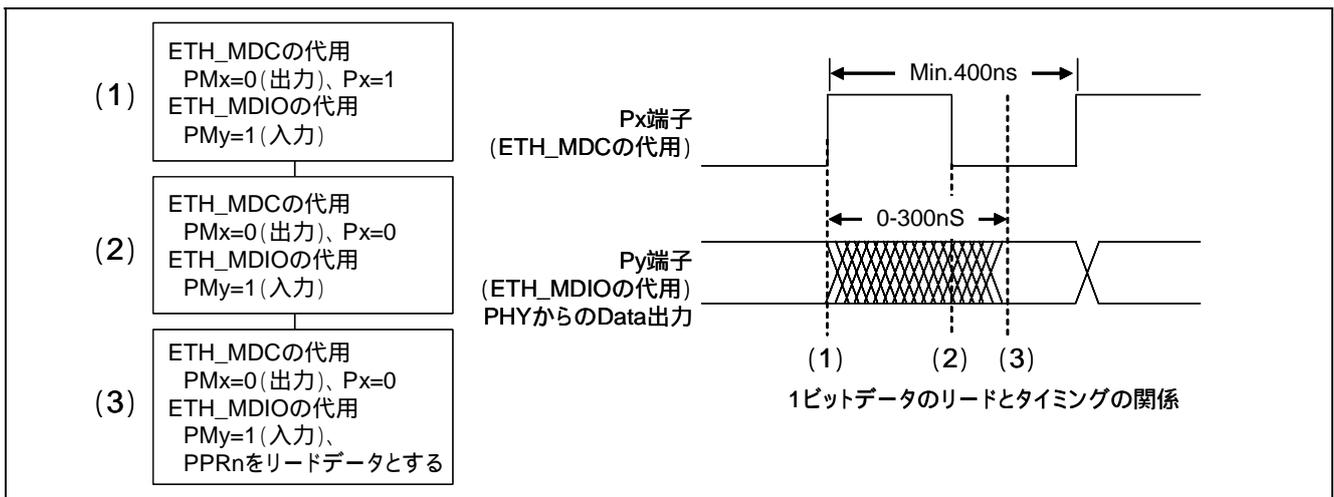


図 3.5 1 ビットデータのリードフロー

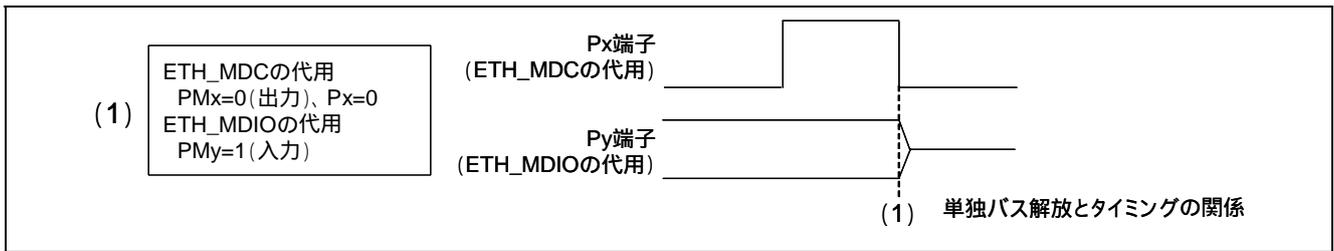


図 3.6 単独バス解放フロー（ライト時の IDLE）

表 3.1 MDC/MDIO 入出力タイミング

項目	記号	Min	Max	単位
MDC High レベルパルス幅	$t_1$	160		ns
MDC Low レベルパルス幅	$t_2$	160		ns
MDC サイクル時間	$t_3$	400		ns
MDIO セットアップ時間	$t_4$	10		ns
MDIO ホールド時間	$t_5$	10		ns
MDIO 出力遅延時間	$t_6$	0	300	ns

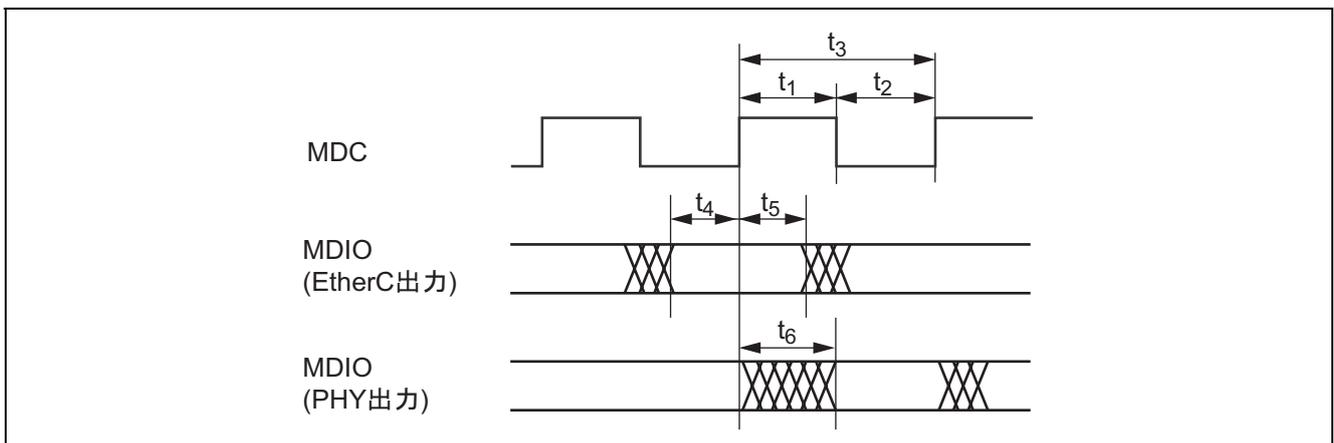


図 3.7 MDC/MDIO 入出力タイミング

### 3.3 PHY-LSI 自動交渉設定の説明

- ether\_phy.c

PHY-LSI のリセット関数（`phy_reset` 関数）と PHY-LSI 初期化関数（`phy_initialize` 関数）が記述されています。図 3.8 に `phy_reset` 関数と図 3.9 に `phy_initialize` 関数のフローを示します。

また、図 3.10～図 3.16 に、これらの関数内から呼び出される処理のフローを示します。

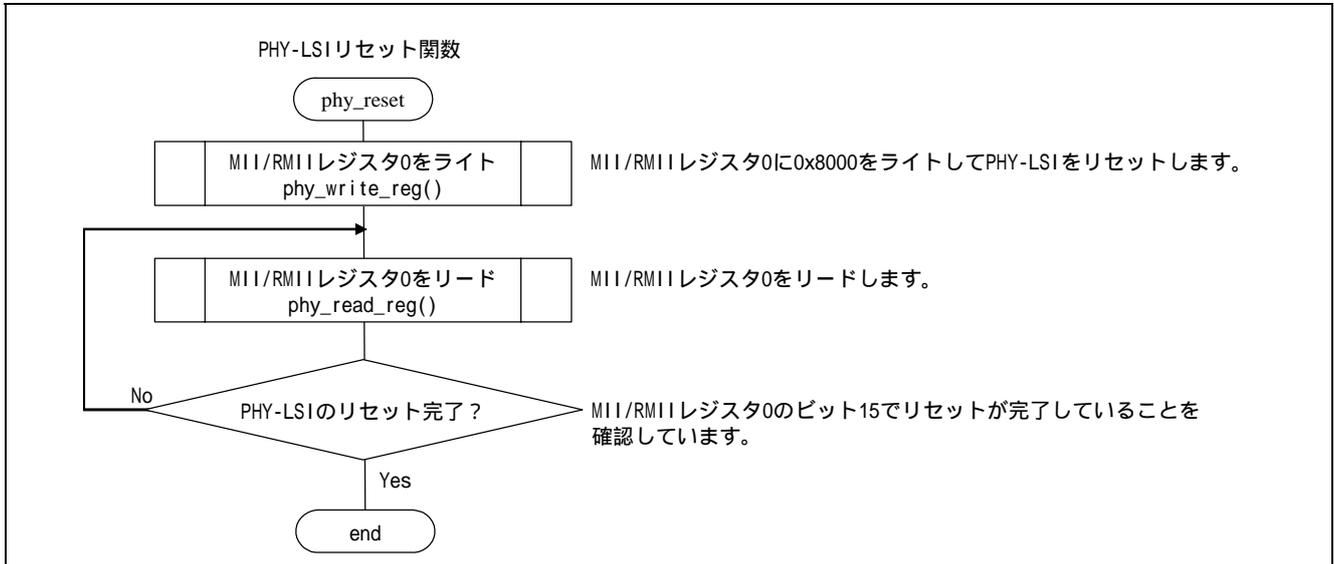


図 3.8 PHY-LSI リセット関数の処理フロー

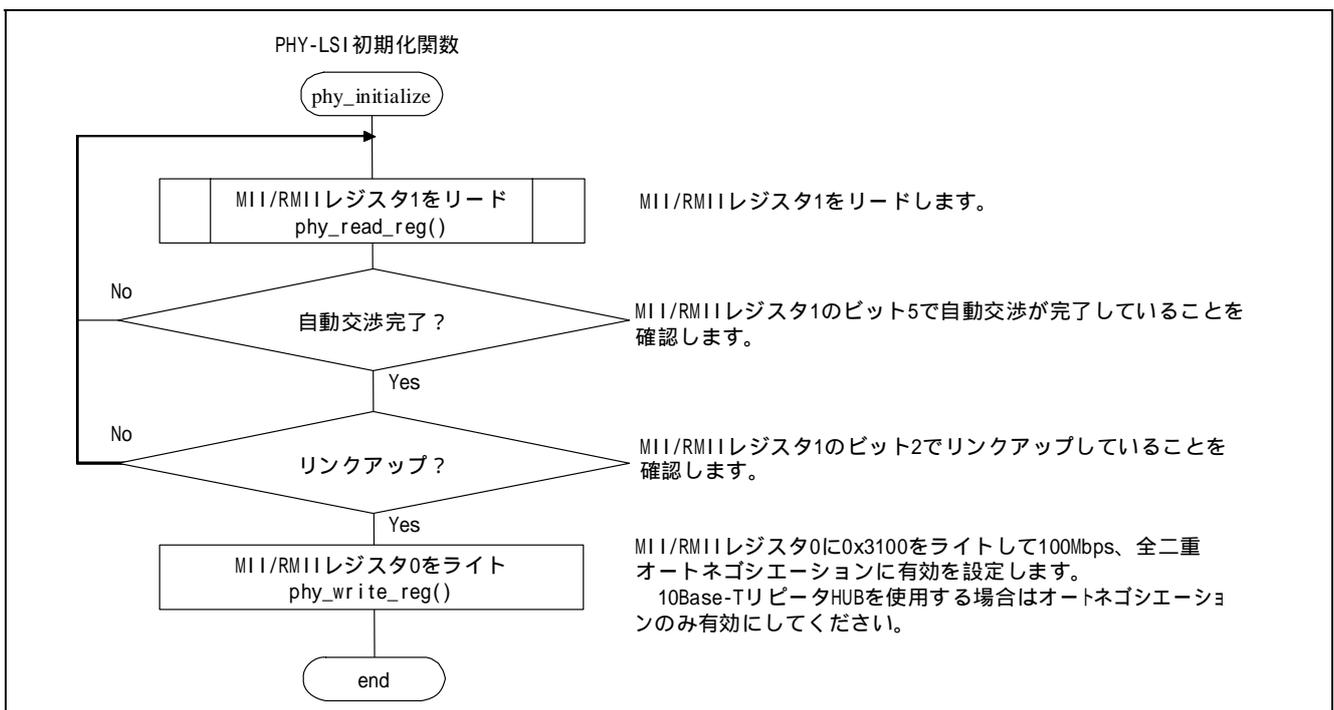


図 3.9 PHY-LSI 初期化関数の処理フロー

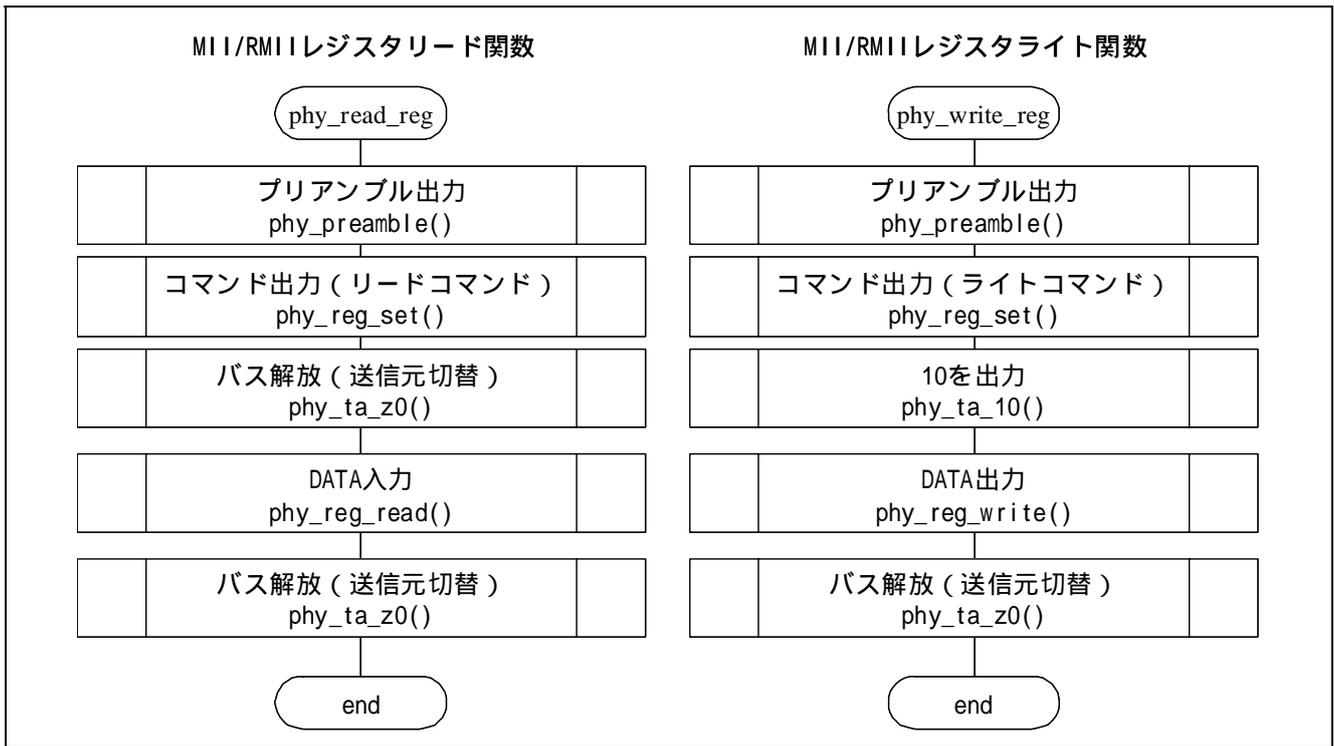


図 3.10 MII/RMII レジスタアクセスの処理フロー(1)

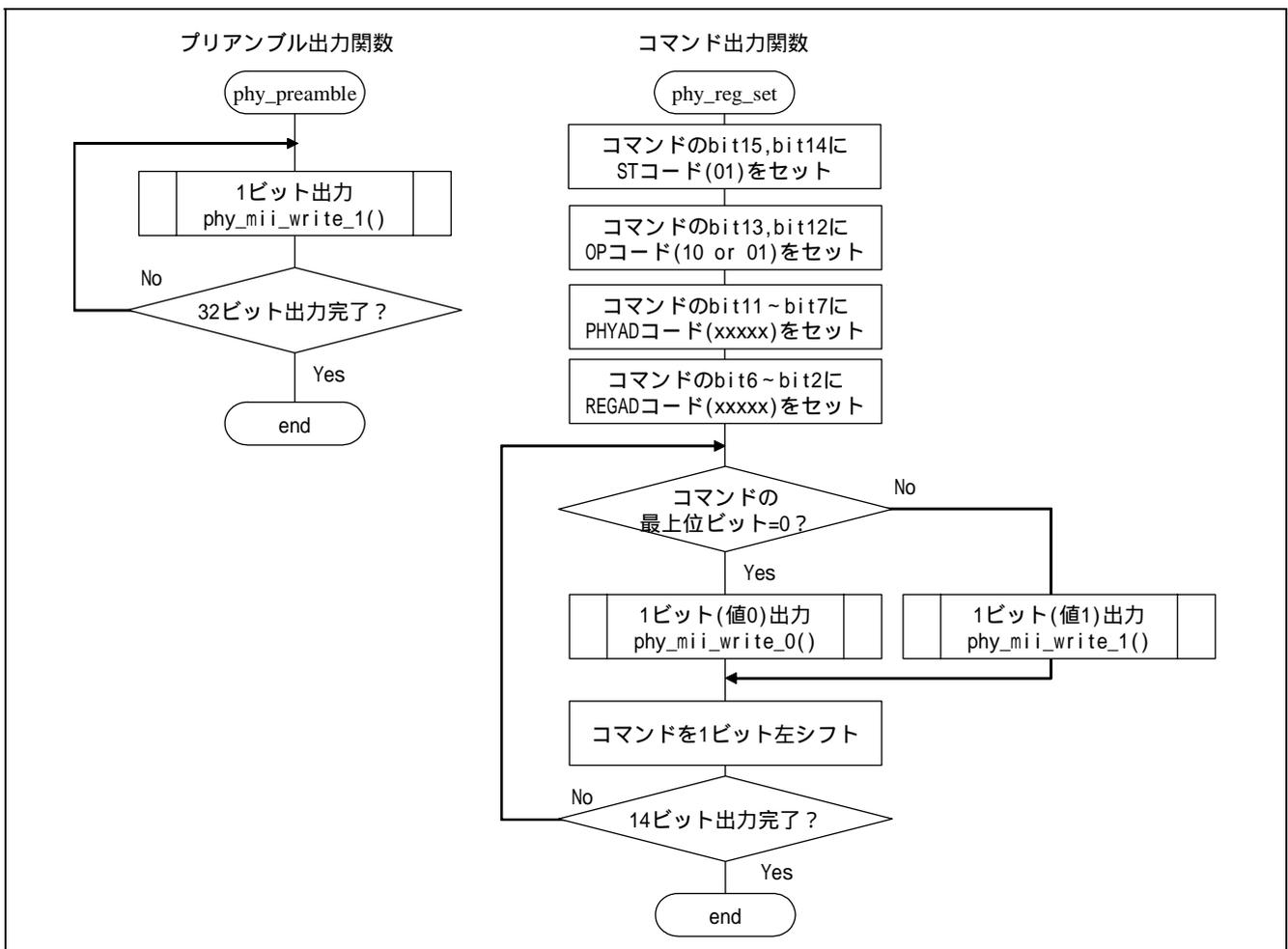


図 3.11 MII/RMII レジスタアクセスの処理フロー(2)

