



本社: 〒184-0012  
 東京都小金井市中町 3-23-17  
 電話/FAX: 050-3588-7915  
 E-mail: sales@dgway.com  
 URL: [www.dgway.com](http://www.dgway.com)

## 特長

- 純ハードワイヤードの TCP/IP プロトコル・スタック
- IPv4 に対応
- 単一セッション接続
- サーバーおよびクライアントの両モード  
(パッシブ/アクティブのオープン/クローズに対応)
- ジャンボ・フレームに対応
- 送信パケット・サイズは送信データ・バスが 64 ビット幅のため 64 ビットの倍数で設定
- 受信データ・バスは 64 ビット幅のため総受信サイズは 64 ビットの倍数
- 送信/受信バッファ・サイズを消費リソースとパフォーマンスから最適点に調整可能
- データ・インターフェイスはシンプルかつ一般的な FIFO インターフェイス
- 単一ポート RAM インターフェイス形式のシンプルな制御インターフェイス
- 10G/25G イーサネット MAC へのインターフェイスとして 64 ビット幅の AXI4 ストリーム
- 156.25MHz 固定の単一クロック・ドメイン
- KC705/VC707/ZC706/KCU105/ZCU102/ZCU106/VCU118 による実機動作リファレンス・デザインを用意
- カスタマイズにより以下の機能は追加実装が可能
  - ・ 64 ビットの倍数に合わない場合のデータ転送
  - ・ ウィンドウ・スケール機能によるバッファ・サイズの拡張
  - ・ 他の方法によるネットワーク・パラメータの設定

コア概要	
コア納品同梱物	
技術資料	データシート、リファレンスデザイン資料
コア納品形態	暗号化した VHDL
制約ファイル	リファレンス・デザインで提供
検証環境	実機動作リファレンス・デザイン
インスタンス化サンプル	VHDL で提供
リファレンス・デザイン	VIVADO プロジェクト コア以外は全てソースコード提供
その他	VCU118, ZCU106, ZCU102, KCU105, KC705, VC707, ZC706 での実機デモ環境
技術サポート	
デザイン・ゲートウェイ・ジャパンによる日本語サポート	

表 1: 7 シリーズ・ファミリ コンパイル結果例

ファミリ	ターゲット・デバイス	Fmax (MHz)	Slice Regs	Slice LUTs	Slices <sup>1</sup>	BRAMTile <sup>2</sup>	Design Tools
Kintex-7	XC7K325TFFG900-2	156.25	3101	3827	1326	36	Vivado2017.4
Zynq-7000	XC7Z045FFG900-2	156.25	3101	3827	1361	36	Vivado2017.4
Virtex-7	XC7VX485TFFG1761-2	156.25	3101	3828	1347	36	Vivado2017.4

表 2: Ultrascale シリーズ・ファミリ コンパイル結果例

ファミリ	ターゲット・デバイス	Fmax (MHz)	CLB Regs	CLB LUTs	CLB <sup>1</sup>	BRAMTile <sup>2</sup>	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	156.25	3106	3808	755	34.5	Vivado2017.4
Zynq-Ultrascale+	XGZU9EG-FFVB1156-2-I	156.25	3106	3806	736	34.5	Vivado2017.4
Virtex-Ultrascale+	XCU9P-FLGA2104-2L	156.25	3106	3807	704	34.5	Vivado2017.4

備考:

1. 実際のリソース消費カウントはユーザロジックやフィット条件等に依存します。
2. ブロックメモリの消費リソース数は送信データ・バッファ 64K バイト、送信パケットバッファ 16K バイト、受信データバッファ 64K バイト時の値となります。バッファ容量の最小設定はそれぞれ 4K バイト、4K バイト、16K バイト(ジャンボフレーム対応のため)です。

## アプリケーション情報

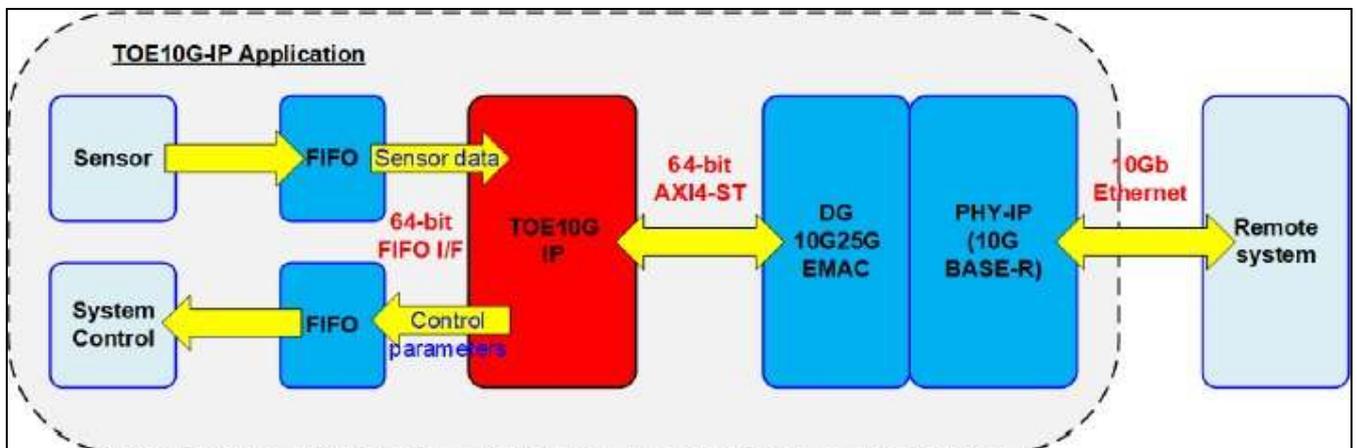


図 1: TOE10G-IP のアプリケーション

10Gb イーサネットは遠隔コントロール・システムにおいて非常に高速にデータを転送できるコミュニケーション・チャンネルです。10Gb イーサネットに TCP/IP プロトコルを適用することでシステムは信頼性を維持したまま大量のデータを超高速に転送することができます。TOE10G-IP コアは 10Gb イーサネットにおいて CPU や外部メモリを使用せずに TCP/IP 転送を実装可能とする IP コアです。このため本 IP コアはビデオ・データ・ストリームやセンサ監視システムなど FPGA で高速にデータを送信または受信するソリューションに適しています。

図 1 にセンサ監視システムのアプリケーション例を示します。センサからのデータは FIFO に蓄えられ本 TOE10G-IP を通して 10Gb イーサネット経由でリモート・システムへ送られます。本 TOE10G-IP コアは単一セッション内での全二重通信をサポートするのでリモート・システム側から 10Gb イーサネットを通してセンサ監視システムを制御するパラメータを送信できます。弊社ウェブサイト TOE10G-IP コアを使った FTP サーバー・デモをご紹介します。FTP サーバー・デモの詳細につきましては弊社までお問い合わせください。

本 TOE10G-IP コアは最速の転送速度を実現するようデザインされているため内部パイプライン・レジスタやバッファからのレイテンシ(遅延)は多少かかります。フィンテック向けのような低遅延を必要とするアプリケーションにおいては弊社の低遅延専用コアをご検討されるようお勧めします。低遅延 IP コアにつきましては以下の専用ウェブサイトをご参照ください。

[低遅延ネットワーク IP コア紹介サイト] [https://dgway.com/Lowlatency-IP\\_X.html](https://dgway.com/Lowlatency-IP_X.html)

## コア概要

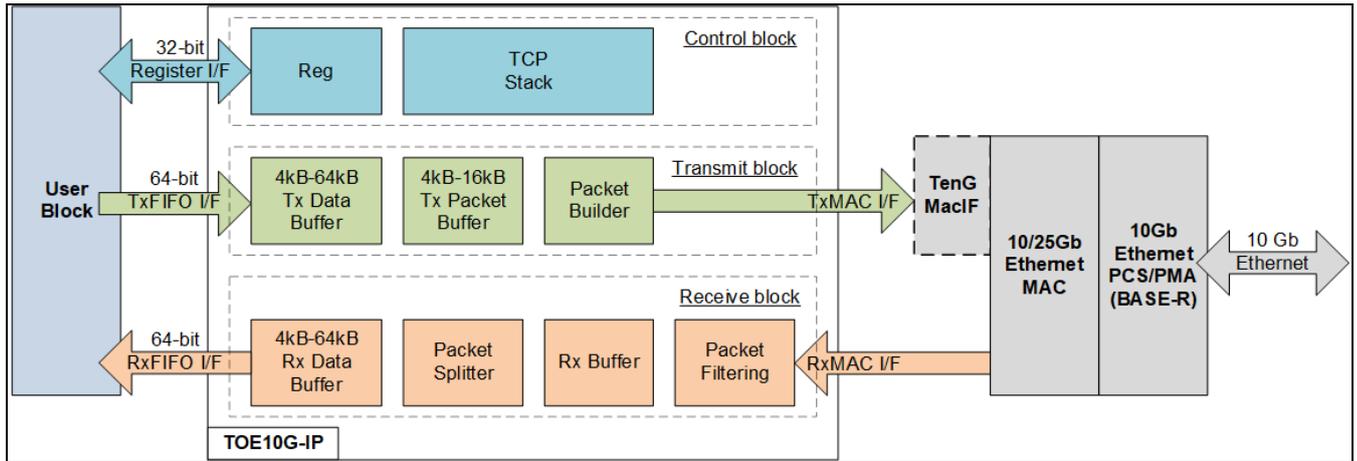


図 2: TOE10GB-IP コアのブロック図

TOE10G-IP コアは TCP/IP スタックを純ハードワイヤード・ロジックで実装しており Xilinx 製 10/25Gb EMAC IP コアおよび低レイヤ・ハードウェアの PCS/PMA(BASE-R)モジュールと接続します。TOE10G-IP コアのユーザ・インターフェイスは制御信号用のレジスタ・インターフェイスとデータ信号用の FIFO インターフェイスの 2 種類あります。

