



**Design Gateway Co.,Ltd**

- 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](mailto:sales@dgway.com)
- URL: [www.dgway.com](http://www.dgway.com)

**特長**

- UDP/IP プロトコル・スタックを実装
- IPv4 に対応
- 単一ポート接続  
(コアを複数インスタンスすることでマルチ・セッションに対応可)
- 8 バイト(64 ビット)単位での送受信パケット・サイズに対応
- 送受信バッファはリソースとパフォーマンスに合わせて最適化調整が可能
- データは標準的な FIFO 接続
- 制御は一般的なレジスタ・インターフェイス接続
- Xilinx 標準 10Gbps イーサネット MAC コアと接続する 64 ビット AXI4 ストリーム
- クロック周波数 156.25MHz の単一クロック・ドメイン
- KCU105/ZCU102 Xilinx 評価ボードによるリファレンス・デザインが提供可能
- IP フラグメントに対応
- 安心の国内サポート

Core Facts	
コアの提供情報	
提供ドキュメント	● データ・シート ● リファレンス・デザイン説明書 ● 実機デモ手順書
コア形態	● 暗号化されたネットリスト
検証方法	● 参照デザインによる実機評価
制約ファイル	● 参照デザインで SCF ファイル
デザイン例使用言語	● VHDL
その他	● Xilinx 各種評価ボード用参照デザイン
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の国内サポート	

表 1: コンパイル結果例 (7 シリーズ)

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices <sup>1</sup>	BRAMTile <sup>2</sup>	Design Tools
Kintex-7	XC7K325TFFG900-2	156.25	1883	2170	782	36	Vivado2017.4
Virtex-7	XC7VX485TFFG1761-2	156.25	1883	2170	796	36	Vivado2017.4
Zynq-7000	XC7Z045FFG900-2	156.25	1883	2177	798	36	Vivado2017.4

備考:

- 3) 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。
- 4) 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)です。

表 2: コンパイル結果例 (UltraScale シリーズ)

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	CLB <sup>1</sup>	BRAMTile <sup>2</sup>	Design Tools
Kintex-UltraScale	XCKU040FFVA1156-2E	156.25	1871	2223	433	34.5	Vivado2017.4
Zynq-UltraScale+	XCZU9EG-FFVB1156-2	156.25	1871	2220	433	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)です。

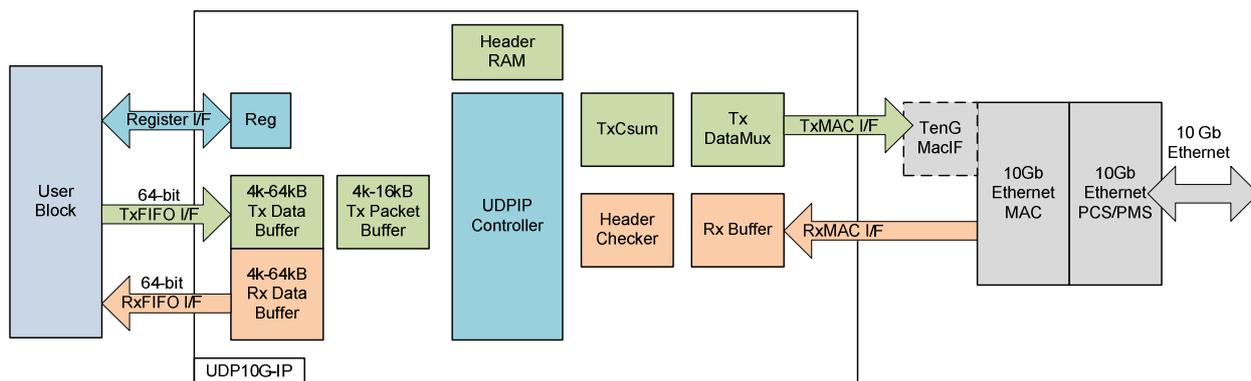


図1: UDP10G-IP コア内部ブロック図

### アプリケーション情報

UDP10G-IP コア(以下本 IP コア)は動画データ・ストリーミングのような UDP/IP プロトコルを使ったネットワーク・アプリケーションにおいて高速でのデータ転送を可能とする機能を提供します。本コアを使うことで、ユーザは CPU を使わずにハードワイヤード・ロジックのみで UDP/IP プロトコルによるデータ転送を可能とします。

### 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、UDP/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、そしてネットワーク・データ転送の物理層として機能します。本 IP コアを使ったシステムにより、UDP/IP プロトコルにてネットワークの外部デバイスとデータ転送を実行できます。

本 IP コアは 3 種類のユーザ・インターフェイスがあり、一つは制御用レジスタ・アクセスのインターフェイスで、他の二つは送信と受信の FIFO インターフェイスです。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 番号等をレジスタ・インターフェイスを介して設定する必要があります。そしてコマンド指示により送信データ・バッファから外部ネットワーク・デバイスへのデータ送信を実行します。また、外部デバイスからの受信データは受信パケットの UDP ヘッダが有効であった場合、UDP データのみを抽出し本 IP コアの受信データ・バッファに格納します。

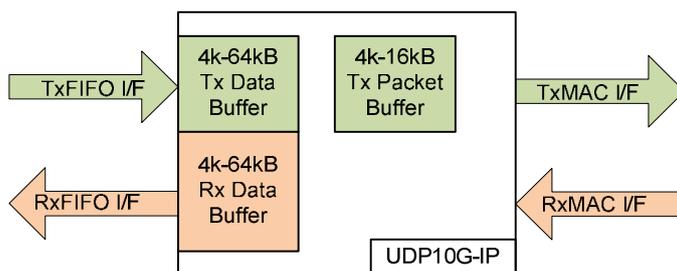
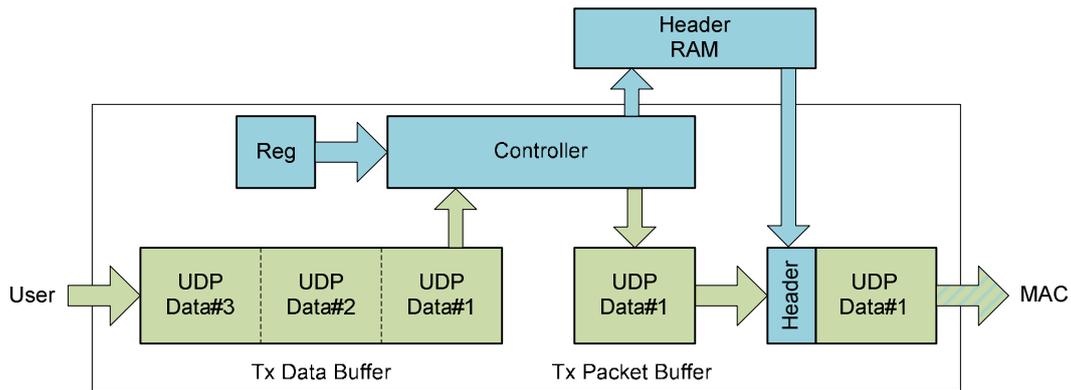