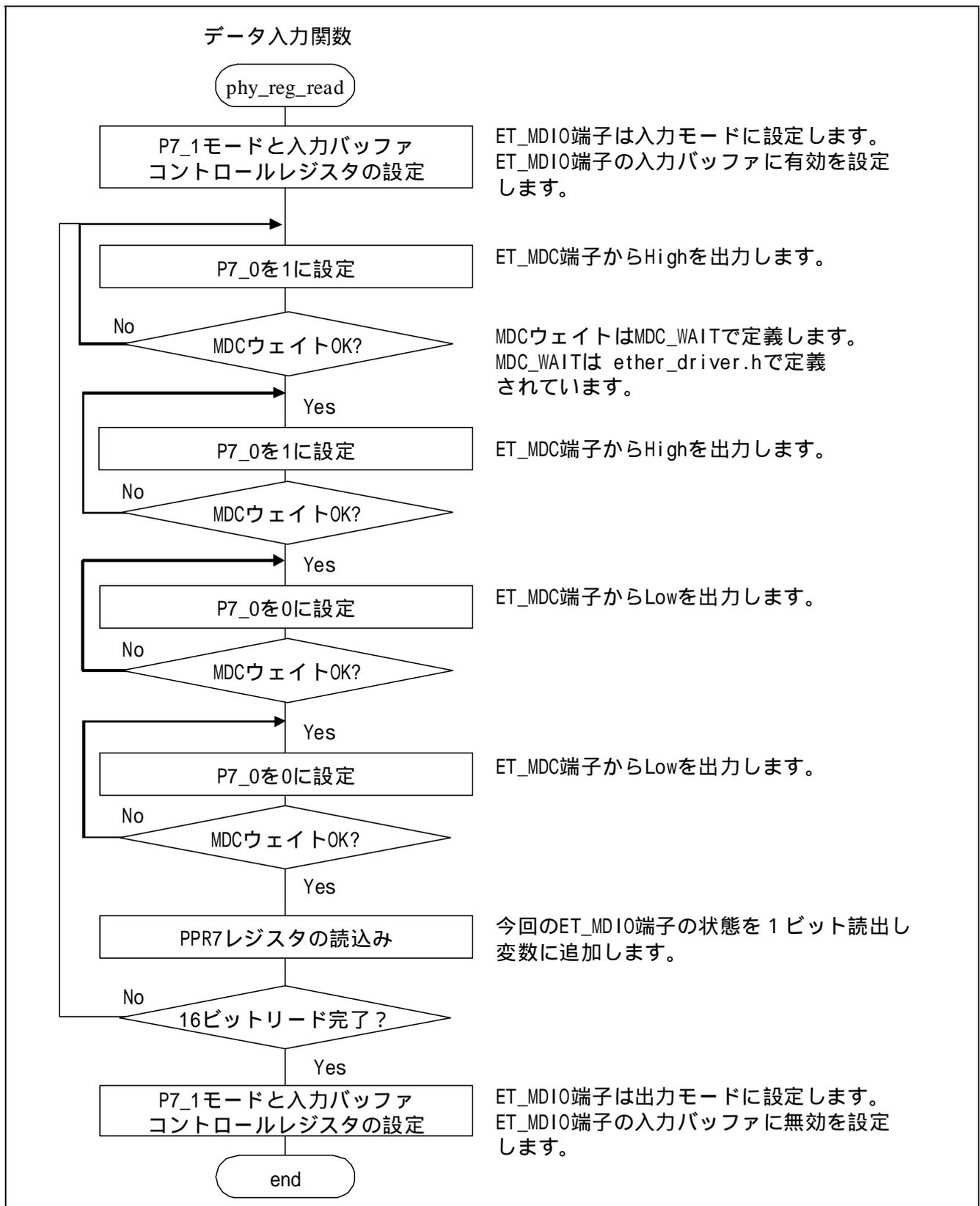


図 3.12 MII/RMII レジスタアクセスの処理フロー(3)

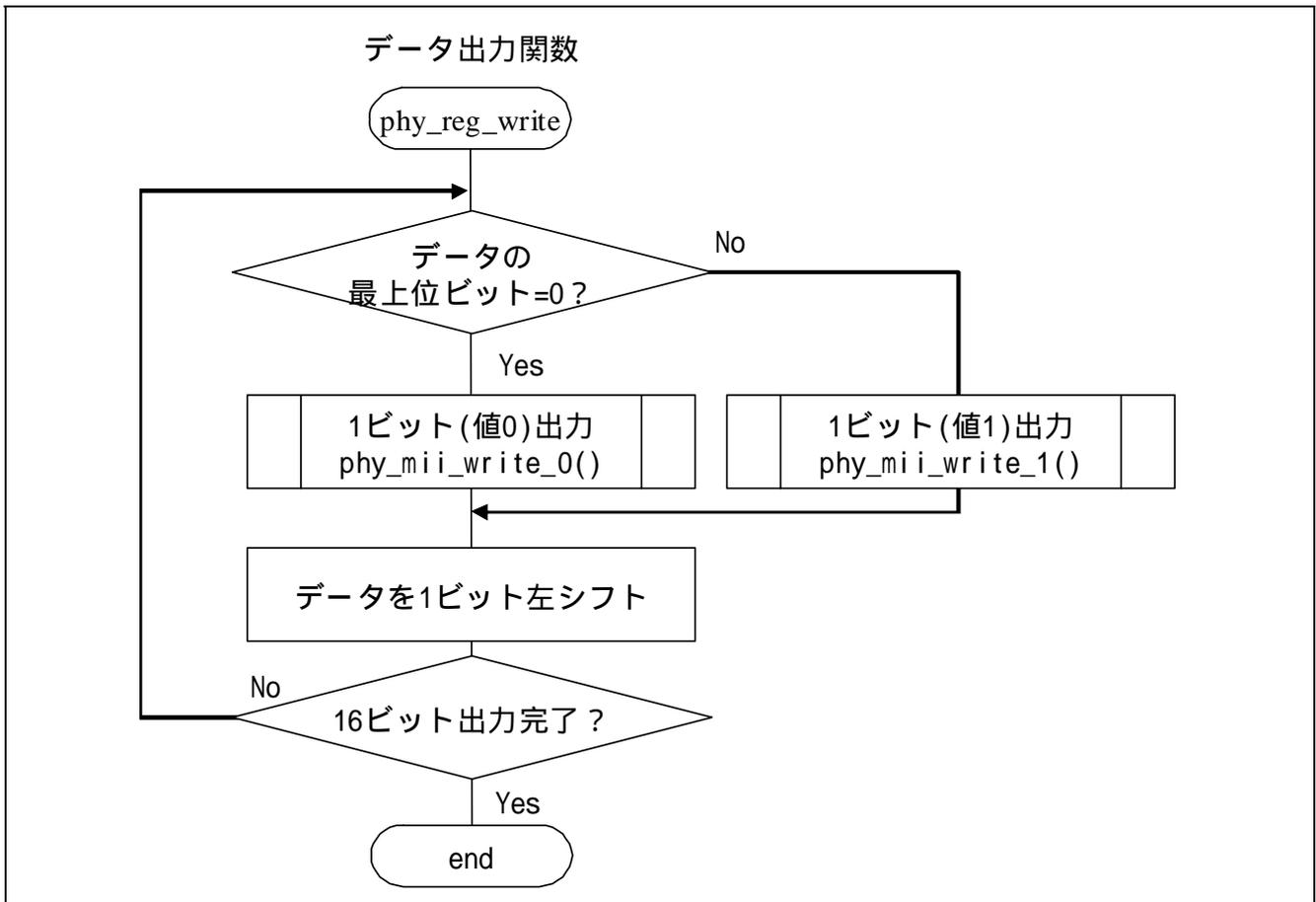


図 3.13 MII/RMII レジスタアクセスの処理フロー(4)

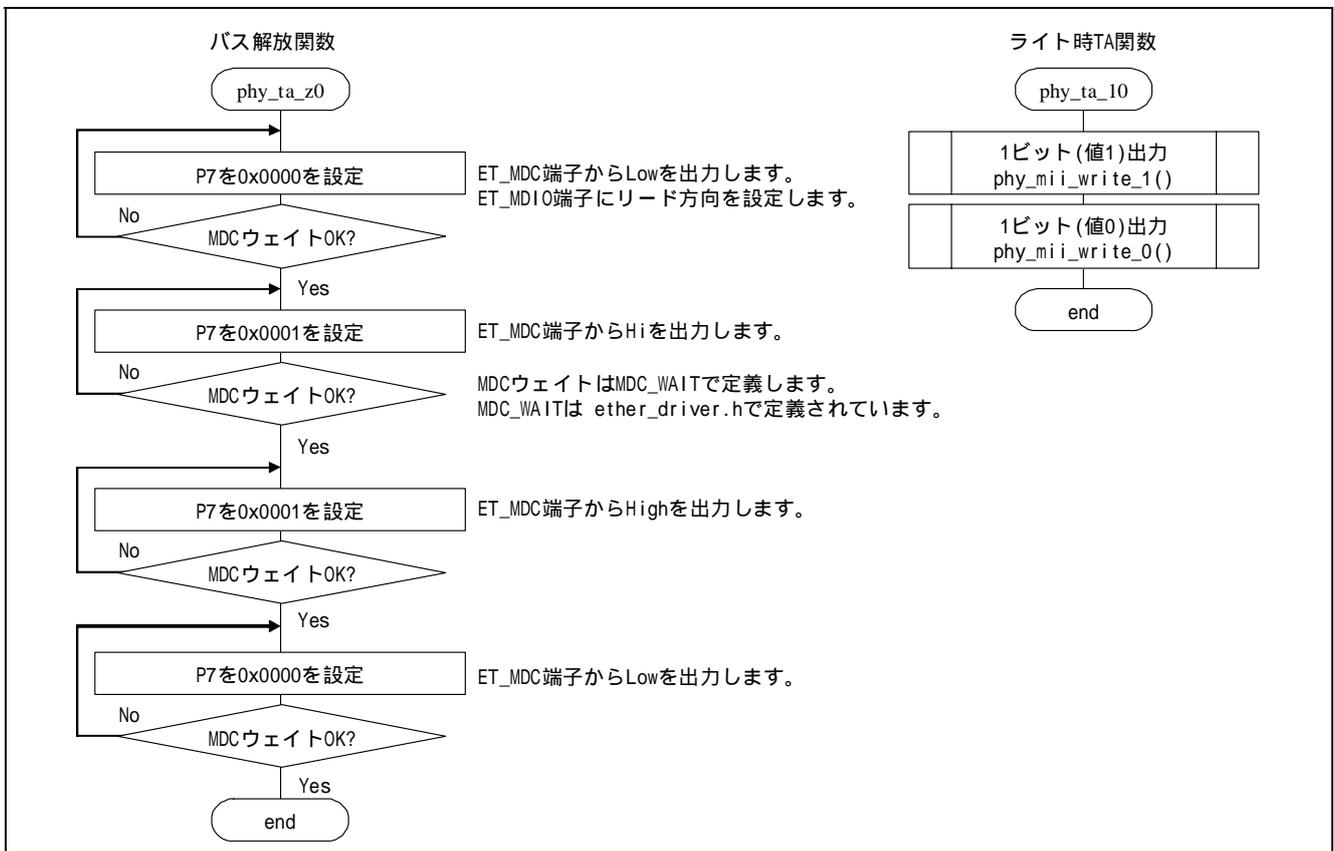


図 3.14 MII/RMII レジスタアクセスの処理フロー(5)

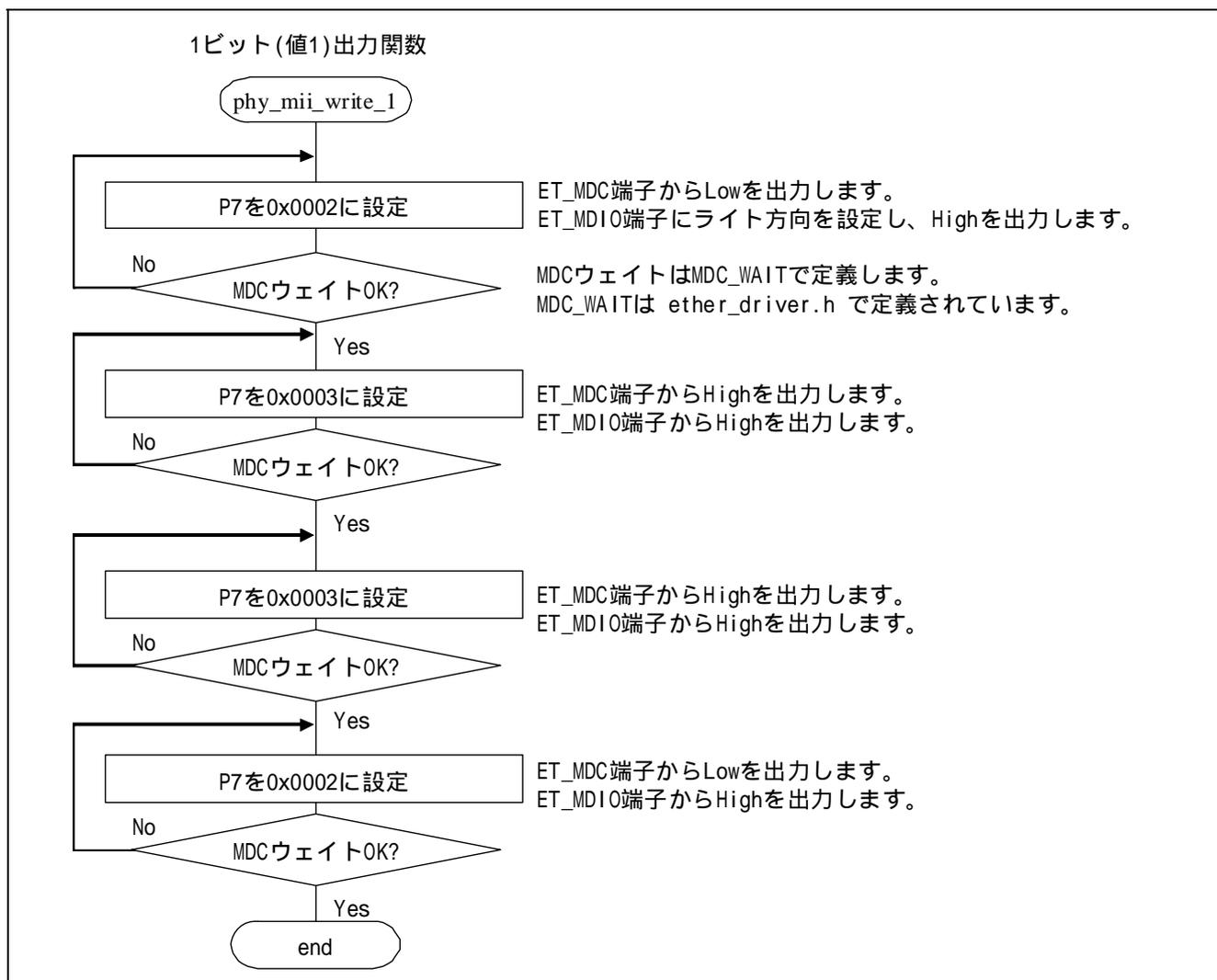


図 3.15 MII/RMII レジスタアクセスの処理フロー(6)

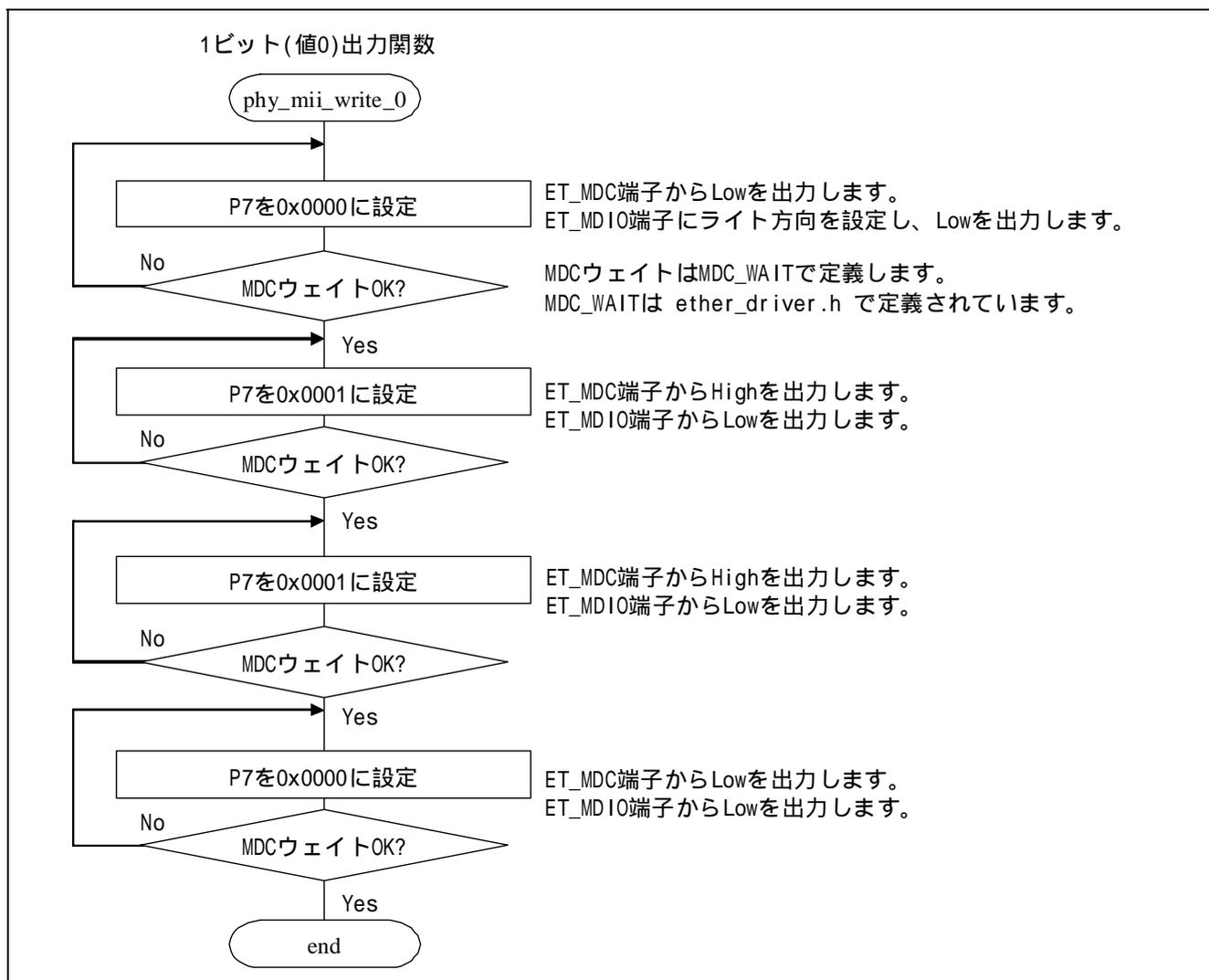


図 3.16 MII/RMII レジスタアクセスの処理フロー(7)

