

ハード・プロセッサ・システム (HPS) は、2つのイーサネット・メディア・アクセス・コントローラ (EMAC) ペリフェラルを提供しています。各 EMAC は、IEEE 802.3 仕様に準拠したイーサネット接続を介して、10/100/1000 Mbps でのデータ送信 / 受信に使用できます。EMAC は、Synopsys® DesignWare® 3504-0 Universal 10/100/1000 Ethernet MAC (DWC_gmac) のインスタンスです。

EMAC には大規模なメモリ・マップドのコントロールおよびステータス・レジスタ (CSR) セットがあり、それらは ARM Cortex™-A9® MPCore™ によってアクセスできます。

この章を理解するには、IEEE 802.3 のメディア・アクセス・コントロール (MAC) の基本事項を熟知している必要があります。

 IEEE 802.3 MAC について詳しくは、IEEE のウェブサイト (standards.ieee.org/findstds/) で使用可能な *IEEE Std 802.3-2008 Part 3: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications* を参照してください。

イーサネット MAC の機能

ここでは、EMAC ペリフェラルでサポートされている機能の概要を示します。

MAC

- IEEE 802.3-2008 対応
- 10/100/1000 Mbps のデータ・レート
- 全二重モードおよび半二重モード
- フロー・コントロール入力デアサーションでのゼロ・クアンタ・ポーズ・フレームの IEEE 802.3x フロー・コントロール自動送信
- 受信したポーズ・コントロール・フレームのユーザーへの転送オプション
- 1000 Mbps の半二重でのパケット・バーストおよびフレーム拡張
- 全二重での IEEE 802.3x フロー・コントロール
- 半二重のバック・プレッシャ・サポート
- IEEE 1588-2002 および IEEE 1588-2008 のネットワーク・クロック同期の精度
- Energy Efficient Ethernet (EEE) 用の IEEE 802.3-az バージョン D2.0

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

Portions © 2011 Synopsys, Inc. Used with permission. All rights reserved. Synopsys & DesignWare are registered trademarks of Synopsys, Inc. All documentation is provided "as is" and without any warranty. Synopsys expressly disclaims any and all warranties, express, implied, or otherwise, including the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, and any warranties arising out of a course of dealing or usage of trade.

†Paragraphs marked with the dagger (†) symbol are Synopsys Proprietary. Used with permission.



- 受信フレーム用の IEEE 802.1Q バーチャル・ローカル・エリア・ネットワーク (VLAN) のタグ検出
- 受信パスでの送信および削除に対するプリアンプルおよび SFD (start-of-frame data) の挿入
- フレームごとにコントロール可能な自動 CRC (Cyclic Redundancy Check) およびパッド生成
- 受信フレームでの自動パッド /CRC 排除のオプション
- 標準的なフレームおよびジャンボ・イーサネット・フレーム (最大 16 KB のサイズ) をサポートしているプログラム可能なフレーム長
- 8 ステップで 40 ~ 96 ビット・タイムのプログラム可能なフレーム間ギャップ (IFG)

PHY インタフェース

- 10/100/1000 用の RGMII (Reduced Gigabit Media Independent Interface)
- Management Data Input/Output (MDIO) (IEEE 802.3) または I²C PHY マネージメント・インタフェース

DMA インタフェース

- 32 ビット・インタフェース
- オptional・バス使用量にプログラム可能なバースト・サイズ
- シングル・チャンネル・モードの送信エンジンおよび受信エンジン
- データ・バッファ・サポートのためのバイト・アラインメントされたアドレッシング・モード
- デュアル・バッファ (リング) またはリンク・リスト (チェイン) ディスクリプタ・チェイニング
- 各転送で最大 8 KB のデータが可能なディスクリプタ

マネージメント・インタフェース

- CSR セットへの 32 ビットのホスト・インタフェース
- 通常モードおよびエラーを含む転送での包括的なステータス・レポート
- さまざまな動作条件用にコンフィギュレーション可能な割り込みオプション
- フレーム毎の送信 / 受信完了の割り込みコントロール
- 送信パケットおよび受信パケットに対して返される個別のステータス

アクセラレーション

- 送信コントロール・プロトコル (TCP)、ユーザー・データグラム・プロトコル (UDP)、またはインターネット・プロトコル (IP) を介したインターネット・コントロール・メッセージ・プロトコル (ICMP) 用の送信および受信のチェックサム・オフロード

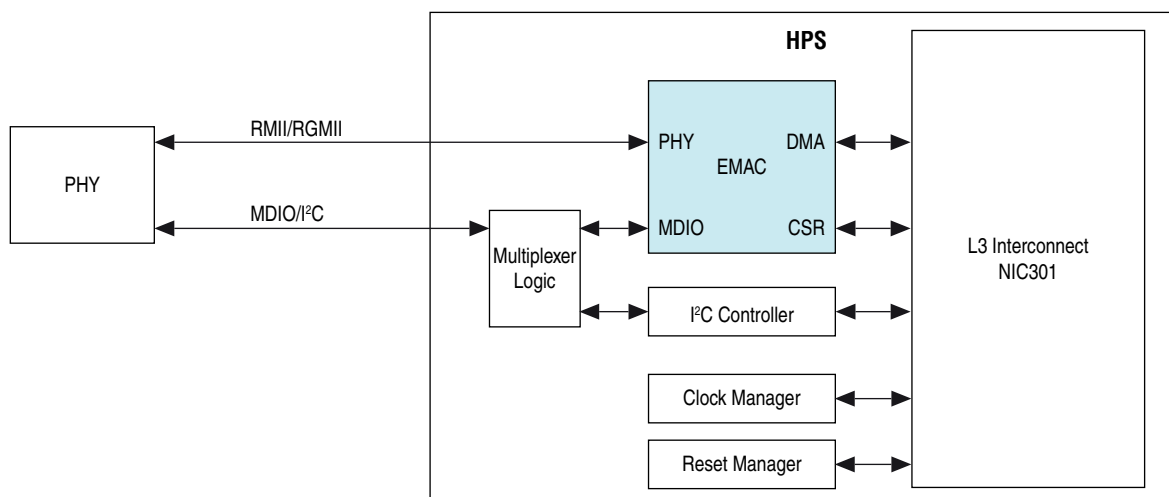
その他の機能

- 柔軟性の高いさまざまなアドレス・フィルタリング・モードのサポート
- バイトごとにマスクのある最大 31 個の追加の 48 ビット完全デスティネーション・アドレス (DA) フィルタ
- バイトごとにマスクのある最大 31 個の 48 ビット・ソース・アドレス (SA) 比較チェック
- マルチキャストおよびユニキャスト DA の 256 ビット・ハッシュ・フィルタ (オプション)
- すべてのマルチキャスト・アドレス指定されたフレームを渡すオプション
- ネットワーク・モニタ用のフィルタがないすべてのフレームを渡すプロミスキャス・モード・サポート
- ステータス・レポートのある受信パケットを (フィルタ毎として) 渡す機能

EMAC のブロック図およびシステム統合

EMAC は、SoC (system on a chip) FPGA デバイスの HPS 部分に統合されます。EMAC は I/O ピンを使用して通信します。17-3 ページの 図 17-1 に、ハイレベルの観点からの EMAC 統合を示します。

図 17-1. EMAC のシステム統合



EMAC から RGMII へのインタフェース

表 17-1 に、PHY データパス I/O を示します。

表 17-1. 外部 PHY データ・インタフェース

EMAC ポート	I/O	幅	説明
clk_tx_i	入力	1	送信クロックです。これは RGMII から提供される送信クロック (1G/100M/10Mbps での 125/25/2.5 MHz) です。EMAC で生成されるすべての PHY 送信信号はこのクロックに同期します。
phy_txd_o	出力	8	PHY 送信データです。これは、MAC で駆動される 8 個の送信データ信号のグループです。RGMII インタフェースのコンフィギュレーションで未使用のビットは、Low に固定されます。 RGMII : ビット [3:0] は RGMII 送信データを提供します。データ・バスは、送信クロック (clk_tx_i) の立ち上がりエッジおよび立ち下がりエッジの両方で変更されます。データの有効性は phy_txen_o で検証されます。clk_tx_i、clk_tx_180_i に同期します。
phy_txen_o	出力	1	PHY 送信データ・イネーブルです。この信号は、EMAC コンポーネントによって駆動されます。 RGMII : この信号は、送信データのコントロール信号 (rgmii_tctl) であり、クロックの両方のエッジで駆動されます。clk_tx_i、clk_tx_180_i に同期します。
rst_clk_tx_n_o	出力	1	送信クロック・リセット出力です。
clk_rx_i	入力	1	受信クロックです。クロック周波数は 1G/100M/10Mbps モードで 125/25/2.5 MHz です。これは外部 PHY から提供されます。EMAC で受信されるすべての PHY 信号はこのクロックに同期します。
phy_rxd_i	入力	8	PHY 受信データです。これは PHY から受信される 8 個のデータ信号のバンドルです。 RGMII : ビット [3:0] は RGMII 受信データを提供します。データ・バスは、受信クロック (clk_rx_i) の立ち上がりエッジおよび立ち下がりエッジの両方でサンプリングされます。データの有効性は phy_rxdv_i で検証されます。clk_rx_i、clk_rx_180_i に同期します。
phy_rxdv_i	入力	1	PHY 受信データ・バリッドです。この信号は PHY によって駆動されます。 RGMII : これは、phy_rxd で受信されるデータを検証するために使用される受信コントロール信号です。この信号は、クロックの両方のエッジでサンプリングされます。clk_rx_i、clk_rx_180_i に同期します。
rst_clk_rx_n_o	出力	1	受信クロック・リセット出力です。
phy_intf_sel_i[1:0]	入力	2	PHY インタフェース選択 : これらのピンは、EMAC の PHY インタフェースのうち 1 つを選択します。これはリセットのアサート中のみにサンプリングされ、その後は無視されます。 ■ 01 : RGMII ■ 00、10 および 11 : 無効
clk_ref_i	入力	1	これは、EMAC への基準クロックです。このクロックは、クロック・マネージャから提供される emac0_clk または emac1_clk です。 システム・マネージャは phy_intf_sel 信号を駆動して、どのクロックが使用されるか制御します。 クロック・レートは 250 MHz です。

PHY マネージメント・インタフェース

HPS は、MDIO または I²C のどちらか一方の PHY マネージメント・インタフェースへのサポートを提供できます。

MDIO インタフェース

MDIO インタフェース信号は、サポートされているすべてのモードで l4_mp_clk に同期します。

表 17-2. PHY MDIO マネージメント・インタフェース

信号	I/O	幅	説明
gmii_mdi_i	入力	1	管理データの入力です。PHY はこの信号を生成して、リード動作中にレジスタ・データを転送します。この信号は gmii_mdc_o クロックに同期して駆動されます。
gmii_mdo_o	出力	1	管理データの出力です。EMAC はこの信号を使用して PHY に対してコントロールとデータの情報を転送します。
gmii_mdo_o_e	出力	1	管理データの出力イネーブルです。このイネーブル信号は、外部 3 ステート I/O バッファから gmii_mdo_o 信号を駆動します。バリッド・データが gmii_mdo_o 信号上で駆動されるときはいつでもこの信号がアサートされます。この信号のアクティブ・ステータスは High です。
gmii_mdc_o	出力	1	管理データ・クロックです。EMAC は、この非周期的なクロックを通して gmii_mdi_i 信号および gmii_mdo_o 信号用のタイミング基準を MII に提供します。このクロックの最大周波数は 2.5 MHz です。このクロックは、クロック・ディバイダを通してアプリケーション・クロックから生成されます。

I²C 外部 PHY マネージメント・インタフェース

いくつかの PHY デバイスは、コントロール・インタフェース用に MDIO の代わりに I²C を使用します。スモール・フォーム・ファクタ・プラグ (SFP) のオプティカル・モジュールまたはプラグ可能モジュールは、このインタフェースを持っているものの間にあることが多いです。

HPS は、PHY デバイスを制御するために 4 つの汎用 I²C ペリフェラルのうち 2 つを使用できます。

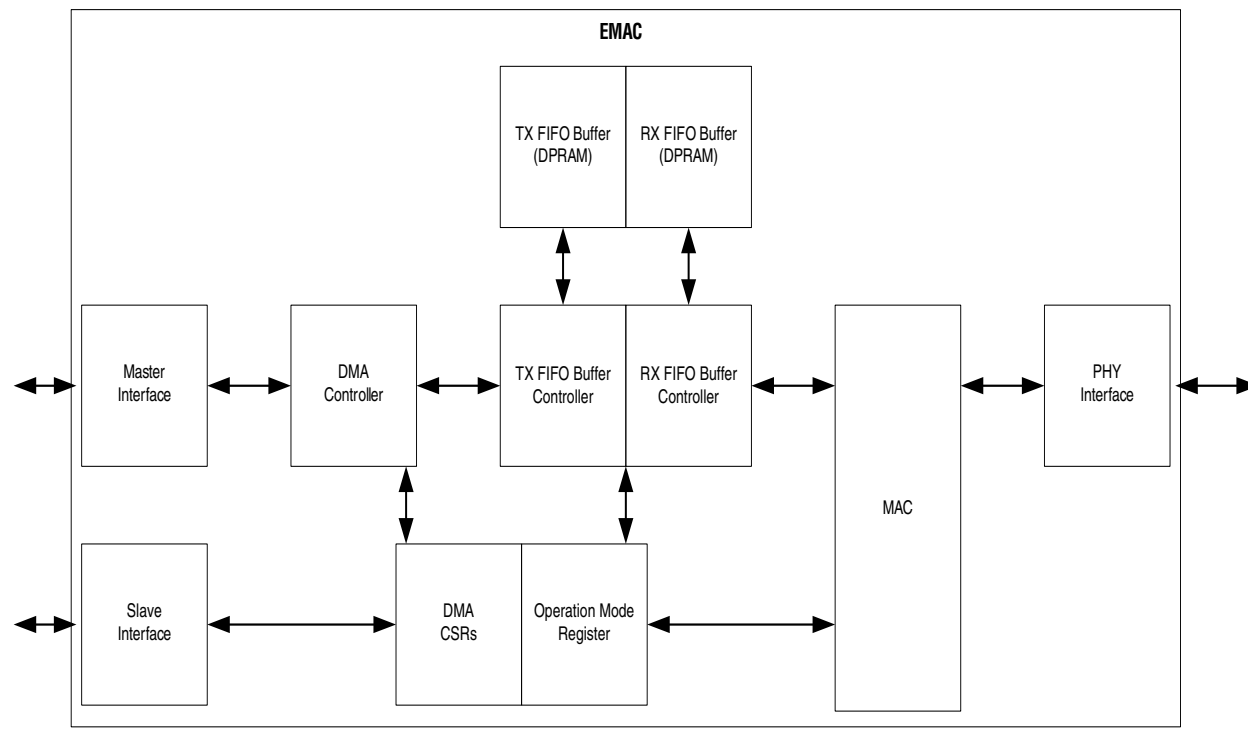
IEEE 1588

EMAC は、すべてのモードでの IEEE 1588 動作を 1 μ s の精度でサポートしています。内部の 2 つの MAC に対するタイム・カウンタ間の同期を維持するために ARM[®] Cortex[™]-A9 マイクロプロセッサ・ユニット (MPU) サブシステムで使用できます。

EMAC の機能の説明

図 17-2 に、それらのインタフェースを持っている EMAC のハイレベルなブロック図を示します。

図 17-2. EMAC のブロック図



MAC へのホスト・インタフェースは 2 つあります。32 ビット・スレーブ・インタフェースのマネージメント・ホスト・インタフェースは、CSR セットにアクセスできます。データ・インタフェースは 32 ビット・インタフェースです。このインタフェースは、ダイレクト・メモリ・アクセス (DMA) コントローラ・チャネルおよび HPS システムの残りの部分の間でのデータ転送を NIC-301 L3 インタコネクトを介して制御します。

MAC コントローラおよびシステム・メモリの間でのデータ転送に最適化されたビルトイン DMA コントローラがあります。DMA コントローラには、個別の送信エンジンと受信エンジン、および CSR セットがあります。送信エンジンは、システム・メモリからデバイス・ポートにデータを転送します。一方、受信エンジンは、デバイス・ポートからシステム・メモリにデータを転送します。このコントローラは、ディスクリプタを使用して、ホストの介入を最小限にとどめてソースからデスティネーションに効率よくデータを移動させます。

また、EMAC には、アプリケーション・システム・メモリと EMAC コントローラの間でイーサネット・フレームをバッファして制御する FIFO バッファ・メモリが含まれています。送信では、イーサネット・フレームは送信 FIFO バッファ (1024 x 42 ビット) に読み出して、最終的に MAC をトリガして転送を実行します。受信するイーサネット・フレームは受信 FIFO バッファに格納されて、DMA コントローラに FIFO バッファのフィル・レベルを示します。そして、DMA コントローラはコンフィギュレーションされたバースト転送を開始します。受信および送信の転送ステータスは、両方とも MAC から取得されて DMA に転送されます。

ホスト・インタフェース

EMAC には、スレーブおよびマスタの 2 本のホスト・インタフェースがあります。マスタは、L3 インタコネクト・ブロックの L3 マスタ・ペリフェラル・スイッチ・インタフェースに接続されます。

スレーブ

EMAC CSR セット・アクセスはスレーブ・インタフェースによって提供されます。スレーブは Level 4 (L4) バスに接続されます。

マスタ

DMA インタフェースはマスタ・インタフェースから提供されます。データ・ディスクリプタおよび実際のデータ・パケットの 2 つのタイプのデータがこのインタフェースで転送されます。このインタフェースは、全二重イーサネット・パケット・トラフィックを非常に効率よく転送します。このポートによって、さまざまな DMA チャンネルから同時にリード・データおよびライト・データを転送できます。送信ディスクリプタの読み出しおよび書き戻しは同時に実行できないため、ここでの例外となります。

DMA 転送は、ソフトウェアのインタフェース上のコンフィギュレーション可能なバースト・トランザクション数に分割されます。dmagrp グループの AXI_Bus_Mode レジスタは、バースト動作をコンフィギュレーションするために使用されます。

インタフェースは、各 DMA チャンネルおよびチャンネル内のそれぞれのリード DMA リクエストまたはライト DMA リクエストに対して固有の ID を割り当てます。個別の ID を持っているデータ転送はリオーダーしてインタリーブできます。

書き込みデータ転送は、一般的に、インタコネクトがデータ・バーストの最後のビートを受け取ってすぐに、OK 応答を持っているポストッド・ライトとして実行されます。しかし、ディスクリプタ（ステータスまたはタイムスタンプ）は、転送完了割り込みロジックとの競合状態を避けるために、常にノンポストッド・ライトとして転送されます。

スレーブはエラー応答を発行する可能性があります。このような事態が発生する場合、EMAC は、元のリクエストを生成した DMA チャンネルをディセーブルして割り込み信号をアサートします。ホストは、DMA を再起動してこの状態から回復するためにハードまたはソフトのリセットを使用して EMAC をリセットする必要があります。

EMAC は、インタフェース上での最大 16 個の未処理のトランザクションをサポートしています。未処理のトランザクションをバッファすることでバック・プレッシャ動作を滑らかにします。システムの高負荷の条件下でリソース競合のボトルネックが発生する場合、この対処は重要です。

キャッシュ・コントロール・インタフェース

システム・マネージャは、このインタフェースを通してマスタ・キャッシュ出力の値を提供します。これらの入力、マスタ転送のキャッシュ可能特性について、このブロックの機能を拡張する L3 インタコネクトへの出力として使用されます。

EMAC の DMA コントローラをコンフィギュレーションしてキャッシュ可能アクセスを実行するには、システム・マネージャにキャッシュ・ビットを設定します。キャッシュ・ビットは、EMAC コントローラのリセット前のブート時のみにアクセスする必要があります。



詳しくは、*Cyclone V* デバイス・ハンドブック Volume 3 の *System Manager* の章を参照してください。

外部 PHY

以下の PHY インタフェースが HPS 用にサポートされています。

- 10/100/1000 の RGMII

また、EMAC には PHY のコンフィギュレーション用とステータス・モニタリング用に使用されるコントロール・インタフェースもあります。この場合、PHY はスレーブ・デバイスです。コントロール・インタフェースには 2 つの選択肢があります。

- MDIO

- I²C インタフェース

MDIO インタフェースは EMAC 内部に組み込まれており、I²C インタフェースは HPS にある I²C ペリフェラルを使用します。これらのインタフェースは EMAC の外部でマルチプレクサ化されます。

送信および受信データの FIFO バッファ

各 EMAC コンポーネントには、関連する送信データおよび受信データの FIFO バッファ・インタフェースがあります。どちらの FIFO バッファ・インタフェースも 1024 x 42 ビットです。FIFO バッファ・ワードは以下によって構成されています。

- データ : 32 ビット
- 側波帯 :
 - EOF (End of frame) : 1 ビット
 - BE (Byte enables) : 2 ビット
 - ECC (誤り訂正コード) : 7 ビット

データおよび側波帯は、7 ビットのシングル誤り訂正とダブル誤り検出 (SEC-DED) コード・ワードによって保護されています。また、これらの FIFO バッファの RAM には、ECC イネーブル・ピン、誤りインジェクション・ピンおよびステータス・ピンが含まれています。イネーブル・ピンおよび誤りインジェクション・ピンはシステム・マネージャによって駆動される入力であり、ステータス・ピンは MPU サブシステムに対して駆動される出力です。

IEEE 1588-2002 タイム・スタンプ

IEEE 1588-2002 規格は、分配されたデバイスのネットワークで高精度なクロックの同期を可能にする PTP (Precision Time Protocol) を定義します。PTP は、マルチキャスト・メッセージをサポートするローカル・エリア・ネットワークでのシステム通信に適用されます。このプロトコルは、システムを同期するための変動的な固有精度、分解能、および安定性を含む不均質システムをイネーブルします。これは、同期が必要なロボットなどの通信マシンの集合体のような自動化システムで頻繁に使用されて、そのため共通のタイム・ベース上で動作します。†

PTP は UDP/IP を介して伝送されます。システムまたはネットワークは、分配されているタイミングおよびクロックの情報によってマスタ・ノードおよびスレーブ・ノードに分類されます。図 17-3 に、PTP メッセージの交換によってスレーブ・ノードをマスタ・ノードに同期させるときの PTP の使用プロセスを示します。†

図 17-3. ネットワーク化されたタイム同期

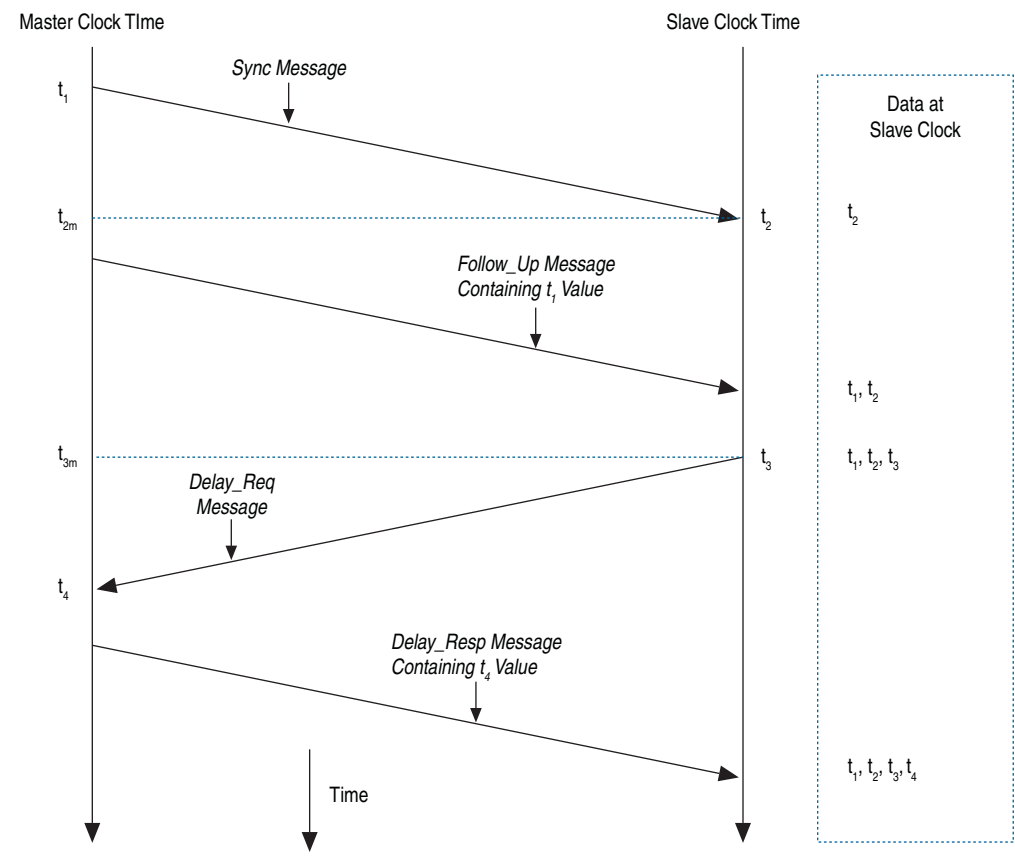


図 17-3 に示されているように、PTP は以下のプロセスを使用します。†

1. マスタは、すべてのノードに対して PTO Sync メッセージを配信します。Sync メッセージには、マスタの基準タイムの情報が含まれています。このメッセージがマスタ・システムを離れるのは t_1 のときです。このタイムは、イーサネット・ポートのために PHY インタフェースでキャプチャされる必要があります。†
2. スレーブは Sync メッセージを受信して、また、そのタイミング基準を使用して正確なタイム t_2 をキャプチャします。†
3. マスタはスレーブに対して、後で使用する t_1 情報が含まれている Follow_up メッセージを送信します。†
4. スレーブはマスタに対して Delay_Req メッセージを送信します。このメッセージは、このフレームが PHY インタフェースを離れた正確なタイム t_3 を記録しています。†
5. マスタはそのメッセージを受信して、システムにそれが入力されたときに正確なタイム t_4 をキャプチャします。†
6. マスタはスレーブに対して、Delay_Resp メッセージで t_4 情報を送信します。†
7. スレーブは、 t_1 、 t_2 、 t_3 、および t_4 の 4 つの値を使用して、そのローカル・タイミング基準をマスタのタイミング基準に同期させます。†

