



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
- URL: 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)
- 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
- クロック周波数 125MHz の単一クロック・ドメイン
- Intel 標準 MAC(IP-TRIETHERNET)と接続する 8ビット Avalon ストリーム・インターフェイス
- CycloneV E/ArriaV GX/Arria 10 SoC 各 Intel 評価ボードで実機動作するリファレンス・デザインを提供
- IP フラグメントをサポート
- 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	● CycloneV E/ArriaV GX/Arria10 SoC ボード用参照デザイン
シミュレーション環境	
ModelSim-Altera 10.1e	
技術サポート	
デザイン・ゲートウェイによる日本語サポート	

表 1: コンパイル結果例

Family	Example Device	Fmax (MHz)	ALMs	Registers ¹	Pin	Block Memory bit ²	Design Tools
StratixIV GX	EP4SGX230KF40C2	125	1,133	1,452	-	1,181,696	QuartusII 16.0
CycloneV E	5CEFA7F3117	125	1,023	1,656	-	1,181,696	QuartusII 16.0
ArriaV GX	5AGXFB3H4F35C5	125	1,038	1,659	-	1,181,696	QuartusII 16.0
Arria10 SX	10AS066N3F40E2SGE2	125	1,051	1,749	-	1,181,696	QuartusII 16.0

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) このサンプルはコアの全 I/O とクロックがチップ外部と直接インターフェイスするケースでのコンパイル結果となります。
- 3) 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、2K バイト、2K バイトです。

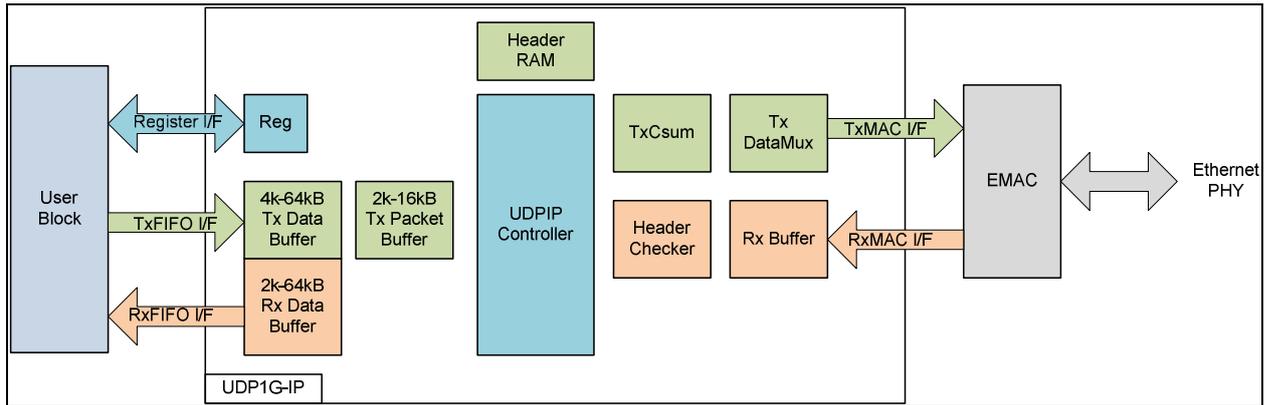


図1：UDP-IPコア内部ブロック図

コア概要

本 UDP1G-IP コアは UDP/IP プロトコルを使った GbE 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 や外部メモリ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。

本コアは Intel 製 MAC コア(IP-TRIETHERNET) と組み合わせることで UDP/IP スタック、トランスポート層、インターネット層、リンク層として機能します。本コアを使ったシステムと外部 PHY チップにより、UDP/IP プロトコルにてネットワークのどのようなデバイスともデータ転送を実行できます。

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 UDP-IP コアの受信データ・バッファに格納されます。

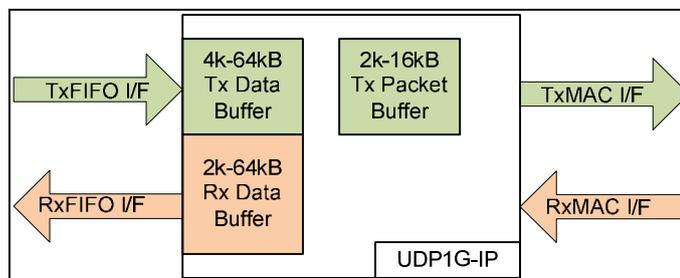


図2：バッファ・サイズ可変の送信データ/送信パケット/受信データ各バッファ

UDP1G-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。送信パケット・バッファ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、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、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。

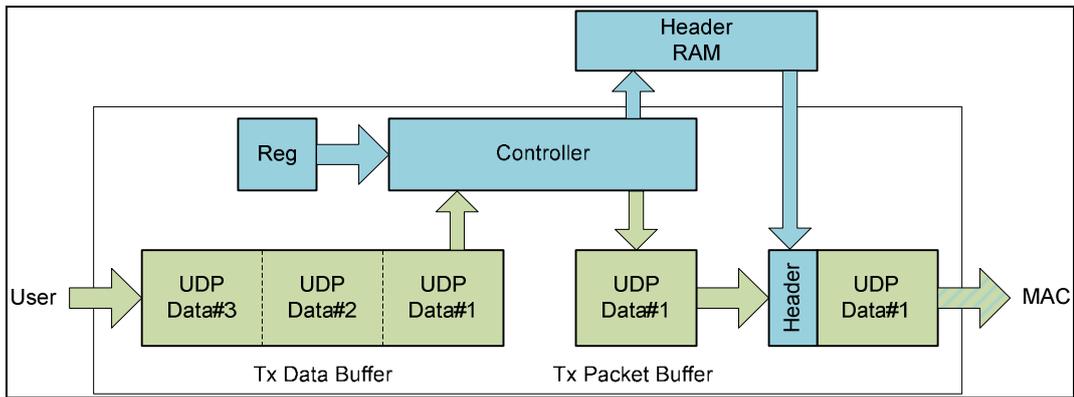


図 3: 送信のデータ・フロー

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 および IP チェックサムはコア内で自動計算されます。レジスタ内のビジイー・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イー・フラグをモニタすることで転送状態が把握できます。IP コアがアイドル状態で送信を実行中でない場合、ユーザ回路は次の送信用として送信パケット・サイズや転送サイズをレジスタ・インターフェイス経由で変更することができます。