### 3.4 PHY-LSI 自動交渉設定の設定内容

表 3.2 に参考プログラムの設定を示します。

表 3.2 参考プログラムの設定

項目	内容
PHY-LSI 型式	Standard Microsystems Corporation 製 LAN8700i
リンクモード	100Mbps (全二重、半二重) および 10Mbps (全二重、半二重)
リンク決定方式	自動交渉(Auto-Negotiation)
PHY アドレス	0x1F <sup>*1</sup>
設定対象の MII レジスタ	Register 0 – Basic Control (アドレス 0x00) Register 1 – Basic Status (アドレス 0x01) Register 4 – Auto Negotiation Advertisement (アドレス 0x04) Register 5 – Auto Negotiation Link Partner Ability (アドレス 0x05)

【注】\*1 V850E2/ML4CPU ボード (型番: R0K0F422C000BR) の設定が 0x1F になっています。使用する PHY アドレスに合わせて変更してください。

### 3.5 PHY-LSI 自動交渉設定の注意点

- 参考プログラムは、PHY のリンク決定方式に自動交渉モードを使用しています。
- 相手先が自動交渉モードの場合は表 3.3 の優先順位に従い、リンクモードが決定します。

表 3.3 リンクモードの優先順位

優先順位		リンクモード
高	1	100Mbps 全二重
	2	100Mbps 半二重
低	3	10Mbps 全二重
	4	10Mbps 半二重

- MII/RMII レジスタのアクセスタイミングを "ether\_driver.h" ファイルの次のマクロ定義で変更することが可能です。設定する値は 1 以上を設定してください。

```
#define MDC_WAIT 2
```

- V850E2/ML4CPU ボード (型番: R0K0F422C000BR) では PHY アドレスが 0x1F に設定されています。参考プログラムでは PHY アドレスを "ether\_driver.h" ファイルの次のマクロ定義で変更することが可能です。

```
#define PHY_ADDRESS ( 31 )
```

## 4. 送受信設定の説明

参考プログラムでは、イーサネット・コントローラ、およびイーサネット・コントローラ専用 DMAC を使用します。

### 4.1 使用機能の動作概要

V850 ファミリでは、イーサネット通信を行う場合、イーサネット・コントローラとイーサネット・コントローラ専用 DMAC を使用します。イーサネット・コントローラは送受信制御を行います。イーサネット・コントローラ専用 DMAC は送信/受信 FIFO とデータ格納先（データ・バッファ）との間で DMA 転送を行います。

#### 4.1.1 イーサネット・コントローラの概要

V850 ファミリは、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・コントローラを内蔵しています。イーサネット・コントローラは同規格に合致する物理層 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。イーサネット・コントローラは MAC 層インタフェースを 1 系統内蔵しています。また、イーサネット・コントローラは内部でイーサネット・コントローラ専用 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。

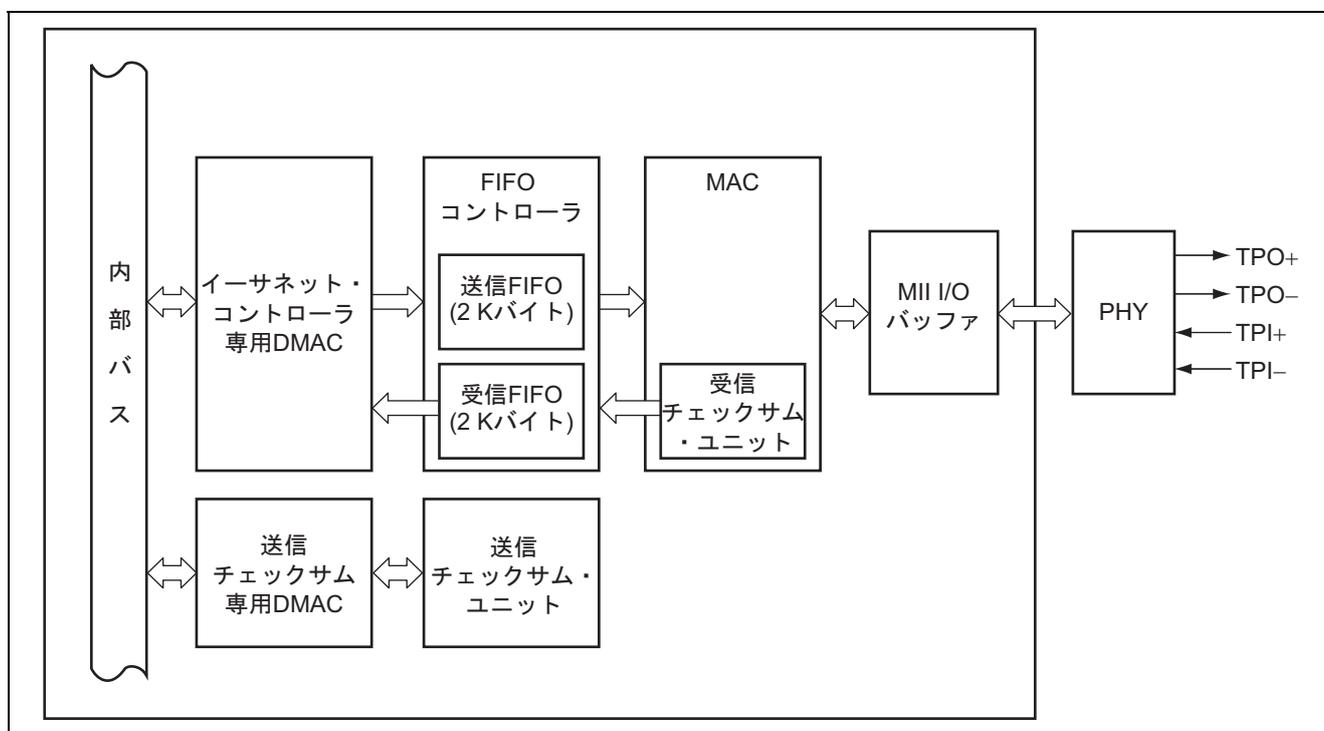


図 4.1 イーサネット・コントローラの構成

#### 4.1.2 イーサネット・コントローラがサポートしているフレーム・フォーマット

イーサネット・コントローラでは、次の3種類のフレーム・フォーマットをサポートします。

- ベーシック・フレーム
- VLAN フレーム
- ポーズ・コントロール・フレーム

##### (1) ベーシック・フレーム

イーサネットで用いられる基本のフレーム・フォーマットで7つの要素から構成されます。

- PA : プリアンブル
- SFD : フレーム開始デリミタ
- DA : デスティネーション・アドレス
- SA : ソース・アドレス
- TYPE/LEN : タイプ/レングス・フィールド
- DATA : データ・フィールド
- FCS : フレーム・チェック・シーケンス



図 4.2 ベーシック・フレームの構造

## (2) VLAN フレーム

イーサネット・コントローラでは VLAN フレーム検出機能をもっており、送信パケットあるいは受信パケットが VLAN フレームとして検出された場合、この受信パケット長を基準にパケット処理を行います。

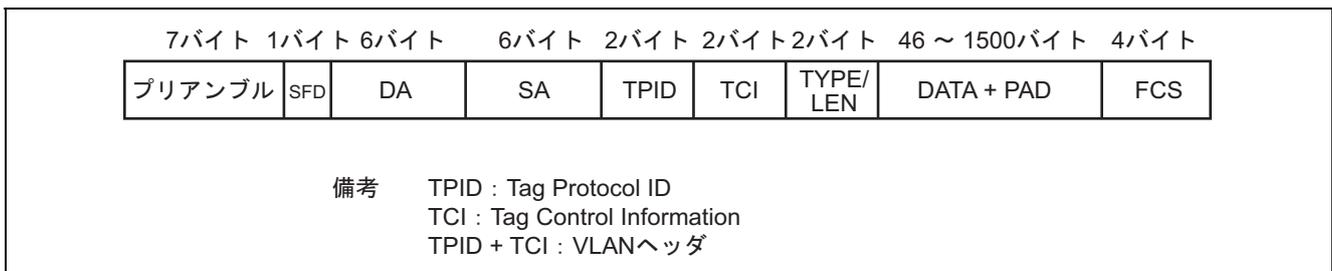


図 4.3 VLAN フレーム構造

## (3) ポーズ・コントロール・フレーム

イーサネット・コントローラには、受信 FIFO 内に残っているデータ量によって、自動的にポーズ・コントロール・フレームを送信する機能があります。

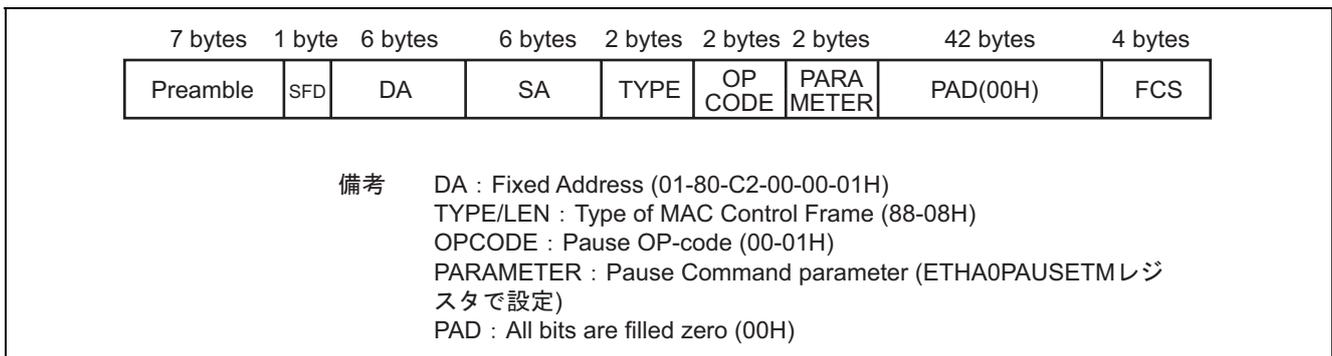


図 4.4 ポーズ・コントロール・フレーム構造

### 4.1.3 イーサネット・コントローラ送信部の概要

イーサネット・コントローラの送信機能は、イーサネット・コントローラ専用 DMAC の DMA 転送により送信 FIFO に取り込んだ送信パケット・データから、IEEE 802.3 規格で定義される送信フレームを生成して PHY に出力します。衝突検出時にはランダム・バックオフ・アルゴリズムによる再送信を実行します。

#### (1) 送信クロック

イーサネット・コントローラは、外部 (PHY) から供給される送信クロック (TXCLK) に同期して動作します。DMA 転送により送信 FIFO に取り込まれた送信パケット・データは、FIFO 内で TXCLK に同期化されて PHY に出力されます。

#### (2) キャリア・センス信号 (CRS)

半二重通信時、イーサネット・コントローラは、FIFO 内に送信データを格納し終わり、送信可能になったときに、キャリアが検出 (CRS = 1) されていた場合、キャリアの終了 (CRS = 0) まで送信を延期します。キャリアの終了後、ETHA0IPGT レジスタによって設定されるパケット間ギャップ (IPG) カウントが満了したあとで送信を開始します。

#### (3) 衝突検出 (COL) と再送信

半二重通信時、イーサネット・コントローラは衝突を検出すると、ジャム・データ (エラーCRC) を送信したあと、送信を中止します。最大衝突検出回数 (初期設定: 15 回) 以内でコリジョン・ウィンドウ内の衝突を検出した場合、ランダム・バックオフ・アルゴリズムによる送信の待機を実行後、送信 FIFO 内のデータを再送信します。

#### (4) パケット間ギャップ (IPG)

イーサネット・コントローラでは、自分または他局の送信が終了した後に IPG のカウントを開始し、自分の送信後で、IPG カウントが ETHA0IPGT レジスタに達する前に、FIFO から次の送信要求を示された場合には、連続 (Back-To-Back) での送信とみなし、カウント終了後ただちに送信を開始します。

#### (5) プリアンブル / CRC / パッド付加

7 バイトのプリアンブルと 1 バイトのフレーム開始デリミタ (SFD) を、FIFO より供給された送信パケットの前に付加します。

ETHA0MACC1.CRCEN ビットに 1 が設定されている場合、送信パケットの終わりに内部で生成されたフレーム・チェック・シーケンス (FCS) を追加します。

#### (6) 送信アボート

次の条件が発生した場合、イーサネット・コントローラは送信をアボートします。

なお、イーサネット・コントローラでは、正常な使用範囲において、送信 FIFO アンダランによるアボートは発生しません。

- 最大衝突回数を超える衝突 (MAX コリジョン)
- コリジョン・ウィンドウ外での衝突 (レイト・コリジョン)
- 過剰送信遅延
- ETHA0LMAX レジスタに設定されるフレーム長を超えるパケットを送信しようとした場合 (ただし、ETHA0MACC1.HUGEN ビットに 1 が設定されている場合、送信フレーム長は制限されません)

#### (7) 全二重動作

ETHA0MACC1.FULLD ビットを 1 に設定すると、全二重動作を可能にします。IPG は、常に ETHA0IPGT レジスタで設定される値となります。

#### (8) フロー・コントロール機能・バック・プレッシャー機能

イーサネット・コントロールは、受信 FIFO と連動したフロー・コントロール機能、バック・プレッシャー機能を持っていますので、受信 FIFO の空き容量が少なくなると自動的に作動して、受信 FIFO オーバフローを極力防止します。

#### 4.1.4 イーサネット・コントローラ受信部の概要

イーサネット・コントローラは、受信フレームから FIFO に対し受信パケットを生成し、SFD の検出、レンジ・フィールド・チェック、FCS チェック、VLAN フレームの検出などを行います。

##### (1) 受信クロック

イーサネット・コントローラは、外部(PHY)から供給される受信クロック (RXCLK) に同期して、データを受信します。

##### (2) MII データの受信

イーサネット・コントローラは、RXDV 信号がアサートされている期間、信号上のデータを受信フレームとして認識し、RXDV 信号がディアサートされたときをフレームの終了とみなします。

##### (3) プリアンプルおよび SFD の検出

イーサネット・コントローラは、受信フレームの先頭においてプリアンプルと SFD を検出、これ以降のデータを受信パケットとします。

##### (4) レンクス・フィールドのチェック

イーサネット・コントローラは受信パケットの長さをカウントし、ソース・アドレスに続く 2 バイトをレンジ・フィールドとみなして、データ・フィールド長のチェックを行います。

##### (5) CRC チェック

イーサネット・コントローラは、受信パケットから 4 バイトのフレーム・チェック・シーケンス(FCS)を計算し、受信パケットの最後に付加されている FCS データと比較します。

##### (6) FIFO へのデータ送出

イーサネット・コントローラは、6 バイト以上のパケットを有効とし、6 バイト未満のパケットは破棄します。

##### (7) ヒュージ・パケットの検出

イーサネット・コントローラは、ETHA0MACC1.HUGEN ビットが 0 であるとき、ETHA0LMAX レジスタで設定される最大フレーム長 (初期値: 1536 バイト) 以下のパケットのみを受信し、それを超えるパケットは途中で打ち切られます。

##### (8) VLAN フレーム検出

イーサネット・コントローラは、受信したパケットすべてに対して VLAN フレームであるかどうかチェックします。

受信したパケットの TPID フィールド(ソース・アドレスに続く 2 バイト)の値が ETHA0VLTP レジスタに設定された値と一致すると、VLAN パケットとして ETHA0RXSTMONI.VLAN フラグをセットします。

#### 4.1.5 イーサネット・コントローラ専用 DMAC の概要

V850 ファミリは、イーサネット・コントローラに直結した専用 DMAC を内蔵しています。バッファ管理の多くの部分をイーサネット・コントローラ専用 DMAC がディスクリプタを用いて制御します。このため 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。

イーサネット・コントローラ専用 DMAC はイーサネット・コントローラと接続され、送受信データを CPU の介在なく効率的な転送をメモリ（データ・バッファ）との間で行います。イーサネット・コントローラ専用 DMAC は、各バッファと対応したディスクリプタと呼ぶバッファポインタなど格納した制御情報を自ら読み込みます。この制御情報に従って送信データを送信データ・バッファから読み込み、受信データは受信データ・バッファにライトします。このディスクリプタを複数個連続して配置（ディスクリプタリスト）することで、送信ならびに受信を連続して実行できます。

表 4.1 にイーサネット・コントローラ専用 DMAC の仕様を示します。

表 4.1 イーサネット・コントローラ専用 DMAC の仕様

項目	内容
DMA 転送モード	シングル転送モード 4 ビート・インクリメント式バースト転送 8 ビート・インクリメント式バースト転送 16 ビート・インクリメント式バースト転送
DMA 転送でアクセス可能な対象	Hバス共有メモリ