PTP 実装のほとんどは、UDP レイヤの上のソフトウェアで行われます。しかし、特別な PTP パケットが PHY インタフェースでイーサネット・ポートに入力されたり出力されたりする場合、正確なタイムをキャプチャするにはハードウェアのサポートが必要となります。PTP 実装を適切かつ高精度に行う上で、このタイミング情報は、キャプチャされてソフトウェアに返される必要があります。†

EMAC は、すべてのモードで IEEE 1588 動作を 1 μ s 精度でサポートすることを意図されています。2 つの EMAC が IEEE 1588 環境で動作している場合、MPU サブシステムは、それら 2 つの MAC 内部のタイム・カウンタ間の同期を維持します。†

FPGA への IEEE 1588 インタフェースによって、FPGA は、`emac_ptp_ref_clk` 入力用の代替ソースを提供できるようになると共に、各 EMAC コントローラからの毎秒のパルス出力をモニタできるようになります。†

EMAC コンポーネントは、ハードウェア支援の IEEE 1588 プロトコル実装を提供します。ハードウェア・サポートはタイムスタンプ・メンテナンス用です。PHY インタフェース上で受信する任意のフレームおよび受信ディスクリプタがこの値で更新されるとき、タイムスタンプが更新されます。また、フレームの SFD が送信されて送信ディスクリプタを更新するときもタイムスタンプが更新されます。†



IEEE 1588-2002 規格について詳しくは、IEEE Standards Association のウェブサイト (standards.ieee.org) で使用可能な *IEEE Standard 1588-2002 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems* を参照してください。†

基準タイミング・ソース

このタイムのスナップショットを得るには、EMAC は基準クロック入力を受け取って、それを使用して基準タイム（64 ビット）を内部で生成し、タイムスタンプをキャプチャします。†

システム・タイム・レジスタ・モジュール

64 ビットのタイムは、このモジュール内で維持されて、入力基準クロックの `osc1_clk` を使用して更新されます。`osc1_clk` クロックはクロック・マネージャから供給され、`emac_ptp_ref_clk` クロックは **FPGA** ファブリックから供給されます。このタイムは、PHY インタフェースで送信または受信されるイーサネット・フレームのスナップショット（タイムスタンプ）を得るためのソースです。

システム・タイム・カウンタは、粗い訂正方法を使用して初期化または訂正されます。この方法では、最初の値またはオフセット値はタイムスタンプ更新レジスタに書き込まれます。初期化では、各 **EMAC** のシステム・タイム・カウンタはタイムスタンプ更新レジスタの値が書き込まれます。一方システム・タイム訂正では、オフセット値がシステム・タイムに加算または減算されます。

細かい訂正方法では、マスタ・クロックに関するスレーブ・クロックの周波数のドリフトは、粗い訂正のときのように、1 クロックではなく、ある期間にわたって訂正されます。これは、線形タイムを維持するのに役立ちます。また、PTP Sync メッセージ・インターバル間の基準タイムにドラスティックな変化（大きなジッタ）をもたらしません。†

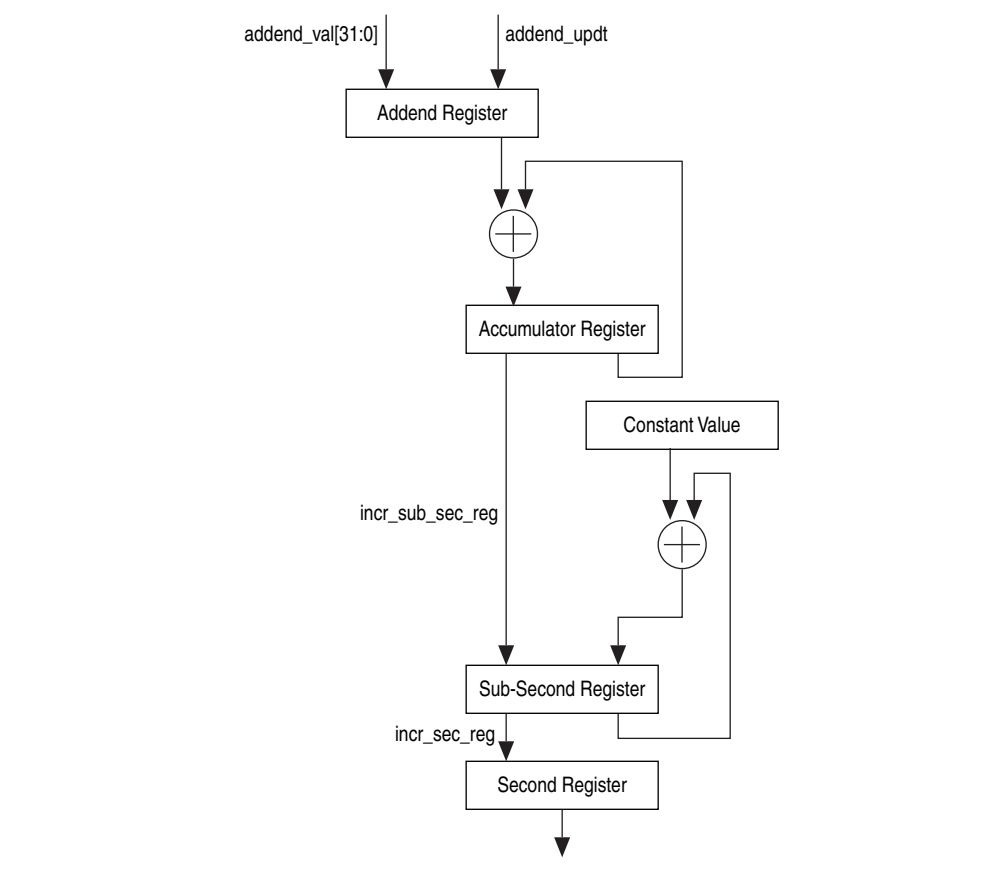
この方法では、 17-4 に示すように、アキュムレータは `Timestamp_Addend` レジスタの内容を合計します。アキュムレータが生成する演算キャリーは、システム・タイム・カウンタをインクリメントするパルスとして使用されます。アキュムレータおよび `Addend` は 32 ビットのレジスタです。ここで、アキュムレータは高精度周波数マルチプライヤまたはディバイダとして動作します。



指定された精度で必要な周波数よりも高い周波数で PTP クロックを接続する必要があります。†

このアルゴリズムは図 17-4 に示されています。†

図 17-4. 細かい（訂正）方法を使用したシステム・タイム更新



システム・タイム更新のロジックは、20 ns の精度を達成するために 50 MHz のクロック周波数を必要とします。周波数の分周比（FreqDivisionRatio）は、必要なクロック周波数に対する基準クロック周波数の比です。したがって、例えば基準クロック（clk_ptp_ref_i）が 66 MHz のとき、この比は 66 MHz / 50 MHz すなわち 1.32 と計算されます。したがって、レジスタに設定されるデフォルトの Addend 値は 232 / 1.32、0xC1F07C1F となります。

基準クロックが低周波数、例えば 65 MHz にドリフトする場合、比は 65 / 50 つまり 1.3 となり、Addend レジスタに設定する値は 232 / 1.30 すなわち 0xC4EC4EC4 となります。基準クロックが高周波数、例えば 67 MHz にドリフトする場合、Addend レジスタを 0xBF0B7672 に設定する必要があります。クロック・ドリフトがない場合、デフォルトの Addend 値の 0xC1F07C1F（232 / 1.32）をプログラムする必要があります。†

図 17-4 では、サブ・セカンド・レジスタを累積するために使用される定数値は十進数の 43 です。これはシステム・タイムで 20 ns の精度を達成します（つまり、20 ns ステップでインクリメントされます）。

ソフトウェアは、Sync メッセージに基づいて周波数のドリフトを計算して、Addend レジスタを更新する必要があります。†

まず、スレーブ・クロックは Addend レジスタの $\text{FreqCompensationValue}_0$ を使用して設定されます。この値は以下のように計算されます。†

$$\text{FreqCompensationValue}_0 = 232 / \text{FreqDivisionRatio} \uparrow$$

最初、 $\text{MasterToSlaveDelay}$ が連続する Sync メッセージ用に同じであると仮定される場合、以下に示すアルゴリズムが適用される必要があります。2 ~ 3 Sync サイクルの後、周波数のロックが起きます。その後、スレーブ・クロックは正確な $\text{MasterToSlaveDelay}$ の値を決定して、その新しい値を使用してマスタと再同期します。†

アルゴリズムは以下の通りです。†

- MasterSyncTime_n のときに、マスタはスレーブ・クロックに Sync メッセージを送信します。ローカル・クロックが SlaveClockTime_n であって MasterClockTime_n を以下のように計算するとき、スレーブはこのメッセージを受信します。†

$$\text{MasterClockTime}_n = \text{MasterSyncTime}_n + \text{MasterToSlaveDelay}_n \uparrow$$

- Sync サイクルのマスタ・クロック・カウン트의 $\text{MasterClockCount}_n$ は、次のように与えられます。†

$$\text{MasterClockCount}_n = \text{MasterClockTime}_n - \text{MasterClockTime}_{n-1}$$

($\text{MasterToSlaveDelay}$ が Sync サイクル n および $n-1$ と同じであると仮定します) †

- 現在の Sync サイクルのスレーブ・クロック・カウン트의 SlaveClockCount_n は、以下のように与えられます。†

$$\text{SlaveClockCount}_n = \text{SlaveClockTime}_n - \text{SlaveClockTime}_{n-1} \uparrow$$

- 現在の Sync サイクルのマスタおよびスレーブのクロック・カウン트의差である ClockDiffCount_n は、以下のように与えられます。†

$$\text{ClockDiffCount}_n = \text{MasterClockCount}_n - \text{SlaveClockCount}_n \uparrow$$

- スレーブ・クロックの周波数スケージングの FreqScaleFactor_n は、以下のように与えられます。†

$$\text{FreqScaleFactor}_n = (\text{MasterClockCount}_n + \text{ClockDiffCount}_n) / \text{SlaveClockCount}_n \uparrow$$

- Addend レジスタの周波数補正值の $\text{FreqCompensationValue}_n$ は、以下のように与えられます。†

$$\text{FreqCompensationValue}_n = \text{FreqScaleFactor}_n \times \text{FreqCompensationValue}_{n-1} \uparrow$$

理論的には、このアルゴリズムは 1 つの Sync サイクルでロックを達成しますが、ネットワーク伝播遅延の変化と動作状態のために、いくつかのサイクルが費やされる可能性があります。†

このアルゴリズムは自動訂正です。何らかの理由で最初にスレーブ・クロックがマスタからの誤った値に設定された場合、アルゴリズムはさらなる Sync サイクルを使ってそれを訂正します。†

送信パス機能

フレームの SFD が PHY インタフェースに送信されると、MAC はタイムスタンプをキャプチャします。タイムスタンプをキャプチャしたいフレームは、フレームごとにコントロール可能です。つまり、キャプチャする必要のあるタイムスタンプかどうか表示するために、各送信フレームをマークできます。MAC は、PTP フレームを識別する送信フレームを処理しません。ユーザーは、タイムスタンプをキャプチャ

するフレームを指定する必要があります。MAC は、FPGA に実装されたハードウェアに対して、フレームの送信ステータスと共にタイムスタンプを返します。送信ディスクリプタのコントロール・ビットを使用できます。MAC は、このように特定の PTP フレームに自動的にタイムスタンプを接続する、対応する送信ディスクリプタ内のソフトウェアに対してタイムスタンプを返します。†

受信パス機能

MAC は、PHY インタフェースで受信するすべてのフレームのタイムスタンプをキャプチャします。DMA は、対応する受信ディスクリプタのソフトウェアに対してタイムスタンプを返します。タイムスタンプは、最後の受信ディスクリプタにのみ書き込まれます。†

タイムスタンプ・エラー・マージン

IEEE 1588 使用によれば、タイムスタンプは PHY インタフェースで送受信されるフレームの SFD でキャプチャされる必要があります。PHY インタフェースの送受信クロックが基準タイムスタンプ・クロック (`osc1_clk`) に同期していないため、タイムスタンプが非同期クロック・ドメイン間に移動するとき、微小なドリフトが生じるようにします。送信パスでは、キャプチャされて通知されるタイムスタンプには 2 つの PTP クロックの最大のエラー・マージンがあります。つまり、キャプチャされるタイムスタンプには、PHY インタフェースで SFD が送信された後に 2 クロック以内に与えられる基準タイミング・ソース値があるということです。†

同様に受信パスでも、エラー・マージンは 3 つの PHY インタフェース・クロック、プラス最大 2 つの PTP クロックです。SFD データが MAC の PHY インタフェースに達する前にこの定常的な遅延がシステム（またはリンク）にあることを PHY インタフェース・クロックが仮定しているために、ユーザーはエラー・マージンを無視できます。†

基準タイミング・クロックの周波数範囲

タイムスタンプの情報は、MAC クロック・ドメインから FPGA クロック・ドメインへ非同期クロック・ドメインをまたいで転送されます。そのため、2 つの連続したタイムスタンプのキャプチャ間に最小限の遅延が必要です。この遅延は、PHY インタフェースの 4 クロック・サイクルと PTP クロックの 3 クロック・サイクルです。2 つのタイムスタンプ・キャプチャの遅延がこの遅延よりも少ない場合、MAC は次のフレーム用のタイムスタンプ・スナッチショットを得ません。†

最大の PTP クロック周波数は、基準タイム (50 MHz のとき 20 ns) の最大限の分解能、および PTP クロックで動作しているロジックで達成可能なタイミング制約によって制限されています。更に、基準タイム・ソースの分解能または精度は、同期の精度を決定づけます。したがって、より高い PTP クロック周波数はより良いシステム性能を提供します。†

最小の PTP クロック周波数は、2 つの連続する SFD バイト間で必要なタイムに依存します。PHY インタフェース・クロック周波数が IEEE 1588 仕様によって固定されているため、表 17-3 に示すように、適切な動作で必要な最小の PTP クロック周波数は MAC の動作モードおよび動作速度によって異なります。†

表 17-3. 最小の PTP クロック周波数の例

モード	2 つの SFD 間の最小ギャップ	最小 PTP 周波数
100 Mbps の全二重動作	168 MII クロック (64 バイト・フレーム用の 128 クロック + 最小 IFG 用の 24 クロック + プリアンブル用の 16 クロック)	$(3 * \text{PTP}) + (4 * \text{MII}) \leq 168 * \text{MII}$ 、つまり ~0.5 MHz $((168 - 4) * 40 \text{ ns} \div 3 = 2180 \text{ ns}$ の期間)
1000 Mbps の半二重動作	24 GMII クロック (衝突のために SFD 直後に送信されるジャム・パターン用の 4 + 12 IFG + 8 プリアンブル) (1)	$3 * \text{PTP} + 4 * \text{GMII} \leq 24 * \text{GMII}$ 、つまり 18.75 MHz

表 17-3 の注：
(1) ジャム・パターンについて詳しくは、IEEE のウェブサイト (standards.ieee.org/findstds/) で使用可能な *IEEE Std 802.3-2008 Part 3: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications* を参照してください。

IEEE 1588-2008 アドバンスド・タイムスタンプ

IEEE 1588-2002 タイムスタンプで説明した基本的なタイムスタンプ機能に加えて、EMAC は IEEE 1588-2008 規格で定義される以下のアドバンスド・タイムスタンプ機能をサポートしています。†

- IEEE 1588-2008 (バージョン 2) タイムスタンプ・フォーマットをサポートしています。†
- すべてのフレームまたは PTP タイプのフレームのみのタイムスタンプを取得するオプションが用意されています。†
- イベント・メッセージのみのタイムスタンプを取得するオプションが用意されています。†
- 一般的なタイプ、境界タイプ、エンド・ツー・エンド・タイプ、またはピア・ツー・ピア・タイプのクロック・タイプに基づいてタイムスタンプを取得するオプションが用意されています。†
- 一般的なクロックおよび境界タイプのマスタまたはスレーブに EMAC をコンフィギュレーションするオプションが用意されています。†
- イーサネットを介して直接送信された PTP メッセージのタイプ、バージョン、および PTP ペイロードを識別し、ステータスを送信します。†
- デジタル・フォーマットまたはバイナリ・フォーマットのサブ・セカンド・タイムを測定するオプションが用意されています。†

 IEEE 1588-2008 規格について詳しくは、IEEE Standards Association のウェブサイト (standards.ieee.org) で使用可能な *IEEE Standard 1588-2008 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems* を参照してください。†

ピア・ツー・ピア PTP トランスペアレント・クロック (P2P TC) メッセージ・サポート

IEEE 1588-2008 バージョンは、SYNC、遅延リクエスト、フォロー・アップ、および遅延応答メッセージに加えてピア・ツー・ピア PTP (Pdelay) メッセージをサポートしています。†

クロック・タイプ

EMAC は、IEEE 1588-2008 規格で定義される以下のクロック・タイプをサポートしています。†

- 一般的なクロック †
- 境界クロック †
- エンド・ツー・エンド・トランスペアレント・クロック †
- ピア・ツー・ピア・トランスペアレント・クロック †

基準タイミング・ソース

EMAC は、IEEE 1588-2008 規格で定義される以下の基準タイミング・ソース機能をサポートしています。†

- 48 ビット・セカンド・フィールド †
- 固定パルス・パー・セカンド出力 †
- フレキシブル・パルス・パー・セカンド出力 †
- 外部イベントを持っている補助スナップショット (タイムスタンプ)

送信パス機能

アドバンスド・タイムスタンプ機能は、代替 (エンハンスド・) ディスクリプタ・フォーマットを通してのみサポートされています。†

受信パス機能

MAC は、受信フレームを処理してバリッド PTP フレームを識別します。以下のオプションを使用することによって、タイムのスナップショットをアプリケーションに送信するように制御できます。†

- すべてのフレーム用にタイムスタンプをイネーブルします。†
- IEEE 1588 バージョン 2 または バージョン 1 のタイムスタンプ用にタイムスタンプをイネーブルします。†
- イーサネットまたは UDP/IP イーサネットを介して直接送信される PTP フレーム用にタイムスタンプをイネーブルします。†
- IPv4 または IPv6 の受信フレーム用にタイムスタンプ・スナップショットをイネーブルします。†
- EVENT メッセージ (SYNC、DELAY_REQ、PDELAY_REQ、または PDELAY_RESP) 用のみにタイムスタンプ・スナップショットをイネーブルします。†
- ノードをマスタまたはスレーブにイネーブルしてタイムスタンプ・タイプを選択します。これは、取得するタイムスタンプのメッセージのタイプを制御します。†

DMA は、対応する送信ディスクリプタまたは受信ディスクリプタ内のソフトウェアに対してタイムスタンプを返します。アドバンスド・タイムスタンプ機能は、32 ビットの代替（エンハンスド・）ディスクリプタのみでサポートされています。†

補助スナップショット

補助スナップショット機能は、システム・タイムのスナップショット（タイムスタンプ）を外部イベントに基づいて格納できるようにします。イベントは、側波帯信号 `ptp_aux_ts_trig_i` の立ち上がりエッジと見なされます。1 つの補助スナップショット入力を使用可能です。補助スナップショット FIFO バッファのデプスは 16 です。†

入力に要したタイムスタンプは共通の FIFO バッファに格納されます。ホストは、この FIFO バッファをトップで読み出すために使用可能な入力のタイムスタンプを確認するためにレジスタを読み出すことができます。MAC はこれらのタイムスタンプを FIFO バッファに格納します。タイムスタンプの 64 ビットのみ FIFO バッファに格納されます。タイムスタンプが格納されると、MAC は割り込みと共にホストに対してこれを表示します。タイムスタンプの値は、FIFO バッファ・レジスタ・アクセスを通して読み出されます。†

IEEE 802.3az Energy Efficient Ethernet

IEEE 802.3-az のバージョン D2.0 で標準化されている Energy Efficient Ethernet（EEE）は、EAC でサポートされています。10/100/1000 Mbps レートでの MAC 動作でサポートされます。EEE は、EMAC がコンフィギュレーションされて全二重モードの動作で RGMII PHY インタフェースと動作する場合のみサポートされています。半二重モードではサポートされません。†



IEEE 802.3az Energy Efficient Ethernet 規格について詳しくは、IEEE 802.3 Ethernet Working Group のウェブサイト（www.ieee802.org/3/）を参照してください。†

EEE は MAC をイネーブルして Low-Power Idle（LPI）モードで動作します。イーサネット・リンクのエンド・ポイントのどちらか一方は、リンクの使用率が低いときに機能をディセーブルして消費電力を節約することができます。MAC は、システムが LPI モードを開始するか終了するか制御して、このことを PHY に対して通信します。†

LPI タイマ

EMAC 内部の 2 つのタイマは LPI モードに関連付けられています。†

- LPI Link Status（LS）タイマ †
- LPI TW タイマ †

LPI LS タイマは ms でカウントし、リンク・ステータスが立ち上がってから終了するまでの時間をカウントします。このタイマは、リンクが終了するとき毎回クリアされ、もう一度立ち上がってソフトウェアにプログラムされた最終値に達するときにインクリメントされます。PHY インタフェースは、最終値に達しない限り LPI パターンをアサートしません。これによって、リモート・ステーション間にリンクが確立した後、LPI パターンがアサートされていないとのために最小限の時間が確保されます。この時間は、IEEE 規格の 802.3-az バージョン D2.0 での 1 秒として定義されます。LPI LS タイマは 10 ビット幅です。そのため、ソフトウェアは最大 1023 ms までプログラムできます。†

LPI TW タイマは μs でカウントし、LPI のデアサートから終了するまでの時間をカウントします。タイマの最終値は、自動ネゴシエートされた時間の解決済み送信 TW の値です。この時間の後、MAC は通常の送信動作を行えるようになります。MAC は μs 単位で LPI TW タイマをサポートしています。LPI TW タイマは 16 ビット幅です。したがって、ソフトウェアは最大 65535 μs までプログラムできます。†

送信 / 受信チャネルの LPI ステートが切り替わるとき、EMAC は LPI 割り込みを生成します。†

チェックサム・オフロード

TCP や UDP などの通信プロトコルは、チェックサム・フィールドを実装しています。これは、ネットワークを介して送信されるデータのインテグリティを決定します。イーサネットでも最も広く使用されているのが IP データグラムを介して TCP および UDP をカプセル化することであるため、EMAC にはチェックサムの計算および送信パスへの挿入、それから受信パスでのエラー検出をサポートするチェックサム・オフロード・エンジン (COE) があります。サポートされているオフロードのタイプは以下の通りです。†

- 送信 IP ヘッダ・チェックサム †
- 送信 TCP/UDP/ICMP チェックサム †
- 受信 IP ヘッダ・チェックサム †
- 受信フル・チェックサム †

フレーム・フィルタリング

EMAC は、受信フレーム用に以下のタイプのフィルタリングを実装しています。†

ソース・アドレスまたはデスティネーション・アドレスのフィルタリング

アドレス・フィルタリング・モジュールは、受信パケットごとのデスティネーションおよびソースのアドレス・フィールドをチェックします。†

- 「ユニキャスト・デスティネーション・アドレス・フィルタ」
- 「マルチキャスト・デスティネーション・アドレス・フィルタ」
- 「ハッシュまたは完全アドレス・フィルタ」
- 「ブロードキャスト・アドレス・フィルタ」
- 「ユニキャスト・ソース・アドレス・フィルタ」
- 「逆フィルタリング動作（最終出力でのフィルタ・マッチ結果の反転）」

ユニキャスト・デスティネーション・アドレス・フィルタ

ユニキャスト完全フィルタリング用として、最大 128 個の MAC アドレスがサポートされています。フィルタは、受信したユニキャスト・アドレスの全 48 ビット、およびプログラムされている MAC アドレスに一致があるか比較します。デフォルトの MacAddr0 は常にイネーブルされていて、他のアドレスの MacAddr1 ~ MacAddr127 は個別のイネーブル・ビットによって選択されます。MacAddr1 ~ MacAddr31 のアドレスでは、関連する受信 DA バイトとの比較中に各バイトをマスクできます。これによって、DA 用のグループ・アドレス・フィルタリングがイネーブルされます。MacAddr32 ~ MacAddr127 のアドレスにはマスク・コントロールがなく、MAC アドレスの全 6 バイトは受信された DA の 6 バイトと比較されます。†

