



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

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)
- 送受信同時の高速データ転送(Full Duplex)対応
- ジャンボ・フレームに対応
- 送受信の各バッファ・サイズが設定可能
- 8 バイト(64 ビット)単位での送受信パケット・サイズ (パケット・サイズと総転送サイズは 8 バイト単位とする必要があります)
- 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
- 制御インターフェイスは汎用的なレジスタ・インターフェイス
- Xilinx 製 10-Gbps イーサネット MAC と接続する 64 ビット AXI4 ストリーム
- 156.25MHz の単一クロック・ドメイン
- KC705/VC707/ZC706/KCU105/ZCU102 による実機動作リファレンス・デザインを用意(Full Duplex の同時送受信デザインを含みます)

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	ユーザ I/F 部のシミュレーション 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	KC705,VC707,ZC706,KCU105/ZCU102 での実機デモ環境
シミュレーション・ツール	
ModelSim SE	
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 ¹	BRAMTile ²	Design Tools
Kintex-7	XC7K325TFFG900-2	156.25	3206	3850	1325	36	Vivado2017.4
Zynq-7000	XC7Z045FFG900-2	156.25	3206	3848	1369	36	Vivado2017.4
Virtex-7	XC7VX485TFFG1761-2	156.25	3206	3847	1359	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 ¹	BRAMTile ²	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	156.25	3211	3786	727	34.5	Vivado2017.4
Zynq-Ultrascale+	XCZU9EG-FFVB1156-2-I	156.25	3211	3785	714	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。

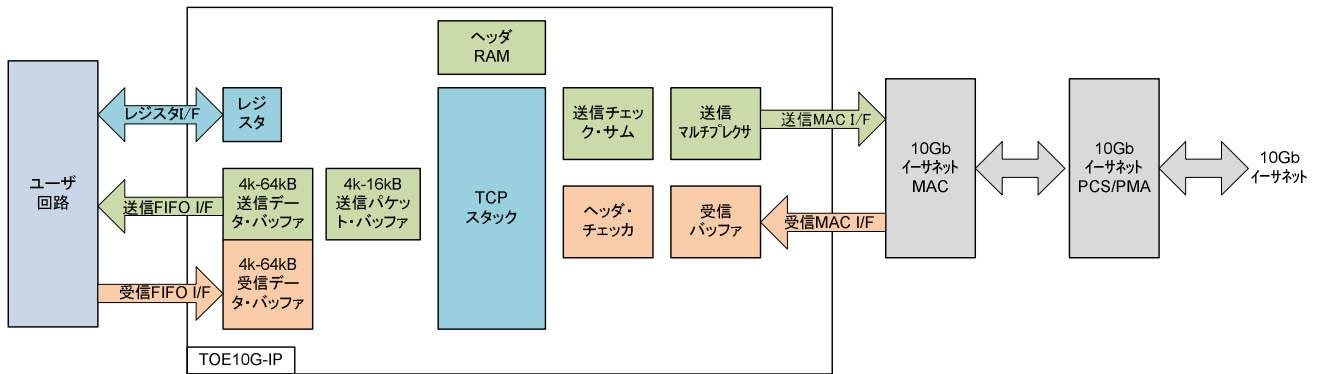


図 1: TOE10G-IP ブロック図

アプリケーション情報

本 TOE10G-IP コア(以下本 IP コアと略します)は 10Gbit イーサネットにて TCP/IP プロトコルによりデータ信頼性を維持しながら同時に高速転送が要求されるネットワーク・アプリケーション向けのコアです。本コアは CPU なしのハード・ロジックのみで TCP/IP による 10Gb イーサネット通信を可能とします。また、データ送信とデータ受信は独立して動作するため、同時送受信(フル・デュプレックス)に対応します。

コア概要

本 IP コアは Xilinx 社から提供される 10GbE EMAC IP コアおよび 10GbE イーサネット PCS/PMA と合わせてすることで、TCP/IP スタック、トランスポート層、インターネット層、リンク層、ネットワーク・データ転送の物理層を構成し動作します。ユーザはこのシステム構成にて 10Gb イーサネット経由で接続したネットワーク・デバイスと TCP/IP プロトコルを用いてデータの送受信が可能となります。