4.1.6 ディスクリプタの概要

イーサネット・コントローラがDMA転送を行うためには、データ・バッファと送受信データの格納アドレス等が書かれたディスクリプタが必要になります。

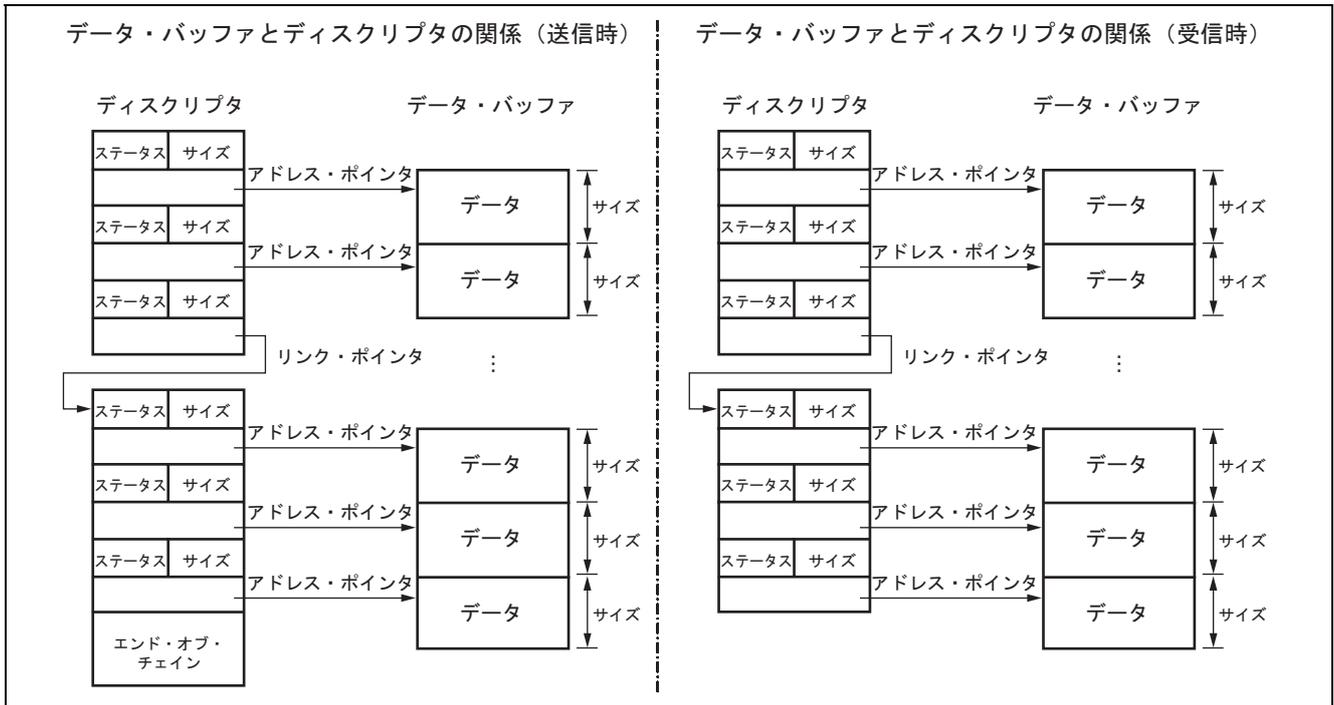


図 4.5 データ・バッファとディスクリプタの関係

## 4.1.7 ディスクリプタの種類

イーサネット・コントローラを使用する上で、送信データ/受信データを格納するメモリ空間が連続していない場合をサポートするために、ディスクリプタ・メカニズムを導入しています。

イーサネット・コントローラが使用するディスクリプタには、次の3種類があります。

- バッファ・ディスクリプタ
- リンク・ポインタ
- エンド・オブ・チェイン

各ディスクリプタは、2ワード（64ビット）のワード・アラインされたデータで構成されます。

## (1) バッファ・ディスクリプタ

バッファ・ディスクリプタのフォーマットは、2ワード（64ビット）で構成され、下位はコントロール・ビット、上位ワードは、このディスクリプタが指すデータ・バッファのスタート・アドレス値を示します。

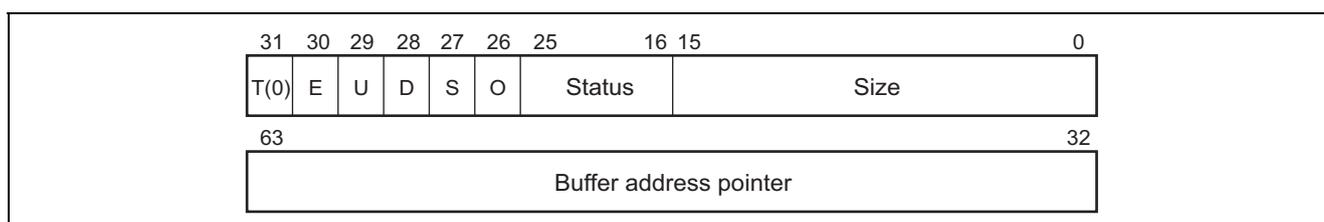


図 4.6 バッファ・ディスクリプタの構成

表 4.2 バッファ・ディスクリプタ・データ・フォーマット

ビット位置	ビット名	意味
63 - 32	BAP	データ・バッファのスタート・アドレスを表すアドレス・ポインタです。
31	T	ディスクリプタ・タイプ ディスクリプタの種類を示します。 バッファ・ディスクリプタの場合、このビットは0で使用します。
30	E	ラスト・バッファ・フラグ パケット・データの最後を表すコントロール・ビットです。 0: 通常のバッファ・データである（最終データではない）ことを示します。 1: 現在のパケットの最終データ・バッファであることを示します。
29	U	ユーズド・ビット DMA が転送済か未転送（転送中を含む）かを示します。 0: 未転送（転送中を含む） 1: 転送済
28	D	データ・バッファのアクセス・エラーを示すビットです。 0: エラーなし 1: データ・バッファのアクセス・エラーが発生した
27	S	受信ステータス情報が Status フィールドに書き込まれていることを示します 0: ステータス情報は含まれていません 1: 受信パケットのステータス情報を含みます 受信時のパケット先頭のディスクリプタ内のコントロール・ビット S のみ有効です。送信時は使用しません。
26	O	受信時のオーバーフローを通知します。 0: オーバーフローなし 1: オーバーフローが発生した 送信時は使用しません。
25 - 16	Status	受信時の Status 情報を示します。コントロール・ビット S が 1 である

		場合、Status フィールドの値が有効です。 送信時は使用しません。
15-0	Size	このディスクリプタが示すバッファ・データのサイズ(バイト単位)を示します

## (2) リンク・ポインタ

リンク・ポインタのフォーマットは、2ワードで構成され、下位ワードはコントロール・ビット、上位ワードは、次のディスクリプタのアドレス値を示します。

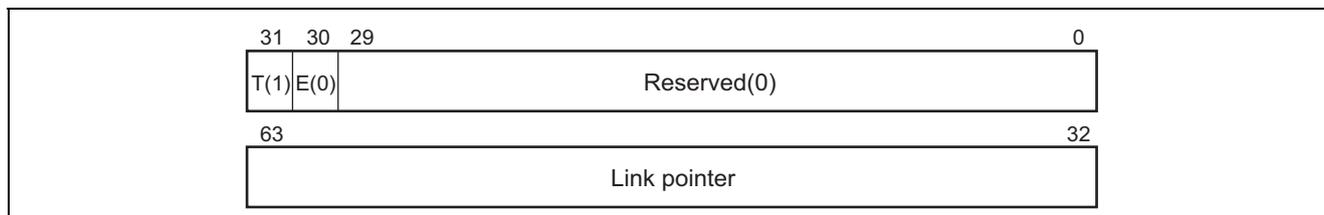


図 4.7 リンク・ポインタの構成

表 4.3 リンク・ポインタ・データ・フォーマット

ビット位置	ビット名	意味
63 – 32	Link Pointer	次のディスクリプタのアドレスを示します。
31	T	リンク・ポインタの場合、このビットは1で使用します。
30	E	リンク・ポインタの場合、このビットは0で使用します。
29 - 0	Reserved	予約領域です。

## (3) エンド・オブ・チェイン

エンド・オブ・チェインのフォーマットは、2ワードで構成され、下位ワードはコントロール・ビット、上位ワードは0を示します。イーサネット・コントローラは、エンド・オブ・チェインを検出すると、DMA 転送を終了し RECI/TECI 割り込みを発生します。

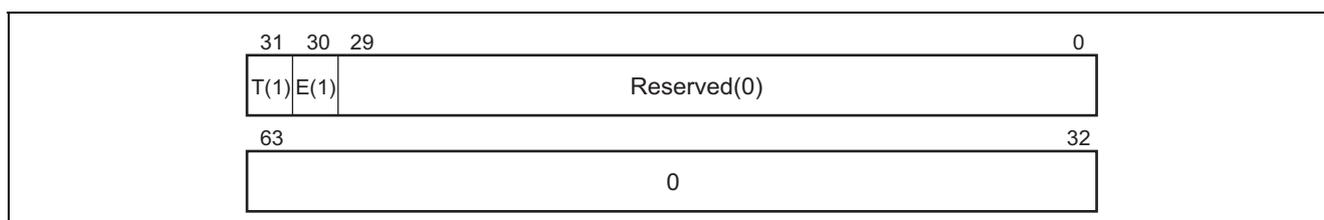


図 4.8 エンド・オブ・チェインの構成

表 4.4 エンド・オブ・チェイン・データ・フォーマット

ビット位置	ビット名	意味
63 – 32	BAP	エンド・オブ・チェインの場合、NULL (オール・ゼロ) を設定します。
31	T	エンド・オブ・チェインの場合、このビットは1で使用します。
30	E	エンド・オブ・チェインの場合、このビットは1で使用します。
29 - 0	Reserved	予約領域です。

#### 4.1.8 ディスクリプタの動作概要

イーサネット・コントローラでは、1回のDMA転送の起動で複数のディスクリプタを逐次処理することが可能です。ETHA0RXDPに受信ディスクリプタ・チェーンの先頭アドレス、ETHA0TXDPに送信ディスクリプタ・チェーンの先頭アドレスをセットし、ETHA0MODEレジスタのRXS、TXSビットをセットすることで、受信DMA転送、送信DMA転送がスタートします。

ディスクリプタ・チェーンの最後はエンド・オブ・チェーンのディスクリプタで終了しなければなりません。

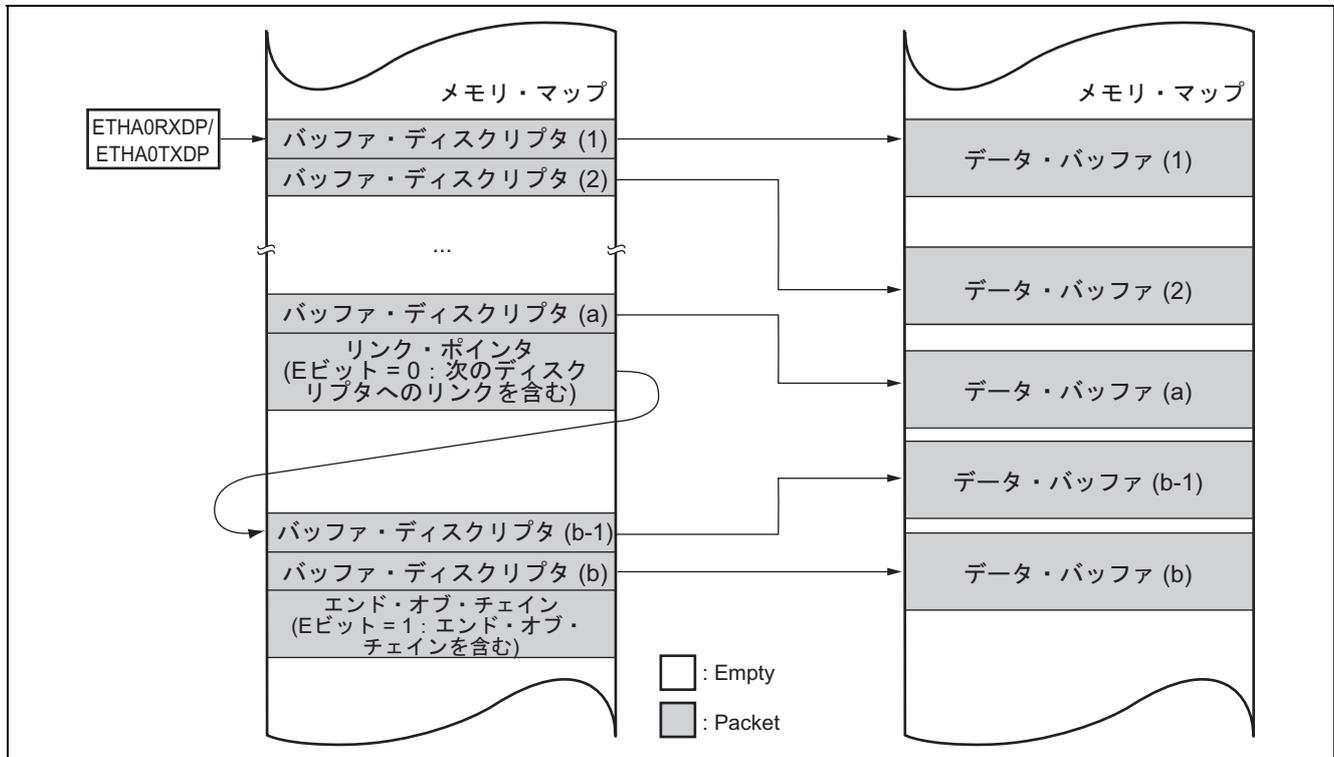


図 4.9 ディスクリプタとデータ・バッファの関係 (チェーン)

4.1.9 ディスクリプタ・チェーンをリング・バッファとした場合

ディスクリプタ・チェーンをリング・バッファにしている場合は、INTSCTX 割り込みの TXI フラグ (INTSCRX 割り込みの RXI フラグ) をトリガにして ETHA0LSTRXDP, ETHA0LSTTXDP を読み出し、ディスクリプタを更新できます。

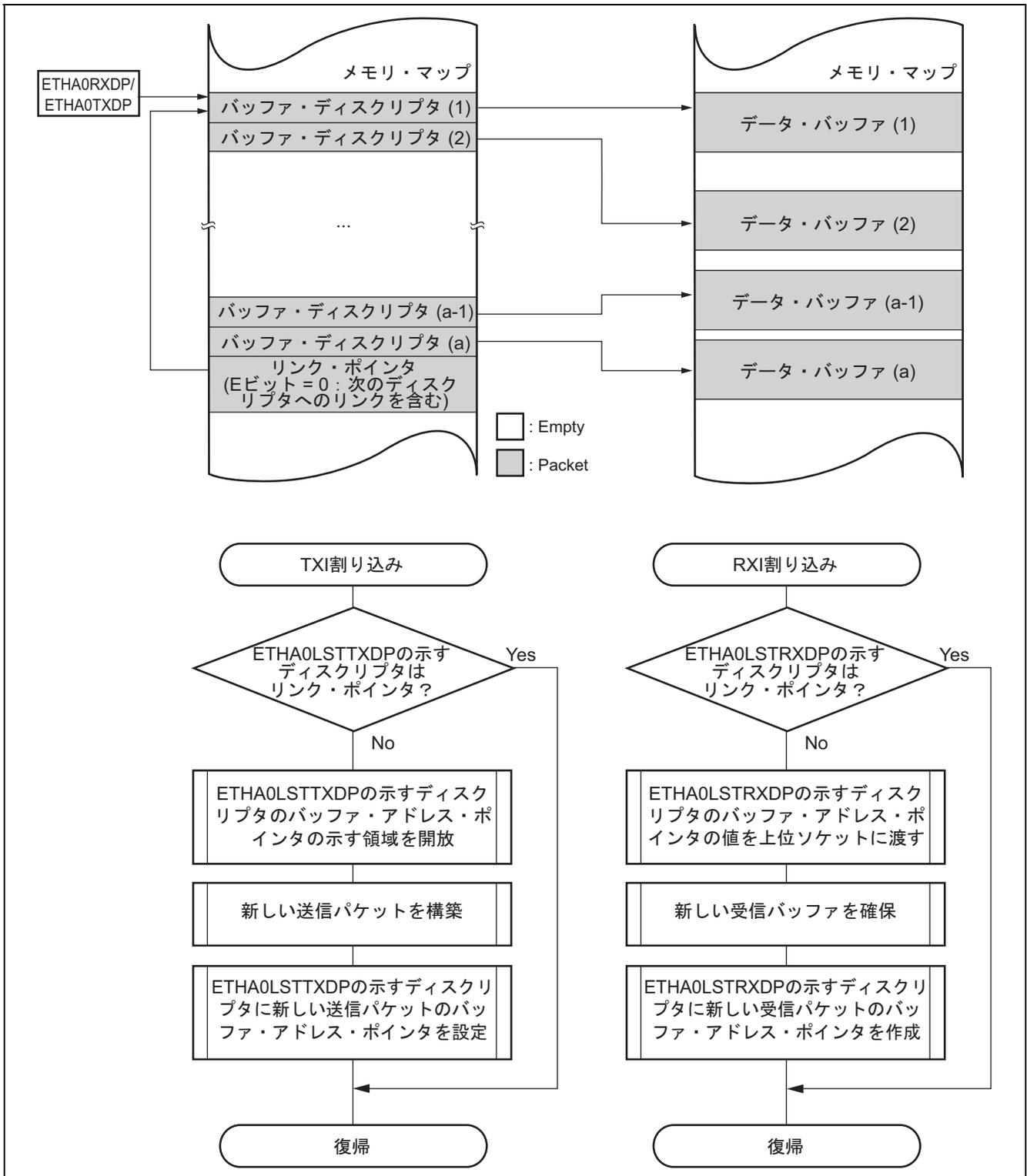


図 4.10 ディスクリプタとバッファの関係 (リング・バッファ)