レジスタ・インターフェイスは 4 ビットのアドレスを持ち最大 16 レジスタへアクセスできますが、各レジスタはネットワーク・パラメータやコマンドやシステム・パラメータを設定します。本 IP コアは単一セッションの通信のみサポートします、このためユーザ側から設定するネットワーク・パラメータは本 IP コアおよび接続相手のターゲット・デバイスで固定値をとります。その後リセット・レジスタにてリセット状態を解除することで IP コアの初期化を開始します。また、このリセット処理はいくつかのネットワーク・パラメータを変更する場合にも必要となります。初期化プロセスは接続ターゲット・デバイスの MAC アドレスを取得するため、サーバーとクライアントの 2 つのモードがあります。初期化プロセス完了後本 IP コアは接続ターゲット・デバイスとのデータ転送が可能となります。

最初のステップとして TCP/IP 規格に従いポートをオープンすることでコネクションを確立する必要があります。本 IP コアは、IP コア側からのポート・オープンによるアクティブ・オープンと、ターゲット・デバイス側からのポート・オープンによるパッシブ・オープンの両方をサポートします。その後コアとターゲットの双方にてデータの送受信が可能となります。データを送信する場合、ユーザ回路は本 IP コアに対して総転送サイズとパケット・サイズを指定する必要があり、データは 64 ビット幅の送信 FIFO 経由で転送します。ターゲット・デバイスからデータを受信する場合、ユーザ回路は本 IP コアから受信 FIFO インターフェイスを介して受信データを読み出します。データの転送が完了した後は、本 IP コアからのポート・クローズによるアクティブ・クローズまたはターゲット・デバイスからのポート・クローズによるパッシブ・クローズによりコネクションは破棄されます。

ユーザによって異なるメモリ・リソースやパフォーマンスの要求に対応するため、本 IP コア内の各バッファ・サイズはユーザ側で選択できます。送信系においては 2 種類のバッファ容量を設定可能です、一つは送信データ・バッファでもう一つは送信パケット・バッファの各容量です。受信系は 1 種類で受信データ・バッファの容量設定が可能です。それぞれの方向ごとにデータ・バッファの容量を大きくするとパフォーマンスも向上します。コア内のハードウェア詳細については次章以降で説明します。

## コアの機能ブロック

本 IP コアは、制御ブロック、送信ブロック、受信ブロックの3ブロックに分かれています。

### 制御ブロック

#### レジスタ

本 IP コアの全てのパラメータは 4 ビット・アドレスで 32 ビット幅データ信号をもつレジスタ・インターフェイスを介して設定されます。レジスタ・インターフェイスのタイミング波形は単一ポート RAM インターフェイスに類似します。アドレスはライトとリードの方向で共有されます。各レジスタの詳細について表 3 に示します。

表 3: レジスタ・マップ定義

レジスタ・アドレス [3:0]	レジスタ名	方向	ビット	説明
0000b	RST	Wr/Rd	[0]	IP リセット。'1'でリセットし'0'でリセットを解除する。初期状態は'1'(リセット状態)。ユーザ回路が動作に必要な全パラメータをレジスタにセットしてから本レジスタに'0'を書き込むことでコアの動作が開始する。ユーザ回路が SML, SMH, DIP, SIP, DPN, SPN レジスタの値を変更する必要がある場合、本レジスタを一旦'1'としコアをリセット状態に移行させてから変更しなくてはならない。
0001b	CMD	Wr	[1:0]	アクティブ・モードでのユーザ・コマンド。"00":データ送信、"10": オープン・コネクション(アクティブ)、"11": クローズ・コネクション(アクティブ)、"01": 未定義(指定しないこと)。本レジスタによってアクティブ・コマンドを開始する前に、ユーザ回路は本レジスタの bit[0]をチェックしてコアは動作中でないことを確認しなくてはならない。本レジスタをセットするとコアは指定されたアクティブ・コマンド動作を自動的に開始する。
		Rd	[0]	システム・ビジー・フラグ。'0': アイドル状態、'1': ビジー状態
			[3:1]	現在の動作状態、"000": データ送信中、"001": アイドル状態、"010": アクティブ・オープン・コネクション、"011": アクティブ・クローズ・コネクション、"100": データ受信中、"101": 初期化中、"110": パッシブ・オープン・コネクション、"111": パッシブ・クローズ・コネクション
0010b	SML	Wr	[31:0]	コアの MAC アドレスの下位 32bit 定義レジスタ。RST レジスタをクリアする前に本レジスタで MAC アドレスを指定する必要がある。
0011b	SMH	Wr	[15:0]	コアの MAC アドレスの上位 16bit 定義レジスタ。RST レジスタをクリアする前に本レジスタで MAC アドレスを指定する必要がある。
0100b	DIP	Wr	[31:0]	接続ターゲット・デバイス側の IP アドレス 32bit を指定する。RST レジスタをクリアする前に本レジスタで IP アドレスを指定する必要がある。
0101b	SIP	Wr	[31:0]	本システム側の IP アドレス 32bit を指定する。RST レジスタをクリアする前に本レジスタで IP アドレスを指定する必要がある。
0110b	DPN	Wr	[15:0]	コネクションを行う接続ターゲット・デバイス側のポート番号を 16bit で指定する。アクティブ・オープンでコネクションを行う場合は RST レジスタをクリアする前に本レジスタでターゲット側のポート番号を指定する必要がある。パッシブ・オープンの場合、受信したオープン・パケットでターゲットのポート番号は自動的に決定される。
0111b	SPN	Wr	[15:0]	本システム側のポート番号を 16bit で指定する。RST レジスタをクリアする前に本レジスタで自身のポート番号を指定する必要がある。
1000b		Wr	[31:0]	送信データ数をバイト単位で指定する、ただし値は 8 バイトの倍数でなくてはならない。有効な値は 8~0xFFFFFFFF8 (最下位 2 ビットは無視される)。CMD レジスタに"00"をセットする前に本レジスタで送信データ数をセットする必要がある。この値はコア内部ロジックにて CMD レジスタがセットされた瞬間にサンプリングされるため、現在の送信が完了していない場合でも次の送信用の値をセットできる。また、再度前回と同じデータ数を送信する場合は本レジスタに再セットする必要はない。
		Rd	[31:0]	まだ送信されていない残りデータ数をバイト単位で表示する。

レジスタ・アドレス [3:0]	レジスタ名	方向	ビット	説明
1001b	TMO	Wr	[31:0]	全てのコマンドにて、受信パケットの待ち時間タイムアウト値を設定する。本レジスタは156.25MHzのカウンタで動作するためタイマ設定値は6.4nsの単位で指定する。本レジスタ値は0x6000以上の値としなくてはならない。
		Rd		各タイムアウト等のステータス、それぞれのビット定義は以下の通り [0] ARPで返信パケットをタイムアウト時間内に受信しなかったタイムアウト後、本IPコアはARP応答を受信するまでARP要求を再送し続ける [1] アクティブ・オープン時にSYNとACKフラグをタイムアウト時間内に受信しなかったタイムアウト後、本IPコアはSYNパケットを最大16回まで再送し、その後はFINパケットを送信してコネクションをクローズする [2] パッシブ・オープン時にACKフラグをタイムアウト時間内に受信しなかったタイムアウト後、本IPコアはSYN ACKパケットを次第16回まで再送し、その後はFINパケットを送信してコネクションをクローズする [3] アクティブ・クローズ時にFINとACKフラグをタイムアウト時間内に受信しなかったタイムアウト後、本IPコアは直ちにRSTパケットを送信しコネクションをクローズする [4] パッシブ・クローズ時にACKフラグをタイムアウト時間内に受信しなかったタイムアウト後、本IPコアはFIN ACKパケットを最大16回まで再送し、その後はRSTパケットを送信してコネクションをクローズする [5] データ送信時にACKフラグをタイムアウト時間内に受信しなかったタイムアウト後、本IPコアは前回送信したパケットを再送する [6] 受信パケットをロストした、受信データFIFOが一杯になった、あるいは誤ったシーケンス番号などの要因により、データ受信中タイムアウトとなった この場合本IPコアは重複ACKパケットを送信し再送を要求する [7] PSH[2]を'1'にセットした状態でのデータ送信コマンド実行中、ターゲットから受信したウィンドウ・サイズが少な過ぎる状態でタイムアウト時間が経過した タイムアウト後本IPコアはTMO[5]のリカバリ処理と同じ前回送信したパケットを再送する [21] 受信したACKパケットのシーケンス番号がスキップしたことを示すロスト・フラグ。その結果、TimerIntがアサートされTMO[6]は'1'となる [22] データ送信が完了していない状態でFINパケットを受信した [23] 受信データFIFOが一杯のため受信パケットが受信できない致命的なエラー [27] 受信パケットのロストを検出した [30] 受信パケット中にRSTフラグが検出された [31],[29:28],[26:24] 内部テスト用信号
1010b	PKL	Wr	[15:0]	バイト単位で指定する送信パケットのデータ長、ただし値は8バイトの倍数でなくてはならない。有効な値は8~16,000(最下位2ビットは無視される)。デフォルト値は1456バイト(非ジャンボ・フレームにて8の倍数での最大サイズ)。 <b>本レジスタの値はデータ転送中(システム・ビジー・フラグ='1')は変更してはならない。次の転送でも同じパケット・サイズの場合、ユーザは本レジスタの値を再セットする必要はない。</b>
1011b	PSH	Wr/ Rd	[2:0]	送信モードの指定(デフォルト値は全ビット'0')、本レジスタはアイドル中(CMDレジスタのbit0='0')のみ変更可能。注3参照 [0] '0': TDL≤PKLの場合(1送信指示が1パケット)に同一パケットを自動再送する '1': TDL≤PKLの場合の自動再送機能を禁止する(1パケットのみの送信とする) [1] PSHフラグをセット、送信時のパケット内TCPヘッダでPSHフラグに'1'をセットする