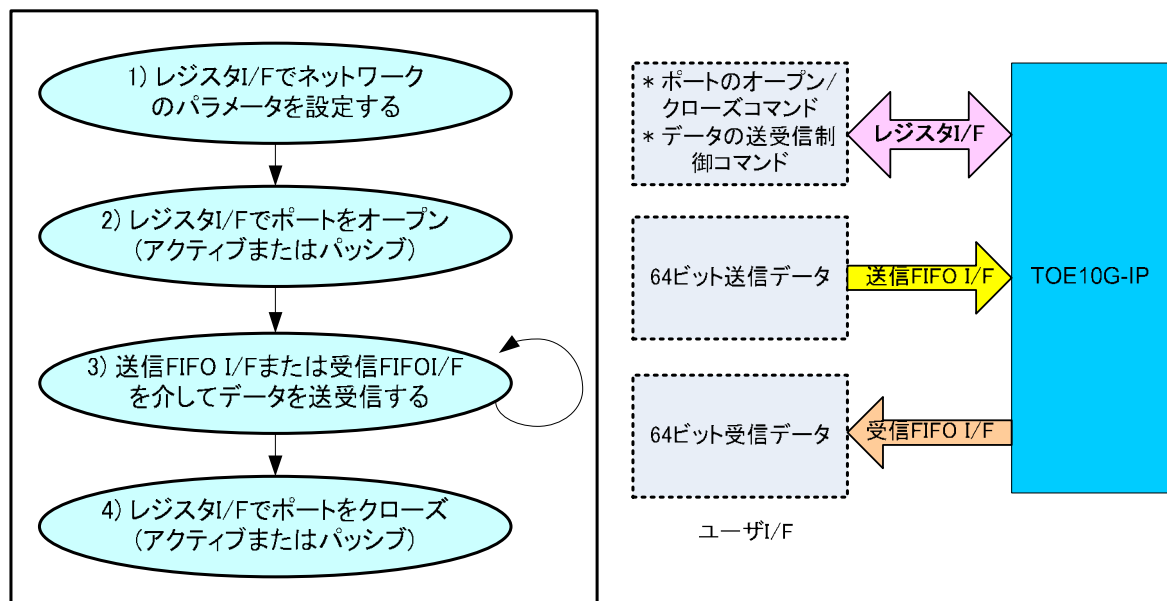


図 2: TOE10GB-IP のユーザ I/F と動作シーケンス

コアとユーザ・ロジック間のユーザ I/F(インターフェイス)は3種類あります。1つはコアの制御/ステータス確認用のレジスタ I/F で、送信データと受信データは FIFO I/F となります。システムの初期化時にユーザはパケット・サイズ、ポート番号、IP アドレスなどのパラメータをレジスタ I/F 経由でセットする必要があります。その後データ転送前にユーザ・ロジック(アクティブ・モード)あるいは接続ターゲット・デバイス(パッシブ・モード)でポートをオープンします。データ送受信の実行において、データのユーザ・ロジック側のインターフェイスは汎用 FIFO 形式となるので、簡単なロジックで実装できます。全てのデータ転送が完了すると、ポートのオープン時と同様、ユーザ・ロジック(アクティブ・モード)あるいは接続ターゲット・デバイス(パッシブ・モード)でポートをクローズします。

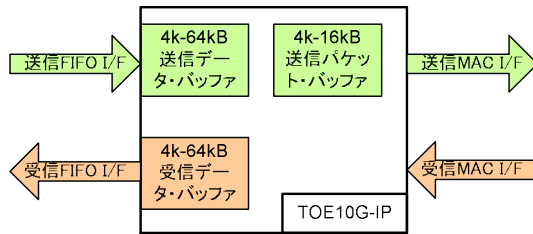


図 3: 送信/受信バッファ・サイズは調整可能

本 IP コアの納品物には、それぞれ異なったサイズの送信データ・バッファ、送信パケット・バッファ、そして受信データ・バッファのネットリストが同梱されます。異なったサイズからユーザ・アプリケーションに合わせて消費リソースを最適化できます。バッファ・サイズを大きくするとメモリ消費リソースも増えますがパフォーマンスが向上します。送信データ・バッファと送信パケット・バッファのサイズは送信パフォーマンスに影響し受信データ・バッファは受信パフォーマンスに影響します。更に送信データ・バッファと送信パケット・バッファのサイズはレジスタ I/F を介してユーザが設定するパケット・サイズにも関係します。送信パケット・バッファは送信パケット・サイズより大きくなくてはなりません、また送信データ・バッファのサイズは送信パケット・サイズより少なくとも 2 倍以上に設定する必要があります。

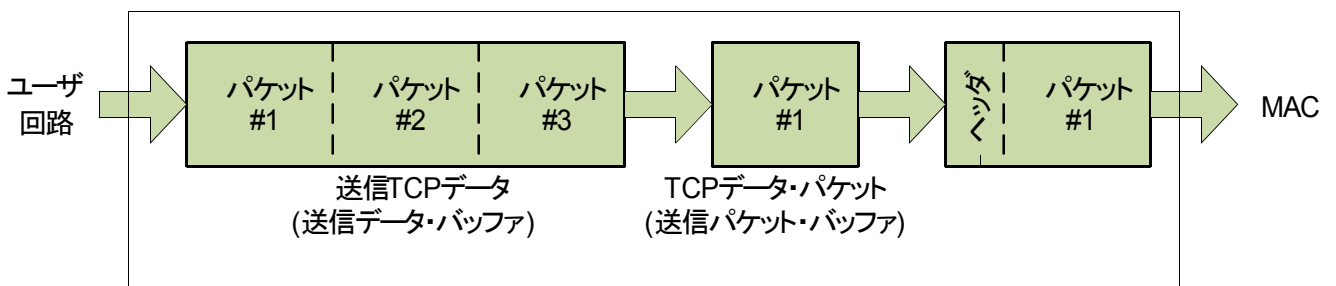


図 4: 送信データ・フロー

データ送信時、送信データ・バッファからの送信データはユーザ回路により設定されたパケット・サイズに分割され送信パケット・バッファに転送されます。そして送信パケット・バッファからの TCP データ・パケットはヘッダ情報が付加されてから EMAC に出力されます。TCP と IP チェックサムは本 IP コア内で自動計算されます。本 IP コアは送信動作中 ACK パケットをモニタします。ACK パケットには受信側で正常に受信できた最新のデータ・サイズ情報や受信バッファの空き容量情報が含まれます。そこで本 IP コアは受信側が新たなデータを受信できる状態であることが確認されれば次のデータ・パケットを送信します。本 IP コアが重複 ACK を受信し受信側でのデータ・ロストが発生した場合はロスト・パケットを再送します。レジスタ内のビジー・フラグは転送データ・サイズがユーザ回路から指定されたサイズと一致し全データ転送が完了するとクリアされます。従ってユーザ回路はこのビジー・フラグをモニタすることで転送ステータスを確認できます。