## 4.1.10 送信ディスクリプタ設定例

図 4.11 に送信バッファを 2 面使用した場合の、ディスクリプタ・チェーン例と動作を説明します。

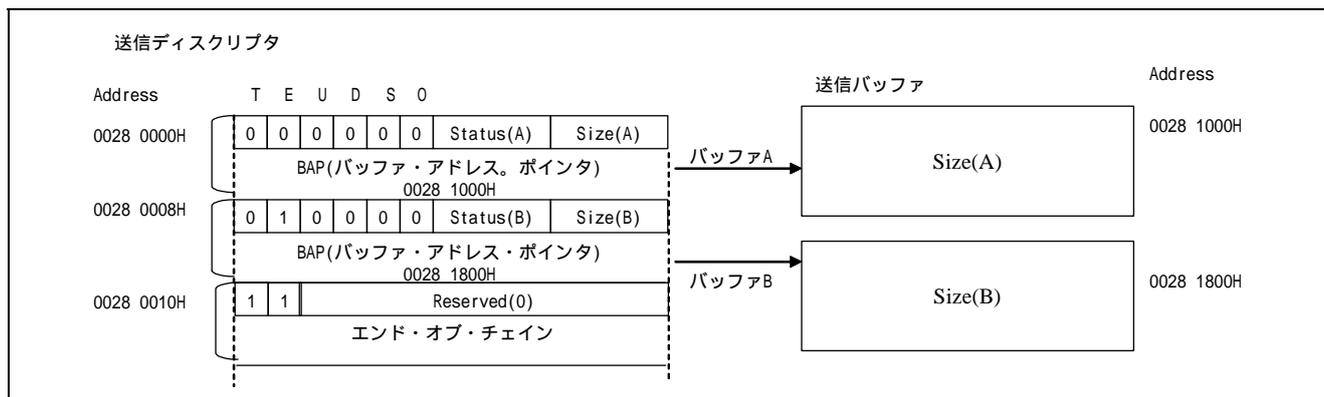


図 4.11 送信ディスクリプタ設定例

- ソフトウェアで、コア・ファンクション設定レジスタ(ETHA0MODEETHA0MODE)のTXSビットをセット(1)します。
- イーサネット・コントローラは、送信ディスクリプタ・ポインタ(ETHA0TXDP)で示されるアドレス(0028 0000H)から先頭ディスクリプタを読み出し、送信ディスクリプタの解析を開始します。
- DMA 転送開始アドレスにバッファ・アドレス・ポインタ(0028 1000H)をセットし、バッファ内のデータをFIFOに転送します。
- 送信ディスクリプタのEビットが0で、最終データではないことを示しているため、次のバッファ・ディスクリプタ(0028 0008H)を読み出し、バッファ・アドレス・ポインタ(0028 1800H)をセットし、バッファ内のデータをFIFOに転送します。
- Eビットが1のバッファ・ディスクリプタが指すバッファ内のデータの送信を完了すると、Uビットをセットし、送信処理が完了します。また、割り込み要求(TXI)が発生します。

## 4.1.11 受信ディスクリプタ設定例

図 4.12 に受信バッファを 2 面使用した場合の、ディスクリプタ・チェーン例と動作を説明します。

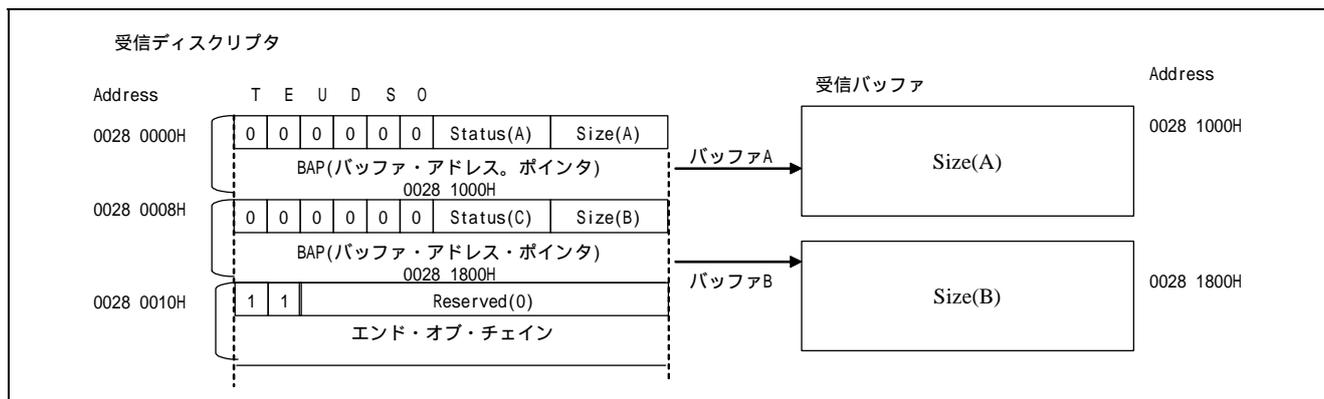


図 4.12 受信ディスクリプタ設定例

- ソフトウェアで、コア・ファンクション設定レジスタ(ETHA0MODEETHA0MODE)のRXSビットをセット(1)します。
- イーサネット・コントローラは、受信ディスクリプタ・ポインタ(ETHA0RXDP)で示されるアドレス(0028 0000H)から先頭ディスクリプタを読み出し、受信ディスクリプタの解析を開始します。
- イーサネット・コントローラは、DMA転送開始アドレスに先頭バッファ・アドレス・ポインタ(0028 1000H)をセットし、FIFO内の受信データをバッファAに転送します。
- その後の受信でバッファAがフルになると次のディスクリプタ(0028 0008H)を読み出し、バッファ・アドレス・ポインタ(0028 1800H)をDMA転送開始アドレスにセットし、FIFO内の受信データをバッファBに転送します。
- 最終ディスクリプタに対して、Eビット、Uビットをセット(1)し、Sizeフィールドに転送したデータのバイト数をライト・バックします。またパケット・データをすべて転送したあと、先頭ディスクリプタの、Uビット、Sビットをセット(1)し、受信ステータス情報をStatus(A)フィールドにライト・バックします。

#### 4.1.12 フレーム送信手順

CPU がイーサネット・コントローラ専用 DMAC の転送対象に、送信ディスクリプタ、および送信データを準備して、送信ディスクリプタ・レジスタ (ETHA0TXDP) を設定し、ETHA0MODEETHA0MODE レジスタの TXS ビットをセットすると、専用 DMAC は、ディスクリプタ・レジスタに設定されたアドレスから、送信バッファ・ディスクリプタをフェッチして、データ・バッファから送信データを読み出し、送信 FIFO に転送します。

FIFO に転送されたデータは、TXCLK に同期化され、プリアンプル、SFD、フレーム・データの順で PHY に出力されます。

ETHA0MACC1 レジスタの CRCEN ビットが設定されている場合、データの終わりに FCS を付加します。

ETHA0MACC1 レジスタの PADEN ビットが設定されている場合、ショート・フレーム送信時に自動的に PAD が付加されます。

現在指しているディスクリプタがフレームの終端を含んでいない場合、次のディスクリプタを読み出し、このディスクリプタが指すデータ・バッファからデータを読み出します。

送信終了後、送信ステータスを最終ディスクリプタに書き込みます。このあと、次の送信バッファ・ディスクリプタをフェッチし、次のデータが送信可能な場合に同様の方法で送信を開始します。

また、各バッファ・ディスクリプタの DMA 処理が完了すると、送信 DMA が完了したことを示す割り込み (TXI) をその都度発生させます。

次の送信バッファ・ディスクリプタがエンド・オブ・チェイン・ディスクリプタの場合は、エンド・オブ・チェイン・ディスクリプタを示す割り込み (TECI) を発生させ、送信 DMA が停止します。

送信 DMA を起動させるには、ETHA0TXDP レジスタとバッファ・ディスクリプタを再設定してください。

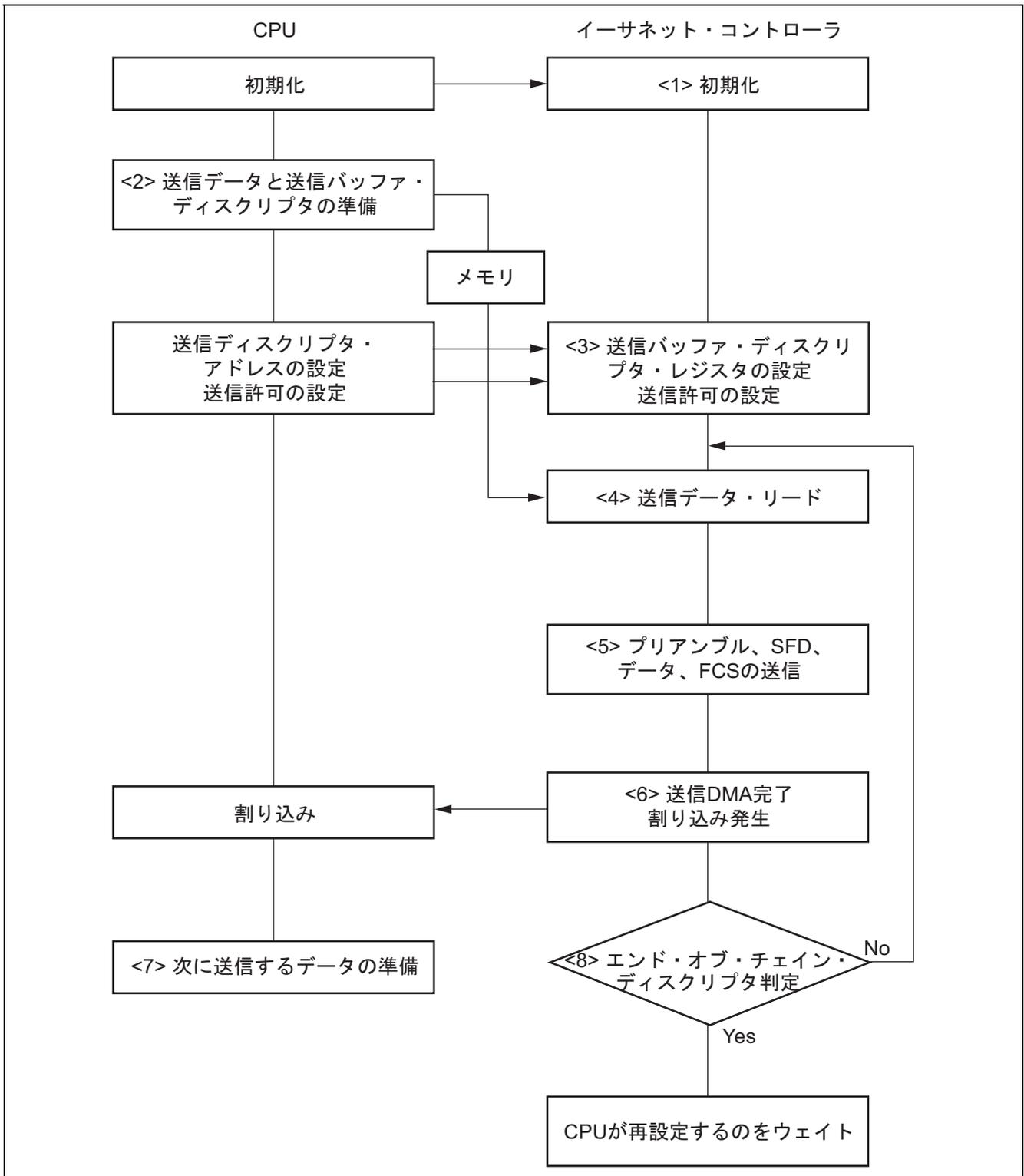


図 4.13 フレーム送信手順

#### 4.1.13 フレーム受信手順

MAC コンフィギュレーション・レジスタ (ETHA0MACC1) の SRXEN ビット (受信許可)、ETHA0MODEETHA0MODE レジスタの RXS ビット (受信 DMA 許可)、受信ディスクリプタ・ポインタ・レジスタ ETHA0RXDP を設定し、MAC がデータを受信するとすぐに受信フレームの処理を開始します。

データを受信すると、プリアンプル、フレーム開始デリミタ (SFD) が有効かをチェックします。

プリアンプルと SFD が有効であると、受信したフレームに対する処理が行われます。

有効なプリアンプルと SFD が見つからなかった場合、フレームは無視されます。

フレームが衝突を起こしたり、アドレス・フィルタリングによってフレームが破棄されたりした場合、受信バッファにはデータが書き込まれません。

正常に受信され、アドレス・フィルタリングにより破棄されなかった受信フレームは、受信バッファ・ディスクリプタが指定するデータ・バッファに転送します。

受信中、イーサネット・コントローラはフレームの長さが適切であることをチェックしています。

フレームの終端になると FCS がチェックされ、バッファ・ディスクリプタに書き込まれます。なお、64 バイト以下のフレーム (ショート・パケット) は DMA 転送されます。

フレームの受信が完了すると、最終ディスクリプタに対して E ビット、U ビットをセット(1)し、Size フィールドに転送したデータのバイト数をライト・バックします。またパケット・データを全て転送したあと、先頭ディスクリプタの U ビット、S ビットをセット(1)し、受信ステータス情報を Status フィールドにライト・バックします。また、割り込み要求 (RXI) が発生します。

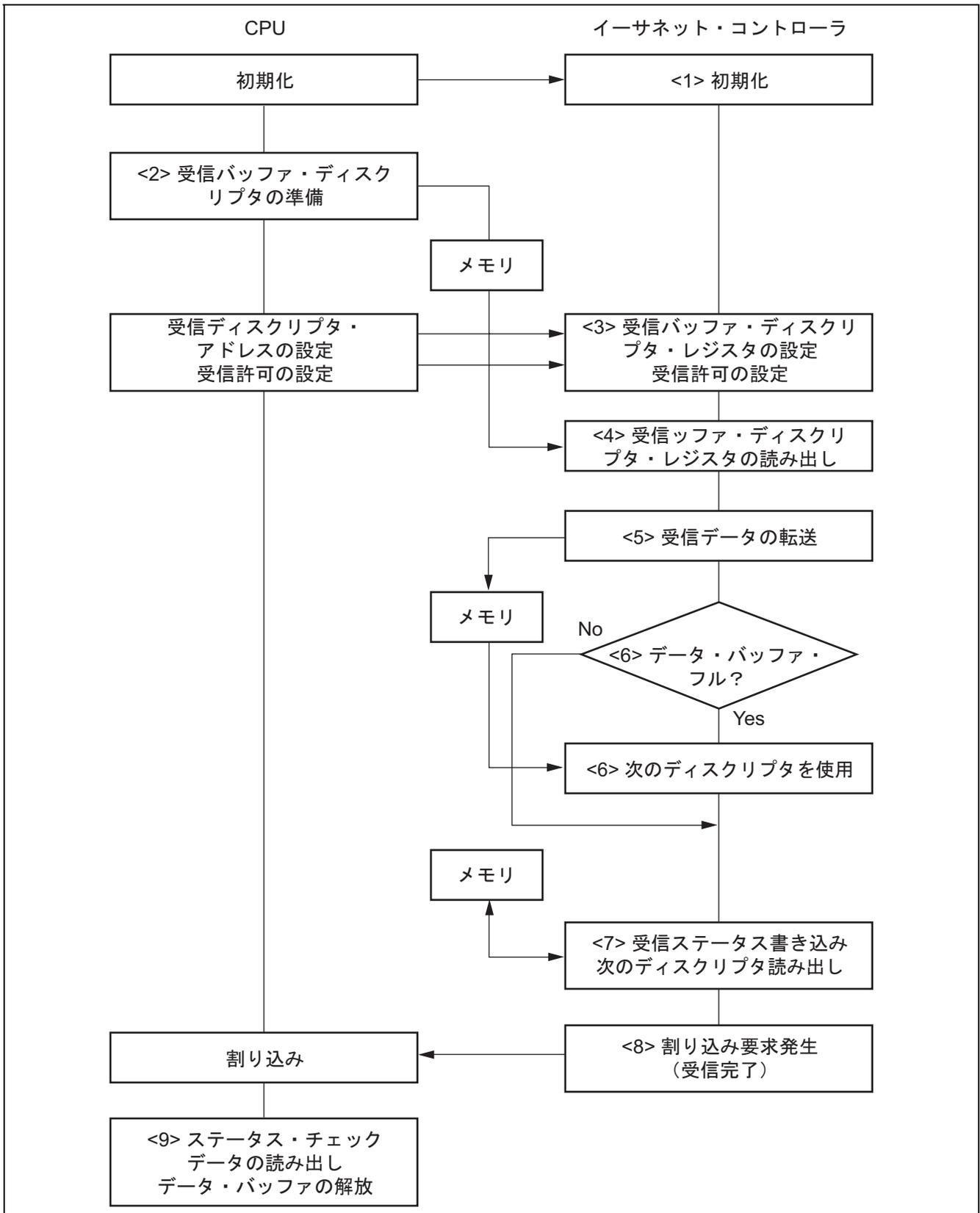


図 4.14 フレーム受信手順

## 4.2 参考プログラムの動作

参考プログラムではメインルーチンで選択したテストタイプにより次の2種類の処理を行います。

- イーサネットフレームを10フレーム送信します。
- イーサネットフレームを10フレーム受信します。

### 4.2.1 参考プログラムの動作（送信時）

送信テストを選択した場合、参考プログラムでは、イーサネット・コントローラおよびイーサネット・コントローラ専用DMACを使用し、対向ホストに向けて10フレーム送信します。

送信ディスクリプタを5面と、1520バイトの送信バッファ（データ・バッファ）を4面用意しています。送信ディスクリプタをリング状にして使用しています。

送信ディスクリプタは5面のうち、1~4面目をバッファ・ディスクリプタとして、5面目をリンク・ポイントとして使用しています。

10フレーム分の送信データを送信バッファへライトした後、転送制御レジスタ（ETHA0TRANSCTL）の送信ステータスビット（TREN\_STA）により10フレームの送信が完了したと判断し、送信テストを終了します。

### 4.2.2 参考プログラムの動作（受信時）

受信テストを選択した場合、参考プログラムでは、イーサネット・コントローラおよびイーサネット・コントローラ専用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を4面と、1520バイトの受信バッファ（データ・バッファ）を3面用意しています。