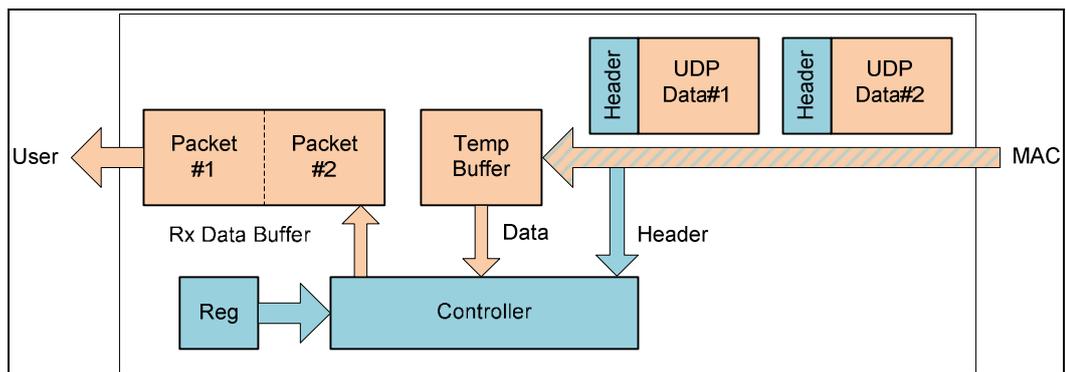


図 4: 受信のデータ・フロー

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になります。

コアの機能ブロック

本 UDP1G-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 2 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

IP コアがリセットから解除('RST レジスタが'1'から'0'へ遷移)されると IP コアはユーザがレジスタ経由で設定した 2 つのモードすなわちサーバー・モードまたはクライアント・モードに従って初期化プロセスを開始します。クライアント・モードでは 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 アドレスを取得します。サーバー・モードでは IP コアは通信ターゲットからの ARP 要求を待機し ARP 応答を返送することで初期化プロセスを完了します。初期化プロセス完了後、コアはアイドル状態となりユーザからのデータ送信指示を待機し、また IP コア自身が受信対象である受信パケットからデータを抽出します。

表 2: レジスタ・マップ

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)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	[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] 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] 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	TDL	Wr	[31:0]	送信データ数をバイト単位で指定する。有効な値は 1~0xFFFFFFFF。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 2: 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 125MHz のカウンタで動作するためタイマ設定値は 8ns 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 [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]	バイト単位で指定する送信パケットのデータ長。1~16000 の範囲で指定する必要がある。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。
1011b	SRV	Wr /Rd	[0]	‘0’:クライアント・モード、IP コアはリセット解除後通信ターゲットに対して ARP 要求を送信し MAC アドレス情報を取得する、ARP 応答を受信すると IP コアのビジーは‘0’ネゲートする。 ‘1’:サーバー・モード、IP コアはリセット解除後通信ターゲットからの ARP 要求を待機する、ARP 要求を受信すると相手の MAC アドレスを抽出し ARP 応答を返送してビジーを‘0’ネゲートする。 デフォルト値=‘0’(クライアント・モード) RST レジスタをクリアする前に本レジスタでサーバー/クライアントのモードを指定する必要がある。また、サーバー・モードの場合は初期化を完了させるためには通信ターゲット側から ARP 要求を行う必要がある。

表 3: 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
11	2kByte	No	Valid	Valid
12	4kByte	Valid	Valid	Valid
13	8kByte	Valid	Valid	Valid
14	16kByte	Valid	Valid	Valid
15	32kByte	Valid	No	Valid
16	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 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 3 に示すように 12(4K バイト)~16(64K 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。この送信データ・バッファのサイズを送信パケット・バッファ・サイズの 2 倍より大きくし十分な容量を確保することで、EMAC へパケットを連続的に送信できます。このバッファは送信時にはユーザ回路と UDP1G-IP コア間のデータ・バッファとしても機能します。

- **送信パケット・バッファ (Tx Packet Buffer)**

このバッファの容量は 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 3 に示すように 11(2K バイト)~14(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+4 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 1 パケット・データを分割します(送信パケット・サイズは PKL レジスタで設定されます)。送信パケット・バッファは 1-2 パケット分のデータを保持します。本送信パケット・バッファのサイズを送信パケット・サイズの 2 倍より大きくしても利点はありません。

- **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。ヘッダ部はユーザ回路によりレジスタ・インターフェイスを介して設定されたネットワーク・パラメータや TxCsum モジュールで計算されたパケットのチェック・サムから構成されます。

- **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。チェックサムの計算が終わるとチェックサム出力はヘッダ RAM へロードされ、送信パケット内のヘッダとなります。

- **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 からのヘッダと送信パケット・バッファ内のデータが結合されます。ヘッダをとデータを含むイーサネット・パケットは EMAC 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 EMAC から送られる受信パケットを一時的に保持します。このバッファの目的は受信データ・バッファへと転送する前にヘッダ・チェックが受信パケットの有効性をチェックするためのものです。有効なパケットの UDP データのみが抽出され受信データ・バッファへと転送されます。