また、ユーザ回路は本 IP コアがデータ転送を実行していないアイドル状態のとき、ポートをクローズせずに次の転送用としてパケット・サイズや総転送サイズを変更することができます。次の送信コマンドを本 IP コアに発行する際にはビジー・フラグが必ず '0' となっていないなくてはなりません。

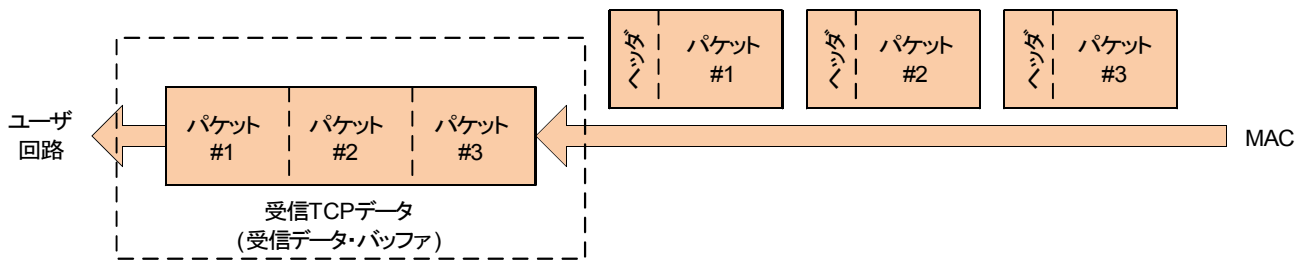


図 5: 受信データ・フロー

データ受信において受信したパケットはまずテンポラリ(一時)バッファに格納されます。そしてヘッダ情報とチェック・サム値をチェックします。ヘッダあるいはチェック・サム値にエラーがあった場合、そのパケットは破棄され受信データ・バッファには転送されません。正しいデータ・パケットを受信すると受信データ・バッファに転送され、また本 IP コアから ACK パケットが送信され、外部ネットワーク・デバイスに対して次のパケット送信を要求します。本 IP コアはそれ以上パケットが受信されず、かつ、受信したパケット・シーケンスが正しかった場合、アイドル状態に戻りビジー・フラグをクリアします。

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

ユーザ回路は TCP/IP オペレーションに関するパラメータをレジスタ・インターフェースにより設定できます。レジスタ・アドレスは全 4 ビット(最大レジスタ数=16)が実装されています。それぞれのレジスタのアドレス・マッピングを表 3 に示します。RST レジスタによるリセットが解除されると、各パラメータを設定したレジスタを初期値として動作が開始されます。

- TCP スタック

ユーザ回路から本 IP コアに対して命令が指示される(アクティブ・モード)と、TCP スタックはユーザ・コマンドを解析し送信ブロックに対してパケットの生成と出力を開始します。また受信ブロックは ACK パケットをモニタします。TCP スタックは送信・受信パケットのシーケンスを制御します。

アイドル常置にて新たなパケットを受信するパッシブ・モードにおいては、TCP スタックは受信ブロックからの受信パケットをデコードします。その次の動作シーケンスは受信パケットの内容に依存します。たとえば ARP 要求を受信すると ARP 応答を実行します。

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が生じた場合、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]
			[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 [22] データ送信が完了していない状況で 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 [21],[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)。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。
1011b	PSH	Wr/ Rd	[1:0]	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': TDLE ≤ 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': TDLE ≤ 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
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 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) 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 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。 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。 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) ただし以下に注意すること 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。 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 '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>

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。

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。本バッファからのデータは送信パケット・バッファに転送され次のパケット送信用に保持されます。

本バッファ・サイズは送信パフォーマンスに影響します。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。このため全プロセスやキャリアでのデータ・レイテンシはパフォーマンスにはそれほど影響しません。

ユーザ回路が総転送サイズの設定数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・サイズは PKL レジスタで設定した送信パケット・サイズと同じかそれ以上の容量とし、送信データ・バッファから転送された1パケット分のデータを格納できるように設定する必要があります。送信パケット・バッファ内のデータは EMAC および接続ターゲットでデータ受信準備が整うと転送されます。現在のパケット送信中に次のパケットが送信データ・バッファから送られてきます。このためパケットは連続して送信することが可能です。

• ヘッダ RAM

この RAM は送信パケットのヘッダ情報を格納するために使われます。ヘッダ RAM 内のパラメータは RST レジスタによりリセット状態が解除された時に更新されます。ターゲット MAC アドレスやターゲット・ポート番号などいくつかのパラメータは ARP 応答(クライアント・モード)や ARP 要求(サーバー・モード)やパッシブ・オープン・パケットなどで更新されます。

• 送信チェックサム

送信パケットが送出される前に本モジュールによりチェックサムが計算されます。そのチェックサム値はヘッダ RAM へセットされます。

• 送信マルチプレクサ

本モジュールによりヘッダ RAM と送信データバッファ内のデータが結合されイーサネット MAC を介して外部に送出されます。

受信ブロック

- **受信バッファ**

このバッファはヘッダ・チェックで処理される前のイーサネット MAC からの全ての受信パケットを一時的に保持します。本バッファの目的はヘッダ・チェックにより受信したパケットが有効であるかの確認完了を待つためのものです。有効な TCP データのみが受信データ・バッファへ転送されます。