ハッシュ・フィルタリング・モードでは、フィルタはユニキャスト・アドレスに対して 64 ビットのハッシュ・テーブルを使用して不完全フィルタリングを実行します。受信されるデスティネーション・アドレスの CRC の上位 6 ビットを使用して、ハッシュ・テーブルの内容をインデックス化します。0 の値は選択されているレジスタのビット 0 を選択します。また、111111 のバイナリ値はハッシュ・テーブル・レジスタのビット 63 を選択します。対応するビットが 1 に設定されている場合はユニキャスト・フレームがハッシュ・フィルタを通過しており、それ以外のときはフレームがハッシュ・フィルタに失敗したことを意味します。†

マルチキャスト・デスティネーション・アドレス・フィルタ

すべてのマルチキャスト・フレームを渡すように MAC をプログラムできます。完全フィルタリング・モードでは、マルチキャスト・アドレスは、プログラムされた MAC デスティネーション・アドレス・レジスタ (1 ~ 31) と比較されます。グループ・アドレス・フィルタリングもサポートされています。ハッシュ・フィルタリング・モードでは、フィルタは 64 ビットのハッシュ・テーブルを使用して不完全フィルタリングを実行します。ハッシュ・フィルタリングでは、受信マルチキャスト・アドレスの CRC の上位 6 ビットを使用して、ハッシュ・テーブルの内容をインデックス化します。値 0 は選択されているレジスタのビット 0 を選択します。また、111111 のバイナリ値はハッシュ・テーブル・レジスタのビット 63 を選択します。対応するビットが 1 に設定されている場合はマルチキャスト・フレームがハッシュ・フィルタを通過しており、それ以外のときはフレームがハッシュ・フィルタに失敗したことを意味します。†

ハッシュまたは完全アドレス・フィルタ

DA がハッシュ・フィルタまたは完全フィルタのどちらか一方に一致するときにフレームを渡すようにフィルタをコンフィギュレーションできます。このコンフィギュレーションは、ユニキャスト・フレームおよびマルチキャスト・フレームの両方に適用されます。†

ブロードキャスト・アドレス・フィルタ

デフォルト・モードでは、フィルタは、どのブロードキャスト・フレームもフィルタしません。しかし、MAC がすべてのブロードキャスト・フレームを拒否するようにプログラムされている場合、フィルタは任意のブロードキャスト・フレームをドロップします。†

ユニキャスト・ソース・アドレス・フィルタ

MAC は、受信フレームのソース・アドレス・フィールドに基づいた完全フィルタリングも実行できます。SA のあるグループ・フィルタリングもサポートされています。アドレスの 1 つ以上のバイトをマスクすることで、アドレスのグループをフィルタできます。†

逆フィルタリング動作（最終出力でのフィルタ・マッチ結果の反転）

デスティネーションおよびソースのアドレス・フィルタリングの両方には、最後の出力でフィルタ一致の結果を反転させるオプションがあります。このモードでは、ユニキャストまたはマルチキャストのデスティネーション・アドレス・フィルタの結果が反転されます。†

VLAN フィルタリング

EMAC は、2 種類の VLAN フィルタリングをサポートしています。†

- VLAN タグ・ベース・フィルタリング †
- VLAN ハッシュ・フィルタリング †

VLAN タグ・ベース・フィルタリング

VLAN タグ・ベース・フレーム・フィルタリングでは、MAC は受信フレームの VLAN タグを比較して、アプリケーションに VLAN フレーム・ステータスを提供します。プログラムされたモードに基づいて、MAC は受信 VLAN タグの下位 12 ビットまたは全 16 ビットを比較して完全な一致を確認します。VLAN タグ・フィルタリングがイネーブルされている場合、MAC は VLAN タグされたフレームを VLAN タグ・マッチ・ステータスと共に転送し、一致していない VLAN フレームをドロップします。VLAN フレームの反転マッチングをイネーブルすることもできます。更に、SVLAN タグされたフレームのマッチングをデフォルトの Customer Virtual Local Area Network (C-VLAN) タグされたフレームと共にイネーブルできます。†

16 ビットのハッシュ・テーブルを使用した VLAN ハッシュ・フィルタリング

MAC には、16 ビットのハッシュ・テーブルを使用した VLAN ハッシュ・フィルタリングが用意されています。また、MAC は VLAN フレームの反転マッチングもサポートしています。反転マッチング・モードでは、フレームの VLAN タグが完全フィルタまたはハッシュ・フィルタに一致している場合、パケットはドロップされる必要があります。VLAN 完全マッチまたは VLAN ハッシュ・マッチがイネーブルされている場合、VLAN ハッシュまたは VLAN 完全フィルタのどちらか一方が一致しているとき、フレームは一致されているものとして見なされます。反転マッチが設定されている場合、完全フィルタまたはハッシュ・フィルタがミスマッチを示すときのみパケットが転送されます。†

レイヤ 3 フィルタおよびレイヤ 4 フィルタ

レイヤ 3 フィルタリングは、ソース・アドレスおよびデスティネーション・アドレスのフィルタリングを指します。レイヤ 4 フィルタリングは、ソース・ポートおよびデスティネーション・ポートのフィルタリングを指します。フレームは、以下の方法でフィルタされます。†

- 一致フレーム †
- 不一致フレーム †
- ノン TCP または UDP IP フレーム †

一致フレーム

MAC は、アプリケーションに対して、イネーブルされているすべてのフィールドに一致するフレームをステータスと共に転送します。MAC は、以下の条件の 1 つに合致しているときのみ、一致したフィールド・ステータスを供給します。†

- イネーブルされているレイヤ 3 およびレイヤ 4 の全フィールドが一致している。†
- イネーブルされているフィールドのうち少なくとも 1 つが一致していて、他のフィールドがバイパスされるかディセーブルされている。†

CSR セットを使用して、フィルタ 3 を通してフィルタ 0 として識別される最大 4 つのフィルタを定義できます。レイヤ 3 およびレイヤ 4 の複数のフィルタがイネーブルされている場合、任意のフィルタ一致が一致として見なされます。2 つ以上のフィルタが一致している場合、フィルタ 0 が最下位でフィルタ 3 が最上位であると、MAC は最下位フィルタのステータスを提供します。例えば、フィルタ 0 とフィルタ 1 が一致している場合、MAC はフィルタ 0 に対応するステータスを供給します。†

不一致フレーム

MAC は、イネーブルされているどのフィールドとも一致しないフレームをドロップします。反転マッチ機能を使用して、IP フィールド上の特別な TCP または UDP を持っているフレームをブロックまたはドロップし、他のフレームをすべて転送できます。フレームがドロップされたとき、適切なアボート・ステータスと共に部分的なフレームを受け取るために、または完全にフレームをドロップするために、EMAC をコンフィギュレーションできます。†

ノン TCP または UDP IP フレーム

デフォルトでは、すべてのノン TCP または UDP IP フレームはレイヤ 3 フィルタとレイヤ 4 フィルタからバイパスされます。オプションとして、IP フレームにわたってすべてのノン TCP または UDP をドロップするように MAC をプログラムできます。†

クロックおよびリセット

イーサネット MAC コントローラは表 17-4 に示すクロックを使用します。

表 17-4. クロック

名称	ノミナルな周波数	機能の使用	説明
clk_ref_i	250 Mhz	EMAC に対する基準クロックです。	クロック・インタフェースから供給される場合、クロックは emac0_clk または emac1_clk です。
clk_tx_i	125/25/2.5 Mhz	自動ネゴシエートの速度が 10/100Mbps に低下します。	
clk_rx_i		PHY が MAC に対してこの基準を提供します。	MAC によって受信されるすべての PHY 信号はこのクロックに同期します。

EEE 用のクロック・ゲーティング

RGMII PHY インタフェースでは、EEE アプリケーション用の送信クロックをゲートできます。詳しくは、17-72 ページの「EEE (Energy Efficient Ethernet) のプログラミング・ガイドライン」を参照してください。

リセット

イーサネット MAC コントローラは表 17-5 に示すリセット信号を使用します。

表 17-5. リセット

名称	ノミナルな周波数	機能の使用	説明
rst_clk_tx_n_o		送信クロック・リセット出力です。	外部 PHY 送信クロック・ドメイン・ロジックをリセットするために使用されます。
rst_clk_rx_n_o		受信クロック・リセット出力です。	外部 PHY 受信クロック・ドメイン・ロジックをリセットするために使用されます。

割り込み

割り込みは、EMAC および外部 PHY デバイスでの特別なイベントの結果として生成されます。割り込みステータス・レジスタは、割り込みをトリガする可能性のあるすべての状態を表示して、どの割り込みが伝播できるか決定します。

イーサネット MAC のプログラミング・モデル

DMA コントローラ

DMA には独立した送信エンジンおよび受信エンジン、そして CSR 空間があります。送信エンジンは、システム・メモリからデバイス・ポートまたは MAC トランザクション・レイヤ (MTL) に対してデータを転送します。一方受信エンジンは、デバイス・ポートからシステム・メモリに対してデータを転送します。DMA コントローラはディスクリプタを使用して、ホスト CPU の関与を最小限に留めてソースからデスティネーションに効率よくデータを移動させます。DMA は、イーサネット内のフレームなどパケット対応のデータ転送のためにデザインされています。DMA コントローラは、フレーム送受信の転送コンプリーションや他の通常 / エラー状態などの状況に際してホスト CPU に割り込めるようにプログラムできます。

DMA およびホスト・ドライバは、2 つのデータ構造を通して通信します。†

- コントロールおよびステータス・レジスタ (CSR) †
- ディスクリプタ・リストおよびデータ・バッファ †

コントロールおよびステータス・レジスタについて詳しくは、17-76 ページの「イーサネット MAC のアドレス・マップおよびレジスタの定義」を参照してください。ディスクリプタについては、17-39 ページの「ノーマル・ディスクリプタ」および 17-53 ページの「代替またはエンハンスド・ディスクリプタ」で説明されています。



ユーザーは RTL コンフィギュレーションで代替ディスクリプタ構造を選択できます。アプリケーションがより大きなバッファ・サイズ (8 KB) を使用できるように、このディスクリプタ構造のコントロール・ビットは再割り当てされます。この代替ディスクリプタ構造のビット・マップについて詳しくは、17-53 ページの「代替またはエンハンスド・ディスクリプタ」を参照してください。17-22 ページの「DMA コントローラ」全体にわたる説明は、この新しい代替ディスクリプタ構造ではなくデフォルトのディスクリプタ構造です。代替ディスクリプタ構造を使用する場合は、17-22 ページの「DMA コントローラ」のディスクリプタに特別なマッピングを無視し、代替ディスクリプタに特別なビット・マップを参照してください。

DMA は、MAC によって受信されるデータ・フレームをホスト・メモリの受信バッファに対して転送し、ホスト・メモリの送信バッファからの送信データ・フレームを (MAC に) 転送します。ホスト・メモリにあるディスクリプタは、これらのバッファに対するポインタとして動作します。†

ディスクリプタのリストは、受信用と送信用の計 2 つあります。各リストのベース・アドレスは、それぞれレジスタ 3 (受信ディスクリプタ・リスト・アドレス・レジスタ) とレジスタ 4 (送信ディスクリプタ・リスト・アドレス・レジスタ) に書き込まれています。ディスクリプタ・リストはフォワード・リンクされています (暗示的または明示的に)。終端のディスクリプタは、リング構造を作成するために最初のエントリに戻ってポイントする可能性があります。受信ディスクリプタおよび送信ディスクリプタ (RDES1[24] および TDES1[24]) の両方にチェーンされている 2 番目のアドレスを設定することで、ディスクリプタの明示的な変更が実現します。ディスクリプタのリストは、ホストの物理メモリのアドレス空間にあります。各ディスクリプタは、2 つのバッファの最大をポイントします。これによって、メモリ内の連続したバッファではなく 2 つのバッファについて、その使用と物理的なアドレス指定が可能になります。†

しかし、ホストの物理メモリ空間にあって、フレーム全体またはフレームの一部で構成されているデータ・バッファは、シングル・フレームを超えられません。バッファにはデータのみが含まれており、バッファ・ステータスはディスクリプタ内で維持されます。データ・チェイニングは、複数のデータ・バッファにまたがるフレームを指します。しかし、単一のディスクリプタは複数のフレームにまたがることはできません。フレーム終端が検出されると、DMA は次のフレーム・バッファにスキップします。データ・チェイニングはイネーブル/ディセーブルできます。†

図 17-5 と図 17-6 に、ディスクリプタのリング構造およびチェーン構造を示します。†

図 17-5. ディスクリプタのリング構造

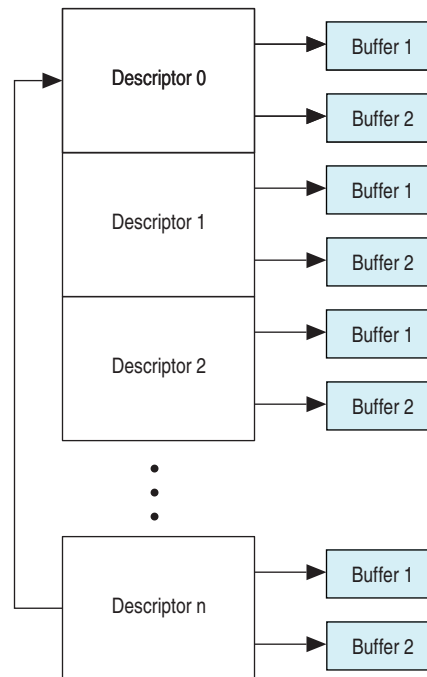
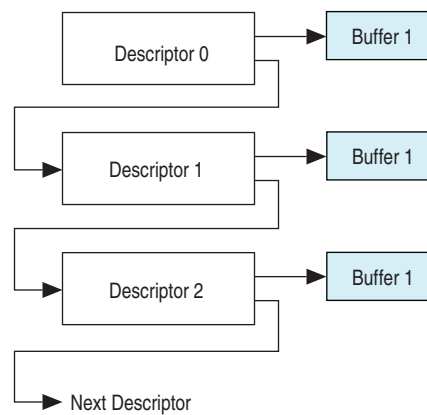


図 17-6. ディスクリプタのチェーン構造



初期化

EMAC の初期化は、次の通りです。

1. ホスト・バス・アクセス・パラメータを設定するためにレジスタ 0 (バス・モード・レジスタ) に書き込みます。†
2. 不要な割り込み原因をマスクするためにレジスタ 7 (割り込みイネーブル・レジスタ) に書き込みます。†
3. 送信 / 受信ディスクリプタのリストを作成し、DMA に各リストの開始アドレスを提供する DMA のレジスタ 3 (受信ディスクリプタ・リスト・アドレス・レジスタ) およびレジスタ 4 (送信ディスクリプタ・リスト・アドレス・レジスタ) に書き込みます。†
4. 目的のフィルタリング・オプションのためにレジスタ 1 (MAC フレーム・フィルタ)、レジスタ 2 (ハッシュ・テーブル・レジスタ)、およびレジスタ 3 (ハッシュ・テーブル Low レジスタ) に書き込みます。†
5. 動作モードをコンフィギュレーションして送信動作をイネーブルするためにレジスタ 1 (MAC フレーム・フィルタ) に書き込みます (ビット 3 がトランスミッタ・イネーブルです)。PS ビットおよび DM ビットは、自動ネゴシエーション結果 (PHY から読み出されます) に基づいて設定されます。†
6. ビット 13 およびビット 1 を設定して送信および受信を開始するためにレジスタ 6 (動作モード・レジスタ) に書き込みます。†
7. 受信動作をイネーブルするためにレジスタ 0 (MAC コンフィギュレーション・レジスタ) に書き込みます (ビット 2 がレシーバ・イネーブルです)。†

送信エンジンおよび受信エンジンがランニング・ステートになり、各ディスクリプタ・リストからディスクリプタを取得するよう試みます。そして、受信エンジンおよび送信エンジンは受信動作および送信動作を処理します。送信プロセスおよび受信プロセスは互いに独立しており、個別に開始 / 終了します。†

ホスト・バス・バースト・アクセス

DMA がレジスタ 0 (バス・モード・レジスタ) の FB ビットを通してマスタ・インタフェースで固定バースト長での転送を実行するようにコンフィギュレーションされている場合、それを試みます。バースト長の最大値はレジスタ 0 (バス・モード・レジスタ) の PBL フィールド (ビット [13:8]) に表示されてそれによって制限されています。受信ディスクリプタおよび送信ディスクリプタは、常に、16 ビットの読み出しで可能な (PBL または $16 * 8$ / バス幅で制限された) 最大バースト・サイズでアクセスされます。

コンフィギュレーションされたバーストに対応する十分な空間が MTL 送信 FIFO バッファで使用可能な場合、または終端フレームまでのバイト数 (コンフィギュレーションされたバースト長よりもそのバイト数が短いとき) で使用可能な場合のみ、DMA がデータ転送を開始します。DMA は、マスタ・インタフェースに対して開始アドレスおよび必要な転送数を示します。インタフェースが固定バースト長でコンフィギュレーションされている場合、INCR4、8、または 16 トランザクションと SINGLE トランザクションの最良の組み合わせを使用してデータを転送します。それ以外の (固定バースト長ではない) 場合、INCR (定義されていない長さ) トランザクションおよび SINGLE トランザクションを使用してデータを転送します。

コンフィギュレーションされたバーストに対応する十分な空間が MTL 受信 FIFO バッファで使用可能な場合、または終端フレームが受信 FIFO バッファで検出される場合 (コンフィギュレーションされたバースト長よりもそのバイト数が短いとき)、DMA がデータ転送を開始します。DMA は、マスタ・インタフェースに対して開始アドレスおよび必要な転送数を示します。インタフェースが固定バースト長でコンフィギュレーションされている場合、INCR4、8、または 16 トランザクションと SINGLE

トランザクションの最良の組み合わせを使用してデータを転送します。インタフェースでの固定バーストが終了する前にフレームの終端に達すると、固定バーストを満たすためにダミー転送が実行されます。その他の場合（レジスタ 0（バス・モード・レジスタ）の FB ビットがリセットされている場合）、INCR（定義されていない長さ）トランザクションおよび SINGLE トランザクションを使用してデータを転送します。

アドレス・アラインメントされたビート用にインタフェースがコンフィギュレーションされている場合、両方の DMA エンジン、開始された最初のバースト転送が、コンフィギュレーションされている PBL 以下のサイズであることを確認します。このようにして、コンフィギュレーションされた PBL にアラインメントされたアドレスで以降のすべてのビートが開始します。インタフェースが INCR16 までしかサポートしていないため、DMA は最大 16 のサイズ（PBL > 16 のため）のビートのみにアドレス・アラインメントできます。

ホスト・データ・バッファ・アラインメント

送信データ・バッファおよび受信データ・バッファには、開始アドレス・アラインメントの制約がありません。例えば、32 ビット・メモリのシステムでは、バッファ用の開始アドレスを 4 バイトのどれかにアラインメントできます。しかし、DMA は常に、バイト・レーンの必要がないダミー・データを持っているバス幅にアドレス・アラインメントされた状態で転送を開始します。これは、通常、イーサネット・フレームの開始または終了の転送で起きます。ソフトウェア・ドライバは、バッファの開始アドレスおよびフレームのサイズに基づいて、ダミーのバイトを破棄する必要があります。†

例：バッファ読み出し

送信バッファ・アドレスが 0x00000FF2（32 ビット・データ・バス用）であり 15 バイトが転送される必要がある場合、DMA はアドレス 0x00000FF0 からフルの 5 つのワードを読み出しますが、MTL 送信 FIFO バッファにデータを転送するとき、余分なバイト（最初の 2 バイト）はドロップまたは無視されます。同様に、最後の転送の終端 3 バイトも無視されます。DMA は、通常、フレームの最後ではない限り MTL 送信 FIFO バッファに対してフルの 32 ビット・データを転送することを確認します。

例：バッファ書き込み

受信バッファ・アドレスが 0x00000FF2（64 ビット・データ・バス用）であり 1 バイトの受信フレームが転送される必要がある場合、DMA はアドレス 0x00000FF0 のフルの 3 ワードを書き込みます。しかし、最初の転送の先頭 2 バイトと 3 番目の転送の終端 6 バイトにはダミー・データが含まれます。

バッファ・サイズの計算

DMA は送受信ディスクリプタのサイズ・フィールドを更新しません。DMA は、ディスクリプタのステータス・フィールド（RDES および TDES）のみ更新します。ドライバはサイズ計算をする必要があります。†

送信 DMA は、MAC に対して正確なバイト数（TDES1 のバッファ・サイズ・フィールドによって示されるバイト数）を転送します。ディスクリプタが先頭（TDES1 の FS ビット）としてマークされている場合、DMA はバッファからの最初の転送をフレームの開始としてマークします。ディスクリプタが終端（TDES1 の LS ビット）としてマークされている場合、DMA はデータ・バッファからの MTL に対する最後の転送をフレームの終端としてマークします。†

受信 DMA は、バッファがフルになるまで、またはフレームの終端が MTL から受信するまでバッファにデータを転送します。ディスクリプタが終端 (RDES0 の LS ビット) としてマークされていない場合、ディスクリプタの関連するバッファはフルで、ディスクリプタの FS ビットが設定されているとき、バッファ内のバリッド・データ数はバッファ・サイズ・フィールドからデータ・バッファ・ポインタ・オフセットを減算した数として正確に表示されます。データ・バッファ・ポインタがデータ・バス幅にアラインメントされている場合、オフセットはゼロです。ディスクリプタが終端としてマークされている場合、バッファはフルではない可能性があります (RDES1 のバッファ・サイズによって表示されるように)。この最後のバッファのバリッド・データの数进行計算するには、ドライバはフレーム長を読み出して (RDES0[29:16] の FL ビット)、このフレームに先行するバッファのバッファ・サイズ合計を減算する必要があります。受信 DMA は、常に、次のフレームの先頭を新しいディスクリプタと共に転送します。†



受信バッファの開始アドレスがシステム・バスのデータ幅にアラインメントされていない場合でも、システムはシステム・バス幅にアラインメントされたサイズの受信バッファを割り当てる必要があります。例えば、アドレス 0x1000 から開始する 1024 バイト (1 KB) の受信バッファをシステムが割り当てる場合、ソフトウェアは受信ディスクリプタのバッファ開始アドレスが 0x1002 オフセットを持つようにプログラムできます。受信 DMA は先頭の 2 つの位置 (0x1000 および 0x1001) にあるダミー・データを持ったこのバッファにフレームを書き込みます。実際のフレームは位置 0x1002 から書き込まれます。そのため、バッファ・サイズが 1024 バイトとしてプログラムされていても、開始アドレス・オフセットのために、このバスで実際に使用可能な空間は 1022 バイトです。†