レジスタ・アドレス [3:0]	レジスタ名	方向	ビット	説明
1011b	PSH	Wr/ Rd	[2:0]	<p>[2] 送信コマンド実行中接続ターゲットからのウィンドウ・サイズがパケット・サイズより小さい(送信できない)状態が継続したことが原因でタイムアウトが発生した場合に、データ・パケットを再送する機能をイネーブルとする。このフラグは相手からのウィンドウ・アップデート・パケットをロストした場合にシステムがハングアップする現象を解消するために設けられた。このデータ・パケット再送によりターゲット・デバイス側ではロストした ACK パケットを再送する。以下の全条件を満たした場合にデータ再送を実行する</p> <p>(1) PSH[2]が'1'にセットされている</p> <p>(2) 現在のコマンドがデータ送信コマンドであり全データの転送が完了していない</p> <p>(3) 受信したウィンドウ・大豆がパケット・サイズより小さい</p> <p>(4) TMO レジスタでセットしたタイマーがタイムアウトとなった</p> <p>本ビット='0'の場合、上記の機能は無効である(デフォルト)</p> <p>本ビット='1'の場合、上記の機能を有効とする</p>
1100b	WIN	Wr/ Rd	[5:0]	<p>ウィンドウ・アップデート・パケットの自動送信を決める 1K バイト単位での閾値</p> <p>初期値は 0(ウィンドウ・アップデート・パケットを自動送信しない)</p> <p>本 IP コアは受信データ・バッファの空き容量が、前回パケットを送信した時点のウィンドウ・サイズより本レジスタで設定した容量分の空きが増えた時点でウィンドウ・アップデート・パケットを自動送信する</p> <p>例えば WIN="000001b"つまり 1K バイトの閾値を設定し前回送信パケットでウィンドウ・サイズが 2K バイトで報告した場合、ユーザ回路から受信データを 1K バイト分リードしてその分受信データ・バッファの空き容量が増えた時点で本 IP コアはウィンドウ・サイズを 3K バイトとしてウィンドウ・アップデート・パケットを自動送信する</p>
1101b	ETL	Wr	[31:0]	<p>拡張された総送信データ数をバイト単位で指定する。本サイズは 8 の倍数でなくてはならず本 IP コアは bit[2:0]を無視する。</p> <p>ユーザ回路は CMD='000'のデータ送信実行中に総送信長を増加するため本レジスタを設定できる、このため総転送サイズは本 IP コアに新たなコマンドを発行せずに送信データ・サイズを追加できる。</p> <p>たとえばデータ転送で TDL=4GB に設定したと仮定する。そして転送残りサイズが 1GB となった時点でユーザ回路は本 ETL に 2GB をセットすることで、6GB 分の連続データ送信が可能となる。(4GB+2GB)</p> <p>ただし以下に注意すること</p> <p>1) ETL レジスタは TDL のリード値(転送残りバイト数)が 128K バイト以上の状態でセットする必要がある。</p> <p>2) ETL レジスタの設定値は(0xFFFF-FFF8 – TDL のリード値)よりも小さい値とし、コア内データ・カウンタがオーバーフローしないようにすること(データ・カウンタは 32 ビットで実装されている)</p>
1110b	SRV	Wr/ Rd	[0]	<p>'0': クライアント・モード。本 IP コアはリセット状態から解放されると、DIP レジスタで設定されたターゲットの IP アドレスから ARP 要求を送信しターゲットの MAC アドレスを取得する</p> <p>'1': サーバー・モード。本 IP コアはリセット状態から解放されると、ターゲットから自分に向けた ARP 要求パケットを待機する。ARP 要求を受信し ARP 応答パケットを送信した後、本 IP コアのビジー状態は'0'に解除される。デフォルト値は'0'(クライアント・モード)</p> <p><b>注意: サーバー・モードにおいては RST レジスタを'1'から'0'とした場合、相手ターゲット・デバイスは ARP 要求を再送し本 TOE10G-IP コアの初期化を完了させなくてはならない。</b></p>
1111b	VER	Rd	[31:0]	IP コアのバージョン情報

注意:

- 1 ターゲットの MAC アドレスはクライアント・モードの場合 ARP の返信パケットにて、サーバー・モードの場合受信した ARP パケットから本 IP コアが自動検出・設定するため、ユーザ回路側でこのパラメータを設定する必要はありません。
- 2 ターゲットのポート番号はパッシブ・オープンの場合は受信パケットから自動設定されます。
- 3 PSH の bit0 における自動再送機能(デフォルトで機能 ON)は、1 パケットのみの送信の場合、受信側の PC の ACK 応答が非常に遅くなる現象を回避するための機能です。その機能が不要な場合このビットを'1'として自動再送機能を禁止してください。

## • TCP スタック

TCP スタックは、すべてのプロセスで他のモジュールを制御するための IP コア のメインコントローラです。IP コアの動作には、IP コアの初期化フェーズとデータ転送フェーズの 2 種類のフェーズがあります。

RST レジスタが'1'から'0'に変化した後、初期化フェーズを開始します。初期化フェーズでは、SRV[0]レジスタで設定されたクライアント・モードとサーバー・モードの 2 つのモードがあります。レジスタ・モジュールからのパラメータは TCP スタックで読み込まれ、送信ブロックと受信ブロックに設定され、モードに続いて初期化処理を完了するためにパケットを転送します。その後、IP コアはデータ転送フェーズに移行します。

TOE10G-IP とターゲット・デバイス間のデータ転送は、ポートを開き、データを転送し、ポートを閉じる 3 つのプロセスで構成されます。本 IP コアは、CMD レジスタ="10"(ポート・オープン)または "11"(ポート・クローズ)を設定すると、SYN または FIN パケットを送信してポートのオープンまたはクローズを開始できます。また、TCP スタックが受信ブロックから SYN または FIN パケットを検出した場合には、パッシブ・モードとしてターゲット・デバイスからポートのオープン/クローズを行うことができます。ポートのオープン/クローズ処理中は、TCP スタックはビジー・フラグを'1'にアサートします。すべてのパケットの転送が終了し、ポートが完全にオープンまたはクローズされると、Busy フラグは'0'にアサートされます。ConnOn 信号を監視することで、ポートがオープンまたはクローズになった状態を確認することができます。ConnOn が'1'にアサートされた場合(ポートが完全にオープンした状態)にデータを転送することができます。データを送信するため、ユーザからのデータは送信データ・バッファと送信パケット・バッファに格納されます。IP コアはパケット・ビルダにてネットワーク・パラメータを読み込んで TCP ヘッダを構築した後、送信ブロックでは、ユーザからのデータを含む TCP パケットをイーサネット MAC 経由でターゲット・デバイスに送信します。ターゲットがデータを正しく受信すると、ACK パケットが受信ブロックに返されます。TCP スタックは送信ブロックと受信ブロックの状態を監視し、データが正常に転送されたことを確認します。データが失われた場合、TCP スタックは現在のデータ転送を一時停止し、送信ブロックでデータの再送処理を開始します。

受信ブロックでデータを受信すると、TCP スタックは受信したデータの順番を確認します。データが正しい順序であれば、送信ブロックで通常の ACK パケットが生成されます。そうでなければ TCP スタックは送信ブロックを制御してターゲット・デバイスへの重複 ACK を生成することで、ロストしたデータの復旧処理を開始します。

表 4: 送信データ/送信パケット/受信データの各バッファで設定できる FIFO サイズ(アドレス・サイズで指定)

有効なビット幅	バッファ・サイズ	送信データ・バッファ	送信パケット・バッファ	受信データ・バッファ
9	4kByte	有効	有効	有効
10	8kByte	有効	有効	有効
11	16kByte	有効	有効	有効
12	32kByte	有効	不可	有効
13	64kByte	有効	不可	有効

## 送信ブロック

送信ブロック内には送信データ・バッファおよび送信パケット・バッファの 2 種類のバッファがあり、それぞれバッファ容量をパラメタライズで選択できます。より大きなサイズに設定すると送信パフォーマンスが向上します。送信データ・バッファおよび送信パケット・バッファの最少サイズは PKL レジスタで設定する送信パケットのサイズによって制限されます。送信データ・バッファからの送信データは送信パケット・バッファにて 1 パケット・サイズに分割されます。そしてコア内で用意された TCP ヘッダが送信パケット・バッファ内の TCP データと結合され完全な TCP パケットが生成されます。送信データ・バッファ内のデータは送信先ターゲット・デバイスからの ACK パケットによりデータ転送が完了したことを確認した後に削除されます。送信コマンドが完了した後、ユーザ回路は次の送信コマンドに向けて PKL レジスタおよび TDL レジスタでパケット・サイズおよび総送信データ・サイズを変更することができます。

---

- **送信データ・バッファ**

送信データ・バッファのサイズは本 IP コアの "TxBufBitWidth" パラメータで指定します。有効な値の範囲は表 4 に示すように 9-13 で 64 ビット・バッファのアドレス・サイズを意味します。本バッファのサイズは PKL レジスタで設定した送信パケット・サイズより 2 倍以上大きいサイズでなくてはなりません。ユーザ回路からの送信データは一旦本バッファに格納されます。このバッファ内データは送信先ターゲットから ACK パケットを受け取りデータ転送が確認されるとフラッシュされます。より大きなサイズに設定するとターゲットからの ACK パケット受信を待たずに連続してデータが送信できます。バッファをクリアするターゲットからの ACK パケットを受信してからユーザ回路から新しい送信データが IP コアに対して書き込まれるまでの時間が次の ACK パケット受信に間に合えば IP コアはユーザ回路からの送信データ書き込み待機時間がなくなります。その結果システムは 10GbE イーサネット接続環境で最高の送信パフォーマンスを達成できます。しかしながらネットワークのキャリアやインターフェイスに大きなレイテンシがあった場合、返信 ACK を受信しバッファをクリアユーザ回路から新しいデータを書き込まれる前に送信バッファ内の全データが送信されることがあります。その結果送信パフォーマンスはレイテンシによって低下します。

ユーザ回路が TDL レジスタに設定した総転送サイズより多くのデータを本バッファに書き込んだ場合、バッファ内に残ったデータは次の転送に使われます。ポートがクローズされるかあるいはリセットを検出すると、格納されていたデータはフラッシュ(クリア)されます。逆にユーザ回路から送信したデータ数が現在のトランザクションに不足していた場合、本 IP コアはパケットを送信せずユーザ回路から必要なデータが書き込まれるまで待機します。

- **送信パケット・バッファ**

送信パケット・バッファのサイズは IP コアの "TxPacBitWidth" パラメータで指定します。有効な値の範囲は表 4 に示すように 9-11 です。本バッファは少なくとも 1 送信パケット分のデータを格納する必要があります、そのためバッファ・サイズは PKL レジスタで設定した送信パケット・サイズ以上の容量とする必要があります。本 PKL レジスタの最大値は(送信パケット・バッファ・サイズ<バイト単位> - 24)となります。