- **ヘッダ・チェック**

受信パケット内のヘッダをチェックしユーザ回路からの設定値と比較します。ヘッダの内容が設定パラメータと適合しなかった場合やチェックサムがエラーであった場合は該当するパケットは破棄されます。また、受信データが重複した場合(受信済みと同一のパケットを再受信した場合)はそのパケットは無視されます。

- **受信データ・バッファ**

受信データ・バッファのサイズは IP コアの "RxBufBitWidth" 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 です。バッファ・サイズは TCP コネクションのウィンドウ・サイズに割り当てられます。本バッファ・サイズを大きくすると接続先ターゲット・デバイスにてネットワーク経路やデータ送信プロセスや受信バッファ・フルなどによって遅延する本 IP コアからの ACK を待たずに連続してデータを送信できるため、受信パフォーマンスが改善されます。また、受信データ・バッファのサイズを増加すると、ネットワークの経路等により受信パケットのシーケンスに狂いが生じた場合でも受信データを正しく並び替える機会も増やすことができます。

ユーザ回路

ユーザ回路はレジスタ I/F を通してパラメータの設定やコア状態のモニタを行い、また、送信 FIFO I/F を介した送信データの書き込みや受信 FIFO I/F を介して受信データの読み出しを行います。ユーザ回路はシンプルなハードウェア・ロジックで実装できます。

10 Gb または 10G/25G イーサネット MAC および PCS/PMA

リファレンス・デザインにおいては Xilinx から提供される 10Gb イーサネット MAC および 10Gb イーサネット PCS/PMA コアが使われます。Ultrascale+ファミリにおいては、10G/25G イーサネット・サブシステムを 10G イーサネットで動作させる必要があります。より詳細については以下のウェブサイトを参照してください。(MAC についてはお客様にて Xilinx 製 IP コアを手配して頂く必要があります)

10G イーサネット

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

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

10G/25G イーサネット

<https://www.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
共通 I/F 信号		
RstB	In	IP コアのリセット: ロウ・アクティブ信号である。
Clk	In	Xilinx ブロックの PHY 層からの 156.25 MHz 固定クロック
ユーザ I/F		
RegAddr[3:0]	In	レジスタの 4bit アドレスバス
RegWrData[31:0]	In	ライト・レジスタの 32bit 書き込みデータ・バス
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き込みを実行する。
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[6:0] 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 コアのビジイー・フラグ
RegDataA8[31:0]	Out	32 ビットの TDL レジスタ(RegAddr=1000b)リード値、ETL レジスタを使う場合に残り転送サイズをモニタするために使う
RegDataA9[31:0]	Out	32 ビットの TMO レジスタ(RegAddr=1001b)リード値、タイムアウト発生時の割り込みステータスとして使う
送信 FIFO I/F		
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き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 書き込みデータ・バス、TOETxFfWrEn に同期する。
受信 FIFO I/F		
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。
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		
rx_axis_tdata[63:0]	In	MAC からの 64bit 受信データ・バス
rx_axis_tvalid	In	受信データの有効信号、rx_axis_tdata に同期
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_tuser	In	受信フレームの最後にそのフレームが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・パケットを意味する
rx_axis_tready	Out	ハンドシェイク信号。rx_axis_tdata が受領されたときにアサートされる。
tx_axis_tdata[63:0]	Out	MAC への 64bit 送信データ・バス
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 に同期
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_tdata に同期
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_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	ハンドシェイク信号。tx_axis_tdata が正常に受信したときにアサートする

タイミング・チャート

IP コアの初期化

RSTレジスタ=0としたあとの初期化プロセスとしては、本 IP コアは SRV レジスタの設定すなわちクライアント・モードかサーバー・モードかによって 2 モードで動作します。