- **ヘッダ・チェック (Header Checker)**

受信パケット内のヘッダにはネットワーク・パラメータとチェックサム値が含まれます。受信したパケットのネットワーク・パラメータが設定された値と異なるか、チェックサムが誤っていた場合受信パケットは破棄されます。

- **受信データ・バッファ (Rx Data Buffer)**

このバッファの容量は 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 3 に示すように 11(2K バイト)~16(64K バイト)です。このバッファはユーザ・ロジックと UDP1G-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、NiosII 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。

トリプル・スピード・イーサネット MAC コア

このモジュールは Intel 社から提供されるソフト IP コアです。詳細については以下の Intel 社サイトを参照してください。

<https://www.altera.co.jp/products/intellectual-property/ip/interface-protocols/m-alt-ethernet-mac.html>

コアの I/O 信号

コアのパラメータを表 4 に、全 I/O 信号を表 5 で説明します。MAC インターフェイスは Intel 製 MAC コアの Avalon ストリーム・インターフェイスと直結できます。

表 4: コアのパラメータ

ジェネリック名	設定範囲	説明
TxBufBitWidth	12-16	送信データ・バッファ・サイズをアドレス・ビット幅で設定します。例えば 12 の場合 4K バイト、16 の場合 64K バイトとなります。
TxPacBitWidth	11-14	送信パケット・バッファ・サイズをアドレス・ビット幅で設定します。例えば 11 の場合 2K バイト、14 の場合 16K バイトとなります。
RxBufBitWidth	11-16	受信データ・バッファ・サイズをアドレス・ビット幅で設定します。例えば 11 の場合 2K バイト、16 の場合 64K バイトとなります。

表 5: コアの I/O 信号

信号名	方向	クロック	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		125 MHz 固定のユーザ I/F および 1Gbps モードでの MAC 通信 I/F 用クロック
ユーザ I/F			
RegAddr[3:0]	In	Clk	レジスタの 4bit アドレスバス
RegWrData[31:0]	In	Clk	ライト・レジスタの 32bit 書込みデータ・バス
RegWrEn	In	Clk	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	Clk	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	Clk	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	Clk	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[6:0]レジスタで割り込み要因を確認することができる。
送信データ・バッファ I/F			
UDPTxFfFull	Out	Clk	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	Clk	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[7:0]	In	Clk	送信データ・バッファの 32bit 書込みデータ・バス、UDPTxFfWrEn に同期する。
受信データ・バッファ I/F			
UDPRxFfRdCnt[15:0]	Out	Clk	受信データ・バッファ内の受信データ総量を示す FIFO データ・カウンタ
UDPRxFfRdEmpty	Out	Clk	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	Clk	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[7:0]	Out	Clk	受信データ・バッファの 32bit 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。

表 5: コアの I/O 信号(続き)

信号名	方向	クロック	説明
MAC I/F 信号			
MacTxSOP	Out	Clk	MAC へのパケット送信開始信号、フレームの先頭バイトが MacTxData から出力されるタイミングで'1'アサートされる。
MacTxData[7:0]	Out	Clk	MAC への送信データ・バス
MacTxEOP	Out	Clk	MAC へのパケット送信終了信号、フレームの最終バイトが MacTxData から出力されるタイミングで'1'アサートされる。
MacTxValid	Out	Clk	MAC への送信データ有効信号、MacTxSOP, MacTxData, MacTxEOP が有効なタイミングで'1'アサートされる。
MacTxReady	In	Clk	MAC からのデータ送信レディ信号、MAC がデータを受信できる状態のときに'1'アサートされる。MacTxReady はパケット送信中には'0'ネゲートされてはならない。
MacRxCik	In		MAC からの受信同期クロック
MacRxSOP	In	MacRxCik	MAC からのパケット受信開始信号、この信号は本 IP コアでは使われていない。
MacRxData[7:0]	In	MacRxCik	MAC からの受信データ・バス
MacRxEOP	In	MacRxCik	MAC からのパケット受信終了信号、フレームの最終バイトが MacRxData 上に出力されるタイミングで'1'アサートされる。
MacRxValid	In	MacRxCik	MAC からの受信データ有効信号、MacRxSOP, MacRxData, MacRxEOP が有効なタイミングで'1'アサートされる。1 パケットの転送中は MacRxValid は連続して'1'アサートされなくてはならない。
MacRxError	In	MacRxCik	MAC からの受信エラー信号、フレーム内にエラーが含まれていた場合にフレーム最終バイトのタイミングで'1'アサートされる。
MacRxReady	Out	RxCik	本 IP コアが MAC からのデータを受信できる状態で'1'アサートされる。この MacRxReady は各受信パケット間のギャップが 3 クロック期間以下の場合に'0'ネゲートされる。詳細については図 13 および図 14 を参照のこと。

タイミング・チャート

IP コアの初期化

ユーザ回路からの RST レジスタ値を '1' から '0' に変更した後の初期化処理において UDP1G-IP コアは SRV レジスタの設定値に従って 2 種類のモードすなわちクライアント・モードかサーバー・モードで動作します。