図 2: 送信/受信バッファのサイズは調整可能

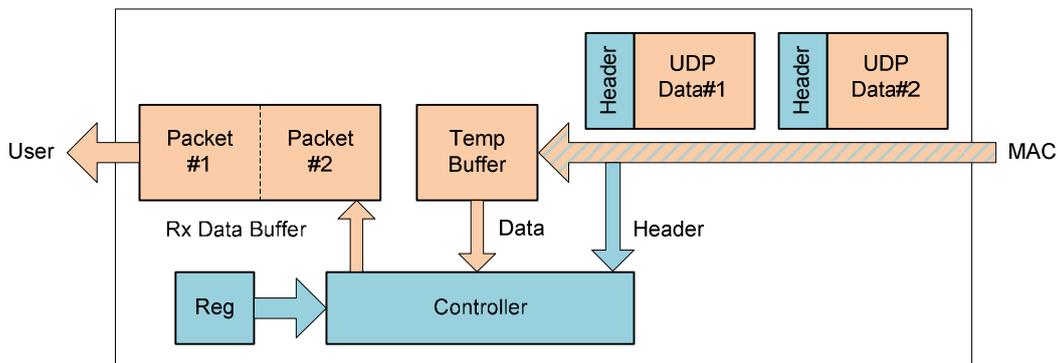
本 IP コアの 3 種類のバッファ(送信データ・バッファ、送信パケット・バッファ、受信データ・バッファ)は IP コアのパラメタライズで設定が可能です。ブロック RAM 消費リソースとパフォーマンスをユーザ・アプリケーションに対して最適化できます。送信データ・バッファと送信パケット・バッファのサイズは、レジスタ・インターフェイスを介して設定可能な送信パケットのサイズに関係します。送信パケット・バッファの容量は必ず送信パケット・サイズよりも大きく設定する必要があります、送信データ・バッファは送信パケット・バッファより少なくとも 2 倍のサイズとする必要があります。

また、パケットの受信とユーザ回路への受信データ転送を同時に行えるようにするため、受信データ・バッファは受信パケット・サイズの 2 倍以上に設定する必要があります。



**図 3: 送信のデータ・フロー**

データの送信時は、送信データ・バッファからのデータはパケット・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。レジスタで設定されたパラメータはパケットのヘッダ作成に使われヘッダ RAM 内に格納されます。送信パケット・バッファからの送信データは MAC へ出力される前にコア内部にてヘッダ RAM のヘッダ・データと結合し UDP パケットが生成されます。UDP チェックサムおよび IP チェックサムはコア内で自動計算されます。レジスタ内のビジー・フラグはユーザによって予め設定された転送サイズ分のデータ転送が完了するとクリアされます。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状態が把握できます。UDP10G-IP コアがアイドル状態に復帰した後ユーザは送信パケット・サイズや総転送長を変更できます。



**図 4: 受信のデータ・フロー**

データの受信時には受信パケットはまずテンポラリ・バッファに一旦格納されます。そして受信パケット内のヘッダとチェックサムが精査され、ヘッダ内容やチェックサムにエラーがあった場合はそのパケットは破棄されるので受信データ・バッファには格納されません。従って有効なデータのみが選別され受信データ・バッファに格納されることとなります。

## 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 UDP/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 ビットです。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。

- UDPIP コントローラ

リセットが解除されると IP コアは ARP 要求を送信し ARP 応答情報から通信ターゲットの MAC アドレスを取得し初期化を完了します。その後コアはアイドル状態となり、ユーザからの外部デバイスへのデータ転送開始指示を待ちます。本 IP コアのコマンドは送信コマンドのみです。

表 3: レジスタ・マップ定義

アドレス [3:0]	レジスタ名	方向	ビット	説明
0000b	RST	Wr /Rd	[0]	本 IP コアのリセット。'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)で、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	[0]	'1'でデータ送信開始 本レジスタによってデータ送信を指示する前に、ユーザ回路は Busy 信号または本レジスタの bit[0]をリードしチェックしてコアが動作中でないことを確認しなくてはならない。
		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、Busy 出力信号と同一
0010b	SML	Wr /Rd	[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 /Rd	[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 /Rd	[31:0]	ターゲット側の 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 /Rd	[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 /Rd	[31:0]	[15:0] 本 IP コアからの送信にて送信先ターゲット側のポート番号を 16bit で指定する。 [31:16] 本 IP コアへの受信にて受信元ターゲット側のポート番号を 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ポート番号を指定する必要がある。
0111b	SPN	Wr /Rd	[15:0]	本 IP コア側のポート番号を 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	TDL	Wr	[31:0]	送信データ数を 8 の倍数バイト単位で指定する。有効な値は 8~0xFFFFFFFF8 (下位 2 ビットは無視される)。CMD レジスタで送信開始を指示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。ユーザが本レジスタでセットした送信データ数はコア内部ロジックでラッチされるため、現在送信中であっても、次の送信のデータ数をセットしておくことが可能である。また、次の送信でも再度同じ送信データ数であ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。

表 3: レジスタ・マップ(続き)

