



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
- 送信バス幅が 64bit のため送信パケット・サイズは 8 バイト(64bit)単位
- 受信バス幅は 64bit のため受信データ・サイズは 8 バイト(64bit)単位
- 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
- Intel 標準 10Gbps イーサネット MAC と 64bit の Avalon ストリーム・インターフェイスで直結
- または低レイテンシの DG 製 DG-10GEMAC-IP コアと同じく 64bit Avalon-ST で直結可能
- 156.25MHz の単一クロック・ドメイン
- Arria10SoC/ Arria10GX 開発キットまたは IntelPAC ボードで実機動作する半二重通信および全二重通信のリファレンス・デザインが提供可

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	Quartus プロジェクト, コア以外は全てソースコード提供
その他	Arria10 SoC / Arria10GX / IntelPAC 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: コンパイル結果例

Family	Example Device	Fmax (MHz)	ALMs ¹	Registers ¹	Pin	Block Memory bit ³	Design Tools
Arria 10 SX	10AS066N3F40E2SGE2	156.25	2,566	3,931	-	1,179,648	QuartusII16.0

注意:

- 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。

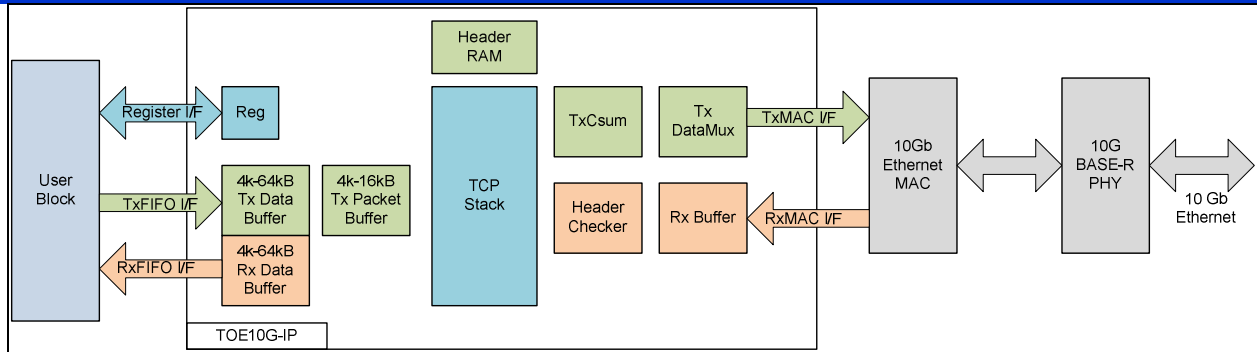


図 1: TOE10G-IP ブロック図

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。本コアを使うことで他のネットワーク・デバイスとの TCP/IP による 10Gb イーサネット通信を CPU や外部メモリを使わず簡単に実装できます。例えばデータ・ロガーのようなアプリケーションに適しています。

コア概要

本 TOE10G-IP コアは TCP スタックを実装するため Intel 社から提供される 10GbE EMAC IP コアまたは、DesignGateway 社から提供する低価格低レイテンシの DG-10GEMAC-IP コアと、トランシーバ部で実装する 10Gb BASE-R PHY と接続することでネットワーク・データ転送の低層でのプロトコルをサポートします。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。

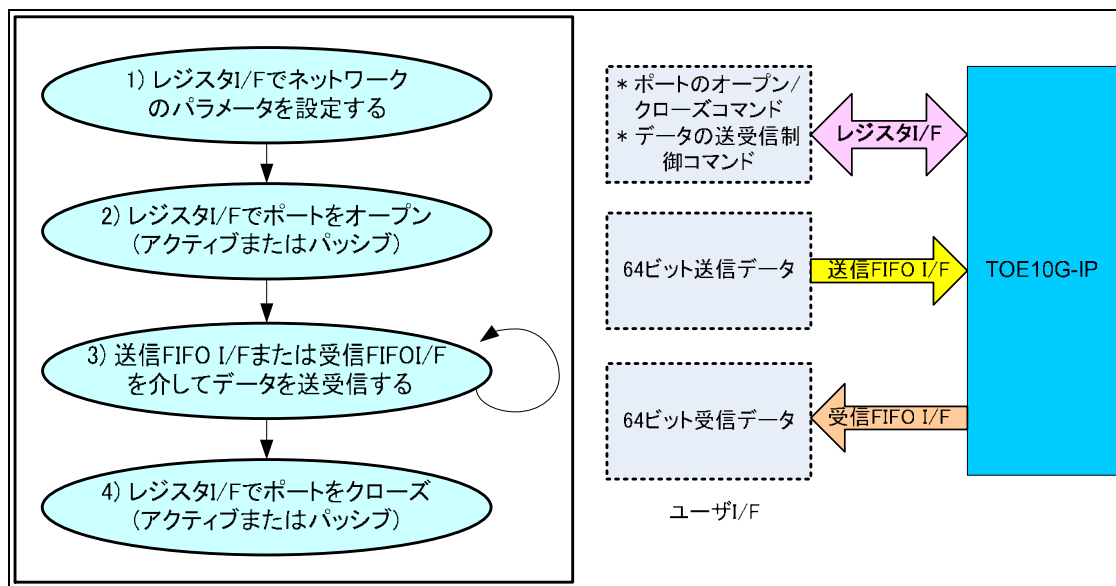


図 2: TOE10G-IP のユーザ I/F と動作シーケンス

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 2 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、もうひとつはデータを送信・受信する FIFO I/F です。システムの初期化時にユーザは MAC アドレス、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。