受信ディスクリプタは4面のうち、1~3面目をバッファ・ディスクリプタとして、4面目をリンク・ポイントとして使用しています。

受信ディスクリプタのUビット、Sビット、Oビット（バッファ・ディスクリプタのビット29、27、26ビット）をチェックし、エラーがなく、かつ、Sizeビット（バッファ・ディスクリプタのビット[15:0]）が0以外であれば、受信ディスクリプタが示すデータ・バッファから受信データを読み出します。

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。

### 4.2.3 参考プログラムの動作環境

図 4.15 に参考プログラムの動作環境を示します。

動作環境の注意として「4.7.1 動作環境の注意点 1」「4.7.2 動作環境の注意点 2」を参照ください。

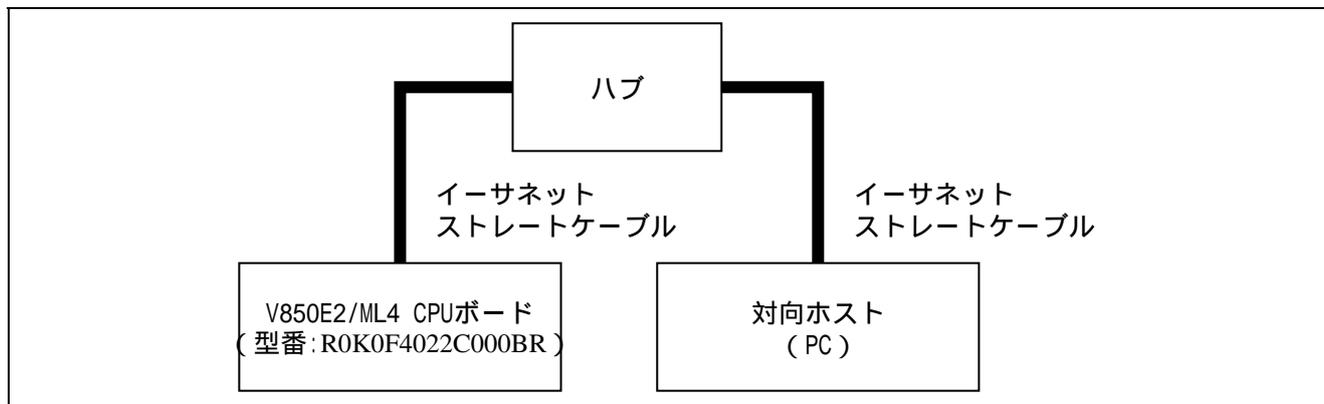


図 4.15 参考プログラムの動作環境

### 4.2.4 イーサネットフレーム・フォーマット

送信データについては、イーサネットフレームのうちプリアンプル、スタートフレームデリミタ (SFD)、および CRC 部を除いた部分を用意してください。ヘッダ部の宛先 MAC アドレスおよび送信元 MAC アドレスは、ご使用になる製品の MAC アドレスに変更してください。なお、イーサネット・コントローラは送信元 MAC アドレスのチェックは行いません。



図 4.16 イーサネットフレーム・フォーマット (送信)

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。

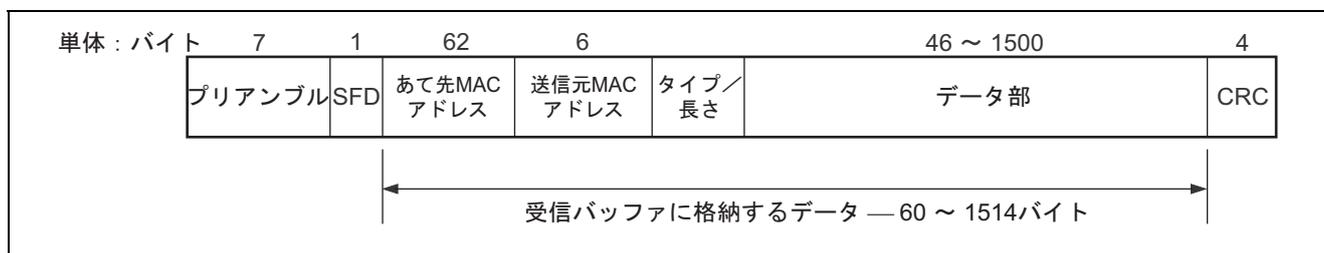


図 4.17 イーサネットフレーム・フォーマット (受信)

### 4.3 参考プログラムのディスクリプタ定義

V850ML4/E2 では、Hバスの共用メモリ領域にディスクリプタとデータ・バッファを確保します。

参考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、ソフトウェアにてリング構造を実現しています。

図 4.18 に参考プログラムでの送信、受信ディスクリプタとデータ・バッファを示します。

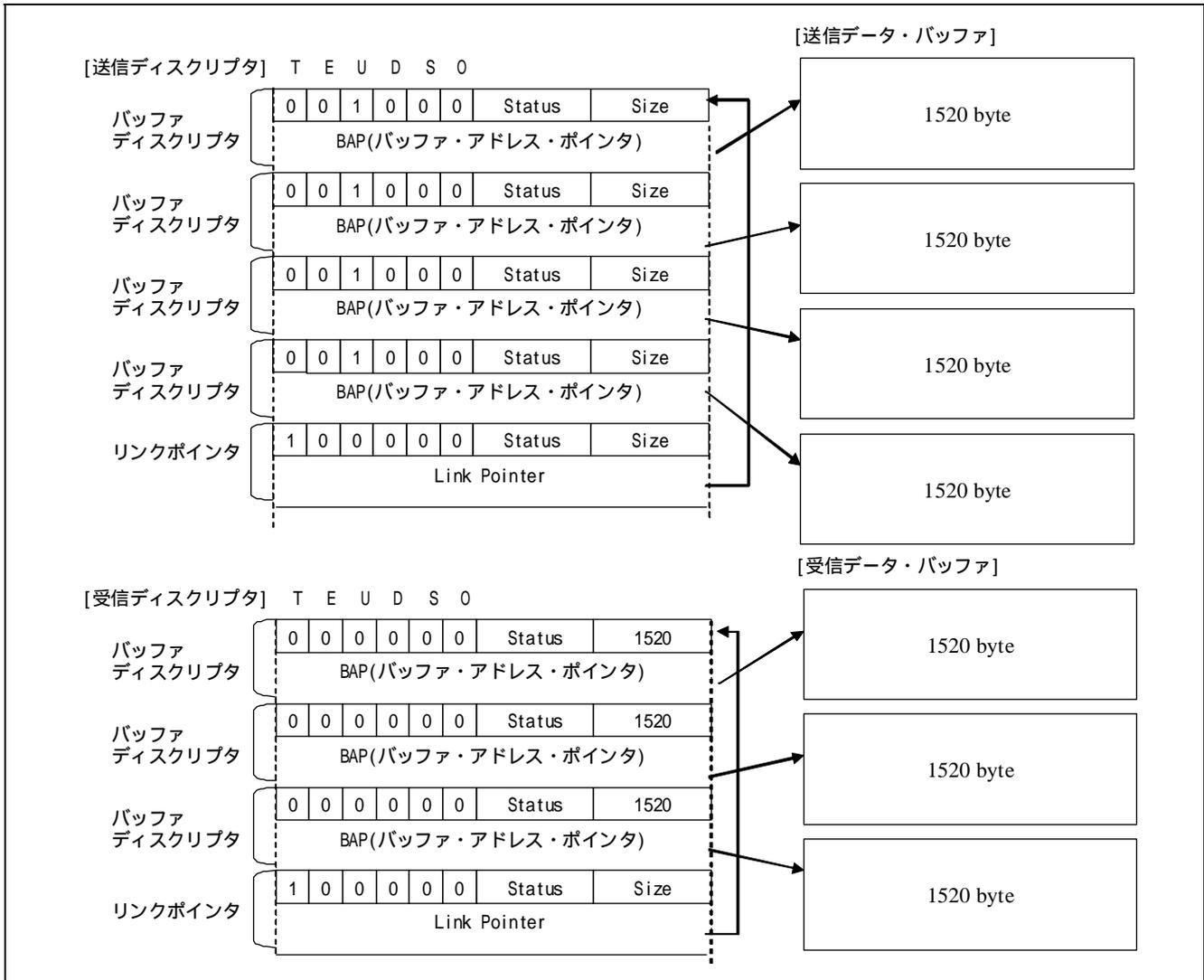


図 4.18 参考プログラムのディスクリプタ設定例

参考プログラムのディスクリプタはコンパイラにより定義されるマクロを使用して、以下のような構造体で定義されています。

```
typedef struct tst_ctrl_desc {
    uint32_t value;          /* descriptor 31- 0 bit      */
    uint8_t* addr;          /* descriptor 63-32 bit     */
} CTRL_DESC;

typedef struct tst_ctrl_buf {
    CTRL_DESC* tx_next;     /* top address of active TX DMA descriptor */
    CTRL_DESC* tx_write;   /* write address of active TX DMA descriptor */
    uint8_t tx_next_buf;   /* write address of active TX DMA buffer */
    CTRL_DESC* rx_read;    /* read address of active RX DMA descriptor */
} CTRL_BUF;
```

“ether\_driver.h”で定義されている次のマクロを変更することでディスクリプタの数、データ・バッファのサイズを変更することが可能です。FRAME\_SIZE はデータ・バッファサイズ、TX\_DESCRIPTOR\_MAX は送信ディスクリプタの数、RX\_DESCRIPTOR\_MAX は受信ディスクリプタの数を設定します。

```
#define FRAME_SIZE      ( 1518 )
#define TX_DESCRIPTOR_MAX ( 5 )
#define RX_DESCRIPTOR_MAX ( 4 )
```

データ・バッファサイズはチェックサム(2 バイト)を除いたサイズを指定します。

#### 4.4 参考プログラムで使用する割り込み

参考プログラムでは、イーサネット・パケット送信割り込み (INTETMTX) の送信 (DMA) エンド・オブ・チェーン割り込み (TECI) を使用します。TECI 割り込みが発生すると、送信ディスクリプタを先頭に戻し送信データがあれば送信を開始します。

#### 4.5 イーサネットドライバ API

TCP/IP スタックのドライバインタフェースとして次の関数が提供されます。これらはルネサス イーサネットデバイスの標準ルネサス API (RAPI) です。

- R\_Ether\_Open
- R\_Ether\_Close
- R\_Ether\_Read
- R\_Ether\_Write

##### 4.5.1 R\_Ether\_Open

R\_Ether\_Open 関数はイーサネット・コントローラと、イーサネット・コントローラ専用 DMAC、PHY-LSI および送受信データ・バッファの初期化を行います。

- プロトタイプ  
int32\_t R\_Ether\_Open(uint32\_t ch, uint8\_t mac\_addr[]);
- 引数  
ch  
— イーサネット・コントローラチャンネル番号の指定  
mac\_addr  
— イーサネット・コントローラの MAC アドレスの指定
- 戻り値  
R\_ETHER\_OK(0) : 正常終了  
R\_ETHER\_ERROR(-1) : エラー発生

- プロパティ
  - “ ether\_driver.h ” ファイルで宣言
  - “ ether\_driver.c ” ファイルで定義
- 説明

R\_Ether\_Open 関数はイーサネット・コントローラおよびイーサネット・コントローラ専用 DMAC を初期化します。イーサネット・コントローラのディスクリプタとデータ・バッファは初期状態にセットアップします。MAC アドレスはイーサネット・コントローラの MAC アドレスレジスタの初期化に使用されません。

初期設定により、PHY-LSI は自動交渉モードに設定されます。

V850E2/ML4 はイーサネット・チャンネルが 1 本だけであり、チャンネル番号に応じてイーサネットドライバの処理を分岐しておりません。チャンネル番号の値によらず正常に動作いたしますが、チャンネル番号は 0 にセットして使用することを推奨いたします。

V850E2/ML4 イーサネットドライバは MAC アドレスが 0 の場合の処理が実装されておりません。MAC アドレスには 0 以外の値を入力するか、MAC アドレスが 0 の場合の処理をお客様で追加して頂く必要があります。

#### 4.5.2 R\_Ether\_Close

R\_Ether\_Close 関数はイーサネット・コントローラおよびイーサネット・コントローラ専用 DMAC 機能を禁止にします。

- プロトタイプ

```
int32_t R_Ether_Close(uint32_t ch);
```
- 引数
  - ch
    - イーサネット・コントローラチャンネル番号の指定
- 戻り値
  - R\_ETHER\_OK(0) : 正常終了
  - R\_ETHER\_ERROR(-1) : エラー発生
- プロパティ
  - “ ether\_driver.h ” ファイルで宣言
  - “ ether\_driver.c ” ファイルで定義
- 説明

R\_Ether\_Close 関数は、イーサネット・コントローラおよびイーサネット・コントローラ専用 DMAC 機能を禁止にします。

V850E2/ML4 はイーサネット・チャンネルが 1 本だけであり、チャンネル番号に応じてイーサネットドライバの処理を分岐しておりません。チャンネル番号の値によらず正常に動作いたしますが、チャンネル番号は 0 にセットして使用することを推奨いたします。

### 4.5.3 R\_Ether\_Read

R\_Ether\_Read 関数はアプリケーションの受信バッファへデータを受信します。

- プロトタイプ

```
int32_t R_Ether_Read(uint32_t ch, void *buf);
```

- 引数

ch

— イーサネット・コントローラチャンネル番号の指定

\*buf

— 受信データ・バッファのポインタ

- 戻り値

0以上の値：受信したバイト数。0は受信データが存在しない場合です。

R\_ETHER\_ERROR(-1)：エラー発生（ハードウェアエラーとソフトウェアエラーを含む）

R\_ETHER\_HARD\_ERROR(-3)：ハードウェアエラー発生（復帰にソフトウェアリセットが必要）

R\_ETHER\_RECOVERABLE(-4)：復帰可能なエラー発生（復帰にソフトウェアリセットが不要）

R\_ETHER\_NODATA(-5)：受信データなし

【注】 参考プログラムではR\_ETHER\_HARD\_ERROR(-3)、R\_ETHER\_RECOVERABLE(-4)、R\_ETHER\_NODATA(-5)の戻り値は使用しておりませんのでご注意ください。

- プロパティ

“ ether\_driver.h ” ファイルで宣言

“ ether\_driver.c ” ファイルで定義

- 説明

R\_Ether\_Read 関数は、受信ディスクリプタが指定するバッファからデータを読みます。読み込んだデータは受信データ・バッファにコピーします。

V850E2/ML4 はイーサネット・チャンネルが1本だけであり、チャンネル番号に応じてイーサネットドライバの処理を分岐しておりません。チャンネル番号の値によらず正常に動作いたしますが、チャンネル番号は0にセットして使用することを推奨いたします。

受信フレームエラーが発生したディスクリプタのデータは破棄し、ステータスをクリアして読み込みを継続します。

#### 4.5.4 R\_Ether\_Write

R\_Ether\_Write 関数はアプリケーションの送信バッファからデータを送信します。

- プロトタイプ

```
int32_t R_Ether_Write(uint32_t ch, void *buf, uint32_t len);
```

- 引数

ch

— イーサネット・コントローラチャンネル番号の指定

\*buf

— 送信するイーサネットデータのポインタ

len

— イーサネットフレームの長さ

- 戻り値

R\_ETHER\_OK(0) : 正常終了

R\_ETHER\_ERROR(-1) : エラー発生

- プロパティ

“ ether\_driver.h ” ファイルで宣言

“ ether\_driver.c ” ファイルで定義

- 説明

R\_Ether\_Write 関数は、送信ディスクリプタが指定するバッファに送信データを書きます。書き込んだデータは、イーサネット・コントローラによって送信されます。

R\_Ether\_Write 関数は、送信完了のチェックを行っていません。

V850E2/ML4 はイーサネット・チャンネルが 1 本だけであり、チャンネル番号に応じてイーサネットドライバの処理を分岐していません。チャンネル番号の値によらず正常に動作いたしますが、チャンネル番号は 0 にセットして使用することを推奨いたします。

送信フレームエラーのチェックを行っていません。

## 4.6 参考プログラムの処理手順

図 4.19、図 4.20 にイーサネットドライバ API を使用した参考プログラムの処理フローと、図 4.21 ~ 図 4.27 にイーサネットドライバ API とその下位関数のフローを示します。

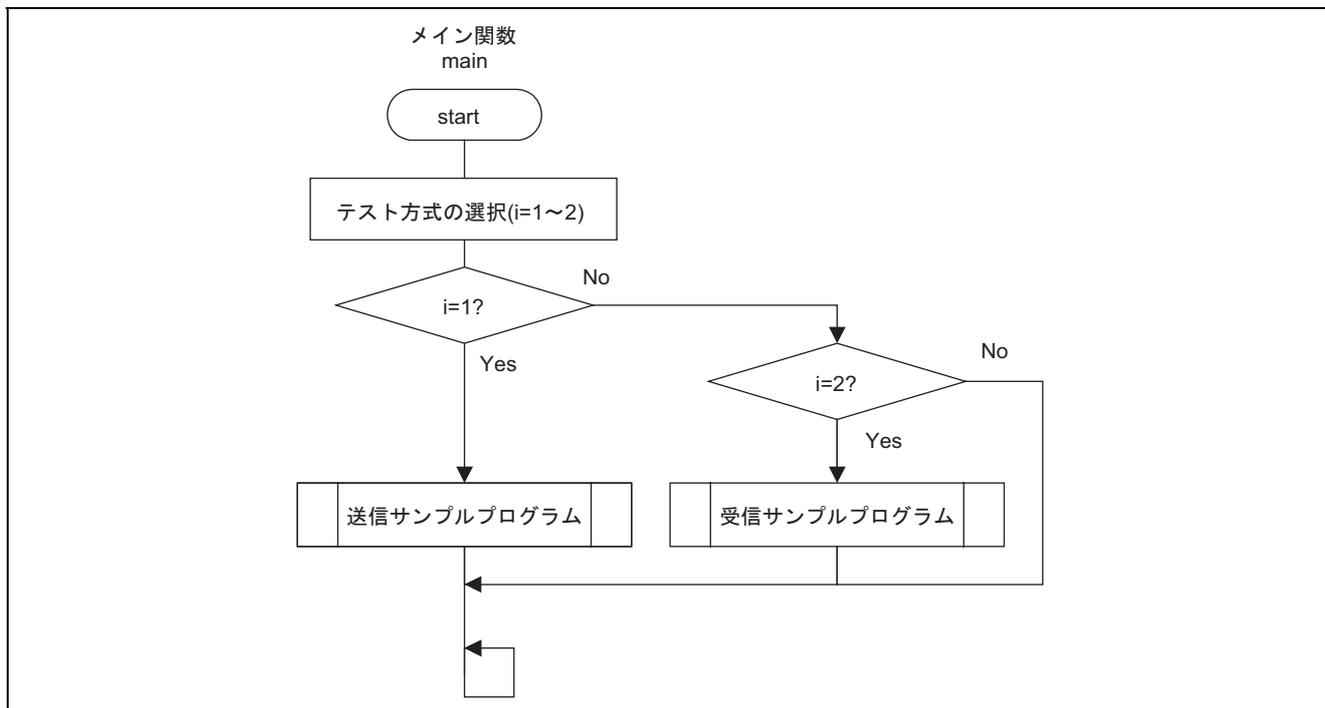


図 4.19 参考プログラムのメイン処理フロー例(1)