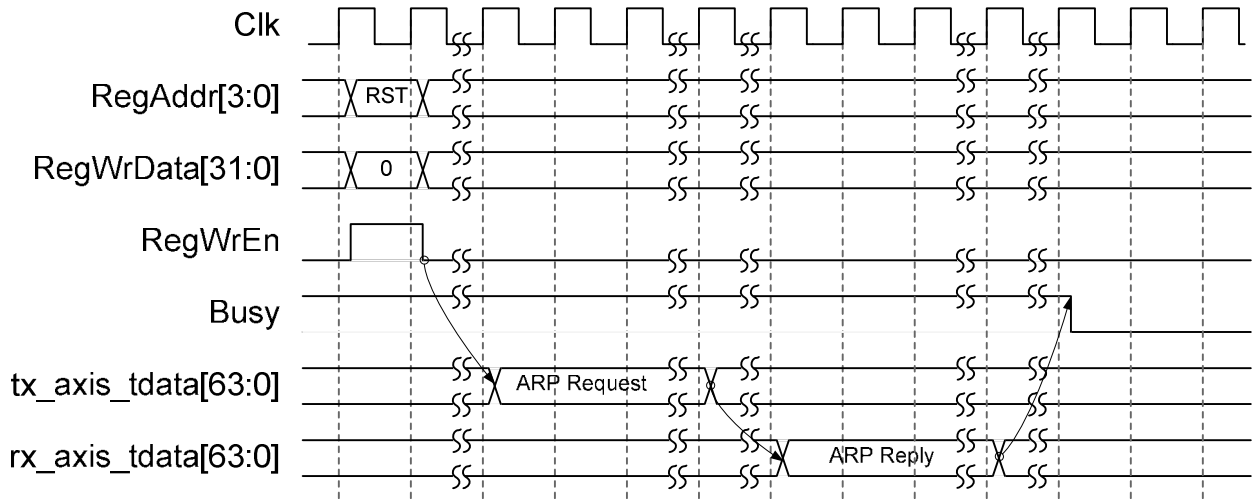


図 6: クライアント・モードの IP コア初期化

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。その後ビジー信号は'0'にネゲートされます。

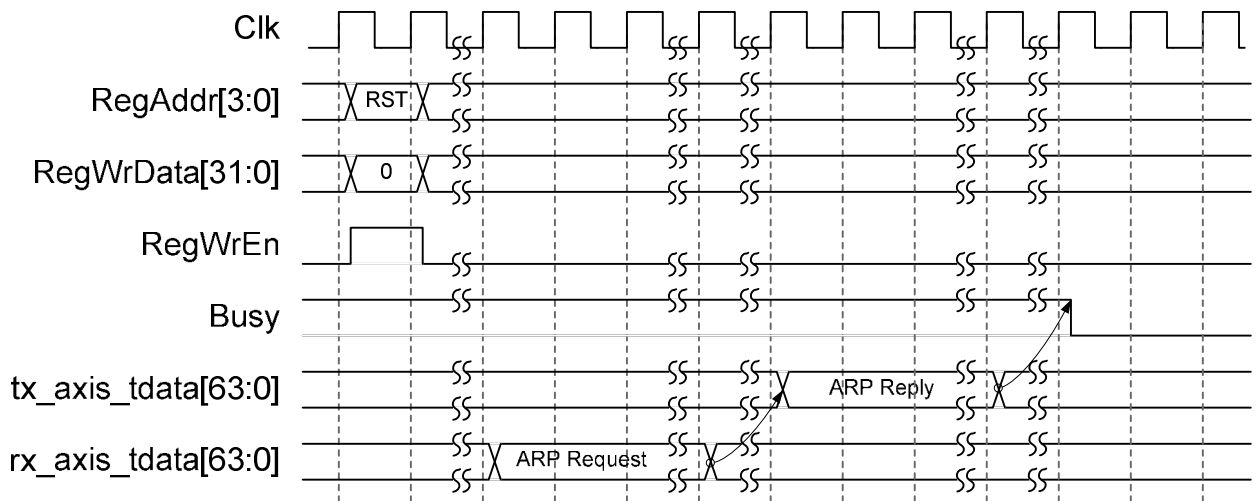


図 7: サーバー・モードの IP コア初期化

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'ネゲートします。

レジスタ・インターフェイス

ユーザ回路からの本 IP コア内部レジスタへのリード・ライト・アクセスは図 8 に示すタイミングにより実行します。アクセス先レジスタのアドレスマップは表 3 で示されます。レジスタへの書込みは RegAddr と RegWrData にそれぞれ有効なライト先のアドレスとデータをセットし 1 クロック期間 RegWrEn='1' とします。レジスタからの読み出しの際は、有効な RegAddr がコアに与えられた次のクロック期間に RegRdData にリードデータが出力されます。

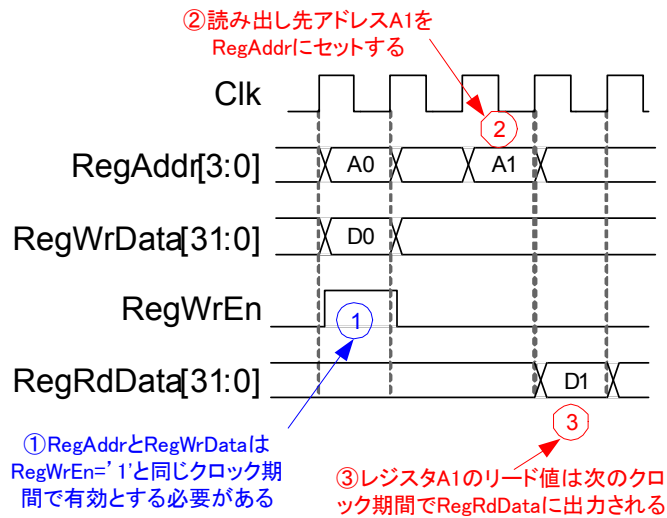
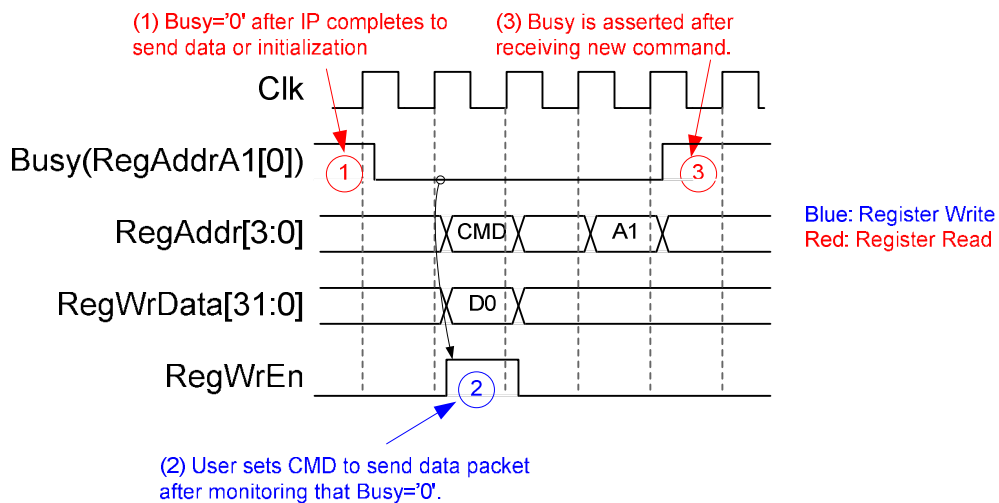