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要求を繰り返す [8] 受信データバッファが一杯のため受信パケットを受け損ねた [9] 受信パケットのチェックサムが間違えていたため受信パケットを破棄した [10] MacRxUserエラーが検出されたため受信パケットを破棄した
1010b	PKL	Wr/Rd	[15:0]	8の倍数バイト単位で指定する送信パケットのデータ長。8~16000の範囲で指定する必要がある(下位2ビットは無視される)。デフォルト値は1472バイト(非ジャンボフレームの最大サイズ)この値はデータ転送(Busyフラグ=1)中に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、コア内部で前の値は保持されているのでユーザ回路は本レジスタを再度セットする必要はない。
1110b	SRV	Wr/Rd	[0]	'0': クライアント・モード、本IPコアはリセット解除後ARP要求を接続相手に向けて送信し、MACアドレスを取得する、相手からのARP応答を受信するとコアのBusyはネゲートする。 '1': サーバー・モード、本IPコアはリセット解除後相手からのARP要求を待機する、ARP要求を受信しARP応答を送信した後にコアのBusyはネゲートする。 このレジスタの初期値は'0'(クライアント・モード)である。

表 4: 各バッファ(TxBuf/TxPac/RxBufBitWidth)の容量パラメータ

有効なビット幅	バッファ容量	送信データ・バッファ有効ビット幅	送信パケット・バッファ有効ビット幅	受信データ・バッファ有効ビット幅
9	4kByte	Valid	Valid	Valid
10	8kByte	Valid	Valid	Valid
11	16kByte	Valid	Valid	Valid
12	32kByte	Valid	No	Valid
13	64kByte	Valid	No	Valid

## 送信ブロック

- **送信データ・バッファ (Tx Data Buffer)**

このデータ・バッファの容量は本 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(4K バイト)~13(64K バイト)で 64 ビット幅バッファのアドレス・サイズに該当します。

このバッファ・サイズは表 3 の PKL レジスタで設定する送信パケット・サイズの少なくとも 2 倍かそれ以上のサイズとする必要があります。送信時は本送信データ・バッファから送信パケット・バッファへ 1 パケット分のデータを転送すると同時にユーザ回路から次のパケット用のデータを受信します。MAC へパケットが出力されると本バッファ内のデータはフラッシュされます。このため現在パケットの送信と次パケットの準備を同時に並行して行うためにパケット・サイズの 2 倍の容量が必要です。このバッファ・サイズを 2 倍より大きくすることによりコア内のユーザ回路からのデータ・フロー制御に貢献します。バッファ内に多数のデータを貯めることで本 IP コアからのデータ送信中にユーザ回路は他の処理を実行できます。

- **送信パケット・バッファ (Tx Packet Buffer)**

このバッファの容量は本 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(4K バイト)~11(16K バイト)です。このバッファ・サイズは PKL レジスタで設定する送信パケット・サイズと同じかそれ以上とする必要があります。送信データ・バッファからの 1 パケット分のデータを格納します。送信パケット・バッファ内のデータは MAC のデータ受信準備が完了するまで保持されます。MAC が長時間データ受信レディとしない場合本バッファは最大 2 データ・パケット分のデータを保持します。このため本バッファにて 2 パケット分以上の残リスペースは使われません。

- **ヘッダ RAM (Header RAM)**

この RAM には送信パケットのヘッダ部を格納します。ユーザ回路から RST レジスタによるリセット解除でヘッダ RAM 内のパラメータは更新されます。いくつかのパラメータ例えばターゲット MAC アドレスは ARP 応答(クライアント・モード)/ARP 要求(サーバー・モード)によって更新されます。

- **送信チェックサム (TxCsum)**

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

- **送信データ・マルチプレクサ (TxDataMux)**

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 (Rx Buffer)**

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。

- **ヘッダ・チェッカ (Header Checker)**

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。ただし IP フラグメントされたパケットの場合 UDP チェックサムは検証されません。チェックサムが適合した場合、UDP データのみが分離され受信データ・バッファに転送されます。

- **受信データ・バッファ (Rx Data Buffer)**

このバッファの容量は本 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(4K バイト)~13(64K バイト)です。このバッファはユーザ・ロジックと本 IP コア間に配置されます。このバッファが一杯の場合、新たに受信したパケットは無視(ロスト)します。このため、新たなパケット受信と前パケットのユーザ回路への転送を同時に実行する場合、少なくとも受信パケット・サイズの 2 倍の容量が必要です。

## 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るので、MicroBlaze などのプロセッサを使わずにシステムを構築することが可能です。

## TenGMacIF

このブロックは本 IP コアと Xilinx 製 10Gb イーサネット MAC コアの送信インターフェイスと接続するためのものです。パケット送信中 Xilinx 10Gb イーサネット MAC コアは tx\_axis\_tready 信号をネゲートすることがあります。ただ本 IP コア内エンジンはパケット送信を連続する必要があります。このためリファレンス・デザインに HDL コードで含まれる本ブロックによって本 IP コアからの連続送信データを一時的に保持し 10Gb イーサネット MAC が送信データの一時停止を要求した場合に対応します。

## 10Gb イーサネット MAC および 10Gb イーサネット PCS/PMA

両ブロックとも Xilinx 社の標準ソフト IP コアです。より詳細については以下の Xilinx 社サイトを参照してください  
[10G Ethernet MACN and PCS/PMA]

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

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

[10G/25G Ethernet Subsystem]

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

## コアの I/O 信号

本 IP コアのパラメータを表 5 に、全 I/O 信号を表 6 で説明します。MAC インターフェイスは 64 ビット AXI4 ストリーム・インターフェイスです。

表 5: コアのパラメータ

ジェネリック名	設定範囲	説明
TxBufBitWidth	9-13	送信データ・バッファ・サイズを 64 ビット幅のデータ・インターフェイスにおけるアドレス・ビット幅で設定します。
TxPacBitWidth	9-11	送信パケット・バッファ・サイズを 64 ビット幅のデータ・インターフェイスにおけるアドレス・ビット幅で設定します。
RxBufBitWidth	9-13	受信データ・バッファ・サイズを 64 ビット幅のデータ・インターフェイスにおけるアドレス・ビット幅で設定します。