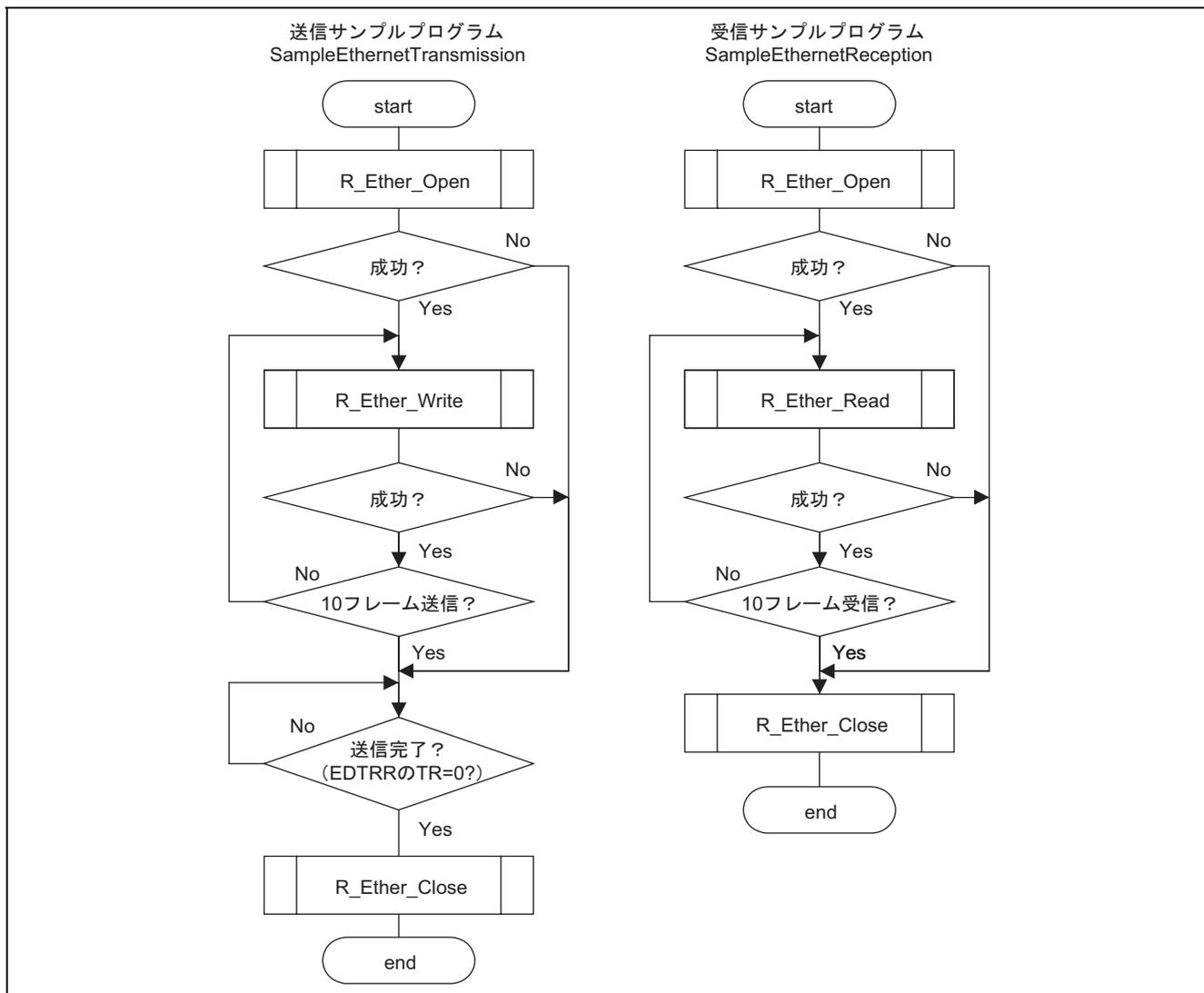


図 4.20 参考プログラムのメイン処理フロー例(2)

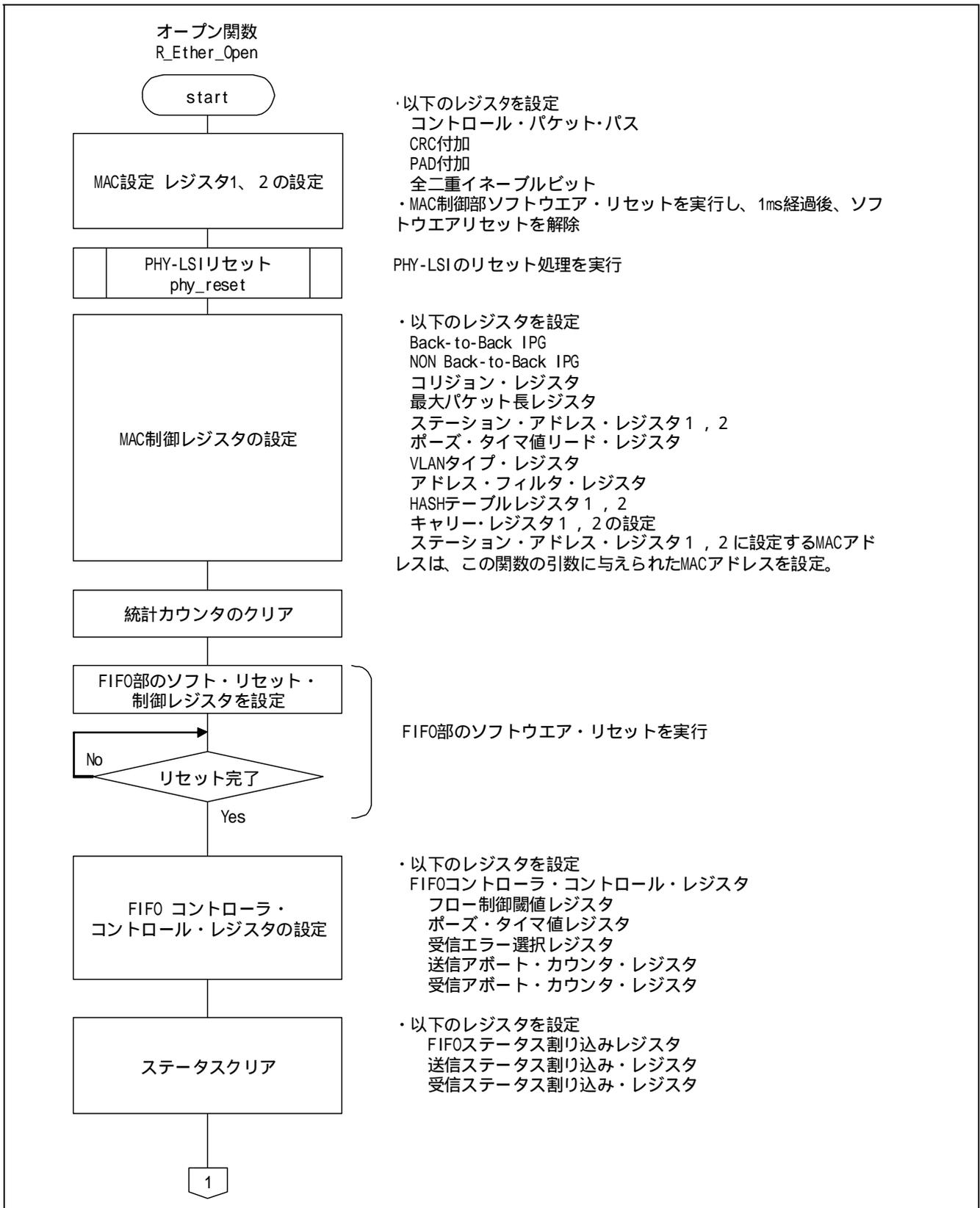


図 4.21 参考プログラムのイーサネット API 処理フロー(1)

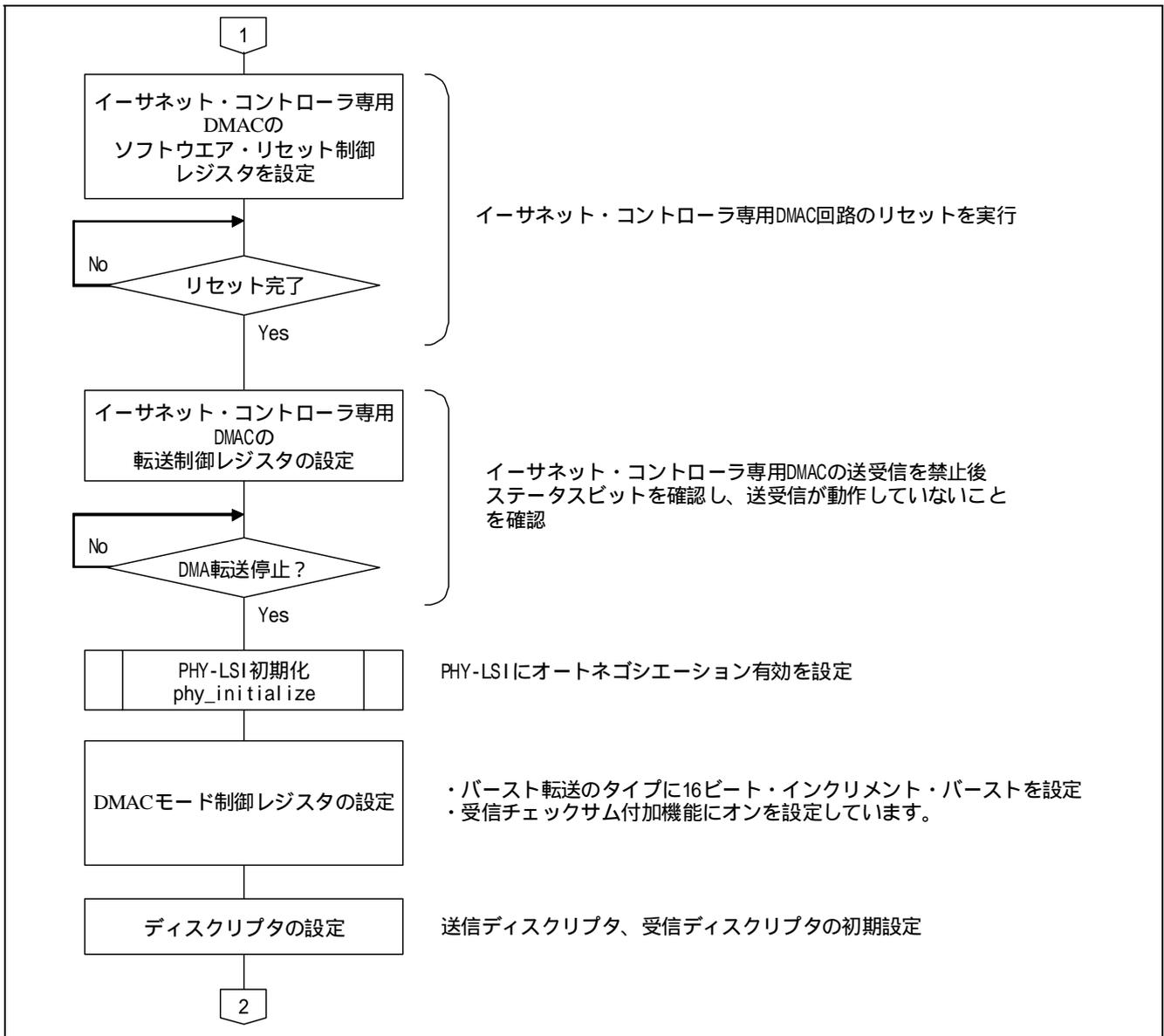


図 4.22 参考プログラムのイーサネット API 処理フロー(2)

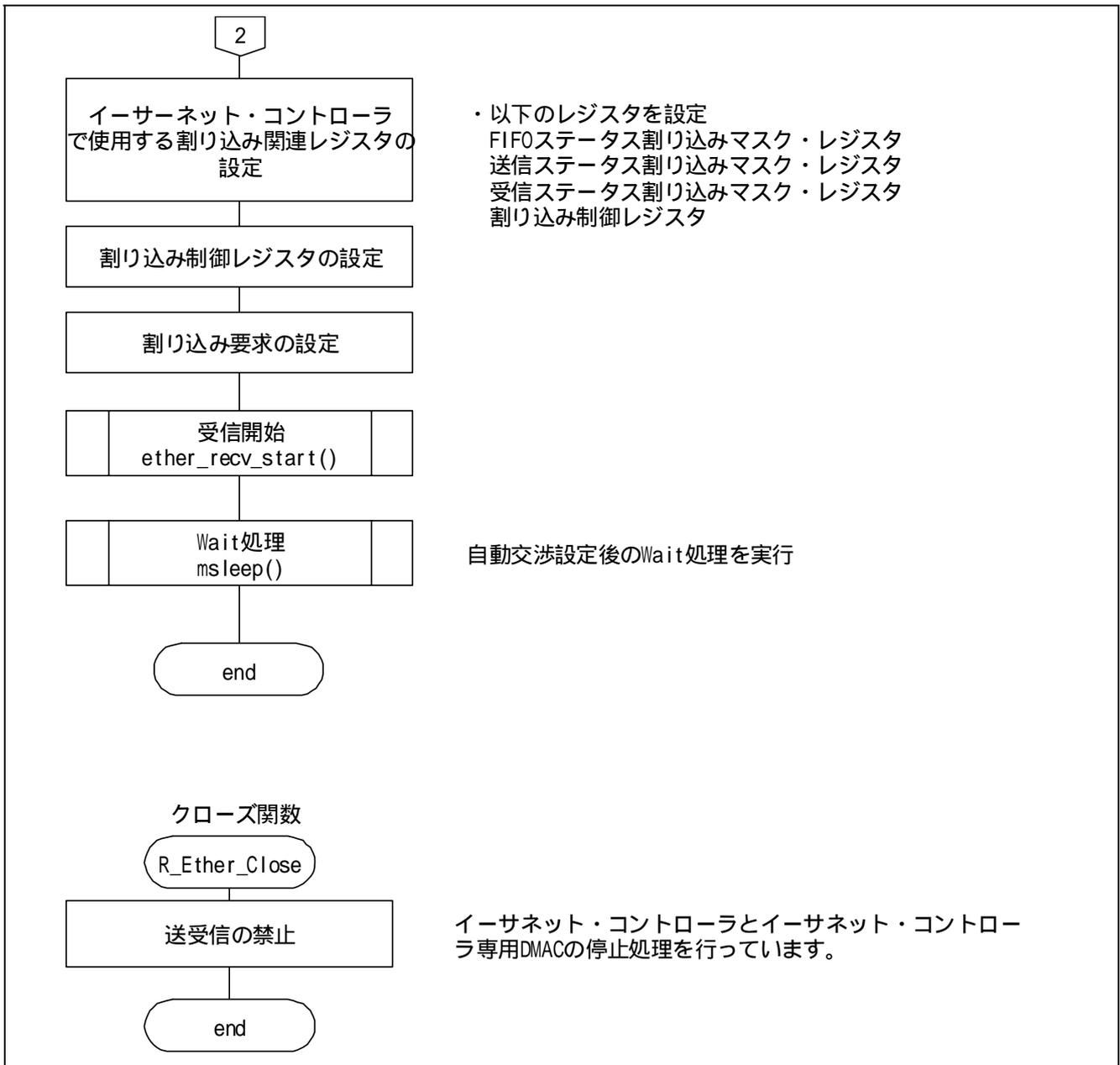


図 4.23 参考プログラムのイーサネット API 処理フロー(3)