- **パケット・ビルダ**

TCP パケットはヘッダとデータから構成されます。パケット・ビルダはレジスタ・モジュールで設定されたネットワーク・パラメータを受け取ります。また、IP チェックサムおよび TCP チェックサムが TCP ヘッダとして計算されます。その後全 TCP ヘッダが生成され送信パケット・バッファ内のデータと結合して EMAC へ転送されます。

## 受信ブロック

受信ブロックにはターゲット・デバイスから受信したデータを格納するため受信データ・バッファを含みます。パケット中のヘッダがレジスタ・モジュール内のネットワーク・パラメータで設定された期待値に一致した場合、データはバッファに格納されます。また、パケット内の IP と TCP のチェックサムは正しくなければなりません。そうでなければ、受信したパケットは破棄されます。より大きなサイズの受信データ・バッファを使用すると、受信パフォーマンスが向上する場合があります。また、TOE10G IP は、1 つのパケットのみがスキップされた場合にパケットの順序を入れ替える機能があります。例えば、パケット#1、#3、#2、#4 の順番で受信した場合、(後から受信した)パケット#2 を(先に受信した)パケット#3 と入れ替えます。ただしパケット#1、#3、#4、#2 のように複数のパケットの順番が入れ替わった場合(この場合パケット#3 と#4 の 2 パケットがパケット#2 よりも先に受信しています)、TOE10G-IP はデータの並び替えを行うことができず、パケット・ロスト状態として検出します。その後、重複 ACK パケットを生成して(ターゲットからの再送を要求して)データ復旧処理を行います。

### • 受信バッファ

前に受信したパケット処理が完了していない場合に EMAC からの受信パケットを一時的に保持するバッファです。

### • パケット・フィルタリング

受信パケットのヘッダは、このモジュールによって検証され、パケットの有効性が確認されます。パケットは以下の条件が満たされた場合に有効となります。

- (1) ネットワーク・パラメータがレジスタ・モジュールの値(MAC アドレス、IP アドレス、ポート番号)と一致している。
- (2) パケットが ARP パケットまたはデータ・フラグメント・フラグのない TCP/IPV4 パケットである。
- (3) IP ヘッダ長と TCP ヘッダ長が有効である(IP ヘッダ長が 20 バイト、TCP ヘッダ長が 20~60 バイト)。
- (4) IP チェックサムと TCP チェックサムが正しい。
- (5) シーケンス番号によってデコードされたデータ・ポインタが有効範囲内である。
- (6) アクノリッジ番号が有効範囲内である。

### • パケット・スプリッタ

このモジュールによりパケット・ヘッダは取り除かれ、TCP データのみが受信データ・バッファに格納されます。

### • 受信データ・バッファ

このバッファ・サイズは IP の "RxBufBitWidth" パラメータで設定します。有効な値は 9~13 で 4Kbyte~64Kbyte のバッファ・サイズです。受信データ・バッファのサイズは、送信パケットのウィンドウ・サイズに適用されます。受信データ・バッファのサイズが十分に大きい場合、ターゲット・デバイスはネットワーク・システム経路によって遅延する可能性のある本 IP コアからの ACK パケットを待たずに多くのデータを TOE10G IP に送信することができます。このため、受信データ・バッファのサイズを大きくすることで、受信性能を向上させることが可能です。

データは、ユーザがデータを読み出すまでバッファに格納されます。ユーザがバッファからデータを読み出さない時間が長い場合、バッファは一杯になり、ターゲット・デバイスはそれ以上のデータを本 IP コアに送信することができません。そのため、ユーザロジックでは、データの準備ができたときに本 IP コアからデータを直ちに読み出すことが推奨されます。受信データ・バッファが一杯にならなければ、ウィンドウ・サイズ・フルによって受信性能が落ちることはありません。

---

## ユーザ回路

ユーザ回路はレジスタ I/F を通してコマンドやパラメータを設定するステート・マシン等で実装できます。また、ステータスをチェックしエラーなく動作が完了したことを確認できます。データ・パスは FIFO と接続することで本 IP コアとデータ送信や受信を行います。

### 10G/25G イーサネット MAC

10/25Gb イーサネット用の MAC 層を実装します。TOE10G-IP と接続するためのユーザ・インターフェースは 64 ビット AXI4 ストリームです。TOE10G-IP はデザイン・ゲートウェイ社の 10G25GEMAC IP と直接接続できますが、Xilinx 社の 10/25G EMAC IP と接続する場合は、小型 FIFO を用いた追加ロジック(図 2 の TenGMaClF)を実装する必要があります。デザイン・ゲートウェイ社では、10G/25G イーサネット・ソリューションで適用可能な 10G25GEMAC-IP コアを提供しています。より少ないレイテンシ時間でリソースを最適化することができます。デザイン・ゲートウェイ社製 10G25GEMAC-IP コアの詳細については、以下の Web サイトを参照してください。

[https://dgway.com/products/IP/GEMAC-IP/dg\\_10g25gemacip\\_data\\_sheet\\_xilinx.pdf](https://dgway.com/products/IP/GEMAC-IP/dg_10g25gemacip_data_sheet_xilinx.pdf)

Xilinx 社からは 10G または 10G/25G イーサネット・サブシステム (イーサネット MAC およびイーサネット PCS/PMA)を提供しており、その機能については以下を参照してください。

10G イーサネット MAC(10GEMAC)

<https://japan.xilinx.com/products/intellectual-property/do-di-10gemac.html>

10G/25G イーサネット・サブシステム

<https://japan.xilinx.com/products/intellectual-property/ef-di-25gemac.html>

### 10G/25G イーサネット PCS/PMA(10GBASE-R)

このモジュールは Xilinx 社 LogiCORE から無償で提供されます、詳細については以下を参照してください。

<https://japan.xilinx.com/products/intellectual-property/10gbase-r.html>

<https://japan.xilinx.com/products/intellectual-property/ef-di-25gemac.html>

## コアの I/O 信号

コアに設定するパラメータおよび I/O 信号をそれぞれ表 5 と表 6 で説明します。本 IP コアの MAC インターフェイスは 64 ビット AXI4 ストリーム・バスです。

表 5: コアのパラメータ

パラメータ名	設定範囲	説明
TxBufBitWidth	9-13	送信データ・バッファのサイズをバッファのアドレス・バスのビット幅で指定します。
TxPacBitWidth	9-11	送信パケット・バッファのサイズをバッファのアドレス・バスのビット幅で指定します。
RxBufBitWidth	9-13	受信データ・バッファのサイズをバッファのアドレス・バスのビット幅で指定します。

表 6: コアの I/O 信号

信号名	方向	説明
<b>共通 I/F 信号</b>		
RstB	In	IP コアのリセット: ロウ・アクティブ信号である。
Clk	In	ユーザ I/F および EMAC I/F と同期する 156.25 MHz 固定のシステム・クロック
<b>ユーザ I/F</b>		
RegAddr[3:0]	In	レジスタの 4bit アドレスバス, ライト動作の場合 RegWrEn='1' で有効
RegWrData[31:0]	In	ライト・レジスタの 32bit 書込みデータ・バス, RegWrEn='1' で有効
RegWrEn	In	レジスタのライト・イネーブル、アドレスおよびデータに有効な値をセットし本信号にパルスを与えることで書込みを実行する。
RegRdData[31:0]	Out	レジスタの 32bit 読み出しデータ・バス、レジスタアドレスをセットしてから 1 クロックのレイテンシ後に有効なリード・データが本バス上に現れる。
ConnOn	Out	コネクション状態('1': コネクションはオープン状態、'0': コネクションはクローズ状態)。
TimerInt	Out	タイマ割り込み、タイムアウト発生時に本信号が 1 クロック期間分 H アサートされる。ユーザ回路は TMO[7:0] レジスタを読むことで割り込み要因を確認することができる。
RegDataA1[31:0]	Out	32 ビットの CMD レジスタ(RegAddr=0001b)リード値、Bit[0]は本 IP コアのビジー・フラグ
RegDataA8[31:0]	Out	32 ビットの TDL レジスタ(RegAddr=1000b)リード値
RegDataA9[31:0]	Out	32 ビットの TMO レジスタ(RegAddr=1001b)リード値
<b>送信 FIFO I/F</b>		
TCPTxFfFlush	Out	コアの送信データ・バッファがクリアされたことを示す。コネクションのクローズ時やリセット実行時に、1 クロック期間 H アサートされる。
TCPTxFfFull	Out	コアの送信データ・バッファの Full フラグ。ユーザ回路は本信号が H アサートされてから 4 クロック期間以内に送信データの書込みを停止しなくてはならない。
TCPTxFfWrEn	In	送信データ・バッファのライト・イネーブル。送信データを書き込む際にアサートする。
TCPTxFfWrData[63:0]	In	送信データ・バッファの 64bit 書込みデータ・バス、TOETxFfWrEn に同期する。
<b>受信 FIFO I/F</b>		
TCPRxFfFlush	Out	コアの受信データ・バッファがクリアされたことを示す。コネクションのオープン時に、1 クロック期間 H アサートされる。
TCPRxFfRdCnt[12:0]	Out	受信データ・バッファ内の受信データ総量を 64bit 単位で示す FIFO データ・カウンタ
TCPRxFfLastRdCnt[2:0]	Out	総受信データ・バイト数が 8 の倍数ではない場合に受信バッファ内の最終 64 ビット(8 バイト)データで有効なバイト数を示す。ユーザ回路は全 8 バイトを受信しないとデータを読み出すことができない。
TCPRxFfRdEmpty	Out	受信データ・バッファの FIFOEmpty フラグ。ユーザ回路は本信号が H アサートされたら直ちにデータの読み出しを停止しなくてはならない。
TCPRxFfRdEn	In	受信データ・バッファの読み出しイネーブル。受信データを読み出す際にアサートする。
TCPRxFfRdData[63:0]	Out	受信データ・バッファの 64bit 読み出しデータ・バス、TCPRxFfRdEn をアサートしてから 1 クロック期間のレイテンシ後に有効なリードデータが出力される。