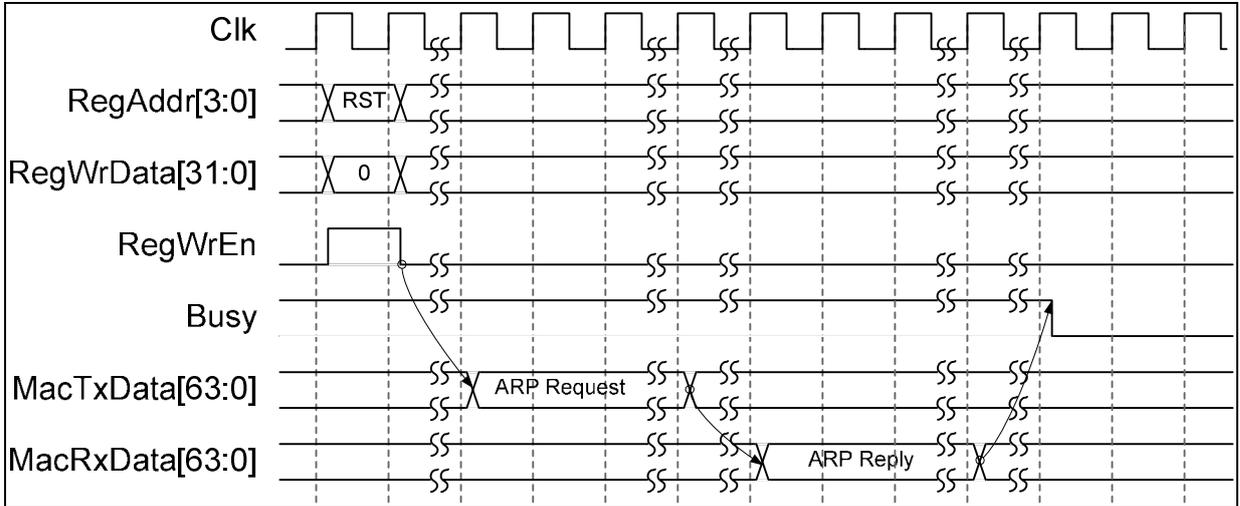


図 5: クライアント・モードの初期化タイミング・チャート

クライアント・モードの場合 UDP1G-IP コアは ARP 要求を発行し通信ターゲットからの ARP 応答を待機します。返送された ARP 応答パケットから通信ターゲットの MAC アドレスを取得します。そしてビジー信号が '0' にネゲートします。

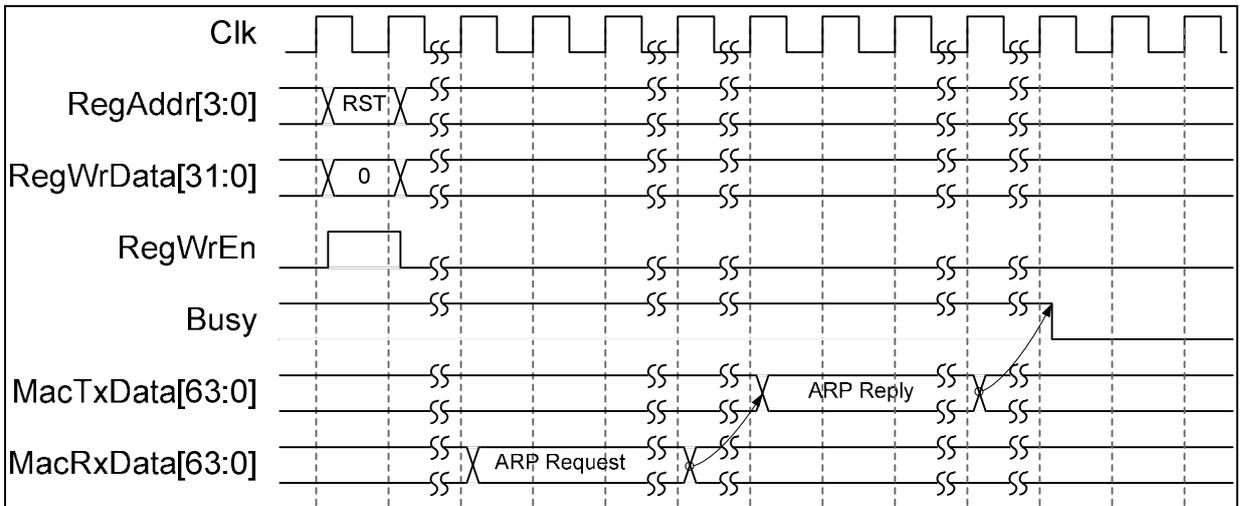
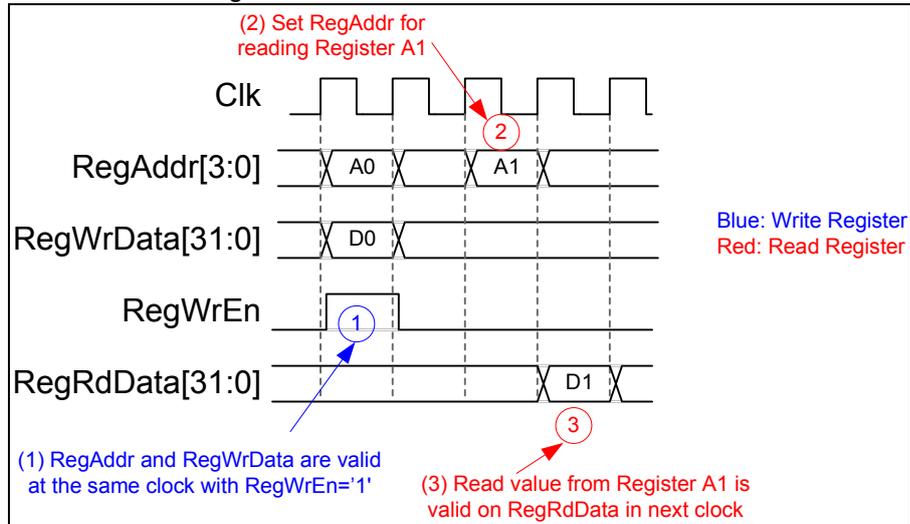