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	本 IP コアのリセット: ロウ・アクティブ信号である。
Clk	In	Xilinx ブロックの物理層から出力される 156.25MHz 固定のクロック
<b>ユーザ I/F</b>		
RegAddr[3:0]	In	レジスタの 4bit アドレスバス
RegWrData[31:0]	In	ライト・レジスタの 32bit 書込みデータ・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。
Busy	Out	コアのビジー状態('0':アイドル状態、'1'コアは初期化中またはビジー状態)。
IntOut	Out	タイムアウト発生または受信パケットの破棄時に本信号が 1クロック期間分 H アサートされる。ユーザ回路は TMO レジスタで割り込み要因を確認することができる。
<b>送信 FIFO I/F</b>		
UDPTxFfFull	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。
UDPTxFfWrEn	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。
UDPTxFfWrData[63:0]	In	送信データ・バッファの 64bit 書込みデータ・バス、UDPTxFfWrEn に同期する。
<b>受信 FIFO I/F</b>		
UDPRxFfRdCnt[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
UDPRxFfRastRdCnt[2:0]	Out	受信データ・バッファ内の最終データでの残留バイト数、0-7 のいずれかの値となる。総受信データ・バイト数が 8 の倍数でない場合はこの値は非ゼロとなる。
UDPRxFfRdEmpty	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。
UDPRxFfRdEn	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。
UDPRxFfRdData[63:0]	Out	受信データ・バッファの 64bit 読み出しデータ・バス、UDPRxFfRdEn 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。

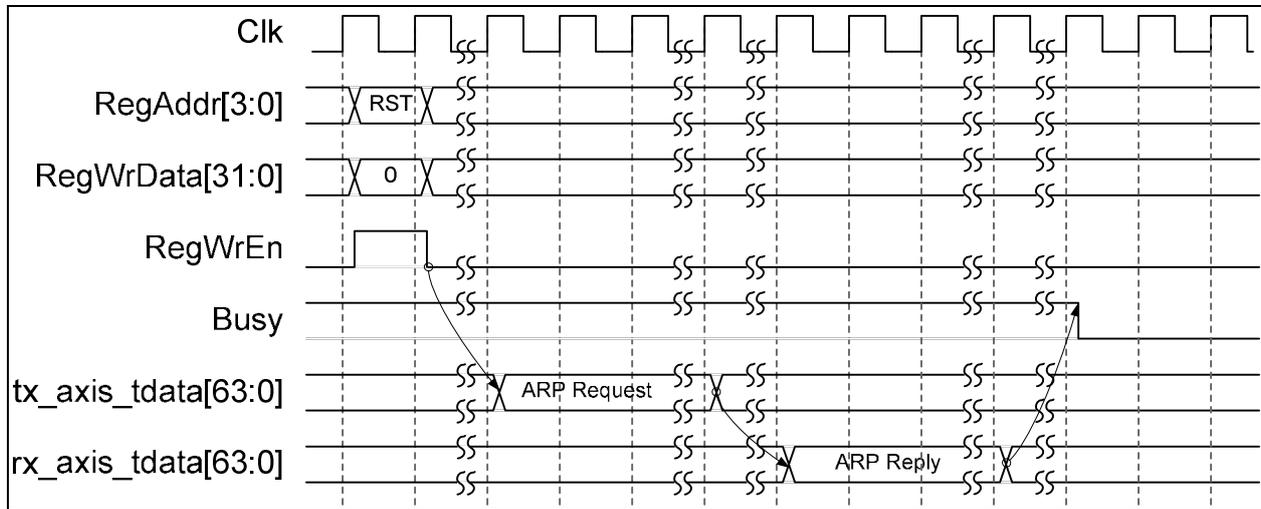
信号名	方向	説明
<b>MAC I/F 信号</b>		
rx_axis_tdata[63:0]	In	受信データ・バス
rx_axis_tvalid	In	受信データ有効信号、rx_axis_tdata 信号に同期。本信号はパケット受信の開始から完了まで連続して'1'にアサートする必要がある。
rx_axis_tlast	In	受信フレーム最終ワードであることを示す信号
rx_axis_tuser	In	受信フレーム最終ワードでフレームにエラーがあったことを示す信号 '1':通常フレーム、'0':エラー・パケット
rx_axis_tready	Out	フレームの最終ワード(rx_axis_tlast='1')受信後に'0'アサートする。この信号は次のパケット・データ受信のため 1 クロック分ネゲートする。
tx_axis_tdata[63:0]	Out	送信データ・バス
tx_axis_tkeep[7:0]	Out	送信データ・バイト・イネーブル、tx_axis_tdata に同期
tx_axis_tvalid	Out	EMAC への送信データ有効信号、tx_axis_tdata に同期
tx_axis_tlast	Out	送信フレームの最終ワードであることを示す制御信号
tx_axis_tuser	Out	エラー状態を示す制御信号、この信号は'0'固定で出力される
tx_axis_tready	In	ハンドシェイク信号。Tx_axis_tdata が受領されたときにアサートされる。この信号は送信パケットの先頭ワードから最終ワードまで連続してアサートされなくてはならない。Xilinx EMAC は本信号をパケット送信中にネゲートするため、本 IP コアと Xilinx EMAC 間に少量のバッファによる接続ロジックを必要とする。