信号名	方向	説明
MAC I/F		
tx_axis_tdata[63:0]	Out	MAC への 64bit 送信データ・バス, tx_axis_tvalid='1' で有効
tx_axis_tkeep[7:0]	Out	送信データのバイト・イネーブル信号、tx_axis_tdata='1' で有効
tx_axis_tvalid	Out	送信データが EMAC に対して有効であることを示す
tx_axis_tlast	Out	フレームの最終バイトであることを示す信号、tx_axis_tvalid='1' で有効
tx_axis_tuser	Out	エラーが発生していることを示す信号。本 IP コアにおいては常に'0'を出力する
tx_axis_tready	In	ハンドシェーク信号。tx_axis_tdata が正常に受信したときにアサートする 本信号はパケット送信中'0'にネゲートしてはならない
rx_axis_tdata[63:0]	In	MAC からの 64bit 受信データ・バス ,rx_axis_tvalid='1'で有効
rx_axis_tvalid	In	受信データの有効信号 rx_axis_tvalid はパケット転送中連続して'1'アサートされなくてはならない
rx_axis_tlast	In	フレームの最終データであることを示す信号,rx_axis_tvalid='1' で有効
rx_axis_tuser	In	受信フレームの最後(rx_axis_tvalid='1' かつ rx_axis_tlast='1')にそのフレームが CRC エラーを含むかどうかを示す信号、'1'は正常パケットで'0'はエラー・パケット
rx_axis_tready	Out	ハンドシェーク信号。rx_axis_tdata が受領されたときに'1'アサートされる。 rx_axis_tready は各受信パケット間のギャップ・サイズとして 2 クロック期間'0'ネゲートする

## タイミング・チャート

### IP コアの初期化

初期化プロセスはユーザ回路が RST レジスタ='0' とすることで開始します。TOE10G-IP コアは SRV[0] レジスタの値に応じて 2 つのモードで動作します、一つはクライアント・モード(SRV[0]='0')で、もう一つはサーバー・モード(SRV[0]='1')です。それぞれのモードにおけるタイミング波形を以下に示します。

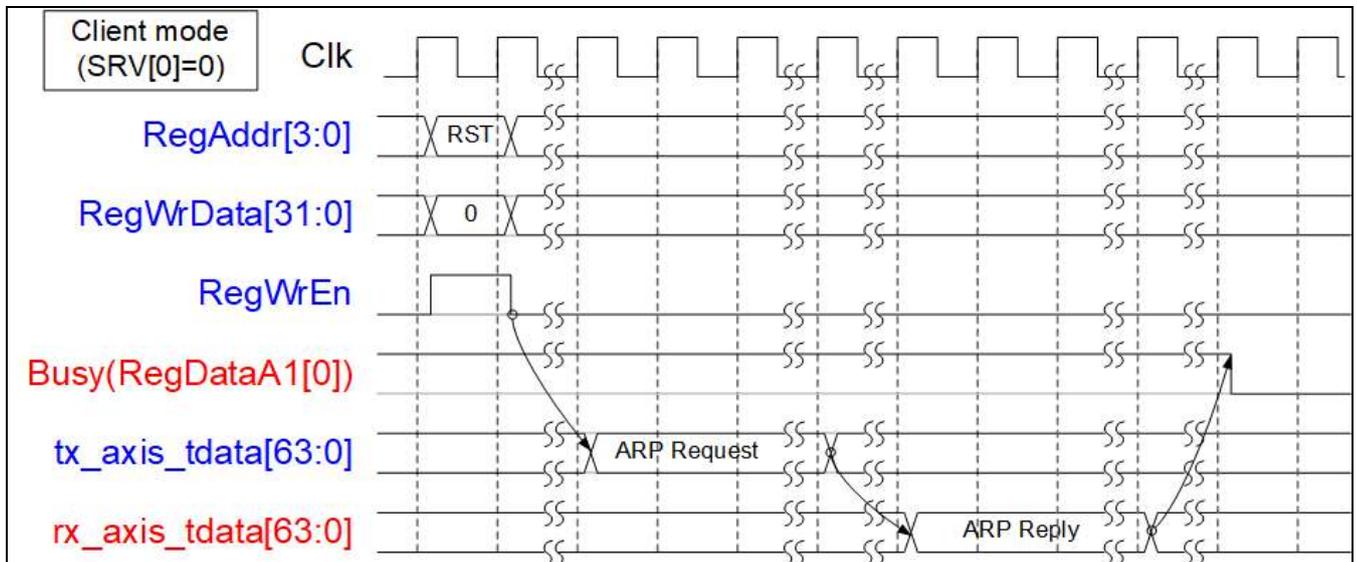


図 3: クライアント・モードの IP コア初期化

クライアント・モードの場合本 IP コアは ARP 要求を送信し、ターゲットからの ARP 応答を待ちます。ターゲット側 MAC アドレスは ARP 応答パケットから抽出します。その後ビジー信号(RegDataA1 レジスタの bit0 は '0' にネゲートされます。

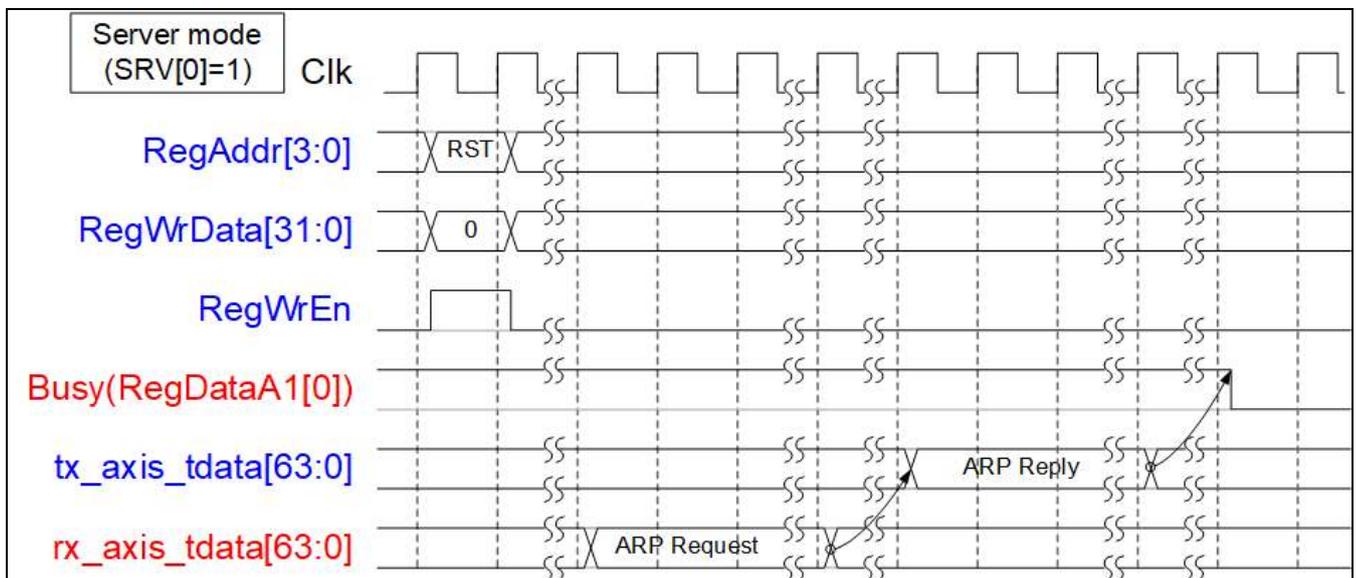


図 4: サーバー・モードの IP コア初期化

サーバー・モードの場合、リセット状態から解放された本 IP コアはターゲットからの ARP 要求を待ちます。ヘッダ情報が本 IP コアに設定されたネットワーク・パラメータに合致した ARP 要求を受け取った場合、本 IP コアは ARP 応答をターゲットに向けて送信します。ターゲット側 MAC アドレスは ARP 要求パケットから抽出します。最後にビジー信号を '0' にネゲートします。

## レジスタ・インターフェイス

全ての制御信号とネットワーク動作のパラメータはレジスタ・インターフェイス経由でセットあるいはモニタされます。レジスタ・インターフェイスのタイミング波形は単一ポート RAM に似ておりライトとリード・アクセスでアドレスを共有します。アドレスに対するリード・データ出力までのリード・レイテンシは 1 クロック期間です。レジスタ・マップは表 3 に示しています。図 5 に示す通りレジスタにライトする場合ユーザ回路は RegAddr と RegWrData に有効な値をセットし RegWrEn='1' とします。レジスタのリードは RegAddr を与えるだけで可能で対応する RegRdData は次のクロック期間で有効な値が出力されます。