図 8: レジスタ I/F のタイミング・チャート

CMD レジスタにおいてはビジーフラグ(RegAddrA1[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。

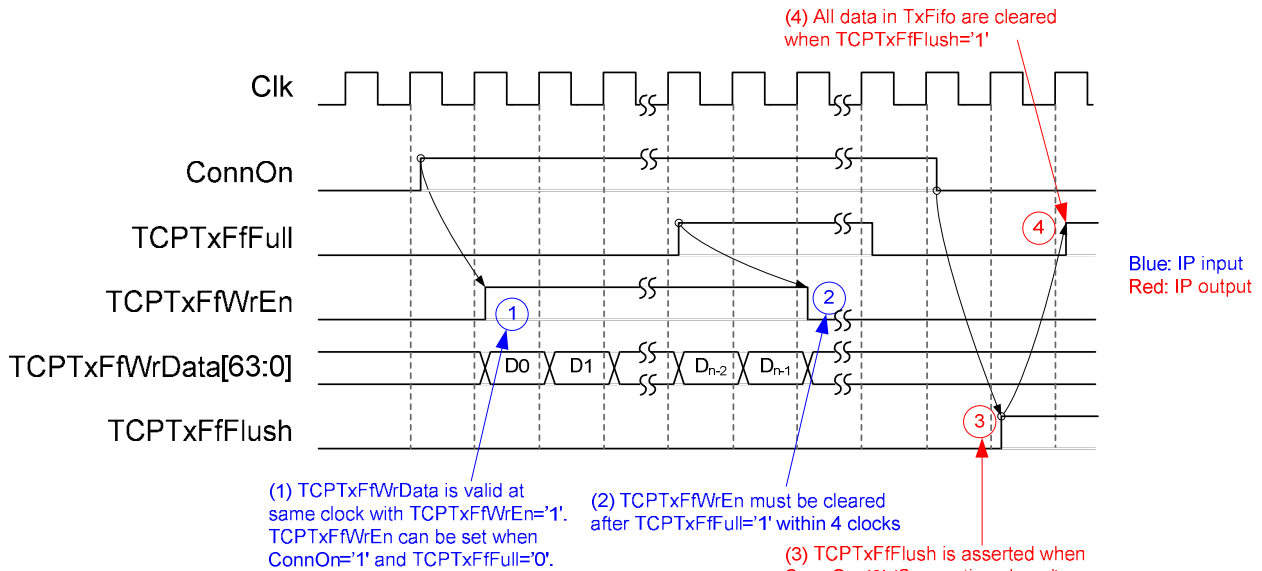


- (1) IP コアのデータ送信または初期化の完了後ビジーが'0'ネゲートされる
- (2) ユーザ回路はビジーをモニターし'0'ネゲートされたことを確認してから送信コマンドが発行できる
- (3) IP コアは新たなコマンドを受信するとビジーをアサートする

図 9: 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

ユーザ回路から本 IP コアを介して送信するデータは図 10 のタイミングに示す FIFO インターフェイスで書き込みます。データを送信する前にユーザ回路は Full フラグ(TCPTxFfFull)が'1'にアサートされておらず、かつ、ConnOn が'1'アサートされていることを確認する必要があります。そして TCPTxFfWrEn='1'とし送信データを TCPTxFfWrData に出力します。TCPTxFfFull が'1'にアサートされた場合は 4 クロック以内にデータの送信を停止しなくてはなりません。また、コネクションがクローズされた場合やリセットされた場合、本 IP コアは TCPTxFfFlush を'1'にアサートし送信 FIFO 内全データがクリアされたことを示し、さらに TCPTxFfFull も'1'にアサートしてユーザ回路からの誤ったライト動作から保護します。