**S**

**タイミング・チャート**

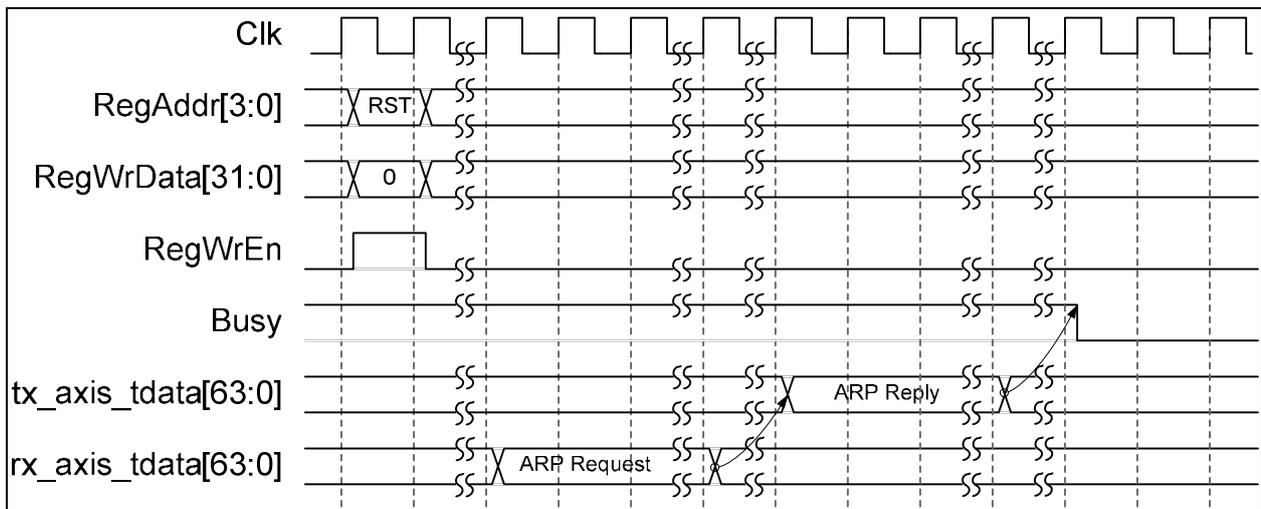
**IP コアの初期化**

ユーザ回路から RST レジスタでリセット状態を解除されると本 IP コアは SRV レジスタの設定値(クライアント・モードまたはサーバー・モード)によって 2 種類のモードで初期化を実行します。



**図 5: クライアント・モードでの 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 MAC アドレスは ARP 応答パケットから抽出されます。その後 Busy 信号は '0' ネゲートされます。

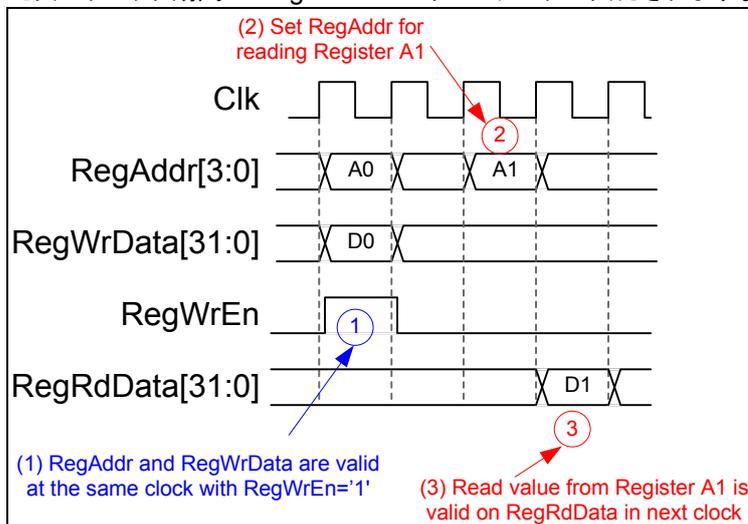


**図 6: サーバー・モードでの IP コア初期化**

サーバー・モードの場合、本 IP コアはリセット解除後ターゲットからの ARP 要求を待ちます。ターゲットとして設定された値と合致したヘッダを含む ARP 要求を受信すると本 IP コアはターゲットに ARP 応答を送信します。ターゲット MAC アドレスは ARP 要求から抽出します。最後に Busy 信号は '0' ネゲートされます。

### レジスタ・インターフェイス

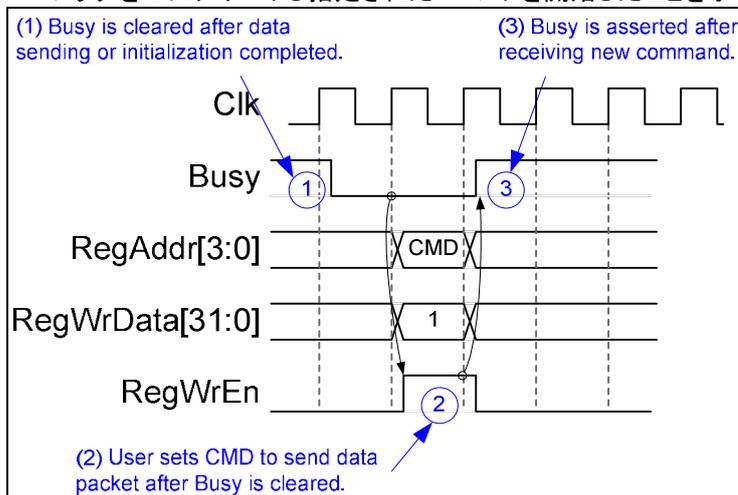
ユーザ回路からの本 IP コア内部へのリードライト・アクセスは図 7 に示すタイミングにより実行します。アクセス先レジスタのアドレスマップは表 3 で示されます。レジスタへの書込みは RegAddr と RegWrData にそれぞれ有効なライト先のアドレスとデータをセットし 1 クロック期間 RegWrEn='1' とします。レジスタからの読み出しの際は、有効な RegAddr がコアに与えられた次のクロック期間に RegRdData にリードデータが出力されます。



- (1) RegAddr と RegWrData は RegWrEn='1' と同じクロック期間で有効とする必要がある
- (2) 読み出し先アドレス A1 を RegAddr にセットする
- (3) レジスタ A1 のリード値は次のクロック期間で RegRdData に出力される

図 7: レジスタ I/F のタイミング・チャート

ユーザ回路は CMD レジスタをセットする前にコアの Busy ピンをモニタするかあるいは CMD レジスタの bit0 をリードすることでビジー・フラグがアサートされていないことを確認する必要があります。CMD レジスタをセットしコマンドを発行すると、図 8 に示すようにコアはビジー・フラグを '1' にアサートし指定されたコマンドを開始したことを示します。