送信

送信機能は、17-39 ページの「送信ディスクリプタ」で詳しく説明されている送信ディスクリプタを使用します。

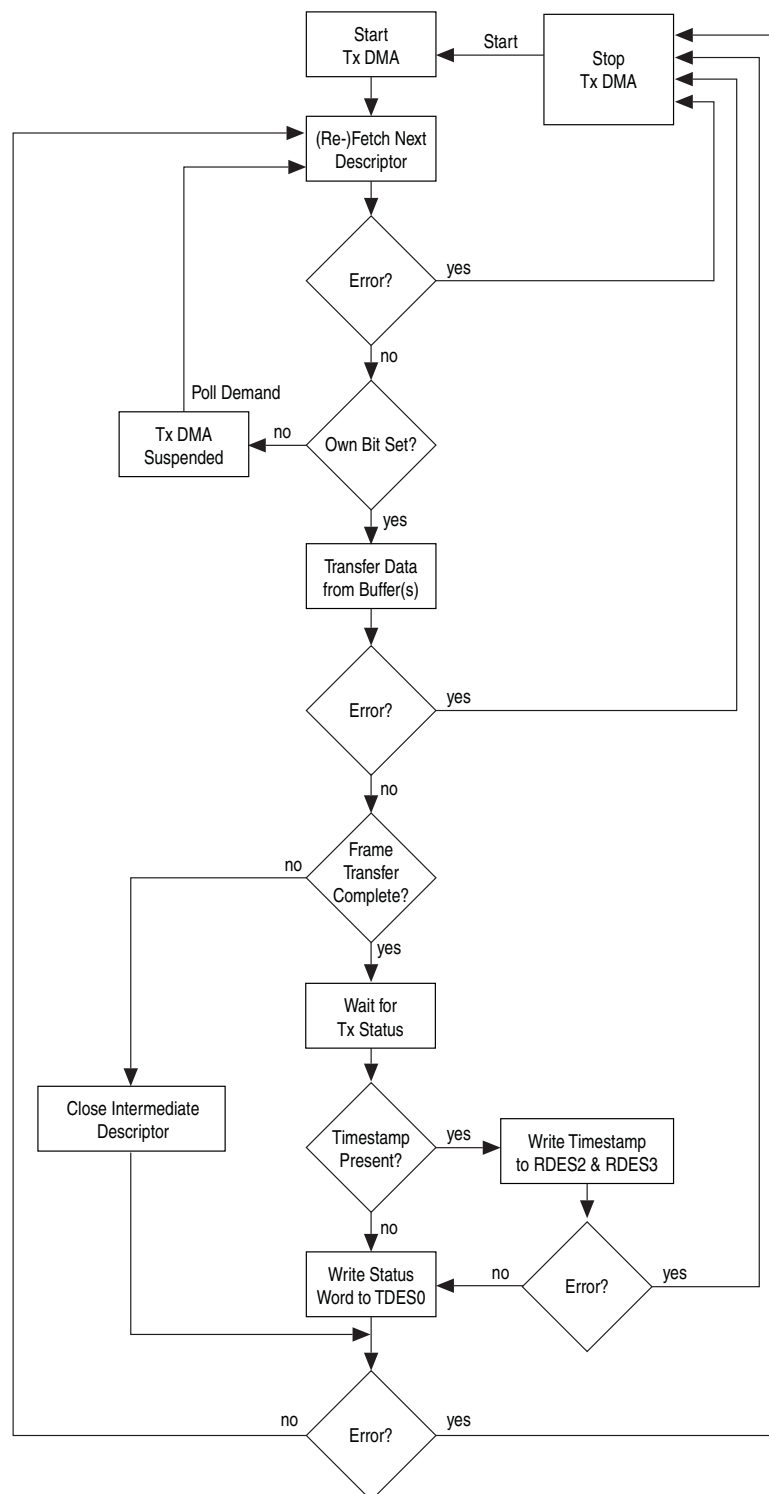
TX DMA 動作：デフォルト (ノン OSF) モード

デフォルト・モードでの送信 DMA エンジン、以下のように進行します。†

1. ホストは、イーサネット・フレーム・データを持っている関連データ・バッファをセット・アップした後、送信ディスクリプタ (TDES0 ~ TDES3) をセット・アップして所有ビット (TDES0[31]) を設定します。†
2. レジスタ 6 (動作モード・レジスタ) のビット 13 (ST) が設定されている場合、DMA が実行状態になります。†
3. 実行状態の間、DMA はフレーム要求送信用に送信ディスクリプタ・リストをポーリングします。ポーリングが開始すると、シーケンシャル・ディスクリプタがリング・オーダまたはチェイン・オーダで続きます。ホストに所有されているものとしてフラグされているディスクリプタを DMA が検出する場合 (TDES0[31] = 0)、エラー状態が発生すると、送信が中断されてレジスタ 5 (ステータス・レジスタ) のビット 2 (送信バッファ使用不可) およびビット 16 (通常割り込み要約) が設定されます。送信エンジンはステップ 9 に進みます。
4. 取得したディスクリプタが DMA に所有されているものとしてフラグされている場合 (TDES0[31] = 1)、DMA は取得したディスクリプタからの送信データ・バッファ・アドレスをデコードします。
5. DMA は、ホスト・メモリから送信データをフェッチして、送信用に MTL に対してデータを転送します。†
6. イーサネット・フレームがデータ・バッファを介して複数のディスクリプタに格納される場合、DMA は中間のディスクリプタを閉じて次のディスクリプタをフェッチします。イーサネット・フレームの終端のデータが MTL に転送されるまでステップ 3、4、および 5 が繰り返されます。†
7. フレーム転送が完了すると、IEEE 1588 タイムスタンプがフレーム用にイネーブルされている場合 (送信ステータスに示されているように)、MTL から得られるタイムスタンプ値は、終端フレーム・バッファが含まれている送信ディスクリプタ (TDES2 および TDES3) に書き込まれます。ステータス情報は、この送信ディスクリプタ (TDES0) に書き込まれます。このステップで所有ビットがクリアされるため、この時点でホストはこのディスクリプタを所有しています。このフレーム用にタイムスタンプがイネーブルされなかった場合、DMA は TDES2 および TDES3 の内容を変更しません。†
8. 終端ディスクリプタの完了時割り込み (TDES1[31]) セットを持っているフレームの送信が完了すると、レジスタ 5 (ステータス・レジスタ) のビット 0 (送信割り込み) が設定されます。そして、DMA エンジンはステップ 3 に戻ります。†
9. 中断状態では、送信ポーリング要求を受信してアンダーフロー割り込みステータスがクリアされると、DMA はディスクリプタを再取得しようと試みます (それによってステップ 3 に戻ります)。†

デフォルト・モードでの TX DMA 送信フローを図 17-7 に示します。†

図 17-7. デフォルト・モードでの TX DMA 動作



TX DMA 動作：OSF モード

実行ステートの間、レジスタ 6（動作モード・レジスタ）のビット 2（OSF）が設定されている場合、送信プロセスは先頭のステータス・ディスクリプタを閉じることなく 2 つのフレームを同時に取得できます。送信プロセスが先頭のフレーム転送を終了すると、2 番目のフレーム用に送信ディスクリプタ・リストをすぐにポーリングします。2 番目のフレームが有効である場合、送信プロセスは先頭のフレームのステータス情報の書き込み前にこのフレームを転送します。†

OSF モードでは、実行ステート送信 DMA は以下のシーケンスを実行します。†

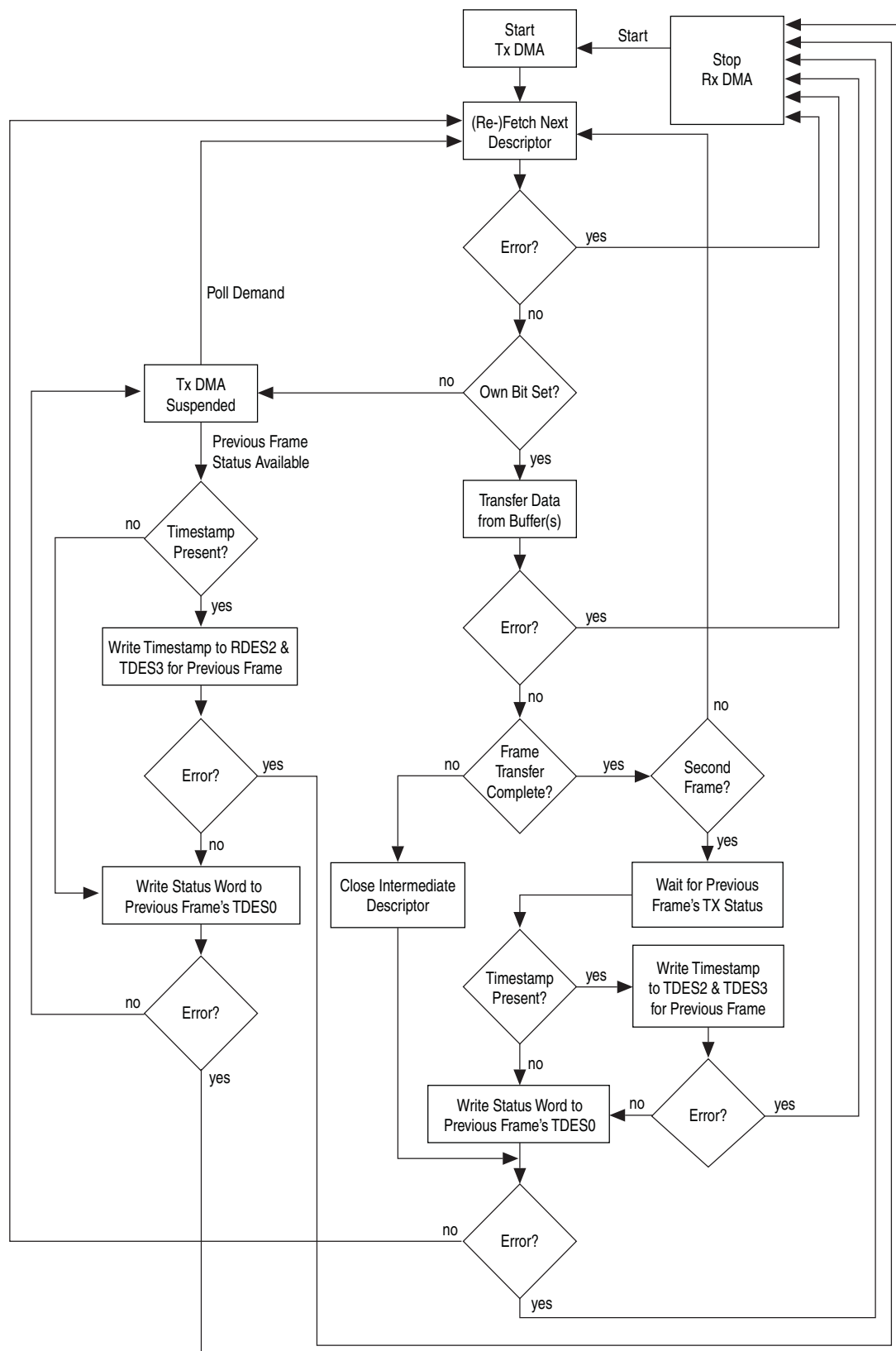
1. DMA は、17-27 ページの「TX DMA 動作：デフォルト（ノン OSF）モード」のステップ 1 ～ 6 に説明されているように実行します。†
2. 前回のフレームの終端ディスクリプタを閉じることなく、DMA は次のディスクリプタをフェッチします。†
3. DMA が取得ディスクリプタを所有している場合、DMA はこのディスクリプタの送信バッファ・アドレスをデコードします。DMA がディスクリプタを所有していない場合、DMA は中断モードになってステップ 7 にスキップします。†
4. DMA は、終端フレームのデータが転送されるまで、フレームが複数のディスクリプタにまたがって分割されている場合は中間のディスクリプタを閉じながらホスト・メモリから送信フレームをフェッチして、そのフレームを MTL に転送します。†
5. DMA は、前回のフレームのフレーム送信ステータスおよびタイムスタンプを待機します。ステータスが使用可能になると、タイムスタンプがキャプチャされたら DMA はそのタイムスタンプを TDES2 および TDES3 に書き込みます（ステータス・ビットで表示されるように）。そして DMA は、クリアされた所有ビットを使用して関連する TDES0 に対してステータスを書き込んで、ディスクリプタを閉じます。前回のフレームでタイムスタンプがイネーブルされていない場合、DMA は TDES2 および TDES3 の内容を変更しません。†
6. イネーブルされている場合、送信割り込みが設定されて、DMA は次のディスクリプタをフェッチしてステップ 3 に進みます（ステータスが通常の場合）。前回の送信ステータスがアンダーフロー・エラーとなってしまった場合、DMA は中断モードになります（ステップ 7）。†
7. 中断モードでは、待機中のステータスとタイムスタンプが MTL から受信されると、DMA はタイムスタンプを TDES2 および TDES3 に書き込んで（現在のフレームにイネーブルされている場合）、その後、関連する TDES0 にステータスを書き込みます。そして、関連する割り込みを設定して中断モードに戻ります。†
8. DMA は、送信ポーリング要求（レジスタ 1（送信ポーリング要求レジスタ））を受信後のみに、中断モードを終了して実行ステートを開始できます（待機中のステータスに応じてステップ 1 またはステップ 2 に進みます）。†



DMA は現在のディスクリプタを閉じる前に次のディスクリプタを前もってフェッチするため、正しく適切に動作するためにディスクリプタ・チェーンは 3 つ以上のディスクリプタを持っている必要があります。†

基本的なフローを図 17-8 に示します。†

図 17-8. OSF モードでの TX DMA 動作



送信フレームの処理

送信 DMA は、プリアンブル、パッド・バイトおよび FCS フィールドを除いて、完全なイーサネット・フレームがデータ・バッファに含まれていることを前提とします。DA、SA、および Type/Len のフィールドにはバリッド・データが含まれています。MAC による CRC または PAD 挿入のディセーブルが必要であることを送信ディスクリプタが示している場合、CRC バイトを含む（プリアンブルを除く）完全なイーサネット・フレームがバッファに含まれている必要があります。†

フレームはデータ・チェイン接続が可能で、複数のバッファをまたぐことができます。フレームは、先頭のディスクリプタ (TDES1[29]) と終端のディスクリプタ (TDES1[30]) によってそれぞれ区切られる必要があります。†

送信開始のとき、先頭のディスクリプタ (TDES1[29]) が設定されている必要があります。これが設定されると、フレーム・データがホスト・バッファから MTL 送信 FIFO バッファに転送されます。同時に、現在のフレームが終端のディスクリプタ (TDES1[30]) をクリアする場合、送信プロセスは次のディスクリプタの取得を試みます。送信プロセスは、このディスクリプタが TDES1[29] をクリアすることを前提とします。TDES1[30] がクリアされている場合、それは中継ぎのバッファを示しています。TDES1[30] が設定されている場合、それはフレームの終端バッファを示しています。†

フレームの終端バッファが送信された後、DMA は、送信ディスクリプタ 1 (TDES1[30]) の終端セグメント・セットを持っている送信ディスクリプタ 0 (TDES0) のディスクリプタ・ワードに最後のステータス情報を書き戻します。このとき、完了時割り込み (TDES1[31]) が設定されている場合、レジスタ 5 (ステータス・レジスタ) のビット 0 (送信割り込み) が設定されて、次のディスクリプタがフェッチされ、プロセスが繰り返されます。†

実際のフレーム送信は、MTL 送信 FIFO バッファがプログラム可能送信スレッシュホールド (レジスタ 6 (動作モード・レジスタ) のビット [16:14]) に達するか、または FIFO バッファがフル・フレームになるか、どちらか一方になった後に開始します。ストア・アンド・フォワード・モード (レジスタ 6 (動作モード・レジスタ) のビット 21) のオプションもあります。DMA がフレーム転送を終了すると、ディスクリプタがリリースされます (所有ビット [31] がクリアされます)。†



フレームの送信および次のフレームを適切にするには、終端のディスクリプタ (TDES1[30]) セットを持っている送信ディスクリプタ用としてバッファ・サイズをゼロでない値に指定する必要があります。†

送信ポーリングの中断

送信ポーリングは、以下の条件のうちいずれか一方によって中断できます。†

- DMA がホストに所有されているディスクリプタを検出する (TDES0[31]=0)。再開するには、ドライバがディスクリプタの所有権を DMA に渡して、ポーリング要求コマンドを発行します。†
- アンダーフローによる送信エラーが検出されると、フレーム送信がアボートされます。適切な送信ディスクリプタ 0 (TDES0) ビットが設定されます。†

最初の条件のために DMA が SUSPEND ステートになると、レジスタ 5 (ステータス・レジスタ) のビット 16 (ノーマル割り込み要約) およびビット 2 (送信バッファ使用不可) が設定されます。2 番目の条件が発生すると、レジスタ 5 (ステータス・レジスタ) のビット 15 (アブノーマル割り込み要約) およびビット 5 (送信アンダーフロー) が設定されて、送信ディスクリプタ 0 に対して中断を引き起こす情報が書き込まれます。†

どちらの場合も、送信リストでの位置は維持されます。維持される位置とは、DMA が閉じた終端ディスクリプタに続くディスクリプタの位置です。†

中断の原因を修正した後に、ドライバは送信ポーリング要求を明示的に発行する必要があります。†

受信

受信機能は、17-43 ページの「受信ディスクリプタ」に詳しく説明している受信ディスクリプタを使用します。

受信 DMA エンジンの受信シーケンスは、17-35 ページの 図 17-9 に示されており、次の手順で進行します。†

1. ホストが受信ディスクリプタ (RDES0 ~ RDES3) を設定し、所有ビット (RDES0[31]) を設定します。†
2. レジスタ 6 (動作モード・レジスタ) のビット 1 (SR) が設定されている場合、DMA は実行ステートになります。実行ステートの間、DMA は受信ディスクリプタ・リストをポーリングして、フリーのディスクリプタの取得を試みます。フェッチされたディスクリプタがフリーではない場合 (ホストによって所有されている場合)、DMA は中断ステートになってステップ 9 に進みます。†
3. DMA は、取得したディスクリプタから受信データ・バッファ・アドレスをデコードします。†
4. 受信フレームは処理され、取得したディスクリプタのデータ・バッファに配置されます。†
5. バッファがフルの場合、またはフレーム転送が完了した場合、受信エンジンは次のディスクリプタをフェッチします。†
6. 現在のフレーム転送が完了すると、DMA はステップ 7 に進みます。DMA が次にフェッチされるディスクリプタを所有しておらずフレーム転送が完了しない場合 (EOF がまだ転送されない場合)、DMA はディスクリプタ・エラー・ビットを RDES0 に設定します (レジスタ 6 (動作モード・レジスタ) のビット 24 でフラッシングがディセーブルされていない限り)。DMA は現在のディスクリプタを閉じ

て、RDES0 値の終端セグメント (LS) ビットをクリアすることで中間体としてそれをマークし (フラッシングがディセーブルされていない場合は終端ディスクリプタとしてマークします)、ステップ 8 に進みます。DMA が次のディスクリプタを所有しているのに現在の転送が完了していない場合、DMA は現在のディスクリプタを中間体として閉じて、ステップ 4 に進みます。†

7. IEEE 1588 タイムスタンプがイネーブルされているとき、DMA は現在のディスクリプタの RDES2 および RDES3 にタイムスタンプを書き込みます (もし使用可能ならば)。そして DMA は、MTL から受信フレームのステータスを受け取って、クリアされた所有ビットと終端セグメントのビット・セットと共に、そのステータス・ワードを現在のディスクリプタの RDES0 に書き込みます。†
8. 受信エンジンは、最新のディスクリプタの所有ビットをチェックします。ホストがディスクリプタを所有している場合、レジスタ 5 (ステータス・レジスタ) のビット 7 (受信バッファ使用不可) が設定されて、DMA 受信エンジンは中断ステートになります (ステップ 9)。DMA がディスクリプタを所有している場合、エンジンはステップ 4 に戻って次のフレームを待ちます。
9. 受信エンジンが中断ステートになる前に、受信 FIFO バッファから一部のフレームがフラッシュされます。フラッシングはレジスタ 6 (動作モード・レジスタ) のビット 24 を使用して制御できます。†
10. 受信ポーリング要求が与えられる場合、または次のフレーム開始が MTL の受信 FIFO バッファから使用可能な場合、受信 DMA は中断ステートから回復します。エンジンはステップ 2 に進んで、次のディスクリプタを再びフェッチします。†

```

graph TD
    Start([Start]) --> StartDMA[Start Rx DMA]
    StartDMA --> FetchNextDesc[(Re-)Fetch Next Descriptor]
    FetchNextDesc --> Error1{Error?}
    Error1 -- yes --> StopDMA[Stop Rx DMA]
    Error1 -- no --> OwnBitSet{Own Bit Set?}
    OwnBitSet -- yes --> FrameDataAvail{Frame Data Available?}
    FrameDataAvail -- no --> WaitData[Wait for Frame Data]
    WaitData --> WriteData[Write Data to Buffer(s)]
    FrameDataAvail -- yes --> WriteData
    WriteData --> Error2{Error?}
    Error2 -- yes --> StopDMA
    Error2 -- no --> FetchNextDescNext[Fetch Next Descriptor]
    FetchNextDescNext --> Error3{Error?}
    Error3 -- yes --> StopDMA
    Error3 -- no --> FrameTransferComplete1{Frame Transfer Complete?}
    FrameTransferComplete1 -- yes --> OwnBitSetForNext{Own Bit Set For Next Descriptor?}
    OwnBitSetForNext -- yes --> CloseRDES0Int[Close RDES0 As Intermediate Descriptor]
    OwnBitSetForNext -- no --> FlushDisabled1{Flush Disabled?}
    FlushDisabled1 -- yes --> SetDescriptorError[Set Descriptor Error]
    FlushDisabled1 -- no --> Error4{Error?}
    Error4 -- yes --> StopDMA
    Error4 -- no --> FrameTransferComplete2{Frame Transfer Complete?}
    FrameTransferComplete2 -- yes --> TimestampPresent{Timestamp Present?}
    TimestampPresent -- yes --> WriteTimestamp[Write Timestamp to RDES2 & RDES3]
    WriteTimestamp --> Error5{Error?}
    Error5 -- yes --> StopDMA
    Error5 -- no --> CloseRDES0Last[Close RDES0 As Last Descriptor]
    CloseRDES0Last --> Error6{Error?}
    Error6 -- yes --> StopDMA
    Error6 -- no --> StopDMA
    FrameTransferComplete2 -- no --> FlushDisabled2{Flush Disabled?}
    FlushDisabled2 -- yes --> SetDescriptorError
    FlushDisabled2 -- no --> Error7{Error?}
    Error7 -- yes --> StopDMA
    Error7 -- no --> FrameTransferComplete3{Frame Transfer Complete?}
    FrameTransferComplete3 -- yes --> FlushRemaining[Flush the Remaining Frame]
    FlushRemaining --> RxDMASuspended[Rx DMA Suspended]
    RxDMASuspended --> PollDemand[Poll Demand/ New Frame Available]
    PollDemand --> FetchNextDesc
    RxDMASuspended --> OwnBitSet
    RxDMASuspended --> FrameTransferComplete1
    RxDMASuspended --> FrameTransferComplete2
    RxDMASuspended --> FrameTransferComplete3
    RxDMASuspended --> Error1
    RxDMASuspended --> Error2
    RxDMASuspended --> Error3
    RxDMASuspended --> Error4
    RxDMASuspended --> Error5
    RxDMASuspended --> Error6
    RxDMASuspended --> Error7
    RxDMASuspended --> StopDMA
  
```

The flowchart illustrates the Rx DMA process, starting with 'Start Rx DMA' and '(Re-)Fetch Next Descriptor'. It includes decision points for 'Error?', 'Own Bit Set?', 'Frame Data Available?', 'Frame Transfer Complete?', 'Flush Disabled?', and 'Timestamp Present?'. The process flows through various states like 'Rx DMA Suspended' and 'Close RDES0 As Intermediate/Last Descriptor' before reaching 'Stop Rx DMA'.

ソフトウェアが CSR を通してタイムスタンプをイネーブルしていれば、バリッド・タイムスタンプ値がフレームに使用不可能な場合（例えば、タイムスタンプが書き込まれる前に受信 FIFO バッファがフルになってしまったなどの理由により）、DMA は RDES2 および RDES3 にすべて 1 を書き込みます。それ以外の場合（つまり、タイムスタンプがイネーブルされていない場合）、RDES2 および RDES3 は変更されません。†

受信ディスクリプタの取得

受信エンジンは、常に受信フレームを見越して外部ディスクリプタを取得するように試みます。以下の条件のいずれかが満たされている場合、ディスクリプタの取得が試みられます。†

- 実行ステートに切り替わった直後に、レジスタ 6（動作モード・レジスタ）のビット 1（開始または停止受信）が設定される。†
- 現在の転送用のフレームが終了する前に、現在のディスクリプタのデータ・バッファがフルになる。†
- コントローラはフレーム受信を完了させるが、現在の受信ディスクリプタを閉じない。†
- ホスト所有のバッファのために受信プロセスが中断し（RDES0[31] = 0）、新規のフレームが受信される。†
- 受信ポーリング要求が発行される。†

受信フレームの処理

フレームがアドレス・フィルタを通過して、フレーム・サイズが MTL の受信 FIFO バッファのコンフィギュレーション可能なスレッシュホールド・バイト・セット以上であるときだけ、またはストア・アンド・フォワード・モードで完全なフレームが FIFO バッファに書き込まれるときだけ、MAC は受信フレームをホスト・メモリに転送します。†

フレームがアドレス・フィルタを失敗すると、MAC ブロックそれ自身にドロップされます（レジスタ 1（MAC フレーム・フィルタ）のビット 31（全受信）が設定されていない限り）。64 バイト未満のフレームは、衝突または早期終端のために、MTL の受信 FIFO バッファから削除できます。†

64（コンフィギュレーション可能なスレッシュホールド）バイトが受信されると、MTL ブロックは DMA ブロックに対して、現在のディスクリプタにポイントされている受信バッファへのフレーム・データ転送を開始するよう要求します。DMA ホスト・インタフェースがデータ転送の受信準備が整った段階で（DMA がホストから送信データをフェッチしていなければ）、DMA は先頭のディスクリプタ（RDES0[9]）を設定してフレームを区切ります。所有（RDES[31]）ビットが 0 にリセットされている場合、データ・バッファが満杯になるか、またはフレームの終端セグメントが受信バッファに転送されると、ディスクリプタがリリースされます。フレームにシングル・ディスクリプタが含まれている場合、終端のディスクリプタ（RDES[8]）と先頭のディスクリプタ（RDES[9]）の両方が設定されます。

DMA は次のディスクリプタをフェッチして、終端のディスクリプタ (RDES[8]) のビットを設定し、前回のフレーム・ディスクリプタの RDES0 ステータス・ビットをリリースします。そして、DMA はレジスタ 5 (ステータス・レジスタ) のビット 6 (受信割り込み) を設定します。ホスト所有としてフラグされたディスクリプタを DMA が検出しない限り、このプロセスが繰り返されます。これが発生すると、受信プロセスはレジスタ 5 (ステータス・レジスタ) のビット 7 (受信バッファ使用不可) を設定して、中断ステートになります。受信リストの位置は維持されます。†

受信プロセスの中断

受信プロセスが中断ステートになっている間に新しい受信フレームが到着すると、DMA はホスト・メモリの現在のディスクリプタを再度フェッチします。ディスクリプタが DMA に所有されている場合、受信プロセスは再び実行ステートになってフレーム受信を開始します。ディスクリプタがまだホストに所有されている場合、デフォルトでは、DMA は MTL RX FIFO バッファのトップで現在のフレームを破棄して、失われたフレーム・カウンタをインクリメントします。MTL EX FIFO バッファに 2 つ以上のフレームが格納されている場合、このプロセスが繰り返されます。†