図 6: サーバー・モードの初期化タイミング・チャート

サーバー・モードの場合 UDP1G-IP コアのリセットが '0' リリースされるとコアは通信ターゲットからの ARP 要求を待機します。パケット・ヘッダが設定されたパラメータと合致する ARP 要求パケットを受信すると UDP1G-IP コアはターゲットに ARP 応答パケットを返送します。ARP 要求パケットからはターゲットの MAC アドレス情報が抽出されます。最後にビジー信号が '0' にネゲートします。

レジスタ・インターフェイス

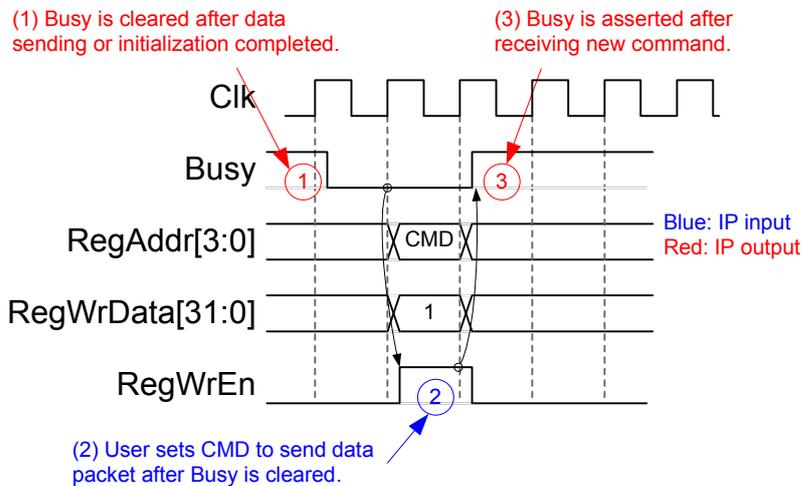
ユーザ回路からの UDP1G-IP コア内部コアへのリードライト・アクセスは図 7 に示すタイミングにより実行します。アクセス先レジスタのアドレスマップは表 2 で示されます。レジスタへの書き込みは RegAddr と RegWrData にそれぞれ有効なライト先のアドレスとデータをセットし 1 クロック期間 RegWrEn='1' とします。レジスタからの読み出しの際は、有効な RegAddr がコアに与えられた次のクロック期間に RegRdData にリードデータが出力されます。



- (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。

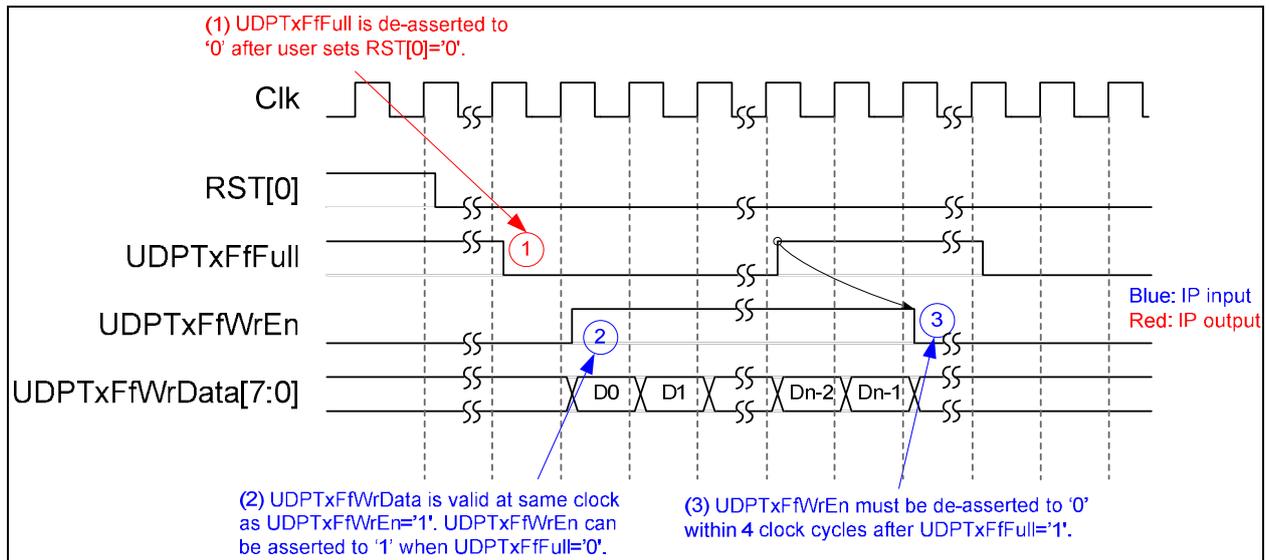


- (1) Busy はコアの初期化完了または前コマンドの完了後にクリアされる
- (2) ユーザ回路は Busy が '0' であることを確認してから CMD レジスタを書き込む
- (3) IP コアがコマンド指示を認識すると Busy をアサートする

図 8: ビジーがネゲート状態で CMD レジスタをセットする

送信 FIFO インターフェイス

ユーザ回路は 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 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。