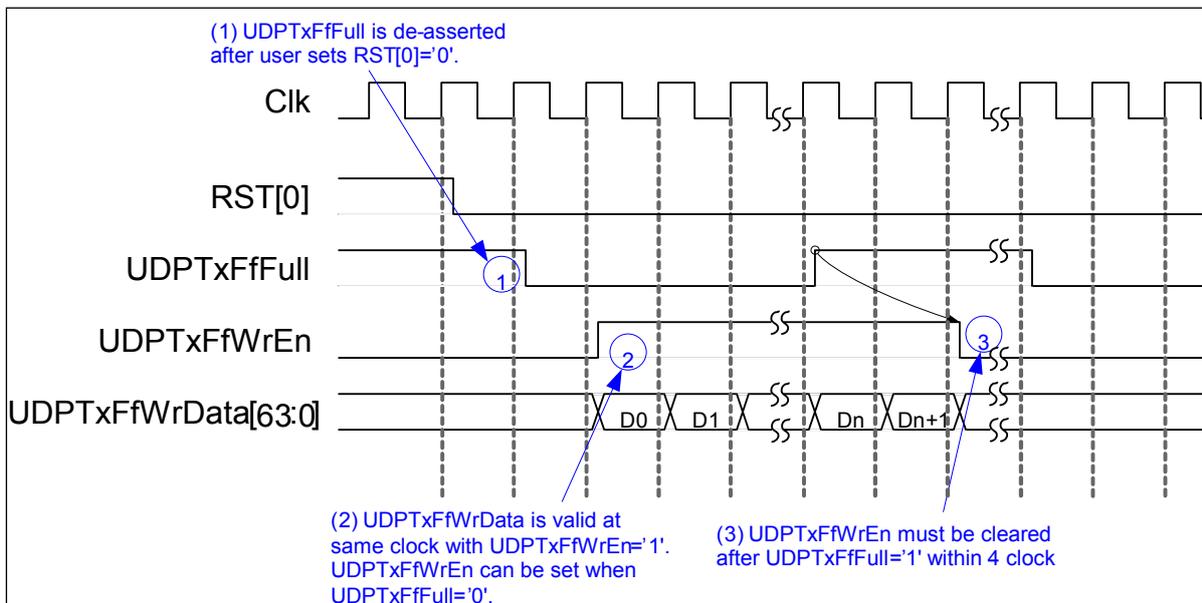


- (1) Busy はコアの初期化完了または前コマンドの完了後にクリアされる
- (2) ユーザ回路は Busy が '0' であることを確認してから CMD レジスタを書き込む
- (3) IP コアがコマンド指示を認識すると Busy をアサートする

図 8: Busy がクリアされた状態からのコマンド発行

**送信 FIFO インターフェイス**

ユーザ回路は本 IP コアに対して図 9 に示すように FIFO インターフェイスでデータを送信できます。データを送る前にユーザ回路は FIFO のフル・フラグ (UDPTxFfFull) をチェックしそれが '1' にアサートされていないことを確認する必要があります。そして書き込みデータの UDPTxFfWrData に同期して書き込みイネーブル信号の UDPTxFfWrEn='1' とします。UDPTxFfFull が '1' となった場合、4 クロック以内に UDPTxFfWrEn による書き込み動作を停止しなくてはなりません。また本 IP コアがリセット状態の場合も UDPTxFfFull はアサートされ、FIFO 内の全データはフラッシュされます。

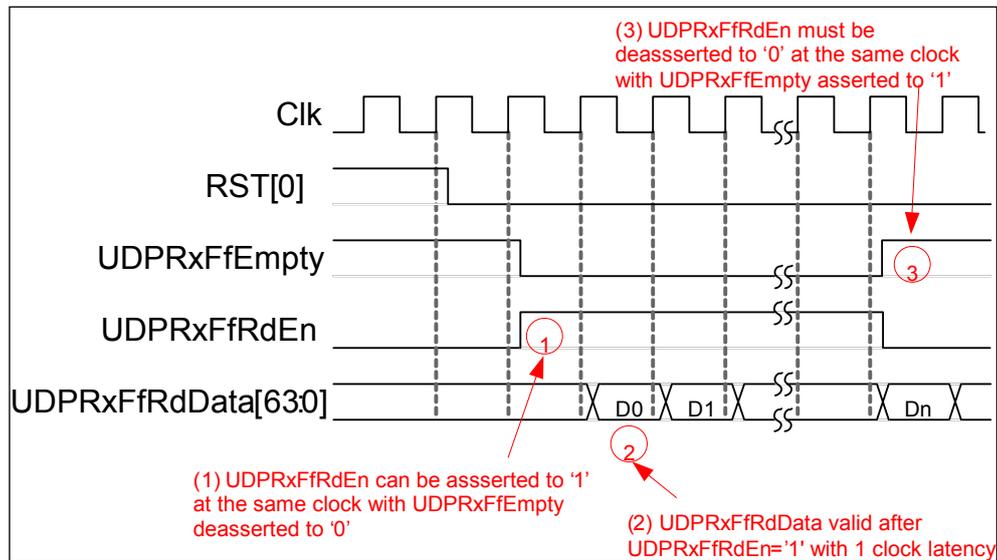


- (1) UDPTxFfFull はユーザ回路から RST[0]='0' とクリアされた後にネゲートされ FIFO データ書き込みが可能となる
- (2) UDPTxFfFull='0' である場合 UDPTxFfWrEn='1' とすることで UDPTxFfWrData を書き込むことができる
- (3) UDPTxFfFull='1' となった場合 4 クロック以内に UDPTxFfWrEn による書き込み動作を停止しなくてはならない

**図 9: 送信データ・バッファ I/F のタイミング・チャート**

### 受信 FIFO インターフェイス

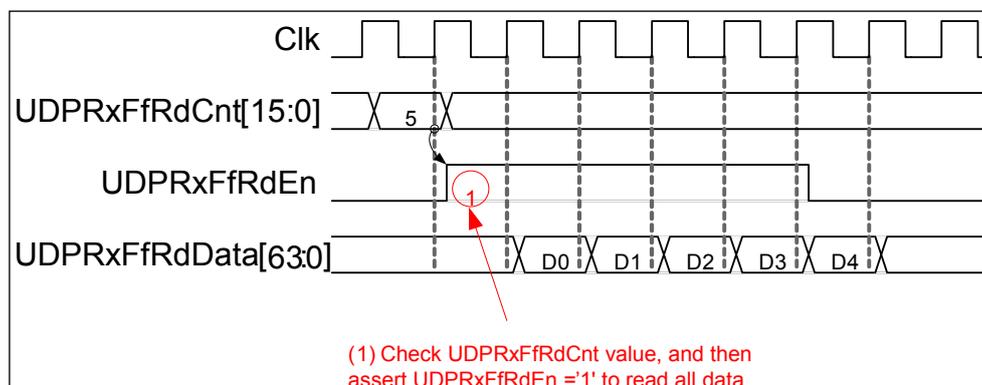
本 IP コアが外部からデータを受信した場合、受信データ・バッファに格納されます。ユーザ回路は図 10 に示すように FIFO インターフェイスでデータを読み出すことができます。ユーザ回路は UDPRxFfEmpty 信号をチェックすることで受信データの格納状態が把握でき、また UDPRxFfEmpty が '0' でない場合に UDPRxFfRdEn をアサートしてその次クロック期間に UDPRxFfRdData で受信データを読み出します。UDPRxFfEmpty が '1' となった場合その同一クロック期間内で UDPRxFfRdEn を '0' にネゲートしデータの読み出しを停止しなくてはなりません。送信データ・バッファと同じように受信データ・バッファも本 IP コアがリセットされると FIFO 内部データをフラッシュします。またコアがリセット中は UDPRxFfEmpty は '1' にアサートされます。