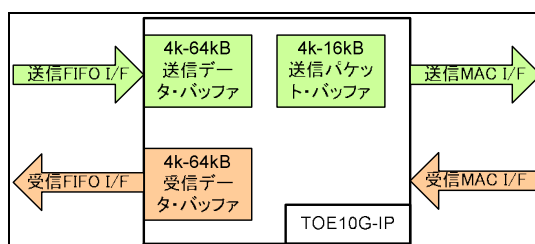


図 3: 送信/受信バッファ・サイズは調整可能

TOE10G-IP コアには、送信データ・バッファ、送信パケット・バッファ、受信データ・バッファの3種類のバッファがあり、IP コアで容量を設定できます。ユーザ・アプリケーションに合わせたサイズを選ぶことで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 PKL レジスタで設定するパケット・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。

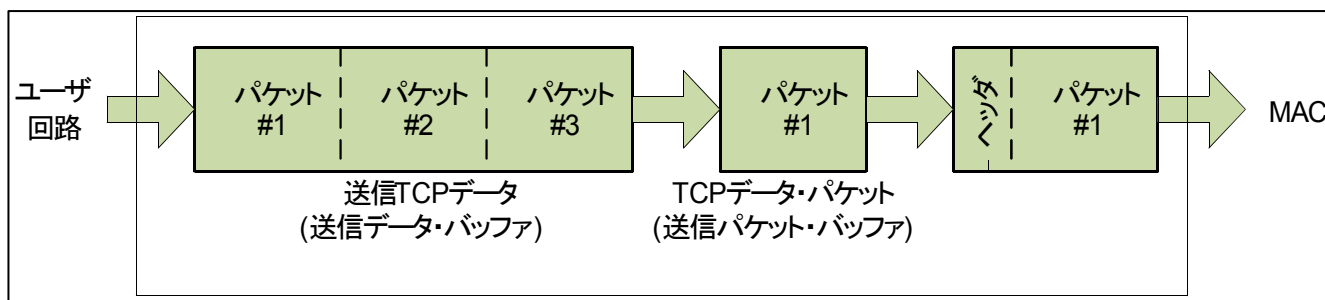


図 4: 送信データ・フロー

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 IP コア内で自動計算されます。コアは受信した ACK パケットのアクノリッジ番号をモニタすることで、すでに相手側で受信したデータを確認します。送信データ・バッファは相手側が全データの受領を完了するとフラッシュします。

TCP 規格によると ACK 番号は前パケットの値と同じです、本 IP コアは受信した ACK 番号を評価し必要に応じて同じパケットを再送します。CMD レジスタを介してモニタできるビジー・フラグは TDL レジスタで設定された膳データの送信が完了すると 0'ネグートします。ユーザはビジー・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 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(PKL)や総転送サイズ(TDL)を変更することができます。

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 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。

- **TCP スタック**

ユーザ回路からアクティブ・コマンドの操作が指示されると 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ポートのオープン・データ送信・ポートのクローズを実行するためパケットの出力を開始します。それと同時に受信ブロックは ACK パケットをデコードしモニタします。その次の動作は受信パケットやコマンドに依存します。たとえば受信パケットが ARP 要求の場合 TCP スタックは送信ブロックによる ARP 応答ステートに遷移します。

表 2: 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, SRV 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	[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 /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]	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 /Rd	[31:0]	コアの 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	[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。 コネクションがパッシブ・モードの場合はターゲットのポート番号は自動的に設定される。ポート番号は受信パケットが正しいネットワーク・パラメータ(SML, SMH, DIP, SIP, SPN レジスタの設定値と一致した場合)に受信パケットのヘッダより抽出される。
0111b	SPN	Wr /Rd	[15:0]	コアのポート番号を 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、ただし TDL は 8 バイトの倍数でなくてはならない。 有効な値は 8~0xFFFFFFFF8。(bit[2:0]はコアによって無視される) 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。その場合本コアは最後のこのレジスタに設定された値を次の送信時に用いる。
		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要求を再送する。 [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16回まで再送し最後に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16回まで再送し最後に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 → 最初のタイムアウト後、本IPコアはRSTパケットを送信して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16回まで再送し最後に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] 受信データFIFOが一杯になった、または誤ったシーケンス番号などの要因による受信パケット・ロストのタイムアウト → タイムアウト後、本IPコアは重複ACKパケットを送信しデータ再送を要求する [7] 送信動作中、相手からの受信ウィンドウ・サイズが少なすぎ(パケットサイズより小さい)ることによるタイムアウト → タイムアウト後、本IPコアは本レジスタbit[5]現象と同様、反転(リバースした)データ・パケットを再送する。本bitはPSHレジスタのbit[2]が'1'にセットされた場合'1'アサートされることがある。 [22] 送信動作中に相手からFINフラグを受信した [23] 受信データFIFOが一杯のため受信パケットを受け損ねた(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], [26:25] 内部テスト状態(不具合発生時にこれらのbit情報も含めてお知らせください)
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、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(ビジー・フラグ='1'中)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。