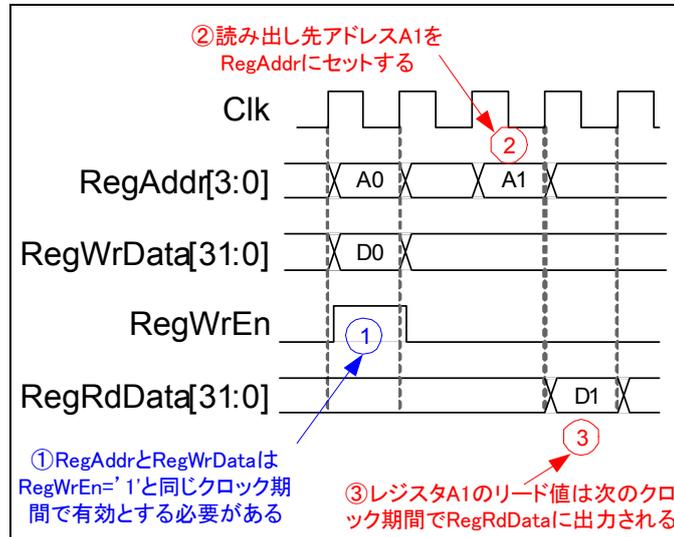
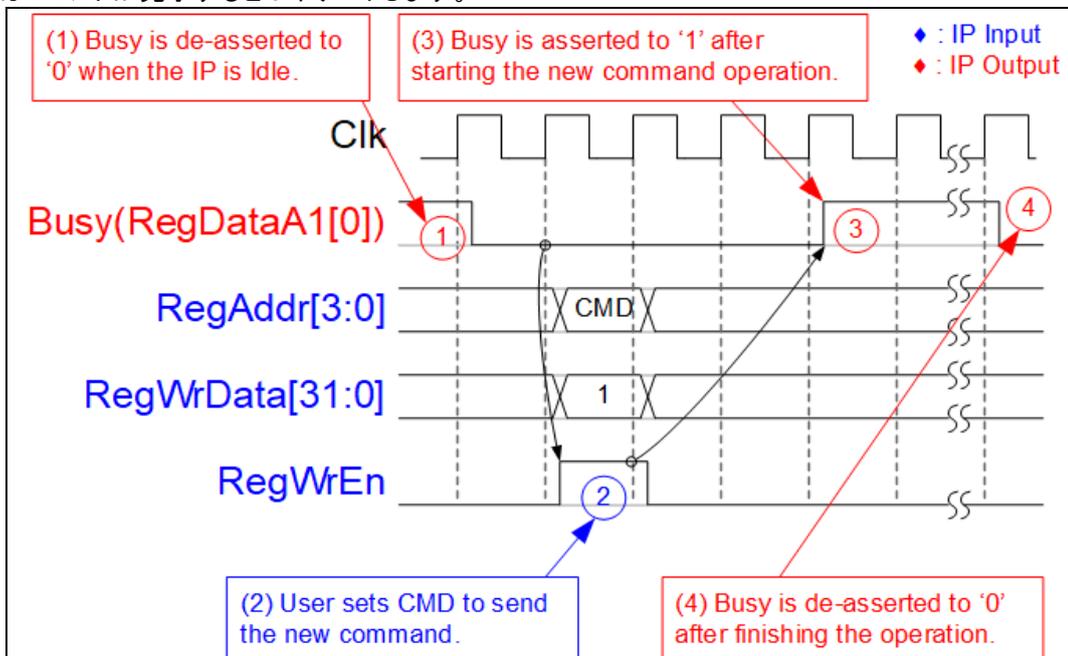


図 5: レジスタ I/F のタイミング・チャート

CMD レジスタにおいてはビジー・フラグ(RegDataA1[0]信号)を確認し、'0'ネゲート(IP コアがアイドル状態)である場合のみセットすることができます。図 9 に示すように CMD レジスタをセットするとビジー・フラグは'1'アサートされます。このビジー・フラグはコマンドが完了すると'0'ネゲートします。

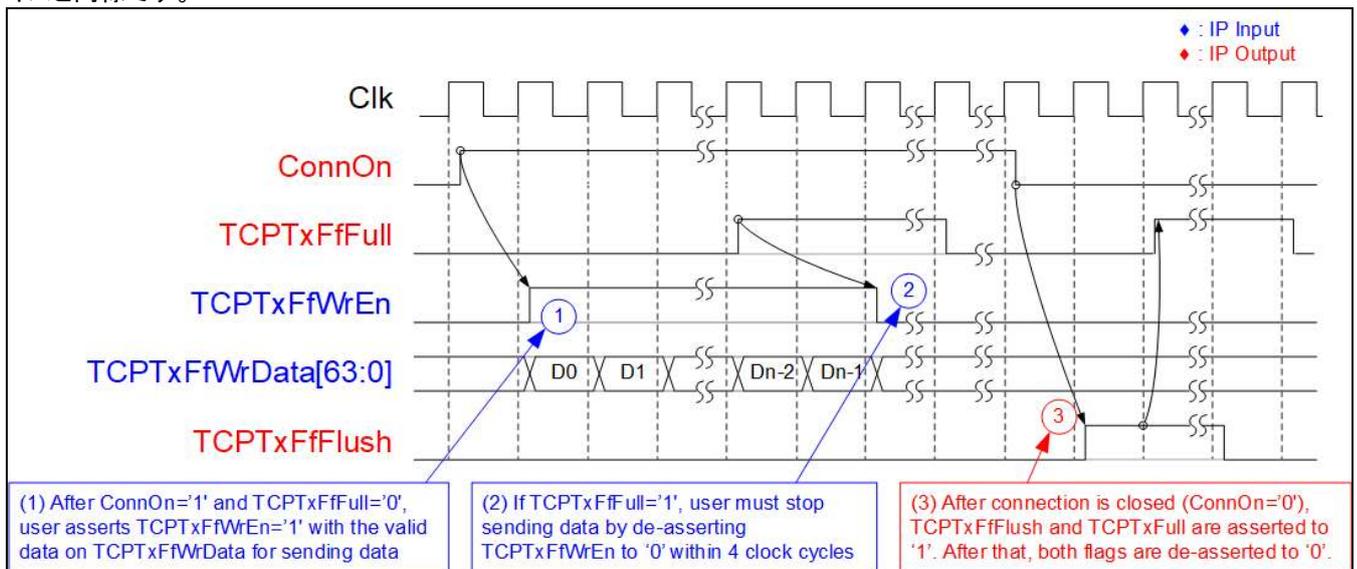


- (1) IP コアがアイドル状態のときビジーは'0'ネゲートされる
- (2) ユーザ回路は(ビジーをモニタし'0'ネゲートされたことを確認してから)送信コマンドが発行できる
- (3) IP コアは新たなコマンド動作を開始するとビジーを'1'アサートする
- (4) コマンド動作が完了した後ビジーは'0'ネゲートされる

図 6: CMD レジスタのライトはビジーがネゲート時のみ可

## 送信 FIFO インターフェイス

送信 FIFO インターフェイスを介して IP コアヘータを送信するためにはフロー制御信号のフル・フラグをモニタする必要があります。ライト信号は図 7 に示すようにライト・データとライト・イネーブルを使った汎用の FIFO のライト・インターフェイスと同様です。



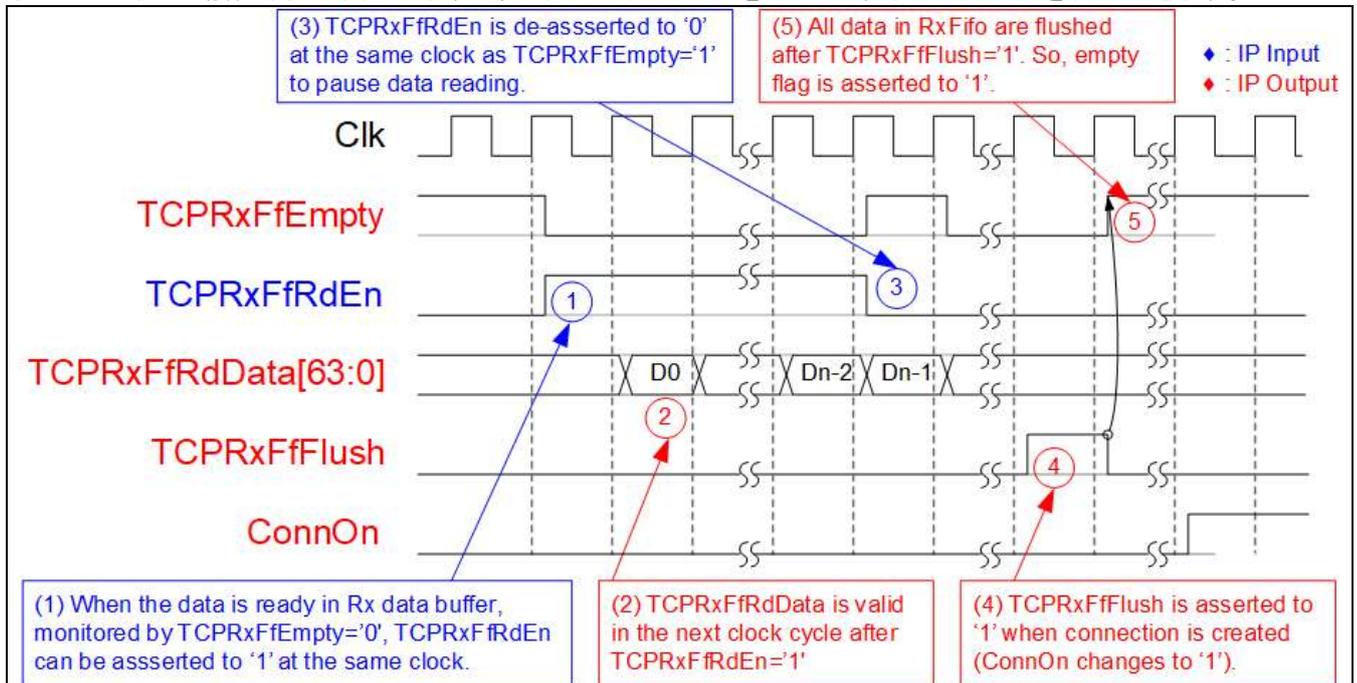
- (1) 送信データの転送は ConnOn='1'かつ TCPTxFfFull='0'となった後ユーザ回路が有効なデータを TCPTxFfWrData にセットし TCPTxFfWrEn='1'アサートすることで行われる
- (2) TCPTxFfFull='1'となった場合ユーザ回路は TCPTxFfWrEn を 4 クロック以内に'0'ネゲートしてデータ送信を停止しなくてはならない
- (3) コネクションがクローズした(ConnOn='0')後 TCPTxFfFlush と TCPTxFfFull が'1'アサートされる、その後どちらのフラグも'0'ネゲートされる

図 7: 送信 FIFO I/F のタイミング・チャート

- (1) データを送信する前にユーザ回路はフル・フラグ(TCPTxFfFull)が'1'アサートされておらず、かつ ConnOn が'1'であることを確認する必要がある。
- (2) TCPTxFfWrEn は TCPTxFfFull が'1'アサートされてから 4 クロック期間以内に'0'ネゲートしデータ送信を一時停止しなくてはならない。
- (3) 全データの送信が完了した後、本 IP コア側から(アクティブ)またはターゲットデバイス側から(パッシブ)によりポートがクローズされる。ポートがクローズされた後は以下の状況となる。
  - a) ConnOn が'1'から'0'ネゲートされる
  - b) TCPTxFfFlush が'1'アサートされ送信 FIFO 内の全データをクリアする
  - c) TCPTxFfFull が'1'アサートされコネクションのクローズ中にユーザ回路がデータを送信することを停止する