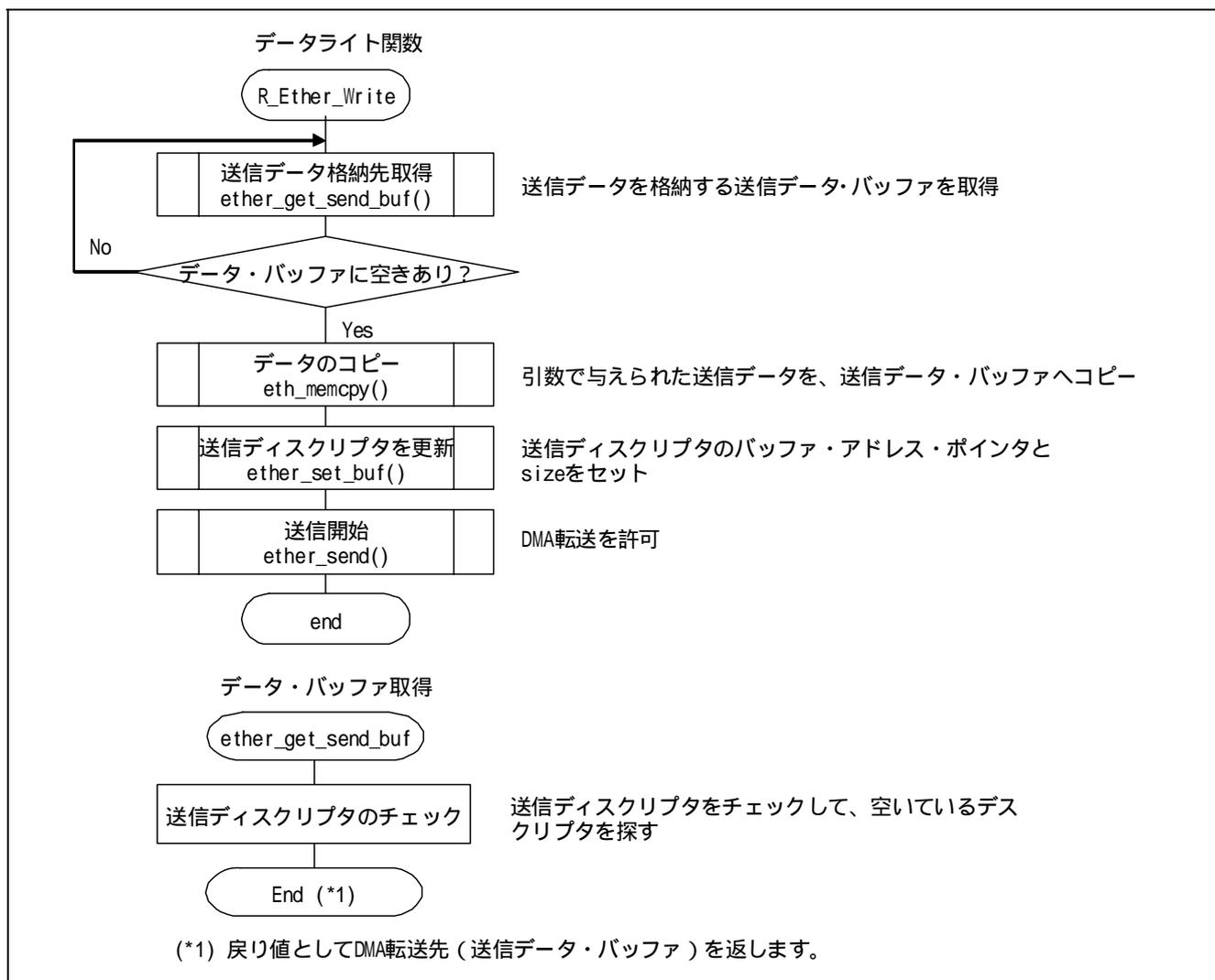


図 4.24 参考プログラムのイーサネット API 処理フロー(4)

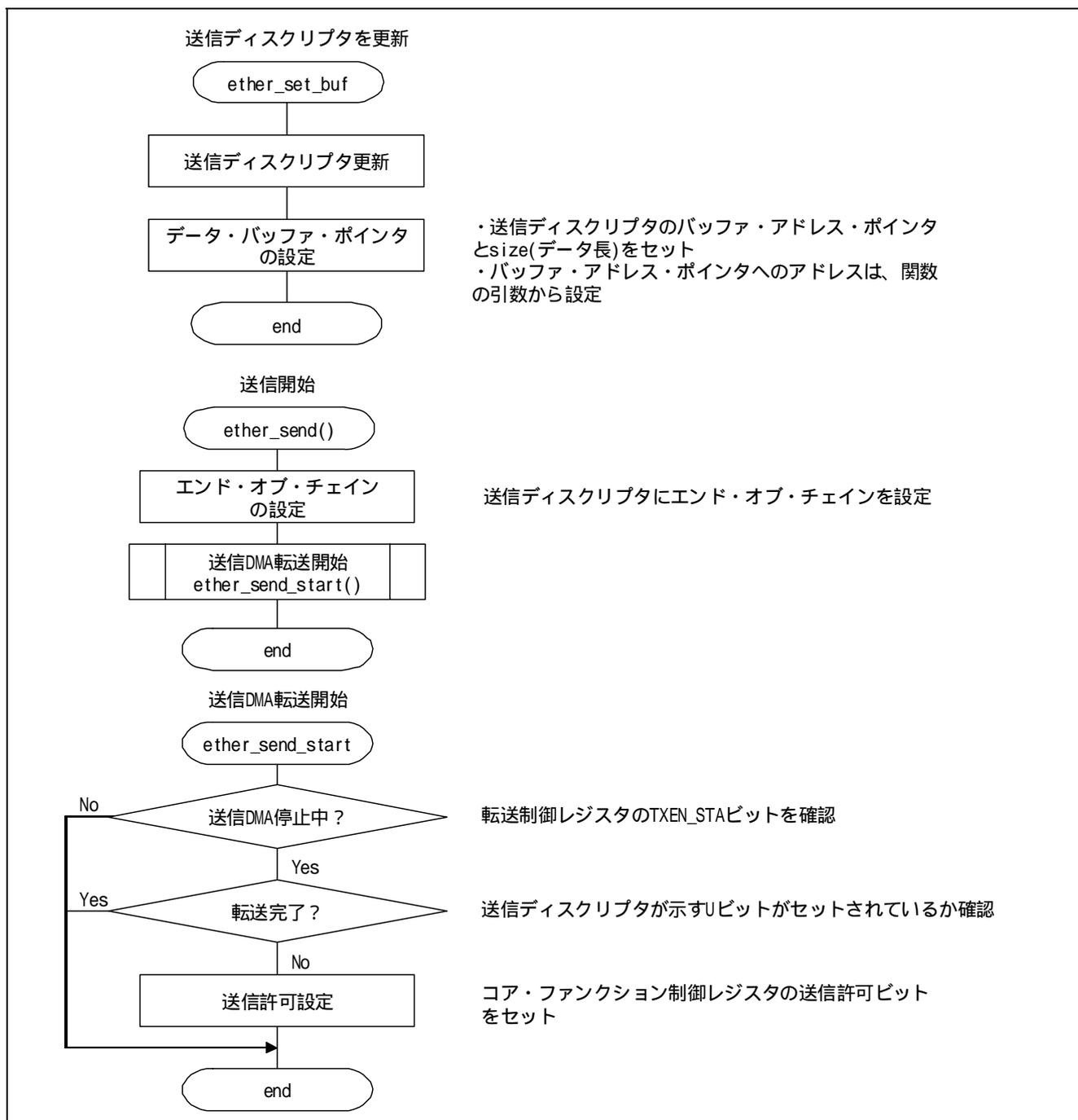


図 4.25 参考プログラムのイーサネット API 処理フロー(5)

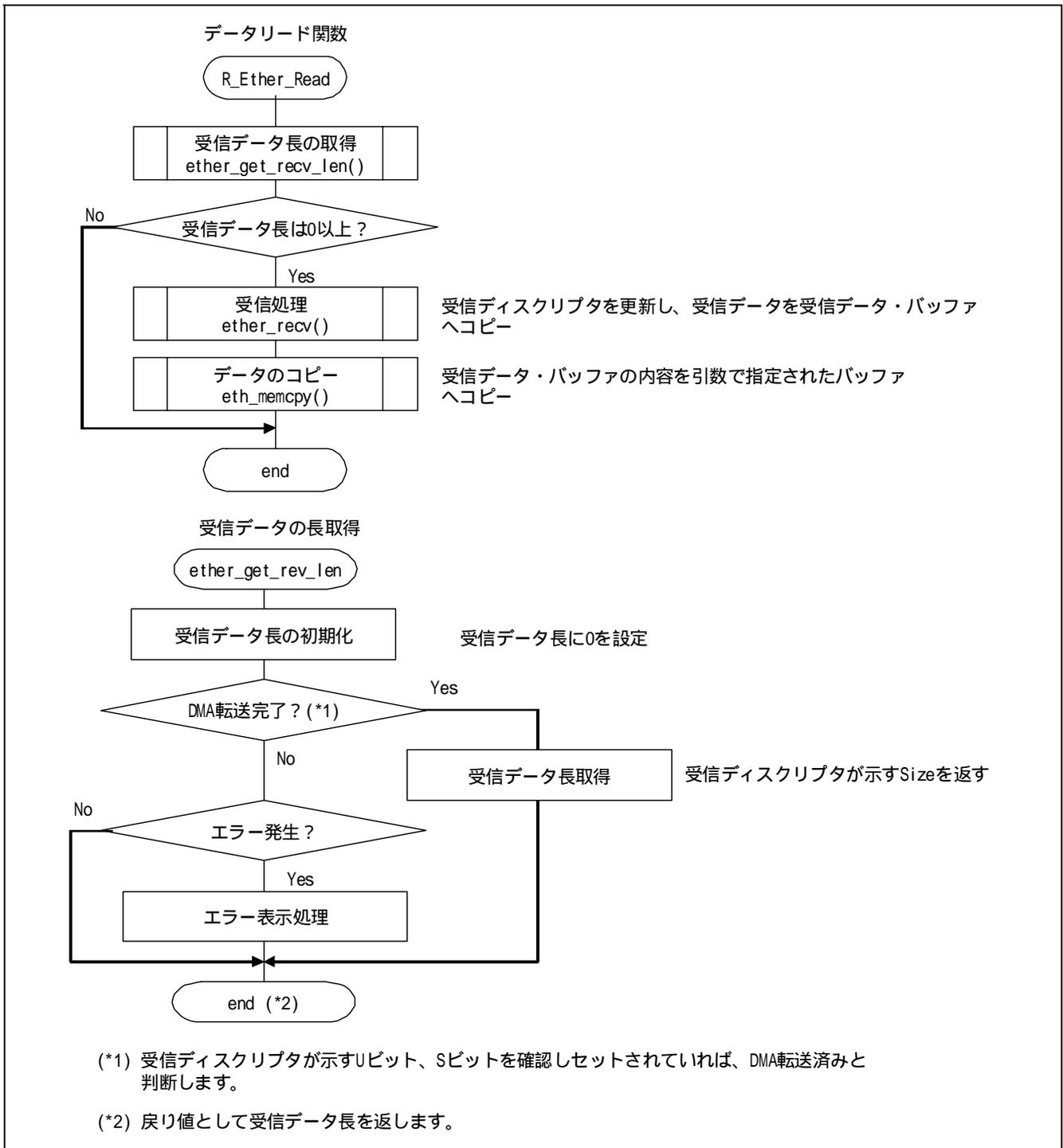


図 4.26 参考プログラムのイーサネット API 処理フロー(6)

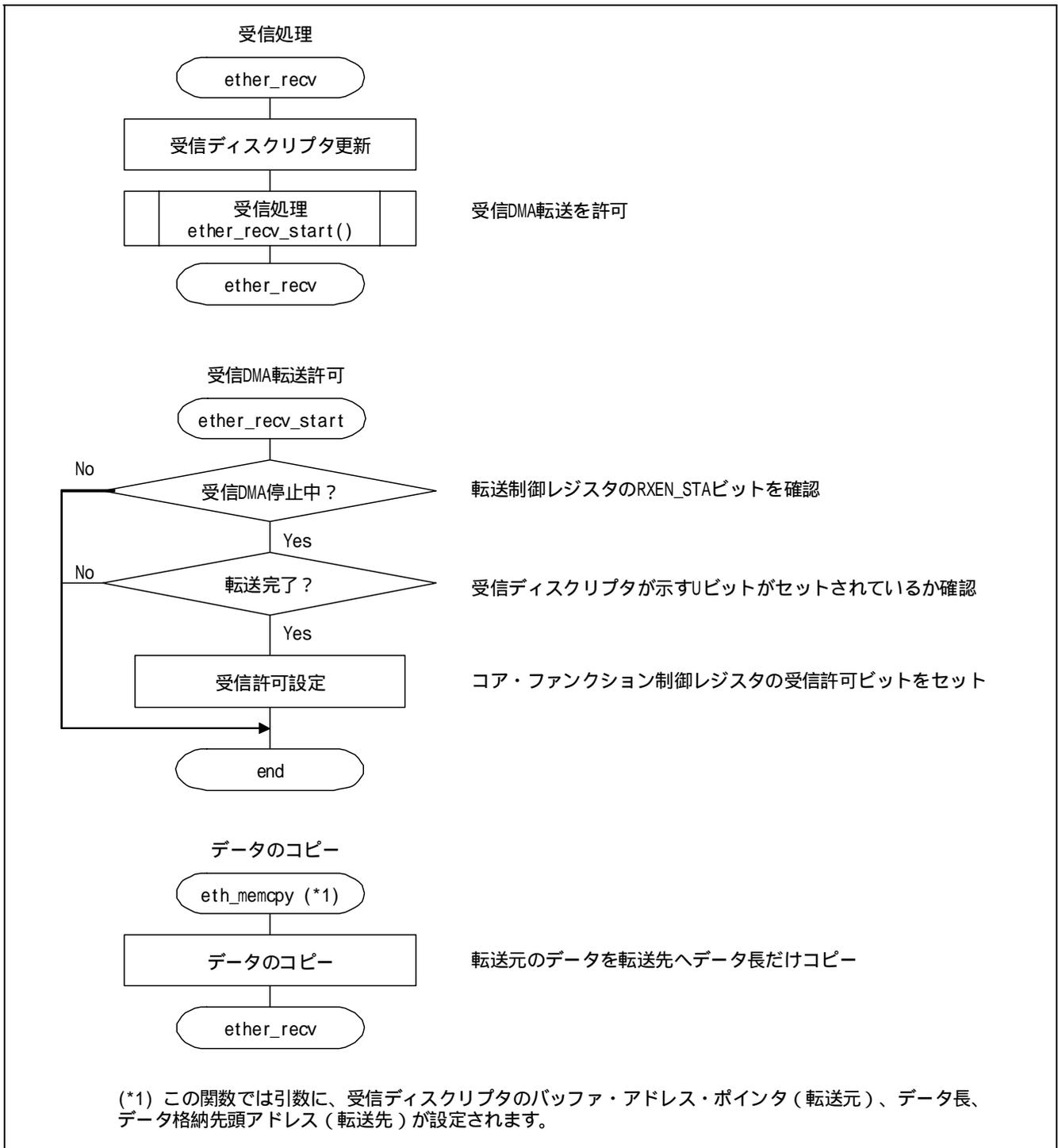


図 4.27 参考プログラムのディスクリプタ処理フロー(7)

## 4.7 送受信設定の注意点

### 4.7.1 動作環境の注意点 1

参考プログラムは通信方式の選択に自動交渉機能を使用しています。V850E2/ML4 と接続相手（図 4.15 ではハブ）の自動交渉完了までの時間差が大きい場合、自動交渉が成功していても通信に失敗することがあります。

自動交渉に成功していても V850E2/ML4 の接続相手が受信できない場合、接続相手が受信可能な状態になるまで V850E2/ML4 側で待ち時間を挿入してください。接続相手が受信可能な状態になるまでの時間はシステムによって異なるため、挿入する待ち時間はお客様で十分に評価してください。

ウェイトする時間を修正する場合、“ether\_driver.c”ファイルの R\_EtherOpen 関数内( 979 行目 )の下記コードの引数を修正ください。

初期設定では 1000 ( 1 秒 ) に設定されています。引数に与える時間は 1 カウント 1ms になっています。

```
msleep(1000);
```

### 4.7.2 動作環境の注意点 2

10Base-T リピータ HUB は Auto-Negotiation に対応していないため、10Base-T リピータ HUB を使用して参考プログラムを動作させる場合、“ether\_phy.c”ファイルの phy\_initialize 関数内 ( 150 行目 ) の下記コードを変更してください。

<変更前>

```
setval = PHY_CONTROL_SPEEDSEL | PHY_CONTROL_DUPLEX | PHY_CONTROL_ANEGENB;
```

<変更後>

```
setval = PHY_CONTROL_ANEGENB;
```

### 4.7.3 エラー処理に関する注意点

参考プログラムは送受信エラー処理が実装されていません。エラー処理が必要になる場合は別途お客様でプログラムを作成して頂く必要があります。

### 4.7.4 送受信バッファ定義 RAM の注意点

V850E2/ML4 のイーサネット・コントローラ専用 DMAC がアクセスできる領域は、H バス共有メモリのみです。イーサネット・コントローラで使用する送受信ディスクリプタとデータ・バッファは、H バス共有メモリに配置してください。

## 5. 参考プログラムのセクション配置

参考プログラムのセクション配置を表 5.1 で示します。

コンパイラに関する詳細は「CubeSuite+ V1.00.00 統合開発環境 ユーザーズマニュアル V850 ビルド編」をご参照ください。

表 5.1 参考プログラムのセクション配置

セクション名	アドレス	サイズ	説明
RESET	0x00000000	0x00000004	リセット
INTTAUA0I0	0x000003b0	0x00000004	TAUA0 チャンネル 0
INTETHA0SRX	0x00000ba0	0x00000004	イーサネット受信パケット読み出し要求
INTETHA0SCRX	0x00000bb0	0x00000004	イーサネット・パケット受信
INTETHA0SCTX	0x00000bc0	0x00000004	イーサネット・パケット送信
INTETHA0RS	0x00000bd0	0x00000004	イーサネット受信ステータス検出
INTETHA0TS	0x00000be0	0x00000004	イーサネット送信ステータス検出
INTETHA0FS	0x00000bf0	0x00000004	イーサネット FIFO ステータス検出
INTETHA0MAC	0x00000c00	0x00000004	イーサネット統計カウンタ・オーバフロー
.pro_epi_runtime	0x00000c08	0x000001e0	プロログ/エピログ・ランタイム呼び出し
.text	0x00000c08	0x000001e0	.text セクション
eth_memory.bss	0xf9800000	0x000029d8	イーサネットディスクリプタとデータ・バッファ
.sdata	0xfedf0000	0x00000030	.sdata セクション (初期値あり変数)
.sbss	0xfedf0030	0x00003cb8	.sbss セクション (初期値なし変数)
.bss	0xfedf3ce8	0x00000200	.bss セクション

## 6. 参考ドキュメント

- V850E2/ML4 ユーザーズマニュアル ハードウェア編 (R01UH0262JJ0001\_V850E2ML4.pdf)
- CubeSuite V1.00.00 統合開発環境ユーザーズマニュアル コーディング編 (r20ut0554jj0100\_qscdxc.pdf)
- CubeSuite V1.00.00 統合開発環境ユーザーズマニュアル V850 ビルド編 (r20ut0557jj0100\_qsb850.pdf)
- Standard Microsystems Corporation の LAN8700/LAN8700i データシート  
(最新版を Standard Microsystems Corporation ホームページから入手してください)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

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

お問い合わせ先

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

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.06.14	—	初版発行

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本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ください。

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. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、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。
2. 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。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じて、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は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、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。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。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、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用用途に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

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。



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/contact/>