MTL EX FIFO バッファのトップで破棄またはフラッシュされたフレームは、フラッシング (レジスタ 6 (動作モード・レジスタ) のビット 24) をディセーブルすることで無視できます。このような場合、受信プロセスは受信バッファ使用不可ステータスを設定し、中断ステートに戻ります。†

割り込み

割り込みは、さまざまなイベントの結果として生成されます。DMA のレジスタ 5 (ステータス・レジスタ) には、割り込みの原因となりうるすべてのビットが含まれています。レジスタ 7 (割り込みイネーブル・レジスタ) には、割り込みの原因となるイベントそれぞれに対するイネーブル・ビットが含まれています。†

割り込みには通常および異常の 2 つのグループがあり、レジスタ 5 (ステータス・レジスタ) に表示されます。割り込みは、関連するビット位置に 1 を書き込むことによってクリアされます。グループ内のイネーブルされた割り込みすべてがクリアされると、関連する要約ビットがクリアされます。両方の要約ビットがクリアされる場合、sbd_intr_o 割り込み信号がデアサートされます。MAC が割り込みアサートの原因である場合、図 17-10 に示すように、レジスタ 5 (ステータス・レジスタ) の GLI、GMI、GPI、TTI、または GLPII のビットのどれかが High に設定されます。

図 17-10. sbd_intr_o 生成⁽¹⁾

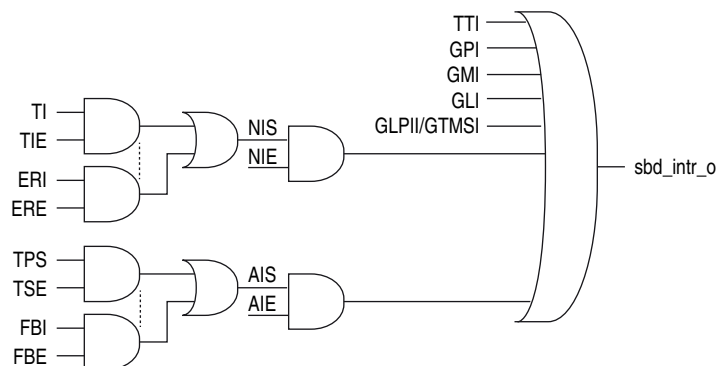


図 17-10 の注：

(1) NIS および AIS の信号は、レジスタ出力の信号です。



レジスタ 5 (ステータス・レジスタ) は割り込みステータス・レジスタです。関連する割り込みイネーブル・ビットがレジスタ 7 (割り込みイネーブル・レジスタ) で設定されている場合のみ、このステータス・レジスタ (レジスタ 5) 内の任意のイベントによって割り込みピン (sbd_intr_o) がアサートされます。†

割り込みはキューイングされません。また、割り込みイベントがドライバの応答よりも早く発生する場合、追加の割り込みは生成されません。例えば、レジスタ 5 (ステータス・レジスタ) のビット 6 (受信割り込み) は、1 つ以上のフレームがホスト・バッファに転送されたことを表示するとします。ドライバは、DMA に所有されている最後に保存された位置から先頭の位置まですべてのディスクリプタをスキャンする必要があります。†

割り込みは、同時に発生する複数のイベントに対して、たった一度のみ生成されます。ドライバは、割り込みの原因を特定するためにレジスタ 5 (ステータス・レジスタ) をスキャンする必要があります。ドライバがレジスタ 5 (ステータス・レジスタ) の適切なビットをクリアした後に新たな割り込みイベントが発生しない限り、割り込みは生成されません。例えば、コントローラはレジスタ 5 (ステータス・レジスタ) のビット 6 (受信割り込み) を生成して、ドライバがレジスタ 5 (ステータス・レジスタ) の読み出しを開始するとします。次に、レジスタ 5 (ステータス・レジスタ) のビット 7 (受信バッファ使用不可) が発生するとします。ドライバが受信割り込みをクリアします。それでも、受信バッファ使用不可割り込みがアクティブであるか待機中であるために、sbd_intr_o 信号はデアサートされません。†

レジスタ 9 (受信割り込みウォッチドッグ・タイマ・レジスタ) のビット 7:0 (割り込みタイマ) は、受信割り込みの柔軟なコントロールを実現するために提供されています。この割り込みタイマがゼロ以外の値でプログラムされている場合、RX DMA がシステム・メモリに対して受信フレームを転送し終わるとすぐに、割り込みタイマがアクティブになります。このとき、受信割り込みは、関連する受信ディスクリプタ (17-47 ページの 表 17-12 の RDES1[31]) でイネーブルされないためアサートされません。このタイマがプログラム値に基づいて動作する場合、関連する RI がレジスタ 7 (割り込みイネーブル・レジスタ) でイネーブルされているとき、RI ビットが設定されて割り込みがアサートされます。フレームがメモリに転送されて RI が設定されている場合、そのディスクリプタにイネーブルされているため、このタイマは動作を開始する前にディセーブルされます。†

DMA へのエラー応答

DMA チャンネルによって開始された任意の転送で、スレーブがエラー応答を返してくる場合、DMA はすべての動作を停止して、レジスタ 5 (ステータス・レジスタ) のエラー・ビットおよび致命的バス・エラー・ビットを更新します。EMAC のソフト・リセット、または EMAC のハード・リセットと DMA の再初期化の後のみ、DMA コントローラが再開します。

ディスクリプタの概要

この項では、HPS の EMAC DMA ディスクリプタについて説明します。

イーサネット・サブシステムの DMA は、17-22 ページの「DMA コントローラ」で説明しているように、ディスクリプタのリンク・リストに基づいてデータを転送します。ディスクリプタは、システム・メモリで作成されます。EMAC DMA は、次の 2 つのタイプのディスクリプタをサポートしています。

- ノーマル・ディスクリプタ — これはデフォルト状態のディスクリプタ・フォーマットで、4 つの DWORDS を持つことができます。このディスクリプタ・タイプについて詳しくは、17-39 ページの「ノーマル・ディスクリプタ」を参照してください。†
- エンハンスド・ディスクリプタ — これは、代替ディスクリプタ・フォーマットで、8 つの DWORDS を持つことができます (32 バイト)。このディスクリプタ・フォーマットについて詳しくは、17-53 ページの「代替またはエンハンスド・ディスクリプタ」を参照してください。

各ディスクリプタには 2 つのバッファ、2 つのバイト・カウンタ・バッファ、2 つのアドレス・ポインタが含まれており、アダプタ・ポートがさまざまなタイプのメモリ管理手法に適合できるようにします。一度コンフィギュレーションすると、ディスクリプタ構造を変更することはできません。†

ディスクリプタのエンディアン

ディスクリプタ・アドレスは、使用されるバス幅にアラインメントされる必要があります (32 ビット・バス、64 ビット・バス、128 ビット・バスでは、それぞれ Word、DWord、または LWord)。データ・バスは、リトル・エンディアンとしてコンフィギュレーションされます。



この項で示す図では、ノーマル送受信ディスクリプタを示します。エンハンスド・ディスクリプタを使用する場合は、ディスクリプタがシステム・メモリに応じて作成されていることを確認する必要があります。エンハンスド・ディスクリプタについて詳しくは、17-53 ページの「代替またはエンハンスド・ディスクリプタ」を参照してください。†

ノーマル・ディスクリプタ

これは、デフォルトでのディスクリプタ・フォーマットで、4 DWORDS を持つことができます。IEEE 1588 アドバンスド・タイムスタンプなどの高度な機能がイネーブルされていないときに、このディスクリプタ・フォーマットを使用できます。

送信ディスクリプタ

EMAC の DMA は、送信フレームに対して少なくとも 1 つのディスクリプタを必要とします。2 つのバッファ、2 つのバイト・カウンタ・バッファ、および 2 つのアドレス・ポインタに加えて、送信ディスクリプタにはコントロール・フィールドがあります。これは、送信フレームごとの MAC 動作を制御するために使用されます。

送信ディスクリプタ 0 (TDES0)

TDES0 には、送信されたフレーム・ステータスおよびディスクリプタ所有権の情報が含まれています。

表 17-6. 送信ディスクリプタ 0 (その 1)

ビット	説明
31	OWN : 所有ビット 設定されている場合、このビットは、ディスクリプタが DMA に所有されていることを示します。このビットがリセットされている場合、ディスクリプタがホストに所有されていることを示します。フレーム送信が完了すると、またはディスクリプタに割り当てられているバッファが空になると、DMA はこのビットをクリアします。フレームの先頭ディスクリプタの所有権ビットは、同じフレームに属する先頭以降の全ディスクリプタが設定された後に設定される必要があります。こうすることによって、ディスクリプタのフェッチとドライバの所有権ビットの設定の間に起きうる競合状態を避けることができます。†
30:18	予約 †
17	TTSS : TX タイムスタンプ・ステータス このステータス・ビットは、関連する送信フレーム用にタイムスタンプがキャプチャされていることを示します。このビットが設定されている場合、TDES2 および TDES3 は送信フレーム用にキャプチャされたタイムスタンプ値を持っています。このフィールドは、ディスクリプタの終端セグメントのコントロール・ビット (TDES1[30]) が設定されている場合のみ、有効です。
16	IHE : IP ヘッダ・エラー 設定されている場合、このビットはチェックサム・オフロード・エンジンが IP ヘッダ・エラーを検出したことを示し、転送されたフレームがチェックサムの挿入のために結果的に変更されなかったことを示します。
15	ES : エラー要約 以下に各ビットのロジカル OR を示します。 ■ TDES0[16] : IP ヘッダ・エラー ■ TDES0[14] : Jabber タイムアウト ■ TDES0[13] : フレーム・フラッシュ ■ TDES0[12] : ペイロード・チェックサム・エラー ■ TDES0[11] : キャリアの喪失 ■ TDES0[10] : キャリアなし ■ TDES0[9] : 遅延衝突 ■ TDES0[8] : 過度の衝突 ■ TDES0[2] : 過度の延期 ■ TDES0[1] : アンダーフロー・エラー †
14	JT : Jabber タイムアウト 設定されている場合、このビットは MAC トランスマッタに Jabber タイムアウトがあったことを示します。レジスタ 0 (MAC コンフィギュレーション・レジスタ) の JD ビットが設定されていない場合のみ、このビットが設定されます。†
13	FF : フレーム・フラッシュ 設定されている場合、このビットは、CPU によって与えられたソフトウェア・フラッシュ・コマンドによって DMA または MAC トランザクション・レイヤ (MTL) がフレームをフラッシュしたことを示します。†

表 17-6. 送信ディスクリプタ 0 (その 2)

ビット	説明
12	12 PCE : ペイロード・チェックサム・エラー 設定されている場合、このビットは、チェックサム・オフロード・エンジンが失敗して、カプセル化された TCP、UDP、または ICMP のペイロードにチェックサムを挿入しなかったことを示します。この失敗は、IP ヘッダのペイロード長フィールドによって示されるように、不適切なバイトによるものか、または、MTL がストア・アンド・フォワード・モードでチェックサムが計算されていない状態で MAC トランスミッタにフレームを転送し始めたことによるもののどちらか一方である可能性があります。後者の状態は、送信 FIFO バッファのデプスが、デッドロックを避けるために送信されるイーサネット・フレームの長さより浅く、ストア・アンド・フォワード・モードであるにもかかわらず FIFO バッファがフルのときに MTL がフレーム転送を開始する場合にのみ発生します。
11	LC : キャリアの喪失 設定されている場合、このビットはフレーム送信中にキャリアの喪失が発生したことを示します (つまり、フレーム送信中に 1 以上の送信クロック期間で gmii_crs_i 信号が非アクティブであったということです)。これは、衝突することなく送信されたフレームのみ、および MAC が半二重モードで動作する場合のみに有効です。†
10	NC : キャリアなし 設定されている場合、このビットは PHY からのキャリア・センス信号が送信中にアサートされなかったことを示します。†
9	予約
8	EC : 過度の衝突 設定されている場合、このビットは、現在のフレームを送信しようと試みて、16 の連続した衝突後に送信がアボートされたことを示します。レジスタ 0 (MAC コンフィギュレーション・レジスタ) のビット 9 (ディセーブル・リトライ) が設定されている場合、このビットは、最初の衝突後に設定されてフレームの送信がアボートされます。†
7	VF : VLAN フレーム 設定されている場合、このビットは、送信されたフレームが VLAN タイプのフレームだったことを示します。†
6:3	CC : 衝突回数 この 4 ビット・カウンタ値は、フレームが送信される前に発生した衝突の回数を示しています。この回数は、過度の衝突ビット (TDES0[8]) が設定されている場合は有効ではありません。†
2	ED : 過度の延期 設定されている場合、ビット 4 (延期チェック) がレジスタ 0 (MAC コンフィギュレーション・レジスタ) で High に設定されているとき、このビットは、24,288 ビット・タイム (1000 Mbps モードまたは Jumbo フレーム・イネーブル・モードでは 155,680 ビット・タイム) を超える過度の延期のために送信が終了したことを示します。†
1	UF : アンダーフロー・エラー 設定されている場合、このビットは、ホスト・メモリからのフレーム到着が遅かったために、MAC がフレームをアボートしたことを示します。アンダーフロー・エラーは、DMA がフレーム送信中に空の送信バッファを検出したことを示します。送信プロセスは中断ステートになり、送信アンダーフロー (レジスタ 5[5]) および送信割り込み (レジスタ 5[0]) の両方を設定します。†
0	DB : 延期ビット 設定されている場合、このビットは、キャリアの存在のために MAC が送信前に延期することを示します。このビットは、半二重モードのみで有効です。†

送信ディスクリプタ 1 (TDES1)

TDES1 には、ディスクリプタのチェイン/リングとフレーム転送を制御するバッファ・サイズおよび他のビットが含まれています。†

 バッファ・サイズの計算について詳しくは、17-26 ページの「バッファ・サイズの計算」を参照してください。†

表 17-7. 送信ディスクリプタ 1 (その 1)

ビット	説明
31	IC : 完了時割り込み 設定されている場合、このビットは、現在のフレームが送信された後に送信割り込み（レジスタ 5[0]）を設定します。†
30	LS : 終端セグメント 設定されている場合、このビットは、バッファにフレームの終端セグメントが含まれていることを示します。このビットが設定されている場合、TBS1 または TBS2 のフィールドはゼロでない値を持っている必要があります。†
29	FS : 先頭セグメント 設定されている場合、このビットは、バッファにフレームの先頭セグメントが含まれていることを示します。†
28:27	CIC : チェックサム挿入コントロール これらのビットは、以下に示すように、IPv4 または IPv6 を介して TCP、UDP、または ICMP をカプセル化するイーサネット・フレームのチェックサムの挿入を制御します。 ■ 0 : 何もしません。チェックサム・エンジンがバイパスされます。 ■ 1 : IPv4 ヘッダ・チェックサムを挿入します。フレームが IPv4 データグラムをカプセル化する場合、この値を使用して IPv4 ヘッダ・チェックサムを挿入します。 ■ 2 : TCP/UDP/ICMP チェックサムを挿入します。チェックサムは、TCP、UDP または ICMP のセグメントのみを介して計算されて、TCP、UDP、または ICMP の擬似ヘッダ・チェックサムが、関連の入力フレームのチェックサム・フィールドに存在しているものとして見なされます。カプセル化されたデータグラムが IPv4 に準拠している場合、IPv4 ヘッダ・チェックサムも挿入されます。 ■ 3 : このエンジンでフルに計算されている TCP/UDP/ICMP チェックサムを挿入します。つまり、チェックサム計算には TCP、UDP、または ICMP の擬似ヘッダが含まれており、入力フレームに関連するチェックサム・フィールドはすべてゼロの値を持っています。カプセル化されたデータグラムが IPv4 に準拠している場合、IPv4 ヘッダ・チェックサムも挿入されます。 チェックサム・エンジンは、TCP、UDP、または ICMP セグメントが IPv4 または IPv6 のどちらかにカプセル化されるか検出し、データに応じて処理します。
26	DC : CRC のディセーブル 設定されている場合、MAC は、送信されたフレームの終端に CRC を追加しません。これは、先頭のセグメント（TDES1[29]）が設定されている場合のみ有効です。†
25	TER : リング終端の送信 設定されている場合、このビットは、ディスクリプタ・リストが最終のディスクリプタに達したことを示します。DMA はディスクリプタ・リングを作成してリストのベース・アドレスに戻ります。†
24	TCH : 2 番目のアドレスへのチェイン 設定されている場合、このビットは、ディスクリプタの 2 番目のアドレスが 2 番目のバッファ・アドレスではなく次のディスクリプタ・アドレスであることを示します。TDES1[24] が設定されている場合、TBS2（TDES1[21-11]）は「don't care」値です。TDES1[25] は TDES1[24] よりも優先されます。†
23	DP : パッドのディセーブル 設定されている場合、MAC は 64 バイトより短いフレームにはパディングを自動的に追加しません。このビットがリセットされていると、DMA は 64 バイトより短いフレームにパディングと CRC を自動的に追加して、DC（TDES1[26]）ビットのステートにかかわらず CRC フィールドが追加されます。これは、先頭セグメント（TDES1[29]）が設定されている場合のみ有効です。†

表 17-7. 送信ディスクリプタ 1 (その 2)

ビット	説明
22	TTSE : 送信タイムスタンプのイネーブル 設定されている場合、このビットは、ディスクリプタに参照される送信フレーム用の IEEE1588 ハードウェア・タイムスタンプをイネーブルします。このフィールドは、先頭セグメント・コントロール・ビット (TDES1[29]) が設定されている場合のみ有効です。†
21:11	TBS2 : 送信バッファ 2 サイズ これらのビットは、2 番目のデータ・バッファをバイトで示します。このフィールドは、TDES1[24] が設定されているときは無効です。†
10:0	TBS1 : 送信バッファ 1 サイズ これらのビットは、先頭のデータ・バッファ・バイト・サイズを示します。このフィールドがゼロの場合、DMA はこのバッファを無視して、TCH (ビット 24) の値に応じてバッファ 2 または次のディスクリプタを使用します。†

送信ディスクリプタ 2 (TDES2)

TDES2 には、ディスクリプタの先頭バッファへのアドレス・ポインタが含まれています。†

表 17-8. 送信ディスクリプタ 2

ビット	説明
31:0	バッファ 1 アドレス・ポインタ これらのビットは、バッファ 1 の物理アドレスを示します。バッファ・アドレス・アラインメントに制約はありません。バッファ・アドレス・アラインメントについて詳しくは、 17-26 ページの「ホスト・データ・バッファ・アラインメント」 を参照してください。†

送信ディスクリプタ 3 (TDES3)

TDES3 には、ディスクリプタの 2 番目のバッファまたは次のディスクリプタのどちらか一方に対するアドレス・ポインタが含まれています。†

表 17-9. 送信ディスクリプタ 3

ビット	説明
31:0	バッファ 2 アドレス・ポインタ (次のディスクリプタ・アドレス) ディスクリプタのリング構造が使用されている場合は、バッファ 2 の物理アドレスを示します。2 番目のアドレス・チェイン (TDES1[24]) ビットが設定されている場合、このアドレスには次のディスクリプタがある物理メモリへのポインタが含まれています。TDES1[24] が設定されている場合のみ、バッファ・アドレス・ポインタはバス幅にアラインメントされる必要があります (LSB は内部で無視されます)。†

受信ディスクリプタ

EMAC の DMA は、フレームを受信するとき少なくとも 2 つのディスクリプタを必要とします。DMA の受信ステート・マシンの、常に、受信するフレームを見越して余分なディスクリプタを取得しようと試みます (受信フレームのサイズは未知です)。RX DMA がディスクリプタを閉じる前に、受信するフレームがない場合でも次のディスクリプタの取得を試みます。

シングル・ディスクリプタ（受信）システムでは、受信バッファが受信するフレームに対応できず次のディスクリプタが DMA に所有されていない場合、サブシステムがディスクリプタ・エラーを生成します。このため、ホストはディスクリプタ・プールまたはバッファ・サイズのどちらか一方を増大するよう強制されます。その他の場合では、サブシステムはすべての受信フレームをドロップし始めます。†

受信ディスクリプタ 0 (RDES0)

RDES0 には、受信フレームのステータス、フレーム長、およびディスクリプション所有権の情報が含まれています。

表 17-10. 受信ディスクリプタ 0 (その 1)

ビット	説明
31	OWN : 所有ビット 設定されている場合、このビットは、ディスクリプタが EMAC の DMA に所有されていることを示します。このビットがリセットされていると、このビットはディスクリプタがホストに所有されていることを示します。DMA は、フレーム受信を完了したとき、またはこのディスクリプタに関連するバッファがフルのとき、このビットをクリアします。
30	AFM : デスティネーション・アドレス・フィルタ失敗 設定されている場合、このビットは、MAC の DA フィルタで失敗したフレームを示します。†
29:16	FL : フレーム長 これらのビットは、ホスト・メモリ（CRC を含む）に対して転送された受信フレームのバイト長を示します。終端ディスクリプタ（RDES0[8]）が設定されていて、ディスクリプタ・エラー・ビット（RDES0[14]）またはオーバーフロー・エラー・ビットのどちらか一方がリセットされている場合、このフィールドは有効です。フレーム長は、IP チェックサム計算（タイプ 1）がイネーブルされていて受信フレームが MAC コントロール・フレームではないとき、イーサネット・フレームに付加される 2 バイトも表示します。 終端ディスクリプタ（RDES0[8]）が設定されている場合、このフィールドは有効です。終端ディスクリプタ・ビットおよびエラー要約ビットが設定されていない場合、このフィールドは現在のフレーム転送での累積バイト数を示します。†
15	ES : エラー要約 以下に各ビットのロジカル OR を示します。 ■ RDES0[0] : ペイロード・チェックサム・エラー ■ RDES0[1] : CRC エラー ■ RDES0[3] : 受信エラー ■ RDES0[4] : ウォッチドッグ・タイムアウト ■ RDES0[6] : 遅延衝突 ■ RDES0[7] : IPC チェックサム（タイプ 2） ■ RDES0[11] : オーバーフロー・エラー ■ RDES0[14] : ディスクリプタ・エラー 終端ディスクリプタ（RDES0[8]）が設定されている場合のみ、このフィールドが有効です。†
14	DE : ディスクリプタ・エラー 設定されている場合、このビットは、フレームが現在のディスクリプタ・バッファの範囲内にフィットせず DMA が次のディスクリプタを所有していないためにフレームが切り捨てられることを示します。フレームは切り捨てられます。このフィールドは、終端ディスクリプタ（RDES0[8]）が設定されている場合のみ有効です。†

表 17-10. 受信ディスクリプタ 0 (その 2)

ビット	説明
13	SAF : ソース・アドレス・フィルタ失敗 設定されている場合、このビットは MAC の SA フィルタに失敗したフレームの SA フィールドを示します。†
12	LE : 長さエラー 設定されている場合、このビットは、受信フレームの実際の長さおよび長さ / タイプ・フィールドがマッチしないことを示します。このビットは、フレーム・タイプ (RDES0[5]) ビットがリセットされている場合のみ有効です。長さエラー・ステータスは、CRC がエラーがあるときは無効です。†
11	OE : オーバーフロー・エラー 設定されている場合、このビットは、RX FIFO バッファでのバッファ・オーバーフローによって受信フレームがダメージを受けていることを示します。 注：このビットは、DMA がアプリケーションに部分的なフレームを転送する場合のみ設定されます。これは、RX FIFO バッファがスレッシュホールド・モードで動作する場合のみ発生します。ストア・アンド・フォワード・モードでは、部分的なフレームのすべてが RX FIFO バッファでドロップされます。†
10	VLAN : VLAN タグ 設定されている場合、このビットは、このディスクリプタにポイントされているフレームが MAC にタグされた VLAN フレームであることを示します。VLAN タグは、レジスタ 7 (VLAN タグ・レジスタ) の設定に基づいて、受信されるフレームの VLAN フィールドのチェックに応じて異なります。†
9	FS : 先頭のディスクリプタ 設定されている場合、このビットは、このディスクリプタにフレームの先頭バッファが含まれていることを示します。先頭バッファのサイズがゼロの場合、2 番目のバッファにフレームの最初が含まれています。2 番目のバッファのサイズもゼロの場合、次のディスクリプタにフレームの最初が含まれています。†
8	LS : 終端のディスクリプタ 設定されている場合、このビットは、ディスクリプタによってポイントされているバッファがフレームの最終バッファであることを示します。†
7	IPC チェックサム・エラー 設定されている場合、IP チェックサム・エンジン (タイプ 1) がイネーブルされているとき、このビットは、MAC によって計算された 16 ビットの IPv4 のヘッダ・チェックサムが受信チェックサム・バイトと一致しなかったことを示します。このビットがこのモードで設定されている場合、ビット 15 (ES) は設定されません。フル・チェックサム・オフロード・エンジン (タイプ 2) がイネーブルされている場合にこのビットが設定されていると、IPv4 または IPv6 のヘッダでエラーが発生していることを示します。このエラーは、イーサネット・タイプ・フィールド値と IP ヘッダ・バージョン・フィールド値の不一致、IPv4 でのヘッダ・チェックサム・ミスマッチ、または期待される IP ヘッダ・バイト値がイーサネット・フレームに欠如していることが原因として考えられます。詳しくは、17-47 ページの 表 17-11 を参照してください。
6	LC : 遅延衝突 設定されている場合、このビットは、半二重モードでフレームを受信しているときに遅延衝突が発生したを示します。†
5	FT : フレーム・タイプ 設定されている場合、このビットは、受信フレームがイーサネット・タイプのフレームであることを示します (LT フィールドは 0x0600 以上です)。このビットがリセットされている場合、受信フレームは IEEE802.3 フレームです。このビットは 14 バイト未満の小型フレームには無効です。詳しくは、17-47 ページの 表 17-11 を参照してください。†