- (1) UDPRxFfRdEn は UDPRxFfEmpty が '0' ネゲートしている同一クロック期間中 '1' にアサートできる
- (2) UDPRxFfRdData は UDPRxFfRdEn が '1' アサートされた次のクロック期間に出力される
- (3) UDPRxFfEmpty が '1' にアサートされた場合その同一クロック期間に UDPRxFfRdEn をネゲートする必要がある

図 10: 受信データ・バッファ I/F のエンpty・フラグについてのタイミング・チャート

受信データ・バッファの状態は UDPRxFfRdCnt をモニターすることでも確認できます。この信号は受信データ・バッファに格納されている全データ数を示します。従って図 11 に示すように総受信データ数と同じ期間 UDPRxFfRdEn を '1' にアサートすることで、全受信データを読み出すことができます。



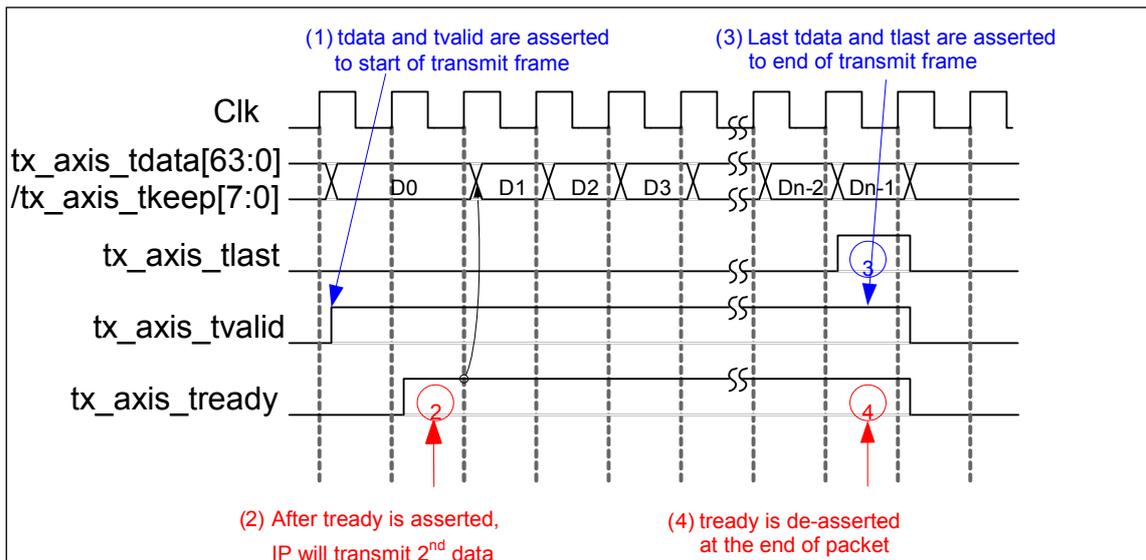
- (1) UDPRxFfRdCnt の値を確認し UDPRxFfRdEn をリード・カウンタ分 '1' アサートする

図 11: 受信データ・バッファ I/F のリード・カウンタについてのタイミング・チャート

## EMAC インターフェイス

パケットを送信する場合本 IP コアはパケット先頭データと合わせて tx\_axis\_tvalid をアサートします。これらの信号は tx\_axis\_tready が '1' アサートされデータ転送要求が認識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の送信まで '1' アサートを維持する必要があります。tx\_axis\_tlast と tx\_axis\_tvalid はパケットの最終ステータスを示すため最後のデータと共に '1' アサートされます。

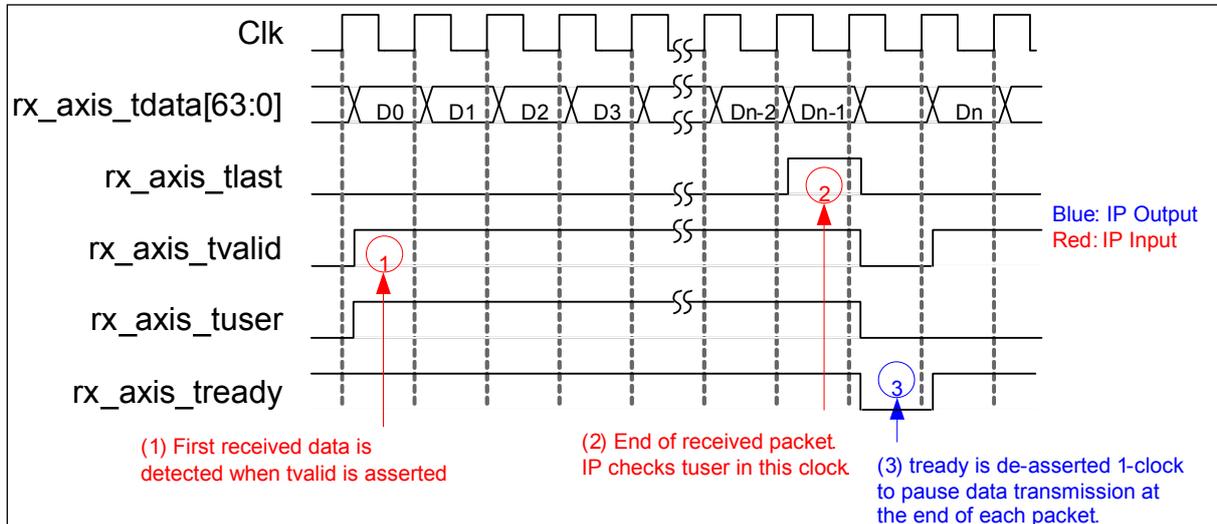
Xilinx 製 10Gb EMAC コアは tvalid が最初にアサートされてから最終データまでの間に tx\_axis\_tready をネゲートすることがあります、このため本 IP コアと Xilinx EMAC 間に少量のバッファを含むアダプタ回路を挿入する必要があります。アダプタ回路の具体的な実装例はリファレンス・デザインにて VHDL ソースコードで提供されます。