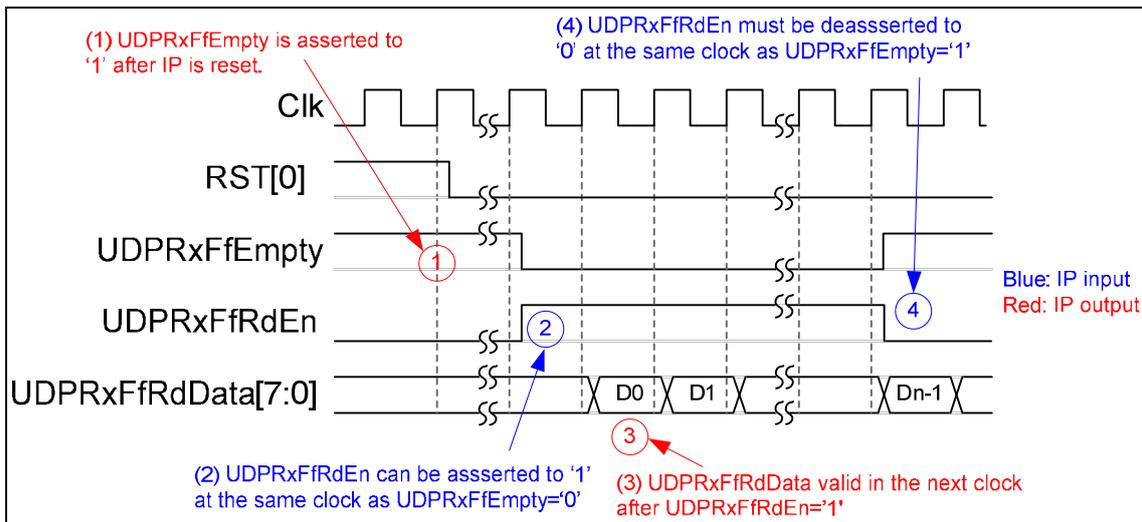


- (1) UDPTxFfFull はユーザ回路から RST[0]='0' とクリアされた後にネゲートされ FIFO データ書き込みが可能となる
- (2) UDPTxFfFull='0' である場合 UDPTxFfWrEn='1' とすることで UDPTxFfWrData を書き込むことができる
- (3) UDPTxFfFull='1' となった場合 4 クロック以内に UDPTxFfWrEn による書き込み動作を停止しなくてはならない

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

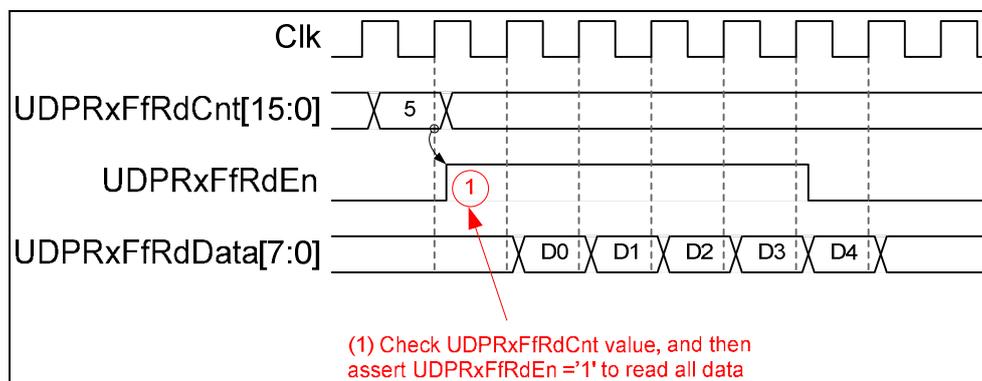
UDP-IP コアが外部からデータを受信した場合、受信データ・バッファに格納されます。ユーザ回路は図 10 に示すように FIFO インターフェイスでデータを読み出すことができます。ユーザ回路は UDPRxFfEmpty 信号をチェックすることで受信データの格納状態が把握でき、また UDPRxFfEmpty が '0' でない場合に UDPRxFfRdEn をアサートしてその次クロック期間に UDPRxFfRdData で受信データを読み出します。UDPRxFfEmpty が '1' となった場合その同一クロック期間内で UDPRxFfRdEn を '0' にネゲートしデータの読み出しを停止し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 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) IP コアがリセット中は UDPRxFfEmpty は '1' アサートされる
- (2) 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
- (3) 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
- (4) 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。