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	[1:0]	<p>本 IP コアが最後のパケットを送信するときの送信モードの指定。デフォルト値は全 bit とも '0'。</p> <p>[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)</p> <p>[1] PSH フラグをセット、送信時の全てのパケットでヘッダ内の PSH フラグに '1' をセットする。 [2] 反転(リバース)データ・パケットの生成イネーブル '1' をセットすると自動的に反転データ・パケットを発生する。このフラグは接続相手デバイスからのウインドウ・アップデート・パケットがロストした場合の対策として用意される。反転データ・パケットを送信すると相手側デバイスで ACK パケットが再生成するのでハングアップを回避できる。この bit[2] 機能は以下の条件で動作する</p> <p>(1) 現在のコマンドがデータ送信でまだ全データ転送が完了していない (2) 受信ウインドウ・サイズがパケット・サイズ以下 (3) TMO レジスタでセットしたタイムアウトが発生</p>
1100b	WIN	Wr /Rd	[5:0]	<p>IP コアがウインドウ・アップデート・パケットを自動送信するときの 1K バイト単位での設定閾値。デフォルト値は 0 (ウインドウ・アップデート・パケットの自動送信を行わない)</p> <p>IP コアは受信データ・バッファの空き容量が、前回送信したパケットでターゲットへ報告したウインドウ情報からこのレジスタで設定した閾値分増加した場合に、自動的にウインドウ・アップデート・パケットを送信する。(この機能はターゲット側から IP コアへデータを送信する場合のフロー制御で必要となる)</p> <p>例えば WIN='000001b'(=1K バイト) で設定し、前回最後に IP コアがターゲットに向けて送信したパケット内で報告したその時点での受信バッファのウインドウ・サイズ(空きバッファ容量)が 2K バイトであったと仮定する。この後ユーザ回路が受信データを読み出した結果 IP コアの受信データ・バッファの空き容量が 1K バイト分増えた時点で、IP コアはターゲットに対してウインドウ・アップデート・パケットを自動的に送り 3K バイトのウインドウ・サイズとなったことを通知できる。</p>
1101b	ETL	Wr	[31:0]	<p>拡張した送信データ数(バイト単位で設定)、この値は 8 バイトの倍数とする櫃よがある。bit[2:0] は本 IP コアによって無視される。</p> <p>ユーザ回路は CMD='00' で送信を実行中にて送信データ数を TDL 以上に増やして送信したい場合に使う、この場合ユーザ回路は IP コアに次のデータ送信コマンドを発行することなく、連続してデータ送信を継続できる。ただし以下の注意点がある</p> <p>1) ETL レジスタは TDL のリード値(残り送信バイト数)が 128K バイト以下でない状態でライトしなくてはならない</p> <p>2) ETL レジスタにセットする値は 0xFFFFF8 から TDL のリード値を差し引いた数以下</p> <p>例えばユーザが TDL に 3.5GByte の送信データ量をセットする。 本 IP コアが 2GByte データを送信した時点(残りサイズが 1.5GByte)でユーザは本 ETL レジスタに 1.5GByte をセットする。そうすることでトータルの送信データ量は 5GByte となるが、すなわち最初に TDL にセットした 3.5GByte に ETL でセットした 1.5GByte が加算される。</p>
1110b	SRV	Wr/ Rd	[0]	<p>サーバー・モードの設定、デフォルト値='0'(クライアント・モード)</p> <p>'0': クライアント・モード、本 IP コアのリセットが '1' から '0' に遷移すると本 IP コアは ARP 要求パケットを送信し ARP 応答パケットから相手側の MAC アドレスを獲得する。本 IP コアが ARP 応答パケットを受信するとビジー・フラグは '0' ネゲートし初期化が完了する。</p> <p>'1': サーバー・モード、本 IP コアのリセットが '1' から '0' に遷移すると本 IP コアは相手からの ARP 要求パケットを待機し相手の MAC アドレスを獲得する。本 IP コアが ARP 要求パケットを受信すると ARP 応答パケットを送信する。その後ビジー・フラグは '0' ネゲートし初期化が完了する。</p> <p>注: 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・モードの場合本 IP コアのリセット解除後、接続相手は本 IP コアに対して ARP 要求パケットを送信しなければ初期化が完了しない。</p>

注意:

1. 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 3: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 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 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が、接続先ターゲット・デバイスからの正常 ACK パケットを受信し前パケットのデータ到達を確認すると送信パケット・バッファに転送されます。そして転送されたパケット・データにはヘッダ等が付加され TCP パケットとして生成されます。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 パケット受信から次パケットの送信までのレイテンシを低減するので、より連続したデータ送信を維持できます。総転送数として設定したよりも多くのデータをユーザ回路から本バッファに送信した場合、その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 3 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までバッファ内で待機してから出力されます。現在のパケットが送信されている間に次のパケット・データが送信データ・バッファから取り出されあらかじめ準備されるため、パケットは連続して送信可能です。

• 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 内の主要なパラメータは 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。一部のパラメータ例えばターゲット MAC アドレス、ターゲット IP アドレス、ターゲット側ポート番号などについてはサーバー・モードで動作する場合は最初の SYN パケット受信時に更新されます。

- **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。計算されたチェックサム出力はヘッダ RAM 内に格納されます。

- **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。ヘッダ・チェックにより受信パケットが有効であるかを確認するまでの間本バッファ内に格納されます。有効な TCP データのみが受信データ・バッファへと転送されます。

- **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。さらに有効なパケットであっても受信済みデータと重複があった場合その部分は破棄されます。

- **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 3 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。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。

10Gb イーサネット MAC および 10Gb BASE-R PHY

このブロックにおいて MAC は Intel 社から提供される 10Gbit EMAC IP コアあるいは DesignGateway 社からオプションで販売される低レイテンシの DG-10GEMAC-IP-A コアを使用します。お客様にて Intel 社または弊社より別途手配して頂く必要がありますのでご注意ください。また、10Gb BASE-R PHY は Intel 社から無償提供されるトランシーバの PHY IP コアを使う必要があります。