表 17-10. 受信ディスクリプタ 0 (その 3)

ビット	説明
4	RWT : 受信ウォッチドッグ・タイムアウト 設定されている場合、このビットは、現在のフレームを受信している間に受信ウォッチドッグ・タイマが終了し、ウォッチドッグ・タイムアウト後に現在のフレームが切り捨てられることを示します。†
3	RE : 受信エラー 設定されている場合、このビットは、フレーム受信中に <code>gmii_rxer_i</code> がアサートされているとき <code>gmii_rxdv_i</code> 信号がアサートされることを示します。エラーは、非常に小さいものか拡張のないものである可能性があり、または拡張中のエラー (<code>rxcl != 0xf</code>) である可能性もあります。†
2	DE : ドリブル・ビット・エラー 設定されている場合、このビットは、受信フレームに非整数の複数のバイトがあることを示します (奇数ニブル)。このビットは MII モードのみで有効です。†
1	CE : CRC エラー 設定されている場合、このビットは、受信フレームで CRC エラーが発生したことを示します。このフィールドは、終端ディスクリプタ (RDES0[8]) が設定されている場合のみ有効です。†
0	RX での MAC アドレスまたはペイロード・チェックサムのエラー 設定されている場合、このビットは、RX MAC アドレス・レジスタ値 (1 ~ 15) がフレームの DA フィールドに一致したことを示します。リセットされている場合、このビットは、RX MAC アドレス・レジスタ 0 の値が DA フィールドに一致したことを示します。フル・チェックサム・オフロード・エンジンがイネーブルされているときにこのビットが設定されている場合、EMAC が計算した TCP、UDP、または ICMP のチェックサムが、カプセル化されている受信 TCP、UDP、または ICMP セグメントのチェックサム・フィールドに一致していないことを示します。このビットは、受信したペイロード・バイトの数が、受信イーサネット・フレームのカプセル化されている IPv4 または IPv6 の長さフィールドが示す値に一致しない場合にも設定されます。詳しくは、 17-47 ページ の 表 17-11 を参照してください。†

フル・チェックサム・オフロード・エンジン（タイプ 2）がイネーブルされている場合、ビット 5、ビット 7、およびビット 0 の組み合わせは表 17-11 で説明する条件を反映します。†

表 17-11. COE（タイプ 2）がイネーブルされているときの受信ディスクリプタ 0

ビット 5： フレーム・ タイプ	ビット 7： IPC チェック サム・エラー	ビット 0： ペイロード・ チェックサ ム・エラー	フレーム・ステータス
0	0	0	IEEE 802.3 タイプのフレーム（長さフィールド値が 0x0600 未満）†
1	0	0	IPv4/IPv6 タイプのフレームで、チェックサム・エラーの検出なし †
1	0	1	IPv4/IPv6 タイプのフレームで、チェックサム・エラーの検出あり（PCE で説明したように）†
1	1	0	IPv4/IPv6 タイプのフレームで、IP ヘッダのチェックサム・エラーの検出あり（IPC CE で説明したように）†
1	1	1	IPv4/IPv6 タイプのフレームで、IP ヘッダおよびペイロードの両方のチェックサム・エラーの検出あり †
0	0	1	IPv4/IPv6 タイプのフレームで、IP ヘッダのチェックサム・エラーはなく、サポートされていないペイロードのためにペイロード・チェックのバイパスあり †
0	1	1	IPv4 でも IPv6 でもないタイプのフレーム（チェックサム・オフロード・エンジンが完全にチェックサムをバイパスする）†
0	1	0	予約 †

受信ディスクリプタ 1（RDES1）

RDES1 には、ディスクリプタ・チェインまたはディスクリプタ・リングを制御するバッファ・サイズと他のビットが含まれています。†

 バッファ・サイズの計算について詳しくは、17-26 ページの「バッファ・サイズの計算」を参照してください。†

表 17-12. 受信ディスクリプタ 1（その 1）

ビット	説明
31	完了時割り込みのディセーブル 設定されている場合、このビットは、このディスクリプタによってポイントされているバッファで終端の受信フレームに対してレジスタ 5（ステータス・レジスタ）のビット 6（RI）が設定されるのを回避します。つまり、そのフレームの RI のために、ホストへの割り込みアサートをディセーブルするということです。†
30:26	予約 †

表 17-12. 受信ディスクリプタ 1 (その 2)

ビット	説明
25	RER : リング終端の受信 設定されている場合、このビットは、ディスクリプタ・リストが終端ディスクリプタに達したことを示します。DMA は、リストのベース・アドレスに戻り、ディスクリプタ・リングを作成します。†
24	RCH : 2 番目のアドレスへのチェイン 設定されている場合、このビットは、ディスクリプタの 2 番目のアドレスは 2 番目のバッファ・アドレスではなく次のディスクリプタ・アドレスであることを示します。RDES1[24] が設定されている場合、RBS2 (RDES1[21-11]) は「don't care」値です。RDES1[25] は RDES1[24] よりも優先されます。†
23:22	予約 †
21:11	RBS2 : 受信バッファ 2 サイズ これらのビットは、2 番目のデータ・バッファ・サイズをバイトで示します。RDES3 (バッファ 2 アドレス・ポインタ) の値がバス幅にアラインメントされていない場合でも、このバッファ・サイズは 4 の倍数である必要があります。バッファ・サイズが 4 の倍数でない場合、予期せぬ動作を引き起こしてしまいます。RDES1[24] が設定されている場合、このフィールドは無効です。
10:0	RBS1 : 受信バッファ 1 サイズ 先頭のデータ・バッファ・サイズをバイトで示します。RDES2 (バッファ 1 アドレス・ポインタ) の値がアラインメントされていない場合でも、バッファ・サイズは 4 の倍数である必要があります。バッファ・サイズが 4 の倍数ではない場合、予期せぬ動作を引き起こしてしまいます。このフィールドがゼロの場合、DMA はこのバッファを無視して、RCH (ビット 24) の値に応じてバッファ 2 または次のディスクリプタを使用します。

受信ディスクリプタ 2 (RDES2)

RDES2 には、ディスクリプタの先頭のデータ・バッファへのアドレス・ポインタが含まれています。†



バッファ・アドレス・アラインメントについて詳しくは、17-26 ページの「[ホスト・データ・バッファ・アラインメント](#)」を参照してください。†

表 17-13. 受信ディスクリプタ 2 (デフォルト動作)

ビット	説明
31:0	バッファ 1 アドレス・ポインタ これらのビットは、バッファ 1 の物理アドレスを示します。次の条件を除いて、バッファ・アドレス・アラインメントに制約はありません。RDES2 値を使用してフレーム開始を格納する場合、DMA はアドレス生成のために RDES2[1:0] でプログラムされた値を使用します。フレーム開始の転送中、DMA は RDES2[1:0] ビットをゼロとして書き込み動作を実行しますが、フレーム・データは実際のバッファ・アドレス・ポインタに従ってシフトされます。アドレス・ポインタがフレームの中間または終端部分に格納されているバッファに対するものである場合、DMA は RDES2[1:0] を無視します。

受信ディスクリプタ 3 (RDES3)

RDES3 には、ディスクリプタの 2 番目のデータ・バッファまたは次のディスクリプタのどちらか一方に対するアドレス・ポインタが含まれています。†

表 17-14. 受信ディスクリプタ 3

ビット	説明
31:0	バッファ 2 アドレス・ポインタ (次のディスクリプタ・アドレス) これらのビットは、ディスクリプタ・リング構造を使用しているときのバッファ 2 の物理アドレスを示します。2 番目のアドレス・チェーン (RDES1[24]) ビットが設定されている場合、このアドレスには次のディスクリプタがある物理メモリへのポインタが含まれています。 RDES1[24] が設定されている場合、バッファ (次のディスクリプタ)・アドレス・ポインタはバス幅にアラインメントされる必要があります (RDES3[1:0] = 0、LSB は内部で無視されます)。しかし、RDES1[24] がリセットされている場合、以下の条件を除いて RDES3 値に制約はありません。RDES3 値を使用してフレーム開始を格納する場合、DMA はアドレス生成のために RDES3[1:0] でプログラムされた値を使用します。アドレス・ポインタがフレームの中間または終端部分に格納されているバッファに対するものである場合、DMA は RDES3[1:0] を無視します。

IEEE 1588-2005 タイムスタンプがイネーブルされているときのディスクリプタ・フォーマット

デフォルトでのディスクリプタ・フォーマット (17-39 ページの「送信ディスクリプタ」および 17-43 ページの「受信ディスクリプタ」とフィールドの記述は、ソフトウェアで作成される場合 (所有ビットが DES0 に設定されている場合) には変更されことなく維持されます。†

しかし、ソフトウェアが IEEE 1588-2005 の機能をイネーブルしていると、DMA がディスクリプタを閉じている場合 (DES0 の所有ビットがクリアされている場合)、DES2 および DES3 のディスクリプタ・フィールド (表 17-15 を参照) は異なる意味を持つことになります。†

DMA は、DES0 の所有ビットをクリアする前にタイムスタンプ値を使用して DES2 および DES3 を更新します。EMAC が 32 ビット・モードで動作している場合、DES2 はタイムスタンプの下位 32 ビット (サブ・セカンド・フィールド、以降の項では TSL と略記) で更新されて、DES3 はタイムスタンプの上位 32 ビット (セカンド・フィールド、以降の項では TSH と略記) で更新されます。

表 17-15. DMA が所有ビットをクリアする場合のディスクリプタ・フィールド

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DES0																																
DES1																																
DES2	タイムスタンプ Low[31:0]																															
DES3	タイムスタンプ High[31:0]																															

以下の項では、このモードでの送受信ディスクリプタに特別な事項を詳しく説明します。†

送信ディスクリプタ

17-49 ページの「IEEE 1588-2005 タイムスタンプがイネーブルされているときのディスクリプタ・フォーマット」の記載事項の変更に加えて、このモードでの送信ディスクリプタには、表 17-16 に示すように、タイムスタンプ用の追加のコントロール・ビットおよびステータス・ビット（それぞれ TTSE および TTSS）があります。ソフトウェアは TTSE ビットを設定して（所有ビットが設定されている場合）、関連するイーサネット・フレームを転送するために EMAC がタイムスタンプを生成するように指示します。タイムスタンプが TDES2 および TDES3 のフィールドで更新されている場合、ディスクリプタが閉じているとき（所有ビットがクリアされているとき）DMA は TTSS ビットを設定します。

表 17-16. 送信ディスクリプタ・フィールド - 通常のフォーマット

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	OWN	RES												TTSS	ステータス [16:0]																	
TDES1	他のコントロール・フィールド									T T S E	バッファ 2 バイト・カウント [21:11]										バッファ 1 バイト・カウント [10:0]											
TDES2	TTSL																															
TDES3	TTSH																															

送信タイムスタンプのコントロール・フィールドおよびステータス・フィールド

これらのフィールドの位置は、通常の送信ディスクリプタとエンハンスド・フォーマットの送信ディスクリプタとは異なります。どちらの場合でも、ディスクリプタを閉じるときディスクリプタによってこのフィールドの値が維持される必要があります。†

デフォルト（通常）のディスクリプタ・フォーマットでは、17-49 ページの表 17-14 および 17-40 ページの表 17-6 で更新される箇所は以下の通りです。†

表 17-17. 送信タイムスタンプ・ステータス - 通常のディスクリプタ・フォーマットの場合（TDES0）

ビット	説明
17	TTSS : 送信タイムスタンプ・ステータス このフィールドは、関連する送信フレーム用にタイムスタンプがキャプチャされたことを示すステータス・ビットです。このビットが設定されている場合、TDES2 および TDES3 の両方は、送信フレーム用にキャプチャされたタイムスタンプ値を持つことになります。 このフィールドは、終端セグメント・コントロール・ビット（ディスクリプタの TDES1[30]）が設定されている場合のみ有効です。†

表 17-18. 送信タイムスタンプ・コントロール – 通常のディスクリプタ・フォーマットの場合 (TDES1)

ビット	説明
22	TTSE : 送信タイムスタンプのイネーブル 設定されている場合、このフィールドは、IEEE1588 ハードウェアがディスクリプタで示される送信フレーム用にタイムスタンプできるようにします。このフィールドは、先頭セグメント・コントロール・ビット (ディスクリプタの TDES1[29]) が設定されている場合のみ有効です。†

送信タイムスタンプ・フィールド

送信ディスクリプタのフォーマットおよびフィールドの記述は、それらがソフトウェアで作成される場合には変更されずに維持されます (所有ビットが設定されている場合)。しかし、DMA が終端ディスクリプタ (TDES1 または TDES0 の LS ビットによって代替ディスクリプタ・フォーマットでマークされているディスクリプタ) を閉じて IEEE 1588 機能がイネーブルされる (所有ビットがクリアされる) 場合、TDES2 および TDES3 のディスクリプタ・フィールドは、フレーム用に使用されているときはタイムスタンプで更新されます。†

表 17-19 および表 17-20 に、ディスクリプタが閉じている場合に異なる意味を持つフィールドを示します。†

表 17-19. 送信ディスクリプタ・フィールド (TDES2)

ビット	説明
31:0	TTSL : 送信フレーム・タイムスタンプ Low このフィールドは、関連する送信フレーム用にキャプチャされたタイムスタンプの最下位 32 ビットを使用して DMA によって更新されます。ディスクリプタの終端セグメント・コントロール・ビット (LS) が設定されている場合のみ、このフィールドはタイムスタンプを持ちます。†

表 17-20. 送信ディスクリプタ・フィールド (TDES3)

ビット	説明
31:0	TTSH : 送信フレーム・タイムスタンプ High このフィールドは、関連する送信フレーム用にキャプチャされたタイムスタンプの最上位 32 ビットを使用して DMA によって更新されます。ディスクリプタの終端セグメント・コントロール・ビット (LS) が設定されている場合のみ、このフィールドはタイムスタンプを持ちます。†

受信ディスクリプタ

受信タイムスタンプ

表 17-21 に、タイムスタンプがイネーブルされている場合の受信ディスクリプタのフォーマットを示します。

表 17-21. 受信タイムスタンプ・フィールド

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDES0																																
RDES1																																
RDES2	受信フレーム・タイムスタンプ Low[31:0]																															
RDES3	受信フレーム・タイムスタンプ High[31:0]																															

17-52 ページの 表 17-22 および 17-52 ページの 表 17-23 に、受信ディスクリプタが閉じていてタイムスタンプがイネーブルされている場合、RDES2 および RDES3 で異なる意味を持つフィールドを示します。

 ソフトウェアがタイムスタンプ機能をディセーブルする場合（レジスタ 448 の TSENA ビットが Low の場合）、DMA は RDES0 を閉じる前にディスクリプタの RDES2 または RDES3 を更新しません。†

表 17-22. 受信ディスクリプタ・フィールド（RDES2）

ビット	説明
31:0	RTSL : 受信フレーム・タイムスタンプ Low このフィールドは、関連する受信フレーム用にキャプチャされたタイムスタンプの最下位 32 ビットを使用して DMA によって更新されます。DMA は、終端ディスクリプタ・ステータス・ビット (RDES0[8]) によって示される受信フレームの終端ディスクリプタのみでこのフィールドを更新します。RDES3 のこのフィールドと RTSH フィールドがすべてゼロの値の場合、タイムスタンプは破壊されているものとして扱われる必要があります。†

表 17-23. 受信ディスクリプタ・フィールド（RDES3）

ビット	説明
31:0	RTSH : 受信フレーム・タイムスタンプ High このフィールドは、関連する受信フレーム用にキャプチャされたタイムスタンプの最上位 32 ビットを使用して DMA によって更新されます。DMA は、終端ディスクリプタ・ステータス・ビット (RDES0[8]) によって示される受信フレームの終端ディスクリプタのみでこのフィールドを更新します。 このフィールドと RDES の RTSL フィールドがすべてゼロの値の場合、タイムスタンプは破壊されているものとして扱われる必要があります。†

代替またはエンハンスド・ディスクリプタ

通常のディスクリプタでは 4 つの DWORDS を持つことができましたが、代替（またはエンハンスド）ディスクリプタ構造では 8 つの DWORDS（32 バイト）を持つことができます。代替ディスクリプタ構造の機能は、以下の通りです。

- 通常のディスクリプタ構造では最大 2,048 バイトのデータ・バッファが可能でしたが、代替ディスクリプタ構造は、最大 8 KB のバッファをサポートするように実装されます（Jumbo フレームで便利です）。
- TDES0、TDES1、RDES0（アドバンスド・タイムスタンプまたは IPC フル・オフロードのコンフィギュレーション）、および RDES1 にコントロール・ビットおよびステータス・ビットの再割り当てがあります。†
- 送信ディスクリプタは、アドバンスド・タイムスタンプが選択されている場合、タイムスタンプを TDES6 および TDES7 に格納します。†
- アドバンスド・タイムスタンプ、IPC フル・チェックサム・オフロード・エンジン、またはレイヤ 3 とレイヤ 4 のフィルタ機能が選択されている場合、この受信ディスクリプタ構造も、拡張ステータス（RDES4）およびタイムスタンプ（RDES6 および RDES7）を格納するために使用されます。†
- ディスクリプタ構造について、以下のオプションから 1 つ選択できます。
 - タイムスタンプがレジスタ 448（タイムスタンプ・コントロール・レジスタ）でイネーブルされている場合、またはチェックサム・オフロードがレジスタ 0（MAC コンフィギュレーション・レジスタ）でイネーブルされている場合、ソフトウェアは 32 バイト（8 DWORDS）のメモリをすべてのディスクリプタに割り当てる必要があります。このため、ソフトウェアはレジスタ 0（バス・モード・レジスタ）のビット 7（代替ディスクリプタ・サイズ）を設定する必要があります。†
 - タイムスタンプまたはチェックサム・オフロードがイネーブルされていない場合、拡張ディスクリプタ（DES4～DES7）は必要ありません。したがって、ソフトウェアは 16 バイト（4 DWORDS）のデフォルト・サイズで代替ディスクリプタを使用できます。そのために、ソフトウェアはレジスタ 0（代替ディスクリプタ・サイズ）のビット 7（バス・モード・レジスタ）をゼロにリセットする必要があります。†
- タイムスタンプまたは受信 IPC フル・チェックサム・オフロード・エンジン（タイプ 2）の機能なしに代替ディスクリプタが選択されている場合、ディスクリプタのサイズは常に 4 DWORDS（DES0～DES3）です。したがって、ソフトウェアは代替ディスクリプタを 16 バイトのデフォルト・サイズで使用できます。†

送信ディスクリプタ

送信ディスクリプタ構造は、17-54 ページの 表 17-24 に示されています。アプリケーション・ソフトウェアは、ディスクリプタを初期化している間にコントロール・ビット TDES0[31:18] をプログラムする必要があります。DMA がディスクリプタを更新する場合、所有ビット以外のすべてのコントロール・ビットを書き戻してステータス・ビット [7:0] を更新します（所有ビットはクリアされます）。トランスミッタ・ディスクリプタのワード 0（TDES0）～ワード 3（TDES3）の内容は、それぞれ 17-55 ページの 表 17-26、17-58 ページの 表 17-27、17-58 ページの 表 17-28、および 17-59 ページの 表 17-29 で説明されています。†

アドバンスド・タイムスタンプのサポートを使用すると、TDES0 のビット 25 (TTSE) を設定することによって、与えられているフレーム用にタイムスタンプのスナップショットをイネーブルできます。ディスクリプタが閉じている場合（つまり所有ビットがクリアされている場合）、タイムスタンプは TDES6 および TDES7 に書き込まれます。このことは、17-54 ページの表 17-24 で示した TDES0 のステータス・ビット 17 (TTSS) によって表示されます。TDES6 および TDES7 の内容については、17-59 ページの表 17-30 および 17-59 ページの表 17-31 で説明しています。†



アドバンスド・タイムスタンプの機能がイネーブルされている場合、DMA が拡張ディスクリプタ・サイズで動作するように、ソフトウェアはレジスタ 0（バス・モード・レジスタ）のビット 7 を設定する必要があります。このコントロール・ビットがリセットされている場合、TDES4 ～ TDES7 のディスクリプタ・スペースは無効です。†

表 17-24. 送信ディスクリプタ・フィールド - 代替（エンハンスド）フォーマット

送信データ・アドレス・ステータス・予約 (エンベデッド) シグナル																																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	OWN	コントロール [30:26]					TTSE	コントロール [24:18]							TTSS	ステータス [16:7]										コントロール/ステータス [6:3]			ステータス [2:0]			
TDES1	コントロール [31:29]		バッファ 2 バイト・カウンタ [28:16]										RES			バッファ 1 バイト・カウンタ [12:0]																
TDES2	バッファ 1 アドレス [31:0]																															
TDES3	バッファ 2 アドレス [31:0] または次のディスクリプタ・アドレス [31:0]																															
TDES4	予約																															
TDES5	予約																															
TDES6	送信タイムスタンプ Low[31:0]																															
TDES7	送信タイムスタンプ High[31:0]																															

表 17-25 に示すように、DMA はシステム・メモリからのディスクリプタの 4 DWORDS を常に読み出すかフェッチして、バッファとコントロールの情報を得ます。†

表 17-25. 代替（エンハンスド）フォーマットでの送信ディスクリプタのフェッチ（読み出し）

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	OWN	コントロール [30:26]					TSE	コントロール [24:18]							ステータス用予約 [17:7]										SLOT 数 [6:3]				ステータス用予約 [2:0]			
TDES1	コントロール [31:29]		バッファ 2 バイト・カウンタ [28:16]										RES			バッファ 1 バイト・カウンタ [12:0]																
TDES2	バッファ 1 アドレス [31:0]																															
TDES3	バッファ 2 アドレス [31:0] または次のディスクリプタ・アドレス [31:0]																															

表 17-26. 送信ディスクリプタ・ワード 0（TDES0）（その 1）

ビット	説明
31	OWN : 所有ビット 設定されている場合、このビットは、ディスクリプタが DMA に所有されていることを示します。このビットがリセットされている場合、ディスクリプタがホストに所有されていることを示します。フレーム送信を完了したとき、またはディスクリプタに割り当てられたバッファが完全に読み出されたとき、DMA はこのビットをクリアします。フレームの先頭ディスクリプタの所有権ビットの設定は、同じフレームに属する以降すべてのディスクリプタが設定された後になされる必要があります。こうすることによって、ディスクリプタのフェッチとドライバによる所有権ビットの設定の間で起きうる競合状態を避けることができます。†
30	IC : 完了時割り込み 設定されている場合、このビットは、現在のフレームが転送された後に送信割り込み（レジスタ 5[0]）を設定します。†
29	LS : 終端セグメント 設定されている場合、このビットは、バッファにフレームの終端セグメントが含まれていることを示します。このビットが設定されている場合、TDES1 の TBS1 または TBS2 のフィールドはゼロでない値を持っている必要があります。†
28	FS : 先頭セグメント 設定されている場合、このビットは、バッファにフレームの先頭セグメントが含まれていることを示します。†
27	DC : CRC のディセーブル このビットが設定されている場合、MAC は、送信されたフレームの終端に CRC を追加しません。これは、先頭のセグメント（TDES1[28]）が設定されている場合のみ有効です。†

表 17-26. 送信ディスクリプタ・ワード 0 (TDES0) (その 2)