## 受信 FIFO インターフェイス

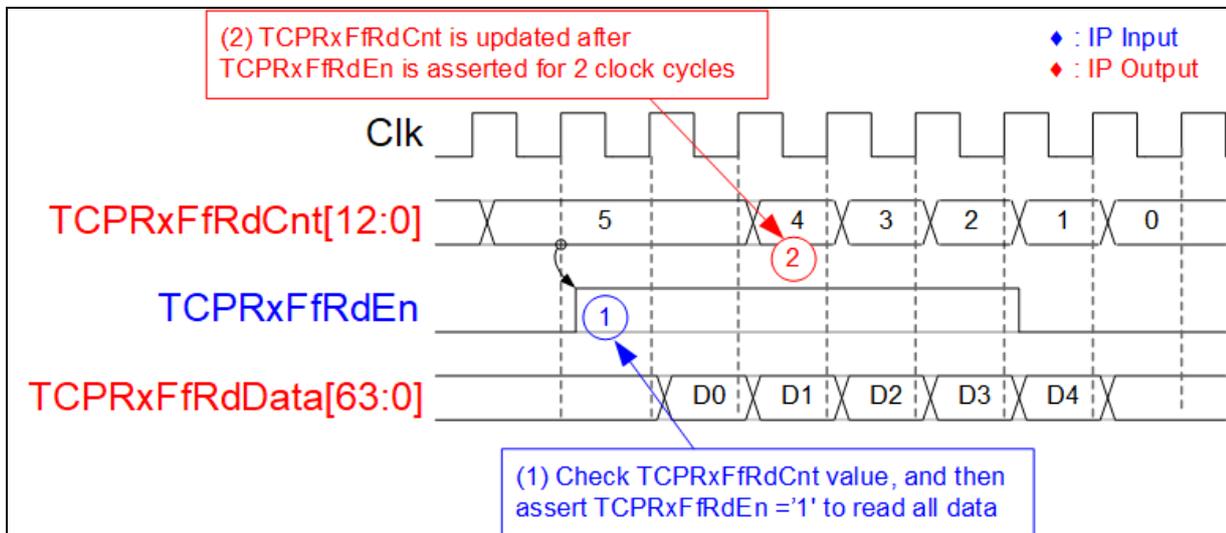
受信データが受信データ・バッファに格納されるとユーザ回路は受信 FIFO インターフェイスを介して受信データ・バッファからデータを読み出すことができます。ユーザ回路はエンプティ・フラグをモニタしてデータの有無を確認します、一般的な FIFO のリード動作と同じで図 8 に示すようにリード・イネーブルをアサートすることでデータをリードできます。



- (1) TCPRxFfRdEn は TCPRxFfEmpty が '0' ネグートされた同じクロック期間に '1' アサートすることができる。
- (2) TCPRxFfRdData は TCPRxFfRdEn='1' となった次のクロック期間で有効な値が出力される。
- (3) TCPRxFfRdEn は TCPRxFfEmpty が '1' アサートされた同じクロック期間でネグートしなくてはならない。
- (4) 受信 FIFO 内の全データを TCPRxFfFlush='1' でクリアするのは次のコネクション・オープン時 (ConnOn が '1' となる)。
- (5) (次のコネクション・オープンで) TCPRxFfFlush='1' で受信 FIFO 全データがクリアされるため、エンプティ・フラグは '1' アサートされる。

**図 8: 受信 FIFO I/F のタイミング・チャート**

- (1) TCPRxFfEmpty をモニタしデータ引き取りが可能かを確認する。データの準備ができたなら (TCPRxFfEmpty='0')、TCPRxFfRdEn を '1' アサートし受信データ・バッファからデータの読み出しが可能。
- (2) TCPRxFfRdData は次のクロック期間で有効となる。
- (3) TCPRxFfEmpty='1' となった場合即座に TCPRxFfRdEn='0' としてデータの読み出し動作を停止しなくてはならない。
- (4) ユーザ回路は次の(新たな)コネクションが確立される前に受信データ・バッファ内の全データを読み出す必要がある。受信データ・バッファ内の全データは次のコネクション確立と同時に TCPRxFfFlush='1' アサートによりクリアされる。次のコネクションが確立されると ConnOn は '0' から '1' に変化する。
- (5) FIFO のフラッシュ動作が完了すると、TCPRxFfEmpty は '1' アサートされる。



- (1) TCPRxFfRdCnt 値をチェックし残りデータ量分のクロック期間 TCPRxFfRdEn='1'でデータを読み出す。  
 (2) TCPRxFfRdCnt は TCPRxFfRdEn='1'アサートの 2クロック後に更新される。

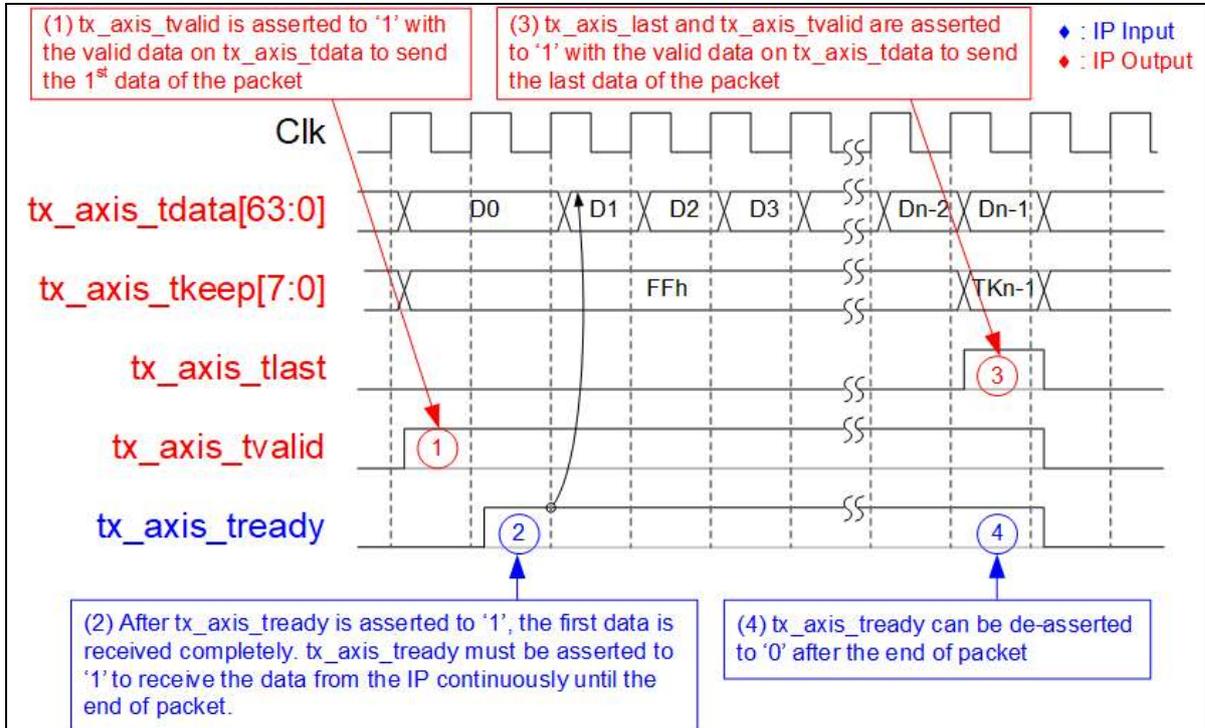
**図 9: FIFO 残量カウンタ情報による受信データの読み出し**

ユーザ回路がデータをバースト・モードで読み出す場合に対応するため、本 IP コアは受信 FIFO インターフェイスを 64 ビット単位で現在の総データ量を示すカウント信号を使うことができます。例えば図 9 のように受信データ・バッファ内に 5 データ(64bit データ x5)格納されているとします。この場合ユーザ回路は TCPRxFfRdEn='1' を 5 クロック期間アサートすることで受信データ・バッファ内の全データを読み出すことができます。注意点として TCPRxFfRdCnt のリード・カウンタ値は TCPRxFfRdEn によるリード・イネーブルのアサートから 2 クロックのレイテンシがあります。

## EMAC インターフェイス

本 IP コアの EMAC インターフェイスは 64 ビット AXI ストリーム・インターフェイスで実装されています。ただ本 IP コアの制約としてパケットの末尾までデータ送信は一時停止できません。このためパケット送信中は継続して tx\_axis\_tready を '1' アサートし続ける必要があります。tx\_axis\_tready は図 10 に示す通りパケットなご伊州データが転送された後で '0' ネゲートできます。

この制約から本 IP コアはデザインゲートウェイ社製 10G25GEMAC IP コアとは直結可能ですが、Xilinx 純正の 10G EMAC IP コアと接続する際には本 IP コアとの間に小容量 FIFO を含む特殊なロジックを必要とします。