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 4 と表 5 で説明します。MAC I/F グループの全信号は Altera 製 10Gb EMAC と直結します。

表 4: 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 5: 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	PHY レイヤから出力される 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レジスタへの書込み動作を実行する。
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)のリード値
RegDataA8[31:0]	Out	32 ビットの TDL レジスタ(RegAddr=1000b)のリード値
RegDataA9[31:0]	Out	32 ビットの TMO レジスタ(RegAddr=1001b)のリード値
送信データ・バッファ I/F		
TCPTxFfFlush	Out	送信データ・バッファをリセットする。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 アサートする。
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。
受信データ・バッファ I/F		
TCPRxFfFlush	Out	受信データ・バッファをリセットする。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	バッファ内受信データ数が 8 の倍数でない場合に受信データ・バッファ内の最終データにおける剰余バイト数。
TCPRxFfRdEmpty	Out	受信データ・バッファの FIFO Empty 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		
MacTxData[63:0]	Out	送信データ・バス。
MacTxEmpty[2:0]	Out	フレームの最終ワードで使われていないバイト数を示す。
MacTxValid	Out	EMAC への送信データ有効信号、MacTxData に同期する。
MacTxSOP	Out	フレームの先頭ワードであることを示す信号。
MacTxEOP	Out	フレームの最終ワードであることを示す信号。
MacTxReady	In	ハンドシェーク信号、MacTxData が正常に受信されたときにアサートする。 この信号はパケット送信期間中には'0'ネゲートしてはならない。
MacRxData[63:0]	In	受信データ・バス。
MacRxValid	In	受信データの有効信号、MacRxData に同期する。 本 MacRxValid 信号はパケット受信中は連続して'1'アサートされ続けなければならない。
MacRxEOP	In	フレームの最終ワードであることを示す信号。
MacRxError	In	受信フレームの最後にそのフレーム内で CRC エラーが検出されたことを示す信号。 '1': エラー・パケット、'0': 通常パケット、MAC 出力の avalon_st_rx_error[1]と接続する。
MacRxReady	Out	ハンドシェーク信号、MacRxData が正常に受領された際にアサートされる。 本 MacRxReady 信号は書く受信パケット間のギャップとして 2 クロック期間'0'ネゲートされる。