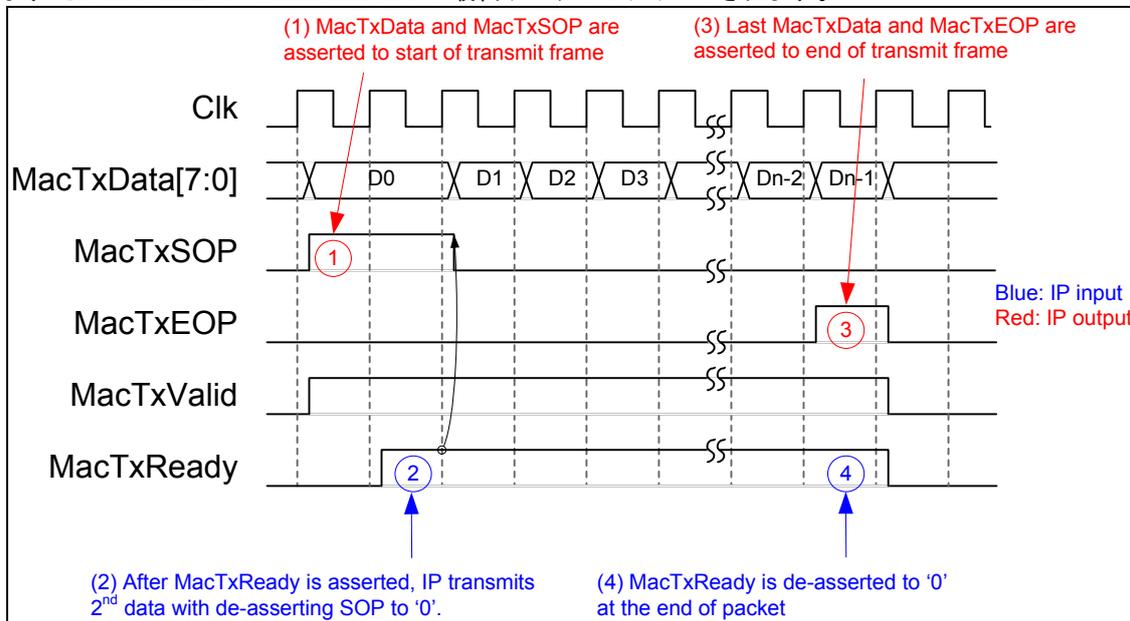


- (1) UDPRxFfRdCnt の値を確認し UDPRxFfRdEn をリード・カウンタ分 '1' アサートする

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

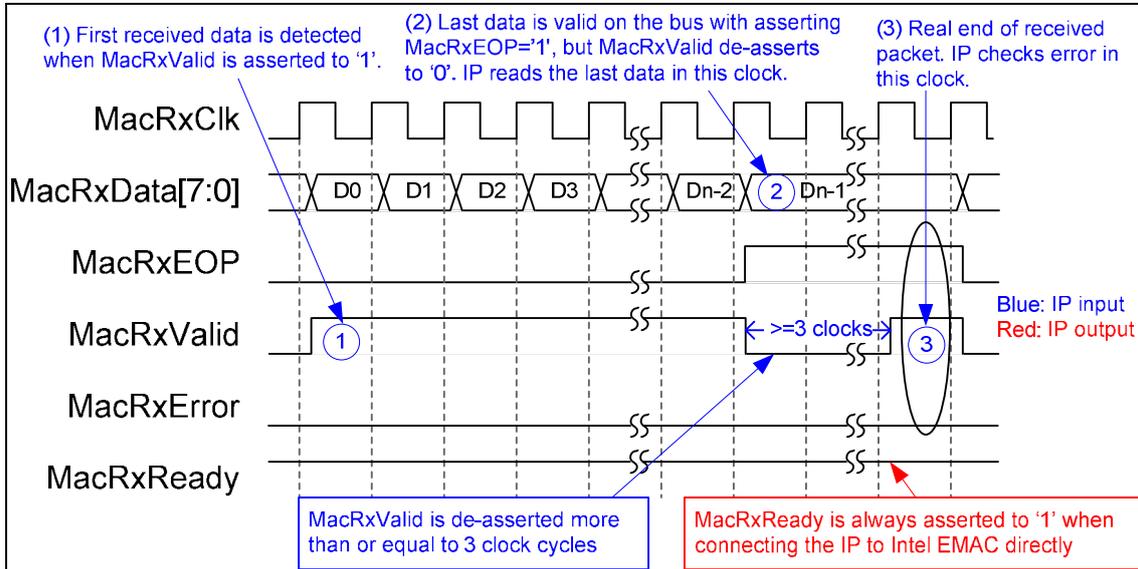
パケットを送信するときに IP コアは MacTxSOP と MacTxValid をアサートし同時に MacTxData 上にパケットの先頭データを出力します。その出力信号はデータ転送要求の応答を示す EMAC からの MacTxReady が '1' アサートされるまで保持します。MacTxReady はパケット転送が完了するまで '1' アサートを継続する必要があります。つまり MacTxValid はパケット最終データまで連続して MacTxData が出力されるよう '1' アサートを維持します。パケットの末尾を示すため MacTxEOP と MacTxValid が最終データで '1' アサートされます。



- (1) MacTxData と MacTxSOP がアサートされフレームの転送が開始する
- (2) MacTxReady がアサートされると IP コアは 2 番目のデータを出力し MacTxSOP はネゲートする
- (3) 送信フレームの最後で MacTxData と MacTxEOP がアサートする
- (4) MacTxReady はパケットの末尾でネゲートする

図 12: MAC I/F 送信側のタイミング・チャート

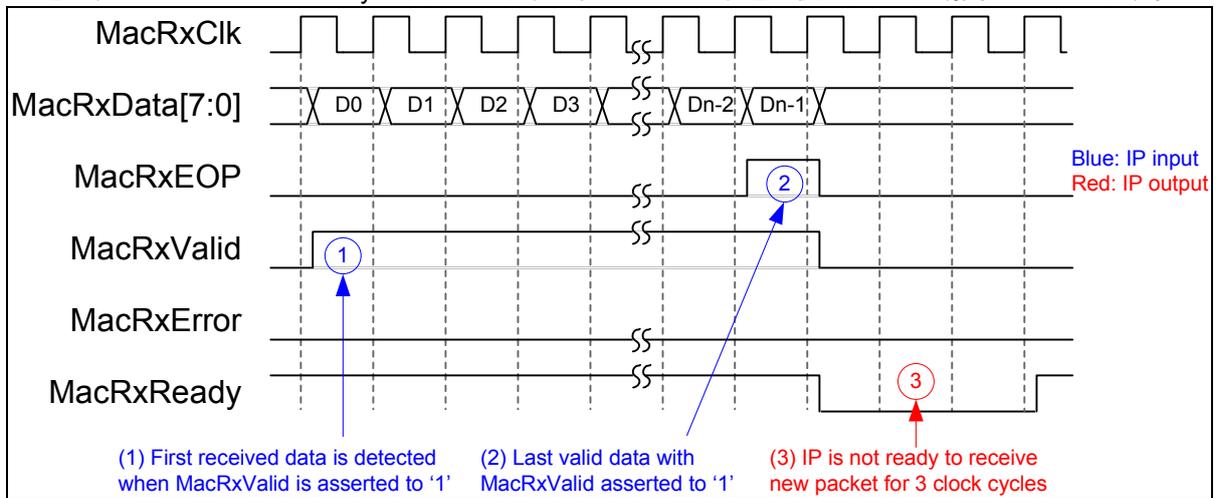
図 13 は受信側のタイミング・チャートです。UDP1G-IP コアは MacRxValid が'0'から'1'アサートする受信フレームの開始をモニタします。MacRxData は MacRxValid='0'かつ MacRxEOP='1'で転送されるパケット最終データまでを連続して受信します。ただ実際のフレーム終了は MacRxEOP='1'かつ MacRxValid='1'です。本当のフレーム終了のタイミングにおいては MacRxError が読み出されパケット状態を評価します。有効な最終データが出力されてから本当のフレーム終了までのクロック期間すなわち MacRxValid のギャップ期間が 3 クロックまたはそれ以上の場合、MacRxReady は常に'1'アサートされます。