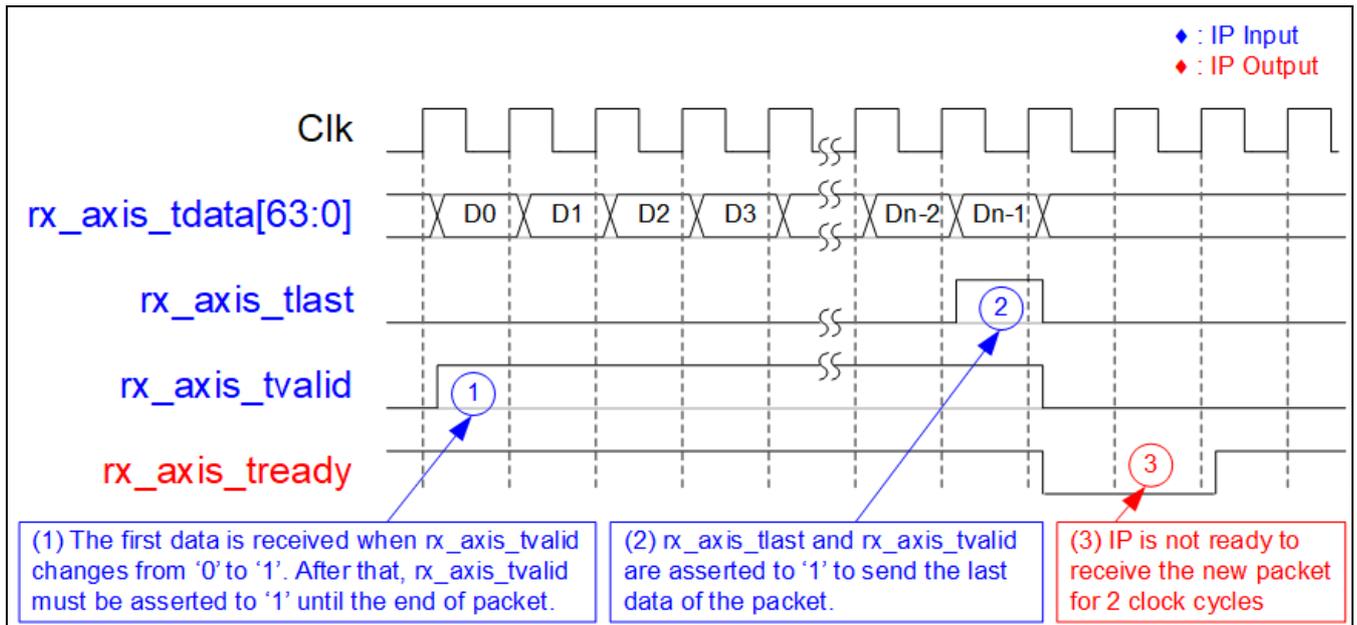


- (1) tx\_axis\_tvalid が tx\_axis\_tdata 上の有効データに合わせて '1' アサートされパケットの先頭データを送信
- (2) tx\_axis\_tready が '1' アサートされた後先頭データ受領が完了する、tx\_axis\_tready は本 IP コアからパケット最終データの受領まで連続して '1' アサートされなくてはならない
- (3) tx\_axis\_tlast が tx\_axis\_tdata 上の最終データに合わせて '1' アサートされパケットの最終データを送信
- (4) tx\_axis\_tready はパケット最終データの後 '0' ネゲート可能

図 10: EMAC インターフェイスの送信タイミング・チャート

- (1) 本 IP コアはパケット先頭データと合わせて tx\_axis\_tvalid をアサートする。先頭データが tx\_axis\_tready の '1' アサートにより EMAC で受け入れられるまで全信号はラッチされる。
- (2) EMAC は先頭データを受け取った後本 IP コアからのパケット最終データまで tx\_axis\_tready は連続して '1' アサートしなくてはならない。本 IP コアは 1 パケットの全データを連続して送信する必要がある。
- (3) パケットの最終データが転送されると tx\_axis\_tlast と tx\_axis\_tready の両方が '1' アサートされる。
- (4) パケット最終データの後 tx\_axis\_tready は '0' ネゲートし次パケット送信まで一時停止できる。

EMAC 受信側インターフェイスにおいては送信側インターフェイスと同様、1 パケット内のデータは連続して受信されなくてはなりません。つまり図 11 に示すようにパケットの先頭データから最終データまでのデータ有効信号(rx\_axis\_tvalid)は常に '1' アサートとなる必要があります。



- (1) rx\_axis\_tvalid が '0' から '1' に変化したときに先頭データが受信される
- (2) rx\_axis\_tlast と rx\_axis\_tvalid の両方が '1' アサートされパケットの最終データがコアへと受信される
- (3) 本 IP コアは次のパケット受信まで 2 クロック期間は受信できない

図 11: EMAC インターフェイスの受信タイミング・チャート

- (1) 本 IP コアは rx\_axis\_tvalid が '0' から '1' に変化し rx\_axis\_tdata 上に有効な先頭データが用意されたときに受信フレームの開始を検出する。
- (2) rx\_axis\_tlast='1'かつ rx\_axis\_tvalid='1'のときにパケット終了を検出する。それと同じクロック期間で最終データが rx\_axis\_tdata で転送される。
- (3) その後本 IP コアは rx\_axis\_tready を 2 クロック期間ネゲートしパケット後処理を行う。このため EMAC はこの 2 クロック期間データ転送を一次停止する必要がある。

---

## コアの使用例

### クライアント・モードの場合(SRV[0]='0')

データ送信とクライアント・モードでの受信における本コアのレジスタ設定シーケンス例を以下に示します。(より詳細についてはリファレンス・デザインのソースコードを参照してください。)

- 1 RSTレジスタを'1'にセットしIPコアをリセット状態とします。
- 2 MACアドレスをSML/SMHに、IPアドレスをDIP/SIPに、ポート番号をSPN/DPN (DPNは本IPコアによるアクティブ・オープンの場合に設定する必要があります)にセットします。
- 3 RSTレジスタを'0'にクリアしリセット状態を解除します、するとIPコアはARP要求を送信し相手側のMACアドレスを受信したARP応答から抽出して獲得します。初期化が完了するとビジー信号(RegAddrA1[0])は'0'にクリアされます。
- 4 コネクションは以下2つのモードで確立されます。
  - a. アクティブ・オープンの場合:CMDレジスタをセットしポートをオープン(IPコアよりSYNパケットが送信)します。その後ビジー・フラグが'0'ネゲートするのを待ちます。
  - b. パッシブ・オープンの場合:"ConnOn"が'1'となる(通信相手からSYNパケットを受信)のを待ちます。
- 5
  - a. データ送信の場合:TDL/PKLレジスタに総転送長およびパケット・サイズをセットしCMDレジスタをセットしてデータ送信を開始します。ユーザ回路は全転送データを送信FIFOに書き込み、CMDレジスタをモニタしてビジー・フラグが'0'にネゲートされるのを待ちます。次の転送を行う場合ユーザ回路はIPコアをリセット状態にせずにTDL/PKLレジスタを変更しCMDレジスタをセットすることができます。
  - b. データ受信の場合:受信FIFOのステータスをチェックし受信FIFOが空になるまでデータを読み出します。
- 6 コネクションの確立と同様コネクションは以下2モードでクローズできます。
  - a. アクティブ・クローズの場合:CMDレジスタをセットしてポートをクローズ(IPコアよりFINパケットが送信)します。その後ビジー・フラグが'0'ネゲートするのを待ちます。
  - b. パッシブ・クローズの場合:"ConnOn"が'0'となる(通信相手からFINパケットを受信)のを待ちます。

### サーバー・モードの場合(SRV[0]='1')

サーバー・モードとクライアント・モードの相違点は初期化プロセスにおける通信相手MACアドレスの取得方法です。クライアント・モードではMACアドレスは本IPコアが送信したARP要求に対して受信したARP応答から抽出します。サーバー・モードではコアに設定されたターゲットIPアドレスと合致したARP要求パケットから抽出します。データの送信・受信プロセスはクライアント・モードと同一です。サーバー・モードでの初期化シーケンスを以下に説明します。

- 1 RSTレジスタを'1'にセットしIPコアをリセット状態とします。
- 2 MACアドレスをSML/SMHに、IPアドレスをDIP/SIPに、ポート番号をSPN/DPNにセットします。
- 3 RSTレジスタを'0'にクリアしリセット状態を解除します、するとIPコアは相手側のMACアドレスを取得するためARP要求の受信を待ちます。そして接続ターゲットに対してARP応答を送信します。初期化が完了するとビジー信号(RegAddrA1[0])は'0'にクリアされます。
- 4 以降のプロセスはクライアント・モードのステップ4~6と同一です。

## コアの検証方法

本 IP コアのユーザ I/F 部についてはシミュレーションによる検証が可能であり、コア全体としては KC705/VC707/ZC706/KCU105/ZCU102/ZCU106/VCU118 ボードにより実機動作での検証が可能です。

## 必要とされる環境と設計スキルについて

本コアの実機動作確認やデザイン・ゲートウェイ社へのサポート依頼には Xilinx 純正の評価ボード (KC705/VC707/ZC706/KCU105/ZCU102/ZCU106/VCU118)が必要となるため、ユーザ側でコア購入時に手配してください。

また、本コアを使ってユーザ・システムを設計・実装するためには、HDL 言語設計技術・TCP/IP プロトコル知識および Vivado/ISE/EDK/SDK によるデザイン実装経験を必要とします。

## 注文情報

本製品は Xilinx 代理店から、あるいはデザイン・ゲートウェイ社から直接購入することが可能です。また、デバイス・ファミリに応じて以下のコアのラインナップが用意されています。

コア型番	対応ファミリ	検証用評価ボード	説明
TOE10G-IP-KT7	Kintex-7	KC705	Kintex-7 対応 TOE10G-IP コア
TOE10G-IP-VT7	Virtex-7	VC707	Virtex-7 対応 TOE10G-IP コア
TOE10G-IP-ZQ7	Zynq-7000	ZC706	Zynq-7000 対応 TOE10G-IP コア
TOE10G-IP-KU	Kintex Ultrascale	KCU105	Kintex Ultrascale 対応 TOE10G-IP コア
TOE10G-IP-ZUP	Zynq Ultrascale+	ZCU102/ZCU106	Zynq Ultrascale+対応 TOE10G-IP コア
TOE10G-IP-VUP	Virtex Ultrascale+	VCU118	Virtex Ultrascale+対応 TOE10G-IP コア

※ 上記以外のデバイス・ファミリ対応状況につきましては以下の問い合わせ URL 経由にてお問い合わせください。

<http://www.dgway.com/Inquiry.html>

## 更新履歴

リビジョン	日付	説明
1.0J	2014/06/05	日本語初期版の作成
1.1J	2014/06/26	10GEMAC の情報を追加
1.2J	2014/09/24	Full Duplex に対応
1.3J	2014/11/14	Zynq-7000 に対応
1.8J	2018-03-19	英語版 Rev1.8 に対応, PSH/WIN/ETL レジスタ, RegDataAx ポート追加 (Rev1.4) CMD[3:1]レジスタ・リード仕様を変更 (Rev1.5) レジスタのリード・バックをサポート (Rev1.6) Kintex Ultrascale (KCU105)対応、TCPRxFfLastRdCnt 信号追加 (Rev1.7) SRV レジスタおよび rx_axis_tready 信号追加 (Rev1.8)
1.13J	2021/2/10	英語版最新リビジョン Rev1.13 を翻訳