- (1) tdata と tvalid がアサートされ送信フレームが開始する
- (2) tready がアサートされると IP コアは 2 番目のデータを送信する
- (3) 転送フレームの末尾で最後の tdata と tlast がアサートされる
- (4) tready はパケットの最後でネゲートする

図 12: EMAC インターフェイスの送信タイミング・チャート

受信側のタイミング・チャートを図 13 に示します。本 IP コアは rx\_axis\_tvalid が '0' から '1' に遷移する受信フレームの開始をモニタします。rx\_axis\_tdata は rx\_axis\_tlast がアサートされ rx\_axis\_tuser が有効な値でパケットの末尾となるまで連続して受信されます。rx\_axis\_tvalid は最初にアサートされてから受信パケットの最後 tlast まで連続して '1' アサートする必要があります。rx\_axis\_tready は各パケット終了後 1 クロック分 '0' ネゲートし、データ転送を一時停止します。



- (1) tvalid がアサートされ先頭受信データを検出する
- (2) IP コアが最終データをリードするためバス上の信号がラッチされる
- (3) tready は各パケット終了後 1 クロック期間ネゲートされデータ転送を一時停止する

図 13: EMAC インターフェイスの受信タイミング・チャート

## コアの制御例

### クライアント・モード(SRV[0]='0')

クライアント・モードでのデータ送信/受信における本 IP コア内レジスタの設定シーケンス例を以下に示します。

- 1) RST レジスタを'1'にセットし本 IP コアをリセット状態とします。
- 2) SML/SMH レジスタで自身の MAC アドレス、DIP/SIP レジスタで相手/自分の IP アドレス、DPN/SPN レジスタでポート番号を設定します。
- 3) RST レジスタ='0'としてコアのリセット状態を解除します、すると UDP10G-IP コアは通信ターゲットに向けて ARP 要求を送信し ARP 応答から MAC アドレス情報を抽出する初期化を開始します。初期化が完了すると Busi 信号は'0'クリアされその後のユーザ回路からのコマンド発行が可能となります。
- 4) a) データ送信の場合、TDL レジスタに総転送数(8 の倍数で設定)、PKL にパケット・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。ユーザ回路は送信 FIFO へ送信データを全て転送し Busy フラグが'0'ネゲートするのを監視します。現在の送信コマンドが完了した後に、次の送信コマンドに向けユーザ回路は IP コアをリセットすることなく、TDL/PKL の値を更新できます。  
b) データ受信の場合、ユーザ回路は受信 FIFO ステータスを監視し受信 FIFO が空の状態であれば受信データをリードします。

### サーバー・モード(SRV[0]='1')

サーバー・モードにおいてはクライアント・モードとの違いは通信ターゲットの MAC アドレスを獲得する初期化プロセスにあります。クライアント・モードでは MAC アドレスは UDP10G-IP コアが送信した ARP 要求後に受信した ARP 応答から抽出します。一方サーバー・モードでは MAC アドレスはターゲット IP アドレスに合致した相手からの ARP 要求での送信元 MAC アドレスから抽出します。初期化後のデータ送信および受信についてはクライアント・モードと同一です。サーバー・モードでの動作シーケンス例を以下に示します。

- 1) RST レジスタを'1'にセットし本 IP コアをリセット状態とします。
- 2) SML/SMH レジスタで自身の MAC アドレス、DIP/SIP レジスタで相手/自分の IP アドレス、DPN/SPN レジスタでポート番号を設定します。
- 3) RST レジスタ='0'としてコアのリセット状態を解除します。UDP10G-IP コアは相手 MAC アドレスを獲得するため ARP 要求を待ちます、そして相手からの ARP 応答を返送します。この初期化が完了すると Busy フラグが'0'ネゲートされます。
- 4) データ送受信はクライアント・モードと同一です。

## コアの検証方法

本 IP コア製品には Xilinx 純正の評価ボードで実機動作する Vivado リファレンス・デザイン・プロジェクトが同梱されているため、実ボードでの動作確認が可能です。また、ドキュメントで示されていない細かい信号タイミング等については、リファレンス・デザインに ChipScope を追加して実機動作させることで、実波形を観測・確認することが可能です。

## 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、Xilinx 社の 10Gb MAC-IP コアも別途必要となりますのでご注意ください。

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・UDP プロトコル知識および Vivado ツール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・ファミリに応じて現在以下のコアのラインナップが用意されています。それ以外のファミリに対応した本 IP コアにつきましては DesignGateway 社までお問い合わせください。

表 6: コアのラインナップ

コア型番	対応ファミリ	Vivado 環境	検証用評価ボード	説明
UDP10G-IP-KU	Kintex UltraScale	Vivado2017.4 又はそれ以降	KCU105	Kintex UltraScale 対応 UDP10G-IP コア
UDP10G-IP-ZUP	Zynq UltraScale+	Vivado2017.4 又はそれ以降	ZCU102	Zynq UltraScale+対応 UDP10G-IP コア
UDP10G-IP-ZQ7	Zynq-7000	Vivado2017.4 又はそれ以降	ZC706	Zynq-7000 対応 UDP10G-IP コア
UDP10G-IP-VT7	Virtex-7	Vivado2017.4 又はそれ以降	VC707	Virtex-7 対応 UDP10G-IP コア
UDP10G-IP-KT7	Kintex-7	Vivado2017.4 又はそれ以降	KC705	Kintex-7 対応 UDP10G-IP コア

## 更新履歴

リビジョン	日時	説明
1.0	Aug-15-2017	New release
1.0J	2017/11/13	日本語初期版を作成
1.1J	2020/07/01	Zynq UltraScale+ファミリに対応