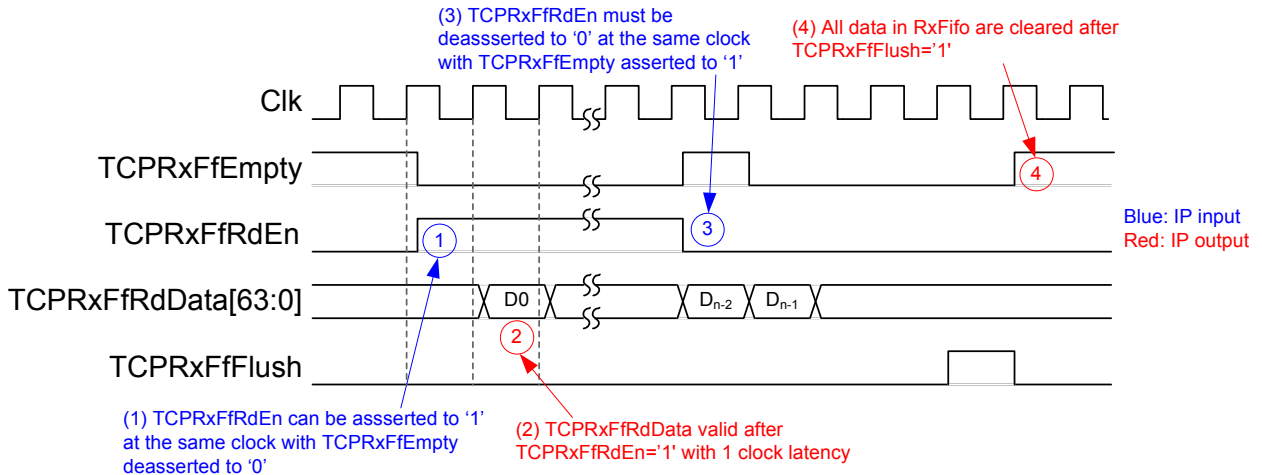


- (1) TCPTxFfWrData は TCPTxFfWrEn='1'と同じクロック期間で有効にする必要がある。TCPTxFfWrEn は ConnOn='1'かつ TCPTxFfFull='0'の場合にイネーブルできる。
- (2) TCPTxFfWrEn は TCPTxFfFull='1'となってから 4 クロック以内にネゲートしなくてはならない。
- (3) TCPTxFfFlush は ConnOn='0'となってから(コネクションがクローズされてから)アサートされる。
- (4) TCPTxFfFlush='1'で送信 FIFO の全データをクリアするとともに TCPTxFfFull='1'として誤ったライト動作から保護する。

図 10: 送信 FIFO I/F のタイミング・チャート

受信 FIFO インターフェイス

本 IP コアがターゲットからデータを受信すると受信データ・バッファに保存されます。ユーザ回路は FIFO インターフェイスを介してバッファ内の受信データを図 11 に示すタイミングで読み出すことができます。ユーザ回路は TCPRxFfEmpty をモニタしてデータの読み出しが可能かどうかを確認します。TCPRxFfEmpty が '0' にクリアされていればデータを読み出すことができます。FIFO 内受信データの読み出しは TCPRxFfRdEn を '1' にセットして行いますが、受信データは次のクロック期間に TCPRxFfRdData に出力されます。データの読み出し中に TCPRxFfEmpty が '1' にアサートされた場合、ユーザ回路は TCPRxFfRdEn を同じクロック期間内で '0' としなくてはなりません。受信データ・バッファ内の全データはコネクションのオープン時にクリアされ、それは TCPRxFfFlush が '1' にアサートされることで検出できます。



- (1) TCPRxFfRdEn は TCPRxFfEmpty が '0' にネゲートされた同じクロック期間に '1' に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' にアサートされた同じクロック期間でネゲートしなくてはならない。
- (4) TCPRxFfFlush='1' (次のコネクション・オープン時にコアがアサートします) で受信 FIFO に残っていた全データはクリアされる。

図 11: 受信 FIFO I/F のタイミング・チャート

受信 FIFO の状態は TCPRxFfRdCnt から確認することができます。この信号は受信 FIFO 内の残りデータ量を示します。このためユーザ回路は図 12 に示すように TCPRxFfRdEn='1' として読み出すデータ量をこの信号の残量情報で決めることができます。