ビット	説明
26	DP : パッドのディセーブル 設定されている場合、MAC は 64 バイトより短いフレームにはパディングを自動的に追加しません。このビットがリセットされていると、DMA は 64 バイトより短いフレームにパディングと CRC を自動的に追加して、DC (TDES0[27]) ビットのステートにかかわらず CRC フィールドが追加されます。これは、先頭セグメント (TDES0[28]) が設定されている場合のみ有効です。†
25	TTSE : 送信タイムスタンプのイネーブル 設定されている場合、このビットは、ディスクリプタに参照される送信フレーム用の IEEE1588 ハードウェア・タイムスタンプをイネーブルします。このフィールドは、先頭セグメント・コントロール・ビット (TDES1[28]) が設定されている場合のみ有効です。
24	予約
23:22	CIC : チェックサム挿入コントロール これらのビットは、チェックサムの計算および挿入を制御します。以下にビット・エンコーディングを示します。 ■ 0 : チェックサム挿入がディセーブルされます。 ■ 1 : IP ヘッダ・チェックサムのみ計算および挿入がイネーブルされます。 ■ 2 : IP ヘッダ・チェックサムおよびペイロード・チェックサムの計算および挿入はイネーブルされますが、擬似ヘッダ・チェックサムはハードウェアで計算されません。 ■ 3 : IP ヘッダ・チェックサムおよびペイロード・チェックサムの計算および挿入がイネーブルされて、擬似ヘッダ・チェックサムがハードウェアで計算されます。 このフィールドは、先頭セグメント・コントロール・ビット (TDES0[28]) が設定されている場合に有効です。
21	TER : リング終端の送信 設定されている場合、このビットは、ディスクリプタ・リストが最終のディスクリプタに達したことを示します。DMA は、ディスクリプタ・リングを作成してリストのベース・アドレスに戻ります。†
20	TCH : 2 番目のアドレスへのチェイン 設定されている場合、このビットは、ディスクリプタの 2 番目のアドレスが 2 番目のバッファ・アドレスではなく次のディスクリプタのアドレスであることを示します。TDES0[20] が設定されている場合、TBS2 (TDES1[28:16]) は「don't care」値です。 TDES0[21] は TDES0[20] よりも優先されます。†
19:18	予約
17	TTSS : 送信タイムスタンプ・ステータス このフィールドは、表示される送信フレーム用にタイムスタンプがキャプチャされたことを示すステータス・ビットとして使用されます。このビットが設定されている場合、TDES2 および TDES3 は送信フレーム用にキャプチャされたタイムスタンプ値を持つことになります。このフィールドは、終端セグメント・コントロール・ビット (TDES0[29]) が設定されている場合のみ有効です。†
16	IHE : IP ヘッダ・エラー 設定されている場合、このビットは MAC トランスミッタが IP データグラム・ヘッダ内でエラーを検出したことを示します。トランスミッタは、アプリケーションから受信するヘッダ・バイト数に対して IPv4 パケットのヘッダ長をチェックして、ミスマッチがあればエラー・ステータスを示します。IPv6 フレームでは、メイン・ヘッダ長が 40 バイトでなければヘッダ・エラーが通知されます。さらに、IPv4 フレームまたは IPv6 フレーム用のイーサネットの長さ / タイプのフィールド値は、パケットと共に受信する IP ヘッダ・バージョンと一致している必要があります。IPv4 フレームでは、ヘッダ長フィールドが 0x5 未満の値を持っている場合、エラー・ステータスも表示されます。†

表 17-26. 送信ディスクリプタ・ワード 0 (TDES0) (その 3)

ビット	説明
15	ES : エラー要約 以下に各ビットのロジカル OR を示します。 ■ TDES0[14] : Jabber タイムアウト ■ TDES0[13] : フレーム・フラッシュ ■ TDES0[11] : キャリアの喪失 ■ TDES0[10] : キャリアなし ■ TDES0[9] : 遅延衝突 ■ TDES0[8] : 過度の衝突 ■ TDES0[2] : 過度の延期 ■ TDES0[1] : アンダーフロー・エラー ■ TDES0[16] : IP ヘッダ・エラー ■ TDES0[12] : IP ペイロード・エラー †
14	JT : Jabber タイムアウト 設定されている場合、このビットは、MAC トランスミッタで Jabber タイムアウトがあったことを示します。このビットは、レジスタ 0 (MAC コンフィギュレーション・レジスタ) のビット 22 (Jabber のディセーブル) が設定されていない場合のみ設定されます。†
13	FF : フレームのフラッシュ 設定されている場合、このビットは、ソフトウェア・フラッシュ・コマンドが CPU によって与えられたことにより DMA または MTL がフレームをフラッシュしたことを示します。†
12	IPE : IP ペイロード・エラー 設定されている場合、このビットは、MAC トランスミッタが TCP、UDP、または ICMP の IP データグラム・ペイロードでエラーを検出したことを示します。 トランスミッタは、アプリケーションから受信される TCP、UDP、または ICMP の実際のパケット・バイト数に対して、IPv4 または IPv6 のヘッダで受信するペイロード長をチェックして、ミスマッチがあればエラー・ステータスを発行します。†
11	LC : キャリアの喪失 設定されている場合、このビットは、フレームの送信中にキャリアの喪失が起きたことを示します (つまり、フレーム送信中に gmii_crs_i 信号が 1 以上の送信クロック期間で非アクティブになったということです)。これは、MAC が半二重モードで動作している場合に衝突することなくフレームが送信されたときのみ有効です。†
10	NC : キャリアなし 設定されている場合、このビットは、送信中に PHY からのキャリア・センス信号がアサートされなかったことを示します。†
9	予約
8	EC : 過度の衝突 設定されている場合、このビットは、現在のフレームを送信しようと試みて、16 の連続した衝突後に送信がアボートされたことを示します。レジスタ 0 (MAC コンフィギュレーション・レジスタ) のビット 9 (再試行のディセーブル) ビットが設定されている場合、このビットは最初の衝突後に設定されて、フレームの送信がアボートされます。†
7	VF : VLAN フレーム 設定されている場合、このビットは、送信フレームが VLAN タイプのフレームであることを示します。†

表 17-26. 送信ディスクリプタ・ワード 0 (TDES0) (その 4)

ビット	説明
6:3	CC : 衝突回数 (ステータス・フィールド) これらのステータス・ビットは、フレームが送信される前に発生した衝突の回数を示します。この回数は、過度の衝突ビット (TDES0[8]) が設定されている場合は無効です。EMAC は、半二重モードのときのみ、このステータス・フィールドを更新します。
2	ED : 過度の延期 設定されている場合、ビット 4 (延期チェック) がレジスタ 0 (MAC コンフィギュレーション・レジスタ) で High に設定されているとき、このビットは、24,288 ビット・タイム (1000 Mbps モード) または Jumbo フレーム・イネーブル・モードでは 155,680 ビット・タイム) を超える過度の延期のために送信が終了したことを示します。†
1	UF : アンダーフロー・エラー 設定されている場合、このビットは、ホスト・メモリからのフレーム到着が遅かったために、MAC がフレームをアボートしたことを示します。アンダーフロー・エラーは、DMA がフレーム送信中に空の送信バッファを検出したことを示します。送信プロセスは中断ステートになり、送信アンダーフロー (レジスタ 5[5]) および送信割り込み (レジスタ 5[0]) の両方を設定します。†
0	DB : 延期ビット 設定されている場合、このビットは、キャリアの存在のために MAC が送信前に延期することを示します。このビットは、半二重モードのみで有効です。†

表 17-27. 送信ディスクリプタ・ワード 1 (TDES1)

ビット	説明
31:29	予約
28:16	TBS2 : 送信バッファ 2 サイズ このフィールドは、2 番目のデータ・バッファをバイトで示します。このフィールドは、TDES0[20] が設定されているときは無効です。詳しくは、17-26 ページの「バッファ・サイズの計算」を参照してください。†
15:13	予約 †
12:0	TBS1 : 送信バッファ 1 サイズ これらのビットは、最初のデータ・バッファ・バイト・サイズをバイトで示します。このフィールドがゼロの場合、DMA はこのバッファを無視して、TCH (TDES0[20]) の値に応じてバッファ 2 または次のディスクリプタを使用します。†

表 17-28. 送信ディスクリプタ 2 (TDES2)

ビット	説明
31:0	バッファ 1 アドレス・ポインタ これらのビットは、バッファ 1 の物理アドレスを示します。バッファ・アドレス・アラインメントに制約はありません。バッファ・アドレス・アラインメントについて詳しくは、17-26 ページの「ホスト・データ・バッファ・アラインメント」を参照してください。†

表 17-29. 送信ディスクリプタ 3 (TDES3)

ビット	説明
31:0	バッファ 2 アドレス・ポインタ (次のディスクリプタ・アドレス) ディスクリプタのリング構造が使用されている場合は、バッファ 2 の物理アドレスを示します。2 番目のアドレス・チェイン (TDES1[24]) ビットが設定されている場合、このアドレスには次のディスクリプタがある物理メモリへのポインタが含まれています。TDES1[24] が設定されている場合のみ、バッファ・アドレス・ポインタはバス幅にアラインメントされる必要があります (LSB は内部で無視されます)。†

表 17-30. 送信ディスクリプタ 6 (TDES6)

ビット	説明
31:0	TTSL : 送信フレーム・タイムスタンプ Low このフィールドは、関連する送信フレーム用にキャプチャされたタイムスタンプの最下位 32 ビットを使用して DMA によって更新されます。ディスクリプタの終端セグメント・ビット (LS) が設定されていてタイムスタンプ・ステータス (TTSS) ビットが設定されている場合のみ、このフィールドはタイムスタンプを持ちます。†

表 17-31. 送信ディスクリプタ 7 (TDES7)

ビット	説明
31:0	TTSH : 送信フレーム・タイムスタンプ High このフィールドは、関連する送信フレーム用にキャプチャされたタイムスタンプの最上位 32 ビットを使用して DMA によって更新されます。ディスクリプタの終端セグメント・ビット (LS) が設定されていてタイムスタンプ・ステータス (TTSS) ビットが設定されている場合のみ、このフィールドはタイムスタンプを持ちます。†

受信ディスクリプタ

受信ディスクリプタの構造は表 17-32 に示されています。アドバンスド・タイムスタンプまたは IPC フル・オフロードの機能が選択されている場合、これはディスクリプタ・データの 32 バイト（8 DWORDS）を持つことができます。これらの機能のうちどちらか一方がイネーブルされている場合、DMA が拡張ディスクリプタ・サイズで動作するように、ソフトウェアはレジスタ 0（バス・モード・レジスタ）のビット 7 を設定している必要があります。このコントロール・ビットがリセットされている場合、RDES0[0] は常にクリアされていて RDES4 ～ RDES7 のディスクリプタ・スペースは無効です。†

表 17-32. 受信ディスクリプタ・フィールド - 代替（エンハンスド）フォーマット

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	OWN	ステータス [30:0]																														
TDES1	CTRL	RES [30:29]	バッファ 2 バイト・カウント [28:16]													Ctrl [15:14]	RES	バッファ 1 バイト・カウント [12:0]														
TDES2	バッファ 1 アドレス [31:0]																															
TDES3	バッファ 2 アドレス [31:0] または次のディスクリプタ・アドレス [31:0]																															
TDES4	拡張ステータス [31:0]																															
TDES5	予約																															
TDES6	送信タイムスタンプ Low[31:0]																															
TDES7	送信タイムスタンプ High[31:0]																															

RDES0 の内容は表 17-33 で説明されています。RDES1 ～ RDES3 の内容については、それぞれ 17-63 ページの表 17-34、17-64 ページの表 17-35、および 17-64 ページの表 17-36 で説明しています。†

表 17-33. 受信ディスクリプタ・フィールド (RDES0) (その 1)

ビット	説明
31	OWN : 所有ビット 設定されている場合、このビットは、ディスクリプタが EMAC の DMA に所有されていることを示します。このビットがリセットされている場合、ディスクリプタがホストに所有されていることを示します。フレーム受信を完了したとき、またはこのディスクリプタに関連するバッファがフルになったとき、DMA はこのビットをクリアします。
30	AFM : デスティネーション・アドレス・フィルタ失敗 設定されている場合、このビットは、MAC の DA フィルタで失敗したフレームを示します。†
29:16	FL : フレーム長 これらのビットは、ホスト・メモリ (CRC を含む) に対して転送された受信フレームのバイト長を示します。終端ディスクリプタ (RDES0[8]) が設定されていて、ディスクリプタ・エラー・ビット (RDES0[14]) またはオーバーフロー・エラー・ビットのどちらか一方がリセットされている場合、このフィールドは有効です。フレーム長は、IP チェックサム計算 (タイプ 1) がイネーブルされていて受信フレームが MAC コントロール・フレームではないとき、イーサネット・フレームに付加される 2 バイトも表示します。 終端ディスクリプタ (RDES0[8]) が設定されている場合、このフィールドは有効です。終端ディスクリプタ・ビットおよびエラー要約ビットが設定されていない場合、このフィールドは現在のフレーム転送での累積バイト数を示します。†
15	ES : エラー要約 以下に各ビットのロジカル OR を示します。 ■ RDES0[1] : CRC エラー ■ RDES0[3] : 受信エラー ■ RDES0[4] : ウォッチドッグ・タイムアウト ■ RDES0[6] : 遅延衝突 ■ RDES0[7] : ジャイアント・フレーム ■ RDES4[4:3] : IP ヘッダまたはペイロードのエラー ■ RDES0[11] : オーバーフロー・エラー ■ RDES0[14] : ディスクリプタ・エラー 終端ディスクリプタ (RDES0[8]) が設定されている場合のみ、このフィールドが有効です。†
14	DE : ディスクリプタ・エラー 設定されている場合、このビットは、フレームが現在のディスクリプタ・バッファの範囲内にフィットせず DMA が次のディスクリプタを所有していないためにフレームが切り捨てられることを示します。フレームは切り捨てられます。このフィールドは、終端ディスクリプタ (RDES0[8]) が設定されている場合のみ有効です。†
13	SAF : ソース・アドレス・フィルタ失敗 設定されている場合、このビットは MAC の SA フィルタに失敗したフレームの SA フィールドを示します。†
12	LE : 長さエラー 設定されている場合、このビットは、受信フレームの実際の長さおよび長さ / タイプ・フィールドがマッチしないことを示します。このビットは、フレーム・タイプ (RDES0[5]) ビットがリセットされている場合のみ有効です。†

表 17-33. 受信ディスクリプタ・フィールド (RDES0) (その 2)

ビット	説明
11	OE : オーバーフロー・エラー 設定されている場合、このビットは、MTL 内でのバッファ・オーバーフローによって受信フレームがダメージを受けていることを示します。 注：このビットは、DMA がアプリケーションに部分的なフレームを転送する場合のみ設定されます。これは、RX FIFO バッファがスレッシュホールド・モードで動作する場合のみ発生します。ストア・アンド・フォワード・モードでは、部分的なフレームのすべてが RX FIFO バッファでドロップされます。†
10	VLAN : VLAN タグ 設定されている場合、このビットは、このディスクリプタにポイントされているフレームが MAC にタグされた VLAN フレームであることを示します。VLAN タグは、レジスタ 7 (VLAN タグ・レジスタ) の設定に基づいて、受信されるフレームの VLAN フィールドのチェックに応じて異なります。†
9	FS : 先頭のディスクリプタ 設定されている場合、このビットは、このディスクリプタにフレームの先頭バッファが含まれていることを示します。先頭バッファのサイズがゼロの場合、2 番目のバッファにフレームの最初が含まれています。2 番目のバッファのサイズもゼロの場合、次のディスクリプタにフレームの最初が含まれています。†
8	LS : 終端のディスクリプタ 設定されている場合、このビットは、ディスクリプタによってポイントされているバッファがフレームの最終バッファであることを示します。†
7	タイムスタンプ使用可能、IP チェックサム・エラー (タイプ 1)、またはジャイアント・フレーム 設定されている場合、アドバンスド・タイムスタンプ機能があるとき、このビットは、ディスクリプタ・ワード 6 (RDES6) およびワード 7 (RDES7) にタイムスタンプが書き込まれるスナップショットを示します。これは、終端ディスクリプタ・ビット (RDES0[8]) が設定されている場合のみ有効です。 IP チェックサム・エンジン (タイプ 1) が選択されているとき、EMAC に計算された 16 ビットの IPv4 ヘッダ・チェックサムが受信チェックサム・バイトと一致しなかったことを示します。 他の場合は、このビットはジャイアント・フレーム・ステータスを示します。ジャイアント・フレームとは、1,518 バイト (VLAN では 1,522 バイト、あるいは MAC コンフィギュレーション・レジスタのビット 27 (2KPE) が設定されている場合では 2,000 バイト) より大きい通常のフレーム、および Jumbo フレーム処理がイネーブルされている場合の 9,018 バイト (VLAN では 9,022 バイト) より大きいフレームです。
6	LC : 遅延衝突 設定されている場合、このビットは、半二重モードでフレームを受信中に遅延衝突が起きたことを示します。†
5	FT : フレーム・タイプ 設定されている場合、このビットは、受信フレームがイーサネット・タイプのフレームであることを示します (LT フィールドは 0x0600 以上です)。このビットがリセットされている場合、受信フレームは IEEE802.3 フレームです。このビットは、14 バイト未満の小型フレームには無効です。
4	RWT : 受信ウォッチドッグ・タイムアウト 設定されている場合、このビットは、現在のフレームを受信している間に受信ウォッチドッグ・タイマが終了し、ウォッチドッグ・タイムアウト後に現在のフレームが切り捨てられることを示します。†
3	RE : 受信エラー 設定されている場合、このビットは、フレーム受信中に gmii_rxdv_i がアサートされているとき gmii_rxer_i 信号がアサートされることを示します。エラーは、非常に小さいものか拡張のないものである可能性があり、または拡張中のエラー (rxd != 0xf) である可能性もあります。†

表 17-33. 受信ディスクリプタ・フィールド (RDES0) (その 3)

ビット	説明
2	DE : ドリプル・ビット・エラー 設定されている場合、このビットは、受信フレームに非整数の複数のバイトがあることを示します (奇数ニブル)。このビットは MII モードのみで有効です。†
1	CE : CRC エラー 設定されている場合、このビットは、受信フレームで CRC エラーが発生したことを示します。このフィールドは、終端ディスクリプタ (RDES0[8]) が設定されている場合のみ有効です。†
0	拡張ステータス使用可能 /RX MAC アドレス 設定されている場合、アドバンスド・タイムスタンプまたは IP チェックサム・オフロード (タイプ 2) のどちらか一方があるとき、このビットは、ディスクリプタ・ワード 4 (RDES4) で拡張ステータスが使用可能であることを示します。これは、終端ディスクリプタ・ビット (RDES0[8]) が設定されている場合のみ有効です。 アドバンスド・タイムスタンプ機能または IPC フル・オフロードが選択されていない場合、このビット RX MAC アドレス・ステータスを示します。設定されている場合、このビットは、RX MAC アドレス・レジスタ値 (1 ~ 15) はフレームの DA フィールドに一致していることを示します。リセットされている場合、このビットは、RX MAC アドレス・レジスタ 0 の値が DA フィールドに一致していることを示します。†

表 17-34. 受信ディスクリプタ・フィールド 1 (RDES1) (その 1)

ビット	説明
31	DIC : 完了時割り込みのディセーブル 設定されている場合、このビットは、このディスクリプタによって示されているバッファで終端の受信フレームに対してステータス・レジスタの RI ビット (CSR5[6]) が設定されるのを回避します。つまり、そのフレームの RI のために、ホストへの割り込みアサートをディセーブルするということです。†
30:29	予約 †
28:16	RBS2 : 受信バッファ 2 サイズ これらのビットは、2 番目のデータ・バッファ・サイズをバイトで示します。RDES3 (バッファ 2 アドレス・ポインタ) の値がバス幅にアラインメントされていない場合でも、このバッファ・サイズは 4 の倍数である必要があります。バッファ・サイズが 4 の倍数でない場合、予期せぬ動作を引き起こしてしまいます。RDES1[14] が設定されている場合、このフィールドは無効です。バッファ・サイズの計算について詳しくは、17-26 ページの「バッファ・サイズの計算」を参照してください。
15	RER : リング終端の受信 設定している場合、このビットは、ディスクリプタ・リストが終端ディスクリプタに達したことを示します。DMA は、リストのベース・アドレスに戻り、ディスクリプタ・リングを作成します。†
14	RCH : 2 番目のアドレスへのチェイン 設定されている場合、このビットは、ディスクリプタの 2 番目のアドレスは 2 番目のバッファ・アドレスではなく次のディスクリプタ・アドレスであることを示します。このビットが設定されている場合、RBS2 (RDES1[28:16]) は「don't care」値です。RDES1[15] は RDES1[14] よりも優先されません。†

表 17-34. 受信ディスクリプタ・フィールド 1 (RDES1) (その 2)

ビット	説明
13	予約 ↑
12:0	RBS1：受信バッファ 1 サイズ 先頭のデータ・バッファ・サイズをバイトで示します。RDES2（バッファ 1 アドレス・ポインタ）の値がアラインメントされていない場合でも、バッファ・サイズは 4 の倍数である必要があります。バッファ・サイズが 4 の倍数ではない場合、予期せぬ動作を引き起こしてしまいます。このフィールドがゼロの場合、DMA はこのバッファを無視して、RCH（ビット 14）の値に応じてバッファ 2 または次のディスクリプタを使用します。バッファ・サイズの計算について詳しくは、17-26 ページの「バッファ・サイズの計算」を参照してください。

表 17-35. 受信ディスクリプタ・フィールド 2 (RDES2)

ビット	説明
31:0	バッファ 1 アドレス・ポインタ これらのビットは、バッファ 1 の物理アドレスを示します。次の条件を除いて、バッファ・アドレス・アラインメントに制約はありません。RDES2 値を使用してフレーム開始を格納する場合、DMA はアドレス生成のために RDES2[1:0] でプログラムされた値を使用します。フレーム開始の転送中、DMA は RDES2[1:0] ビットをゼロとして書き込み動作を実行しますが、フレーム・データは実際のバッファ・アドレス・ポインタに従ってシフトされます。アドレス・ポインタがフレームの中間または終端部分に格納されているバッファに対するものである場合、DMA は RDES2[1:0] を無視します。バッファ・アドレス・アラインメントについて詳しくは、17-26 ページの「ホスト・データ・バッファ・アラインメント」を参照してください。

表 17-36. 受信ディスクリプタ・フィールド 3 (RDES3)

ビット	説明
31:0	バッファ 2 アドレス・ポインタ（次のディスクリプタ・アドレス） ディスクリプタのリング構造が使用されている場合は、バッファ 2 の物理アドレスを示します。2 番目のアドレス・チェイン（RDES1[24]）ビットが設定されている場合、このアドレスには次のディスクリプタがある物理メモリへのポインタが含まれています。 RDES1[24] が設定されている場合、バッファ（次のディスクリプタ）・アドレス・ポインタはバス幅にアラインメントされる必要があります（RDES3[1:0] = 0、LSB は内部で無視されます）。しかし、RDES1[24] がリセットされている場合、以下の条件を除いて RDES3 値に制約はありません。RDES3 値を使用してフレーム開始を格納する場合、DMA はアドレス生成のために RDES3[1:0] でプログラムされた値を使用します。アドレス・ポインタがフレームの中間または終端部分に格納されているバッファに対するものである場合、DMA は RDES3[1:0] を無視します。

書き込まれる拡張ステータスは、表 17-37 に示されています。拡張ステータスは、使用可能な IPC またはタイムスタンプに関連したステータスがある場合のみ書き込まれます。拡張ステータスの可用性は RDES0 のビット 1 で示されます。このステータスは、アドバンスド・タイムスタンプまたは IPC フル・オフロードの機能が選択されている場合のみ使用可能です。†

表 17-37. 受信ディスクリプタ・フィールド 4 (RDES4) (その 1)

ビット	説明
31:28	予約 †
27:26	レイヤ 3 およびレイヤ 4 のフィルタ数の一致 これらのビットは、受信フレームに一致するレイヤ 3 およびレイヤ 4 のフィルタ数を示します。 ■ 00 : フィルタ 0 ■ 01 : フィルタ 1 ■ 10 : フィルタ 2 ■ 11 : フィルタ 3 このフィールドは、ビット 24 およびビット 25 が High に設定されている場合のみ有効です。2 つ以上のフィルタが一致している場合、これらのビットは最低のフィルタ数のみ示します。†
25	レイヤ 4 フィルタの一致 設定されている場合、このビットは、受信フレームがレイヤ 4 ポート数フィールドの 1 つと一致していることを示します。このステータスは、以下のうち 1 つの条件のとき与えられます。 ■ レイヤ 3 フィールドはイネーブルされておらず、イネーブルされているレイヤ 4 フィールドのすべてが一致している場合。 ■ イネーブルされているレイヤ 3 とレイヤ 4 のすべてのフィルタ・フィールドが一致している場合。 2 つ以上のフィルタが一致している場合、このビットはビット [27:26] で示されるフィルタのレイヤ 4 フィルタ・ステータスを示します。†
24	レイヤ 3 フィルタの一致 設定されている場合、このビットは、受信フレームが、イネーブルされているレイヤ 3 の IP アドレス・フィールドの 1 つと一致していることを示します。 このステータスは、以下のうち 1 つの条件のとき与えられます。 ■ イネーブルされているレイヤ 3 のすべてのフィールドが一致しており、イネーブルされているレイヤ 4 のすべてのフィールドがバイパスされる場合。 ■ イネーブルされているすべてのフィルタ・フィールドが一致している場合。 2 つ以上のフィルタが一致している場合、このビットはビット [27:26] で示されるフィルタのレイヤ 3 フィルタ・ステータスを示します。†
23:15	予約
14	タイムスタンプのドロップ 設定されている場合、このビットは、タイムスタンプがこのフレーム用にキャプチャされましたがオーバーフローのために MTL RX FIFO バッファでドロップされたことを示します。
13	PTP バージョン 設定されている場合、このビットは受信 PTP メッセージが IEEE 1588 バージョン 2 のフォーマットを持っていることを示します。リセットされている場合、メッセージはバージョン 1 のフォーマットを持っていることになります。

表 17-37. 受信ディスクリプタ・フィールド 4 (RDES4) (その 2)