タイミング・チャート

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。

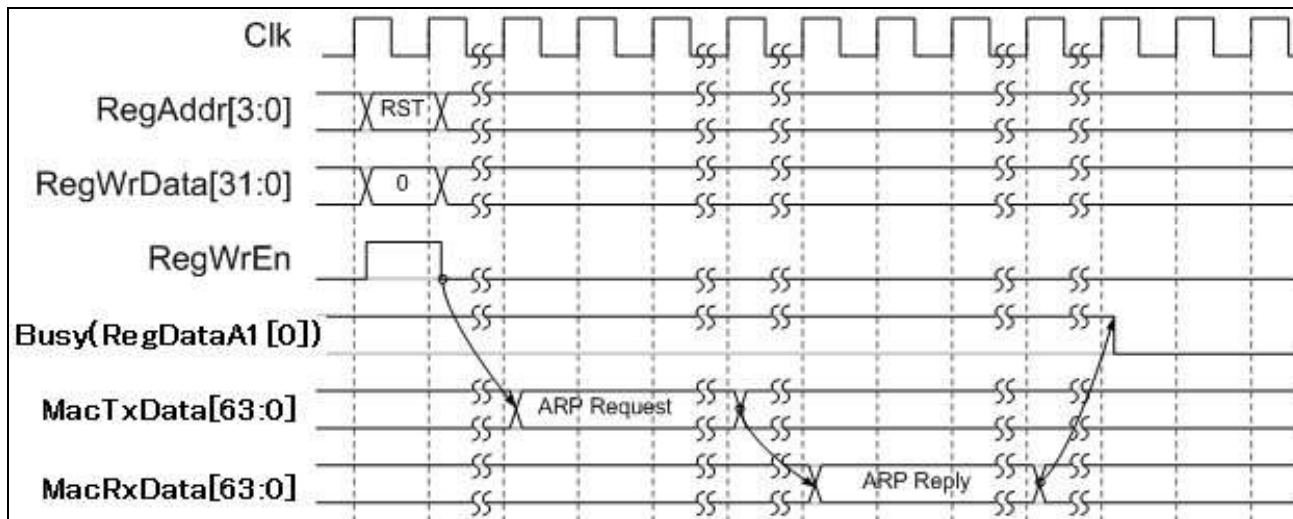


図 5: クライアント・モードの IP コア初期化

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。

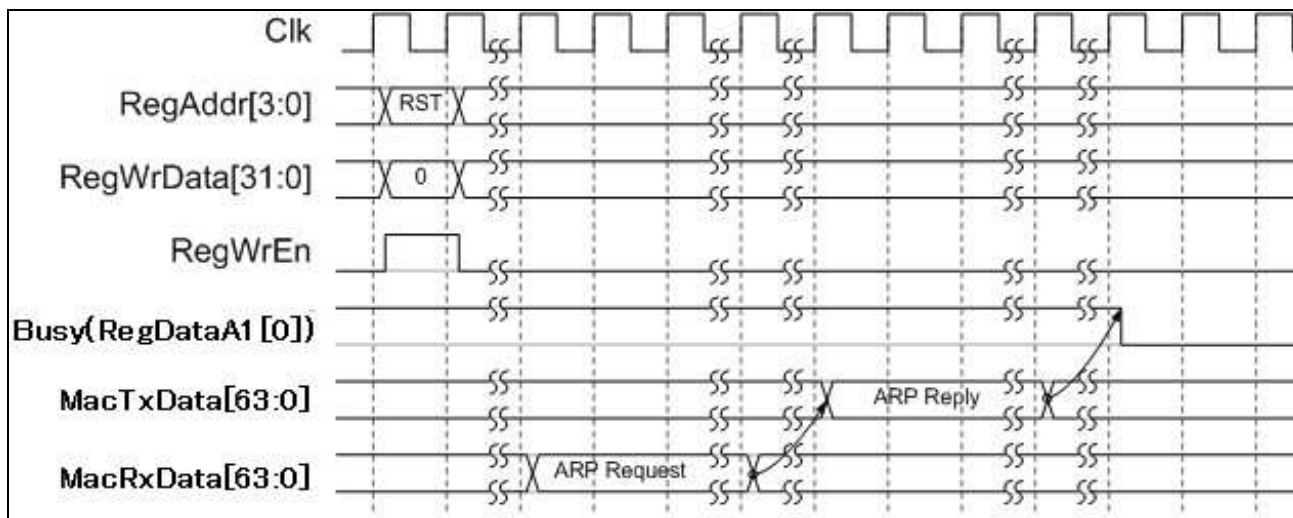


図 6: サーバー・モードの IP コア初期化

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 IP コア内部レジスタへのリード・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。

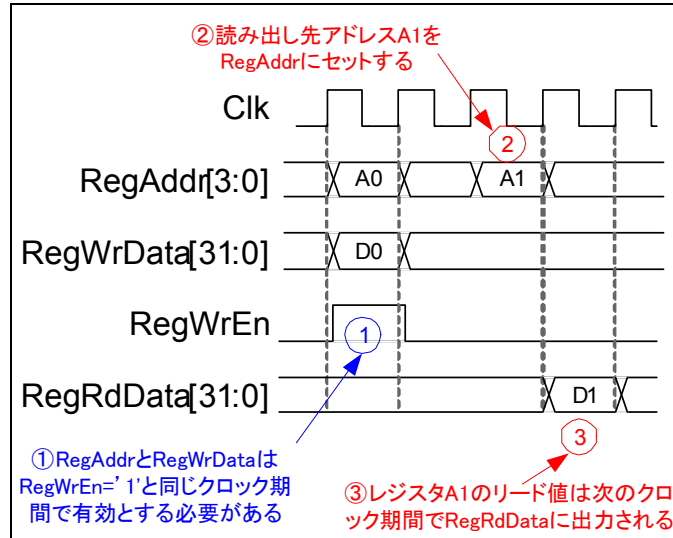
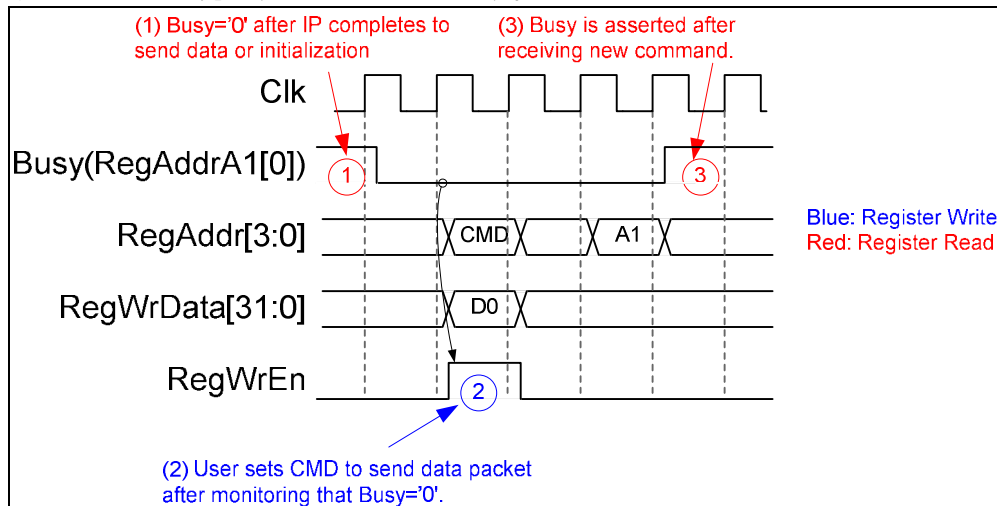


図 7: レジスタ I/F のタイミング・チャート

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 8 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 8: 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 FIFO インターフェイスを介して本 IP コアへ送信するデータのタイミング・チャートを図 9 に示します。