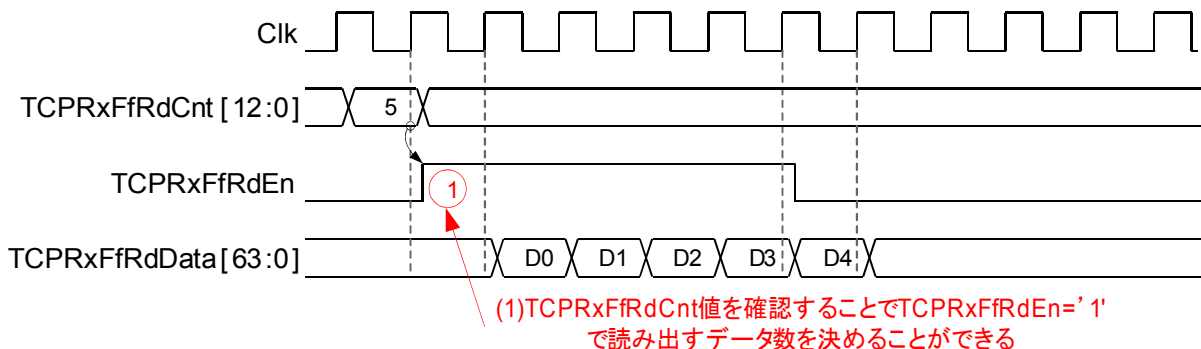


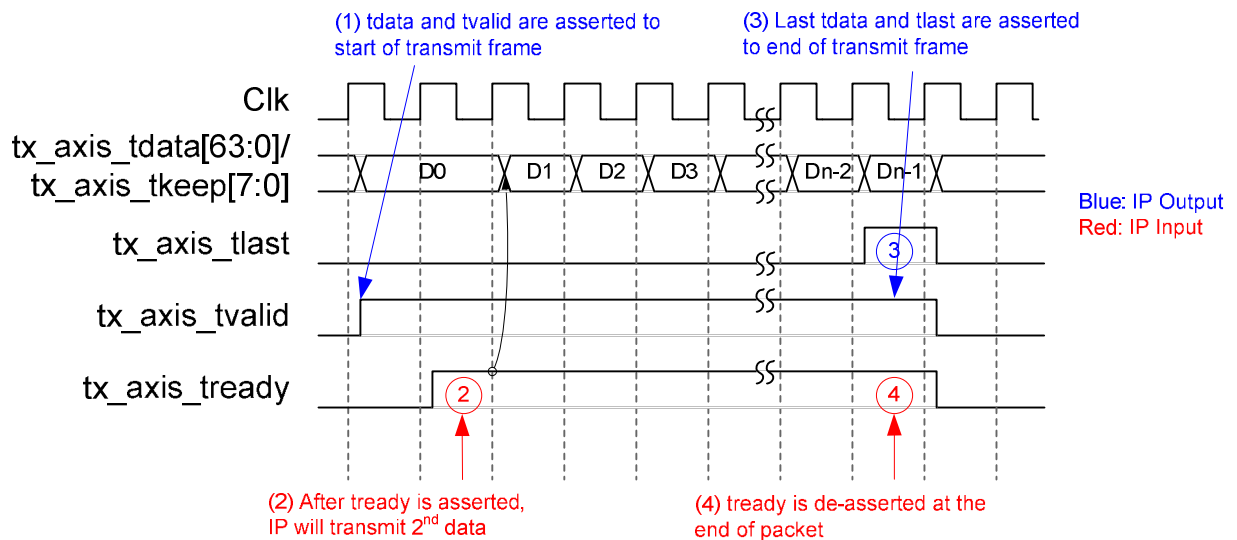
図 12: FIFO 残量カウンタ情報による受信データの読み出し

EMAC インターフェイス

本 IP コアがパケットを送信する際は、パケットの先頭データで tx_axis_tvalid をアサートします。その信号は tx_axis_tready が '1' アサートされデータ転送要求が確認されるまで保持されます。その後 tx_axis_tready はパケット最終データまで連続して '1' アサートされ続け全パケットが連続して転送されなくてはなりません。パケットの最終データを示すため、tx_axis_tlast と tx_axis_tvalid がともに '1' アサートされます。

Xilinx の EMAC 新バージョンは tx_axis_tready がパケットの先頭と末尾の間でネゲートされる場合があります。本 IP コアはこの転送中のネゲートには対応できません。このため本 IP コアと Xilinx EMAC 間に小規模なバッファを介して接続する必要があります。その接続ロジックはリファレンス・デザインにて HDL コードで提供されます。

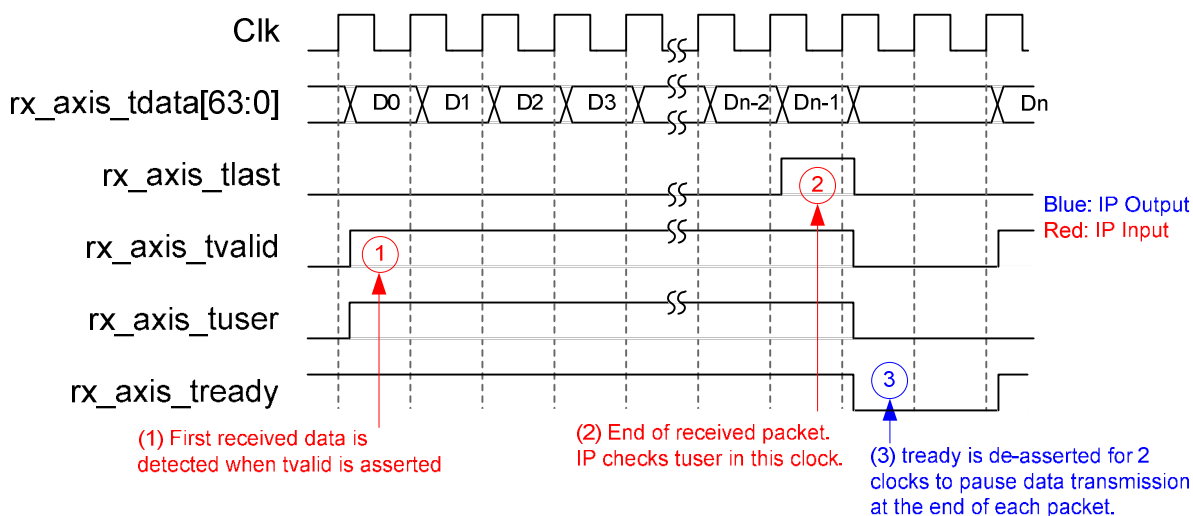
また、10G/25G Xilinx イーサネット MAC は(パケット・サイズが 60 バイト以下の場合に必要な)ゼロ・パディング機能が内蔵されていません。このため、Ultrascale+ デバイスにおけるリファレンス・デザインでは総転送サイズが 60 バイト以下の場合にゼロ・パディングを追加するロジックを実装しています。



- (1) tdata と tvalid がアサートされフレーム転送が開始する
- (2) tready がアサートされた後本 IP コアは次のデータを転送する
- (3) フレーム転送の最終データが tdata と tlast で出力される
- (4) パケット最終データで tready がネゲートされる

図 13: EMAC インターフェイスの送信タイミング・チャート

図 14 に受信側のタイミング・チャートを示します。本 IP コアは rx_axis_tvalid が '0' から '1' アサートされるのをモニタし、rx_axis_tdata がフレーム受信の開始からパケット最終データでアサートされる rx_axis_tlast まで連続して受信します。このため、rx_axis_tvalid は 1 パケットの全データ転送中(最初の tvalid から tlast まで連続して)常時 '1' アサートされている必要があります。1 パケットの受信後、rx_axis_tready は 2 クロック期間ほど '0' ネゲートすることでデータ転送を一旦休止し受信パケットの検証を完了させます。このため次のパケットは tready=0' ネゲート中に転送することはできません。



- (1) tvalid がアサートされることで受信データの先頭を検出する
- (2) 受信パケットの終了、本 IP コアは tuser 信号をチェックする
- (3) 各パケット終了から 2 クロック期間ほど tready はネゲートし転送を休止する

図 14: EMAC インターフェイスの受信タイミング・チャート

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。
 - 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。

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。

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 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)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	Zynq 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)