- (1) MacRxValid がアサートされると受信パケットの先頭データとして検出する
- (2) MacRxEOP='1'で有効な最終データが現れるが MacRxValid はネゲート状態, IP コアはこの時点でデータを取得する
- (3) 本当のフレーム終了タイミング、ここで IP コアはエラーの有無をチェックする
MacRxValid は 3 クロックまたはそれ以上の期間ネゲートする
IP コアが Intel EMAC と直結する場合 MacRxReady は常に'1'アサートされる

図 13: Intel EMAC 直結時の MAC I/F 受信側のタイミング・チャート

ユーザのアプリケーションによっては UDP1G-IP コアが Intel EMAC と直結せず、Avalon-ST バスを介して他のモジュールと接続される場合があります。この場合 MacRxValid はパケット末尾まで連続して '1' アサートすることができます。UDP1G-IP コアは各受信パケットごとに必ず 3 クロック期間またはそれ以上のギャップが必要とされることに注意してください、このため MacRxReady は図 14 に示すようにフレーム末尾の後 3 クロック期間ネグートします。



- (1) MacRxValid がアサートされると受信パケットの先頭データとして検出する
- (2) MacRxEOP='1'かつ MacRxValid='1'で有効な最終データが現れる
- (3) IP コアは(前パケット受信完了後)3 クロック期間は次のパケットを受信できない

図 14: Avalon-ST バス経由で EMAC を接続する場合の MAC I/F 受信側のタイミング・チャート

コアの使用例

クライアント・モード(SRV[0]='0')

クライアント・モードにおけるデータ送信および受信でのレジスタ設定シーケンス例を以下に示します。

- 1) RSTレジスタ='1'としてIPコアを強制リセット状態とします。
- 2) 自身のMACアドレスをSML/SMHへ、IPアドレスをDIP/SIPへ、ポート番号をDPN/SPNへ設定します。
- 3) RSTレジスタ='0'としてリセットを解除します、するとIPコアは初期化を開始しARP要求パケット送ることによって接続相手のMACアドレスをARP応答パケットから取得します。初期化が完了するとビジー信号は'0'ネゲートします。
- 4) a. データ送信の場合、TDLレジスタに総転送バイト数をセットしPKLレジスタにパケット・サイズをセットします。次にCMDレジスタでデータ送信開始を指示します。送信データは送信FIFOへと転送されます。ビジー・フラグはがモニタされ'0'ネゲートされるのを待機します。次の転送においてはユーザ回路はTDL/PKLレジスタの設定やCMDレジスタでの送信指示はIPコアをリセットせずに実行可能です。
b. データ受信の場合、ユーザ回路は受信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リードします。

サーバー・モード(SRV[0]='1')

クライアント・モードとクライアント・モードの違いは接続相手のMACアドレスを取得する初期化プロセスです。クライアント・モードでは接続相手のMACアドレスはUDP1G-IPコアがARP要求を送信した後に受信されたARP応答パケットから抽出されます。一方サーバー・モードでは接続相手のMACアドレスは接続相手として設定されたIPアドレスに合致したARP要求パケット受信時にそのパケットから抽出します。データの送信あるいは受信プロセスはクライアント・モードと同じです。サーバー・モードでの動作シーケンスの例を以下に説明します。

- 1) RSTレジスタ='1'としてIPコアを強制リセット状態とします。
- 2) 自身のMACアドレスをSML/SMHへ、IPアドレスをDIP/SIPへ、ポート番号をDPN/SPNへ設定します。
- 3) RSTレジスタ='0'としてリセットを解除します、するとIPコアは初期化を開始しARP要求パケット受信を待機します。そしてARP応答パケットを接続相手へと返送します。初期化が完了するとビジー信号は'0'ネゲートします。
- 4) 初期化完了以降はクライアント・モードと同じです。

コアの検証方法

本 UDP1G-IP コア製品には Intel 純正の評価ボードで実機動作する Quartus 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 SignalTAP 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 UDP1G-IP 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 Intel 純正の対応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、Intel 社の MAC-IP コア(IP-TRIETHERNET)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 Quartus 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 Intel 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 UDP1G-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	対応評価ボード名	対応評価ボード型番
UDP1G-IP-C5	Cyclone V	CycloneV E FPGA 開発キット	DK-DEV-5CEA7N
UDP1G-IP-A5	Arria V	ArriaV GX スタータ開発キット	DK-START-5AGXB3N
UDP1G-IP-A10	Arria10	Arria10SoC 開発キット	DK-SOC-10AS066-A

改版履歴

リビジョン	日付	履歴
1.0	Feb-27-2017	New release
1.1	Mar-1-2017	Update Figure11
1.1J	2017/03/09	Rev1.1 に対応した日本語翻訳版を発行
1.2J	2018/08/28	SRV レジスタおよび MacRxReady を追加した最新版コアに対応