ビット	説明
12	PTP フレーム・タイプ 設定されている場合、このビットは、PTP メッセージがイーサネットを介して直接送信されたことを示します。このビットが設定されておらずメッセージ・タイプがゼロでない場合、PTP メッセージは UDP-IPv4 または UDP-IPv6 を介して送信されることを示します。IPv4 または IPv6 の情報は、ビット 6 およびビット 7 から得られます。
11:8	メッセージ・タイプ これらのビットは、受信メッセージのタイプを与えるためにエンコードされています。 ■ 0000 : 受信 PTP メッセージなし ■ 0001 : SYNC (すべてのクロック・タイプ) ■ 0010 : Follow_Up (すべてのクロック・タイプ) ■ 0011 : Delay_Req (すべてのクロック・タイプ) ■ 0100 : Delay_Resp (すべてのクロック・タイプ) ■ 0101 : Pdelay_Req (ピア・ツー・ピアの透過クロック) ■ 0110 : Pdelay_Resp (ピア・ツー・ピアの透過クロック) ■ 0111 : Pdelay_Resp_Follow_Up (ピア・ツー・ピアの透過クロック) ■ 1000 : アナウンス ■ 1001 : マネージメント ■ 1010 : シグナリング ■ 1011-1110 : 予約 ■ 1111 : 予約メッセージ・タイプの PTP パケット
7	IPv6 パケットの受信 設定されている場合、このビットは、受信したパケットが IPv6 パケットであることを示します。このビットは、レジスタ 0 (MAC コンフィギュレーション・レジスタ) のビット 10 (IPC) が設定されている場合のみ更新されます。
6	IPv4 パケットの受信 設定されている場合、このビットは、受信したパケットが IPv4 パケットであることを示します。このビットは、レジスタ 0 (MAC コンフィギュレーション・レジスタ) のビット 10 (IPC) が設定されている場合のみ更新されます。
5	IP チェックサムのバイパス 設定されている場合、このビットは、チェックサム・オフロード・エンジンがバイパスされていることを示します。
4	IP ペイロード・エラー 設定されている場合、このビットは、EMAC が計算した 16 ビットの IP ペイロード・チェックサム (つまり、TCP、UDP、または ICMP のチェックサム) が、受信セグメント内の関連するチェックサム・フィールドに一致していないことを示します。TCP、UDP、ICMP のセグメント長が IP ヘッダ・フィールドのペイロード長の値に一致していない場合も設定されます。このビットは、ビット 7 またはビット 6 のどちらか一方が設定されている場合に有効です。

表 17-37. 受信ディスクリプタ・フィールド 4 (RDES4) (その 3)

ビット	説明
3	IP ヘッダ・エラー 設定されている場合、このビットは、EMAC によって計算された 16 ビットの IPv4 ヘッダ・チェックサム受信チェックサム・バイトと一致しないこと、または IP データグラム・バージョンがイーサネット・タイプの値と一致していないことを示します。このビットは、ビット 7 またはビット 6 のどちらか一方が設定されている場合に有効です。
2:0	IP ペイロード・タイプ これらのビットは、受信チェックサム・オフロード・エンジン (COE) によって処理された IP データグラムにカプセル化されているペイロードのタイプを示します。また、COE は、IP ヘッダ・エラーまたはフラグメント化した IP のために IP データグラムのペイロードが処理されない場合、これらのビットをゼロに設定します。 ■ 0 : Unknown または IP ペイロード未処理 ■ 1 : UDP ■ 2 : TCP ■ 3 : ICMP ■ 4-7 : 予約 このビットは、ビット 7 またはビット 6 のどちらか一方が設定されていない場合に有効です。

RDES6 および RDES7 には、タイムスタンプのスナップショットがあります。タイムスタンプのスナップショットの可用性は、RDES0 ディスクリプタのビット 7 によって表示されます。RDES6 および RDES7 の内容は、それぞれ表 17-38 および 17-67 ページの表 17-39 に示す通りです。†

表 17-38. 受信ディスクリプタ・フィールド 6 (RDES6)

ビット	説明
31:0	RTSL : 受信フレーム・タイムスタンプ Low このフィールドは、関連する受信フレーム用にキャプチャされたタイムスタンプの最下位 32 ビットを使用して DMA によって更新されます。DMA は、終端ディスクリプタ・ステータス・ビット (RDES0[8]) によって示される受信フレームの終端ディスクリプタのみでこのフィールドを更新します。†

表 17-39. 受信ディスクリプタ・フィールド 7 (RDES7)

ビット	説明
31:0	RTSH : 受信フレーム・タイムスタンプ High このフィールドは、関連する受信フレーム用にキャプチャされたタイムスタンプの最上位 32 ビットを使用して DMA によって更新されます。DMA は、終端ディスクリプタ・ステータス・ビット (RDES0[8]) によって示される受信フレームの終端ディスクリプタのみでこのフィールドを更新します。†

DMA の初期化

この項では、適切なシーケンスで DMA/MAC レジスタを初期化する方法を説明します。DMA を初期化するには、以下のステップを実行します。

1. ソフトウェアをリセットします。これは、EMAC の内部レジスタとロジックのすべてをリセットすることです（DMA レジスタ 0（バス・モード・レジスタ）– ビット 0）。†
2. リセット・プロセスの完了を待ちます（DMA レジスタ 0（バス・モード・レジスタ）のビット 0 をポーリングするのを待ちます）。リセット動作が完了するとクリアされます。†
3. レジスタ 11（AHB または AXI のステータス）のビットをポーリングして、前回開始したトランザクションまたは処理中のトランザクションがすべて完了していることを確認します。



ソフトウェアのリセット後に（パフォーマンスの理由で）アプリケーションがレジスタをポーリングできない場合、DMA 動作をトリガする前に、次のステップに進んで（ステップ 12 で説明しているように）このレジスタをチェックすることを推奨します。†

4. 以下のフィールドをプログラムして、DMA レジスタ 0（バス・モード・レジスタ）に値を設定することによってバス・モード・レジスタを初期化します。†
 - a. 混合バーストおよび AAL
 - b. 混合バーストまたは不定バースト †
 - c. バースト長の値およびバースト・モードの値 †
 - d. ディスクリプタ長（リング・モードが使用されている場合のみ有効）†
 - e. TX および RX の DMA アービトレーション手法 †

5. レジスタ 10 (AXI バス・モード・レジスタ) のインタフェース・オプションをプログラムします。固定バースト長がイネーブルされている場合、バスで可能な最大のバースト長を選択します (ビット [7:1])。†
6. 送受信に適切なディスクリプタ・チェーンを作成します。また、受信ディスクリプタが DMA に所有されていることを確認します (ディスクリプタのビット 31 が設定されている必要があります)。OSF モードが使用されている場合、少なくとも 2 つのディスクリプタが必要です。ディスクリプタについて詳しくは、[17-39 ページの「ノーマル・ディスクリプタ」](#)および [17-53 ページの「代替またはエンハンスド・ディスクリプタ」](#)を参照してください。†
7. ソフトウェアがディスクリプタを再使用する前に、チェーン内で 3 つ以上の異なる送信または受信ディスクリプタを作成することを確認します。†
8. 送受信ディスクリプタのベース・アドレス (それぞれレジスタ 3 (受信ディスクリプタ・リスト・アドレス・レジスタ) およびレジスタ 4 (送信ディスクリプタ・リスト・アドレス・レジスタ)) を使用して、送受信ディスクリプタ・リスト・アドレスを初期化します。†
9. 以下のフィールドをプログラムして、レジスタ 6 (動作モード・レジスタ) の動作モードを初期化します。
 - a. 送受信ストア・アンド・フォワード †
 - b. 送受信スレッシュールド・コントロール (RTC および TTC) †
 - c. ハードウェア・フロー・コントロールのイネーブル †
 - d. MTL 送受信 FIFO バッファ用のフロー・コントロールのアクティベーションおよびディアクティベーション・スレッシュールド (RFA および RFD) †
 - e. エラー・フレームおよび小規模フレームの転送のイネーブル †
 - f. OSF モード †
10. 設定されているステータス・レジスタのビット (割り込みビットのみ) に書き込むことで、割り込みリクエストをクリアします。例えば、ビット 16 に 1 を書き込むことで、通常の割り込み要約はこのビット (DMA レジスタ 5 (ステータス・レジスタ)) をクリアします。†
11. レジスタ 7 (割り込みイネーブル・レジスタ) をプログラムすることで割り込みをイネーブルします。†

 ステップ 3 を実行していない場合のみステップ 12 を実行します。†

12. レジスタ 11 (AHB または AXI ステータス) を読み出して、前回のトランザクションがすべて完了していることを確認します。†

 レジスタ 11 (AHB または AXI ステータス) を読み出すときに前回のトランザクションがまだ処理中の場合は、マスタ・インタフェースによって処理されるスレーブ・コンポーネントをチェックすることが推奨されます。

13. コントロール・レジスタ (DMA レジスタ 6 (動作モード・レジスタ)) の SR (ビット 1) および ST (ビット 13) を設定することにより、送受信 DMA を開始します。†

MAC の初期化

以下の MAC 初期化の動作は、DMA 初期化の後に実行できます。DMA を設定する前に MAC を初期化すると、DMA がアクティブになった後に MAC レシーバのみ (以下の最後のステップ) がイネーブルされます。その他の場合、受信フレームは RX FIFO バッファを埋めてオーバーフローになります。†

1. EMAC レジスタ 4 (GMII アドレス・レジスタ) をプログラムして、外部 PHY のマネージメント・サイクルを制御します。例えば、物理層アドレス PA (ビット 15 ~ 11) です。また、PHY に書き込んで PHY から読み出すために、ビット 0 (GMII ビジー) を設定します。†
2. レジスタ 4 (GMII アドレス・レジスタ) のビット 15 ~ 11 で適切なアドレス値を指定することで、リンク・アップ、動作速度、および動作モード用の PHY からのレジスタ 5 (GMII データ・レジスタ) の 16 ビット・データを読み出します。†
3. MAC にアドレス・レジスタ (レジスタ 16 (MAC アドレス 0 High レジスタ) およびレジスタ 17 (MAC アドレス 0 Low レジスタ)) を提供します。MAC の 128 のアドレスがサポートされているため、その MAC アドレスをプログラムする必要があります。
4. レジスタ 2 (ハッシュ・テーブル High レジスタ) およびレジスタ 3 (ハッシュ・テーブル Low レジスタ) をプログラムします。
5. 以下のフィールドをプログラムして、受信するフレーム用にレジスタ 1 (MAC フレーム・レジスタ) で適切なフィルタを設定します。†
 - a. 全受信 †
 - b. プロミスキャス・モード †
 - c. ハッシュまたは保護フィルタ †
 - d. ユニキャスト、マルチキャスト、ブロードキャスト、およびコントロール・フレーム・フィルタ設定 †
6. 以下のフィールドをプログラムして、レジスタ 6 (フロー・コントロール・レジスタ) で適切なフロー・コントロールを設定します。†
 - a. ポーズ・タイムおよび他のポーズ・フレーム・コントロール・ビット †
 - b. 送受信フロー・コントロール・ビット †
 - c. フロー・コントロール・ビジー / バックプレッシャのアクティベート †

7. 適用可能な場合は、必要に応じて、コンフィギュレーション用に割り込みマスク・レジスタ・ビットをプログラムします。†
8. レジスタ 0 (MAC コンフィギュレーション・レジスタ) で適切なフィールドをプログラムします。例えば、送信と Jabber ディセーブル中のフレーム間ギャップです。自動ネゴシエーションに基づいて、二重モード (ビット 11) またはポート選択 (ビット 15) を設定できます。†
9. レジスタ 0 (MAC コンフィギュレーション・レジスタ) でビット 3 (TE) およびビット 2 (RE) を設定します。†



EMAC DMA がアクティブに送受信しているときは、コンフィギュレーション (二重モード、速度、ポート、またはループバックなど) を変更しないでください。ソフトウェアがこれらのパラメータを変更できるのは、EMAC DMA のトランスミッタおよびレシーバがアクティブでないときのみです。

通常を送受信動作の実行

通常の動作では、以下のステップを実行します。†

1. 通常を送受信割り込み用に、割り込みステータスを読み出します。そしてディスクリプタをポーリングし、ホストに所有されているディスクリプタ (送信または受信のどちらか一方) のステータスを読み出します。†
2. ディスクリプタに適切な値を設定し、送受信ディスクリプタが DMA に所有されていてデータの送受信を再開することを確認します。†
3. ディスクリプタが DMA に所有されていない場合 (またはディスクリプタが使用可能ではない場合)、DMA は中断ステートになります。TX/RX ポーリング要求レジスタ (レジスタ 1 (送信ポーリング要求レジスタ) およびレジスタ 2 (受信ポーリング要求レジスタ)) にゼロを書き込むことによってディスクリプタをフリーにしポーリング要求を発行して、送信または受信を再開できます。†
4. ホストの現在のトランスミッタ/レシーバ・ディスクリプタのアドレス・ポインタの値は、デバッグ・プロセス用に読み出せます (レジスタ 18 (現在のホストの送信ディスクリプタ・レジスタ) およびレジスタ 19 (現在のホストの受信ディスクリプタ・レジスタ))。†
5. ホストの現在の送信バッファ・アドレス・ポインタおよび受信バッファ・アドレス・ポインタの値は、デバッグプロセス用に読み出せます (レジスタ 20 (現在のホストの送信バッファ・アドレス・レジスタ) およびレジスタ 21 (現在のホストの受信バッファ・アドレス・レジスタ))。†

送信の停止と開始

送信を停止させるときには、以下のステップを実行します。†

1. レジスタ 6（動作モード・レジスタ）のビット 13（送信コマンドの開始または停止）をクリアすることによって、送信 DMA をディセーブルします（適用可能な場合）。†
2. それまでのフレーム送信が完了するのを待ちます。レジスタ 9（デバッグ・レジスタ）の適切なビットを読み出すことによってこのことをチェックできます。†
3. レジスタ 0（MAC コンフィギュレーション・レジスタ）のビット 3（TE）およびビット 2（RE）をクリアすることで、MAC トランスミッタおよび MAC レシーバをディセーブルします。†
4. （レジスタ 9（デバッグ・レジスタ）を読み出すことによって）RX FIFO バッファのデータがシステム・メモリに転送されていることを確認してから、受信 DMA をディセーブルします（適用可能な場合）。†
5. TX FIFO バッファおよび RX FIFO バッファの両方が空になっていることを確認します。†
6. 動作を再開させるには、まず DMA を開始して、それから MAC のトランスミッタおよびレシーバをイネーブルします。†

EEE（Energy Efficient Ethernet）のプログラミング・ガイドライン

TX LPI モードの開始と終了

EMAC では EEE（Energy Efficient Ethernet）機能が使用可能です。これを使用するには、EMAC の初期化中に以下のステップを実行します。

1. MDIO インタフェースを介して PHY レジスタを読み出して、リモート・エンドが EEE 機能を備えているか確認し、タイマ値のネゴシエーションを行います。†
2. MDIO インタフェースを介して PHY レジスタをプログラムします（LPI モードで RX クロックを停止させるかどうか表示する `RX_CLK_stoppable` ビットを含む）。†
3. レジスタ 13（LPI タイマ・コントロール・レジスタ）でビット [16:5] の LST、およびビット [15:0] の TWT をプログラムします。†
4. MDIO インタフェースを使用して PHY チップのリンク・ステータスを読み出して、レジスタ 12（LPI コントロールおよびステータス・レジスタ）のビット 17（PLS）を更新します。PHY チップのリンク・ステータスが変更されるときはいつでも、この更新が必要です。†
5. レジスタ 12（LPI コントロールおよびステータス・レジスタ）のビット 16（LPIEN）を設定して、MAC を LPI ステータスにします。処理中の送信が完了すると、MAC は LPI モードになってビット 0（TLPIEN）を設定します。†



TX FIFO バッファでキューに保留中のフレームがすべて送信完了した後のみ MAC を LPI ステートにするには、レジスタ 12（LPI コントロールおよびステータス・レジスタ）のビット 19（LPITXA）を設定する必要があります。†



LPI ステート中に送信クロックをオフにするには、クロック入力をゲートするために `sbd_tx_clk_gating_ctrl_o` を使用します。†

 TLPI ステート中に CSR クロックをオフにする、またはシステムの残りの部分を電源オフにするには、レジスタ 12 (LPI コントロールおよびステータス・レジスタ) の TLPIEN 割り込みが生成されるまで待ちます。LPI ステートから抜け出すには、ステップ 6 を実行する前にクロックを復元させます。†

6. レジスタ 12 (LPI コントロールおよびステータス・レジスタ) のビット 16 (LPIEN) をリセットして、MAC を LPI ステートから抜け出させます。†

TLPIEX 割り込みステータス・ビットを設定して送信を再開する前に、MAC は、ビット [15:0] の TWT でプログラムされたタイムを待ちます。†

LPI モードでの CSR クロックのゲート・オフ

MAC が低電力アイドル (LPI) モードのとき、CSR クロックをゲート・オフして電力を節約できます。†

RX LPI モードでの CSR クロックのゲート・オフ

MAC が PHY から LPI パターンを受信するとき、以下の動作を実行します。†

1. MAC RX は LPI モードになって、RX LPI エントリ割り込みステータス [レジスタ 12 の RLPIEN 割り込み (LPI_Control_Status)] が設定されます。†
2. 割り込みピン (sbd_intr_o) がアサートされます。ホストがレジスタ 12 (LPI_Control_Status) を読み出すと、sbd_intr_o 割り込みはクリアされます。†

sbd_intr_o 割り込みがアサートされて MAC TX も LPI モードになると、CSR クロックをゲート・オフできます。CSR クロックをゲート・オフするときに MAC TX が LPI モードでない場合、MAC トランスミッタでのイベントは CSR で通知または更新されません。†

CSR クロックを復元するには、MAC が lpi_intr_o で LPI 終了割り込みを (clk_rx_i に同期させて) アサートした後に、LPI モード終了の表示が PHY から届くまで待機します。レジスタ 12 が読み出されると lpi_intr_o 割り込みがクリアされます。†

TX LPI モードでの CSR クロックのゲート・オフ

レジスタ 12 (LPI コントロールおよびステータス・レジスタ) のビット 16 (LPIEN) が設定されている場合、以下の動作を実行します。†

1. 送信 LPI エントリ割り込み (レジスタ 12 の TLPIEN ビット) が設定されます。†
2. 割り込みピン (sbd_intr_o) がアサートされます。ホストがレジスタ 12 を読み出すと、sbd_intr_o 割り込みがクリアされます。†

sbd_intr_o 割り込みがアサートされて MAC RX も LPI モードになると、CSR クロックをゲート・オフできます。CSR クロックをゲート・オフするときに MAC RX が LPI モードでない場合、MAC レシーバでのイベントは CSR で通知または更新されません。†

CSR クロックを復元するには、MAC が TX LPI モードから抜け出すときに CSR クロックをオンにします。†

CSR クロックが再開した後、レジスタ 12 (LPI コントロールおよびステータス・レジスタ) のビット 16 (LPIEN) をリセットして、MAC を LPI モードから抜け出させます。†

フレキシブル PPS (Pulse-Per-Second) 出力のプログラミング・ガイドライン

PPS でのシングル・パルスの生成

PPS でシングル・パルスを生成するには、以下の動作を実行します。†

1. レジスタ 459 (PPS コントロール・レジスタ) のビット [6:5] である TRGTMODSEL の 11 および 10 (割り込み用) をプログラムします。これにより、MAC は PPS 信号出力の開始時間にターゲット・タイム・レジスタ (レジスタ 455 および 456) を使用するようになります。†
2. ターゲット・タイム・レジスタ (レジスタ 455 および 456) の開始時間の値をプログラムします。†
3. レジスタ 473 (PPS0 幅レジスタ) で PPS 信号出力の幅をプログラムします。†
4. レジスタ 459 (PPS コントロール・レジスタ) のビット [3:0] の PPSCMD を 0001 にプログラムします。これにより、MAC は、ターゲット・タイム・レジスタ (レジスタ 455 および 456) でプログラムした時間に PPS 信号出力で信号パルスを生成するようになります。†

PPSCMD が実行されると (PPSCMD ビット = 0)、プログラムされた開始時間が経過する前にキャンセル開始コマンド (PPSCMD=0011) を与えることでパルス生成をキャンセルできます。また、前もって次のパルスの動作をプログラムすることも可能です。次のパルスをプログラムするには、次の動作を実行します。†

1. ターゲット・タイム・レジスタ (レジスタ 455 および 456) で次のパルスの開始時間をプログラムします。この時間は、前のパルスの立ち下がりエッジの時間よりも長い必要があります。†
2. 次の PPS 信号出力の幅をレジスタ 473 (PPS0 幅レジスタ) でプログラムします。†
3. レジスタ 459 (PPS コントロール・レジスタ) のビット [3:0] の PPSCMD をプログラムして、前のパルスがデアサートされた後にシングル・パルスを生成します。これにより、MAC は、ターゲット・タイム・レジスタでプログラムされている時間に、PPS 信号出力でシングル・パルスを生成するようになります。前のパルスが Low になる前にこのコマンドを発行すると、新しいコマンドが前のコマンドを上書きして、EMAC は 1 つの拡張パルスのみ生成する可能性があります。

PPS でのパルス・トレーニングの生成

PPS でパルス・トレーニングを生成するには、以下の動作を実行します。†

1. レジスタ 459 (PPS コントロール・レジスタ) のビット [6:5] である TRGTMODSEL の 11 または 10 (割り込み用) をプログラムします。これにより、MAC は PPS 信号出力の開始時間にターゲット・タイム・レジスタ (レジスタ 455 および 456) を使用するようになります。†
2. ターゲット・タイム・レジスタ (レジスタ 455 および 456) で開始時間の値をプログラムします。†
3. レジスタ 473 (PPS0 幅レジスタ) で PPS 信号出力でのパルス・トレーニングの間隔値をプログラムします。†
4. レジスタ 473 (PPS0 幅レジスタ) で PPS 信号出力の幅をプログラムします。†
5. レジスタ 459 (PPS コントロール・レジスタ) のビット [3:0] の PPSCMD を 0010 にプログラムします。これにより、MAC はターゲット・タイム・レジスタ (レジスタ 455 および 456) でプログラムされた開始時間に PPS 信号出力でパルス・トレーニングを生成するようになります。デフォルトでは、PPS パルス・トレーニングは、「STOP Pulse train at time」コマンドまたは「STOP Pulse Train immediately」で停止しない限り、フリー・ランニングです。†
6. ターゲット・タイム・レジスタ (レジスタ 455 および 456) で停止の値をプログラムします。ターゲット・タイム・レジスタ (レジスタ 455 および 456) を再びプログラミングする前に、レジスタ 456 (ターゲット・タイム・ナノ秒レジスタ) のビット 31 (TSTRBUSY) がリセットされていることを確認します。†
7. レジスタ 459 (PPS コントロール・レジスタ) の PPSCMD フィールド (ビット 3:0) を 0100 にプログラムします。これにより、ステップ 6 でプログラムされている特定の停止時間の経過後、PPS 信号出力のパルス・トレーニングが停止します。†

PPSCMD フィールドを 0101 にプログラミングすることで、いつでもパルス・トレーニングを停止できます。同様に、停止時間 (ステップ 6 でプログラミングした時間) の経過前に PPSCMD フィールドを 0110 にプログラミングすることで、パルス・トレーニングの停止コマンド (ステップ 7 でのコマンド) をキャンセルできます。プログラムされている開始時間 (ステップ 2) の経過前に PPSCMD フィールドを 0011 にプログラミングすることで、パルス・トレーニングの生成をキャンセルできます。†

PPS への影響なしでの割り込みの生成

レジスタ 459 (PPS コントロール・レジスタ) のビット [6:5] の TRGTMODSEL によって、以下のどれか 1 つを実行するようにターゲット・タイム・レジスタをプログラムできます。†

- 割り込みのみを生成する。†
- 割り込みと PPS の開始時間および停止時間を生成する。†
- PPS の開始時間および停止時間を生成する。†

ターゲット・タイム・レジスタ (レジスタ 455 および 456) をプログラムして、割り込みイベントのみを生成します。†

1. レジスタ 459 (PPS コントロール・レジスタ) のビット [6:5] である TRGTMODSEL の 00 (割り込み用) をプログラムします。これにより、MAC はターゲット・タイム・レジスタ (レジスタ 455 および 456) をターゲット・タイム割り込みに使用するようになります。†
2. ターゲット・タイム・レジスタ (レジスタ 455 および 456) でターゲット・タイムの値をプログラムします。これにより、ターゲット・タイムを経過するときに MAC は割り込みを生成するようになります。ビット [6:5] の TRGTMODSEL が変更される場合 (例えば PPS を変更するために)、割り込み生成は新しいモードと新たにプログラムされたターゲット・タイム・レジスタ値で上書きされます。

イーサネット MAC のアドレス・マップおよびレジスタの定義

 アドレス・マップおよびレジスタの定義は、このハンドブックの volume に付属の [hps.html](#) ファイルにあります。ファイルを開くにはリンクをクリックします。

モジュールの説明およびベース・アドレスを見るには、スクロールして以下のモジュール・インスタンスのリンクをクリックします。

■ **emac0**

■ **emac1**

そしてレジスタおよびフィールドの説明を見るには、スクロールしてレジスタ名をクリックします。レジスタのアドレスは、各モジュール・インスタンスのベース・アドレスに相対的なオフセットです。

 すべてのモジュールのベース・アドレスは、*Cyclone V* デバイス・ハンドブック Volume 3 の *Introduction to the Hard Processor System* の章にも示されています。

改訂履歴

表 17-40 に、本資料の改訂履歴を示します。

表 17-40. 改訂履歴

日付	バージョン	変更内容
2012 年 11 月	1.2	マイナーな更新。
2012 年 5 月	1.1	プログラミング・モードの項の追加。
2012 年 1 月	1.0	初版。