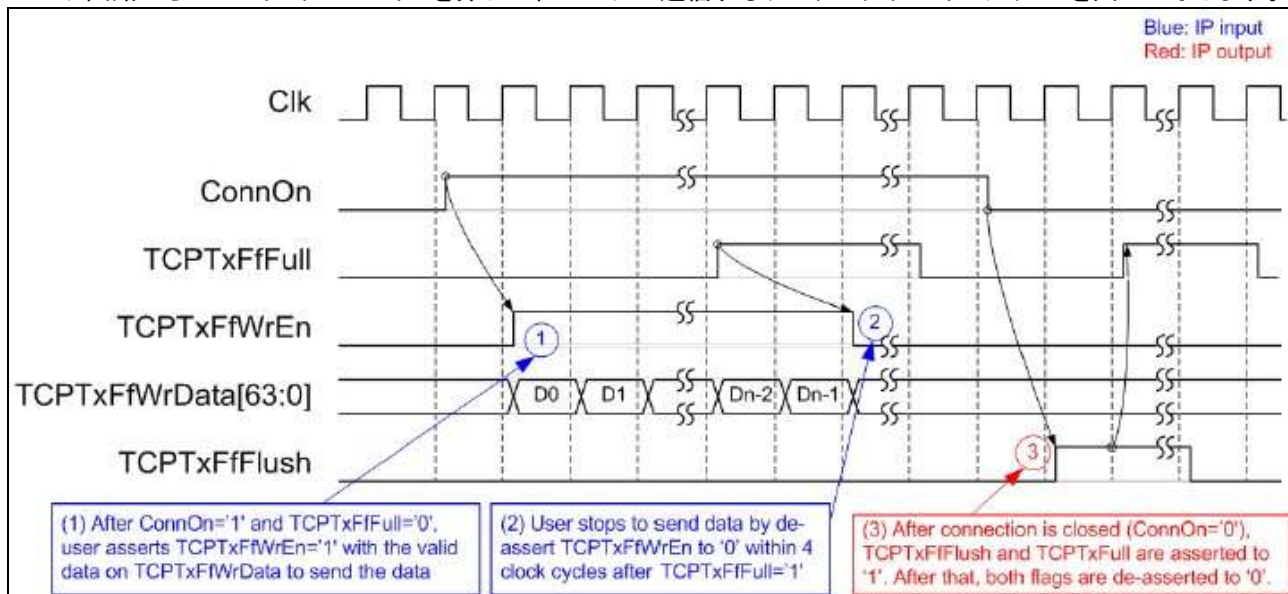


図 9: 送信 FIFO I/F のタイミング・チャート

- (1) データを送信する前にユーザ回路は 2 つの信号をチェックする必要があります。ひとつは Full フラグ (TCPTxFfFull) が '1' アSSERTされていないことです。更に ConnOn が '1' である必要があります。この 2 信号の条件が合致していれば TCPTxFfWrEn='1' アSSERTすることで TCPTxFfWrData を書き込むことができます。
- (2) TCPTxFfWrEn は TCPTxFfFull='1' アSSERTされてから 4 クロック以内に '0' ネグートしなくてはなりません。
- (3) コネクションがクローズされると本 IP コアにより TCPTxFfFlush が '1' アSSERTされ Tx FIFO 内のデータはクリアされます。その後 TCPTxFfFlush と TCPTxFfFull は '0' ネグートされ次の送信データを書き込む準備ができます。

受信 FIFO インターフェイス

本 IP コアがターゲットから受信したデータはパケットから抽出され受信データ・バッファに保存されます。ユーザ回路は図 10 に示すように FIFO インターフェイスを介してバッファ内の受信データを読み出すことができます。

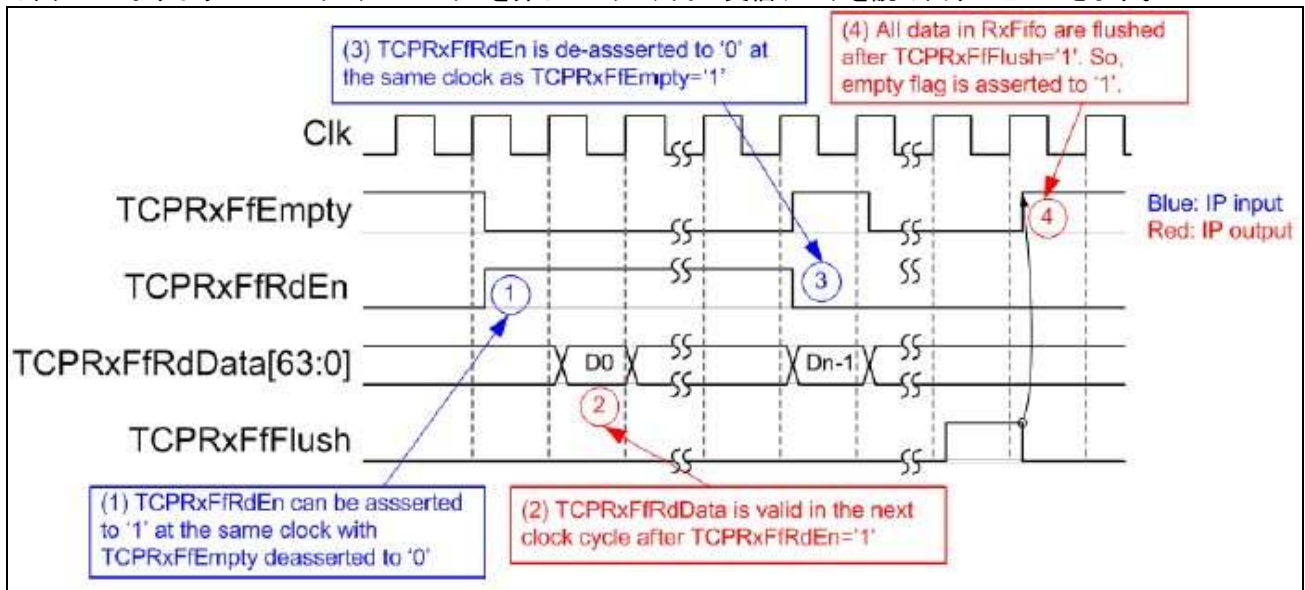


図 10: エンプティ・フラグによる受信データ・バッファのタイミング・チャート

- (1) データを TCPRxFfEmpty で有効受信データの有無を確認できます。TCPRxFfEmpty が '0' ネゲートされている場合データをリードできます。TCPRxFfRdEn を '1' アサートし受信データ・バッファからリードします。
- (2) TCPRxFfRdData は次クロック期間で有効なデータが出力されます。
- (3) TCPRxFfEmpty='1' となった場合同じクロック期間で直ちに TCPRxFfRdEn='0' ネゲートし FIFO のアンダーフローを防止する必要があります。
- (4) (次の)コネクションがオープンされると受信データ・バッファ内の全受信データはクリアされます。受信データ・バッファのクリア動作は TCPRxFfFlush 信号をモニタすることで確認可能です。

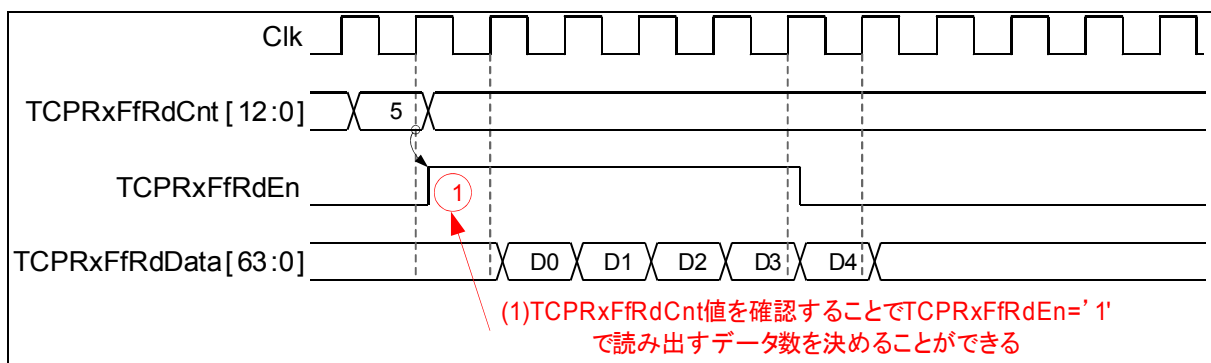
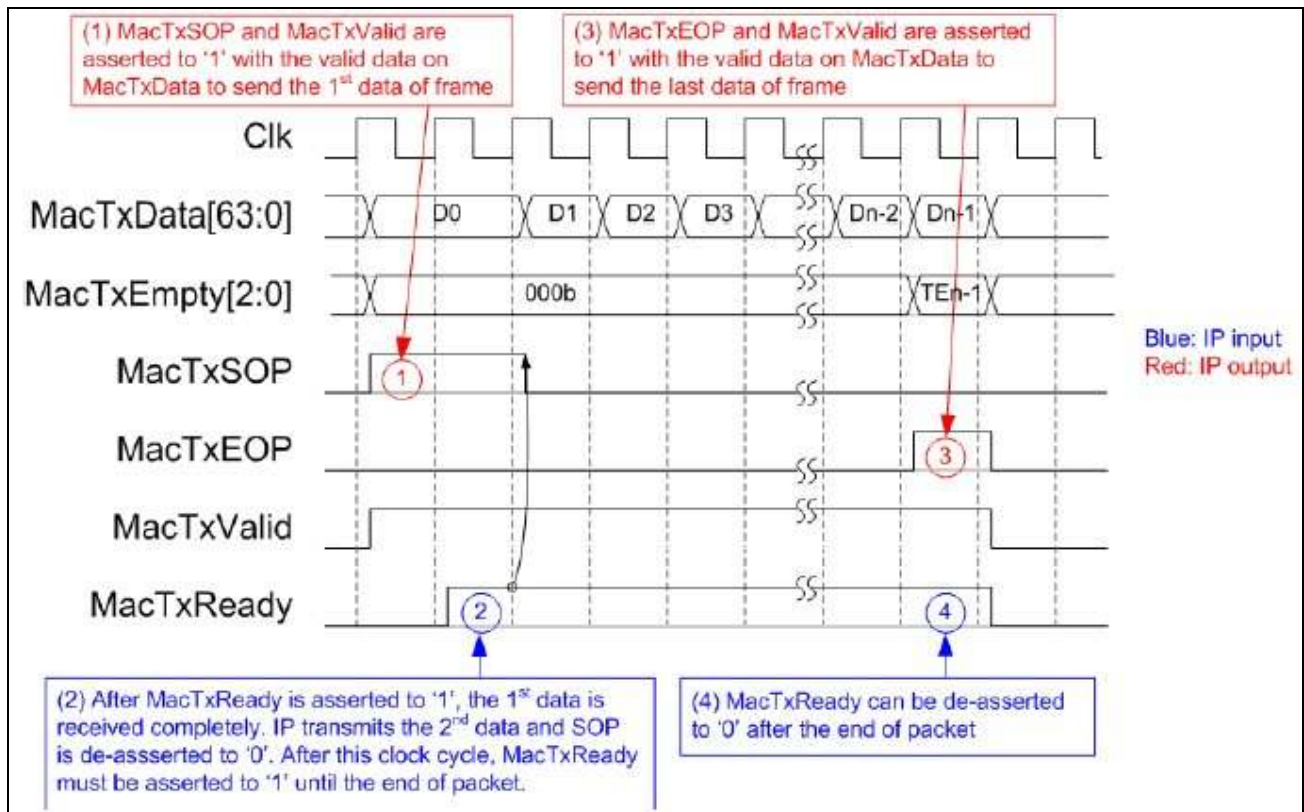


図 11: FIFO 残量カウンタ情報による受信データの読み出し

その他の方法として、受信データ・バッファの状態は 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 64 ビット単位で示します。このためユーザ回路は図 11 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。

EMAC インターフェイス

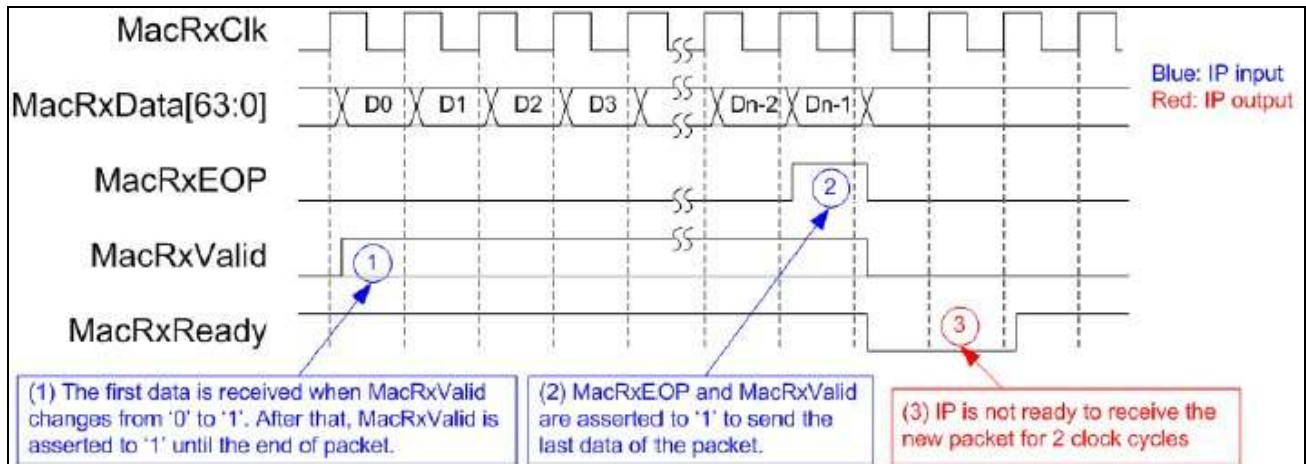
本 IP コアの EMAC インターフェイスは Intel 製 EMAC IP コアと互換性があります。図 12 に示すとおり本 IP コアがパケットを送信する際には MacTxSOP と MacTxValid をパケットの先頭データでアサートします。全信号は EMAC からの MacTxReady が '1' アサートされデータ転送要求を確認するまで保持されます。MacTxready はパケット転送の最後まで連続して '1' アサートし続けなくてはなりません。送信の最終データにて MacTxEOP と MacTxValid を '1' アサートすることでパケット終了のステータスを示します。



- (1) MacTxSOP と MacTxValid が '1' アサートされフレーム先頭データを MacTxData で送信転送する。
- (2) MacTxReady が '1' アサートされると先頭データ転送が完了する。本 IP コアは 2 番目のデータを転送し SOP はネゲートされる。このクロック期間後 MacTxReady はパケット終了まで連続して '1' アサートを維持しなくてはならない。
- (3) MacTxEOP と MacTxValid を '1' アサートし MacTxData でパケット最終データを送信する。
- (4) MacTxReady はパケット終了後 '0' ネゲートできる。

図 12: EMAC インターフェイスの送信タイミング・チャート

図 13 に Intel 製 EMAC 仕様に沿った受信側のタイミング・チャートを示します。本 IP コアは MacRxValid が '0' から '1' アサートされることを検出してパケットの受信を開始します。MacRxData はパケット最終データの MacRxEOP が '1' アサートされるまで (MacRxValid='1' を) 連続して MacRxData で受信します。パケットがすべて受信し終わると本 IP コアは MacRxReady を 2 クロック期間ネゲートしパケットの終了処理を行います。このため EMAC は 2 パケットをギャップなしで送信することはできません。少なくとも 2 クロック期間のギャップが必要となります。



- (1) MacRxValid が '0' から '1' アサートされることで先頭データを受信する。
- (2) MacRxEOP と MacRxValid の両方を '1' アサートしてパケット最終データを転送する。
- (3) 本 IP コアは 2 クロック期間次のパケット受信できない

図 13: 本 IP コアが Avalon-ST バスと接続されている場合の 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 (SRV[0]='0')

SRV[0]レジスタがデフォルトの'0'とした場合のデータ送信/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='1'としてIPコアをリセット状態とします。
- 2) 自身の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='0'にクリアしリセット状態を解除します。IPコアはARP要求を自動送信しターゲットのMACアドレス情報を取得します。ARPプロセスが完了するとビジー・フラグ(CMD[0])が'1'から'0'に遷移します。
- 4) a. アクティブ・オープンの場合: CMDレジスタをセットしてポートをオープン(SYNパケットを送信)します。
b. パッシブ・オープンの場合: "ConnOn"が'1'となる(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。ビジーフラグがクリアされると次の送信操作が可能となります。
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) コネクションのオープンと同様以下2つのモードでコネクションを破棄します。
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(FINパケットを送信)します。
b. パッシブ・クローズの場合: "ConnOn"が'0'となる(FINパケットを受信)のを待ちます。

サーバー・モード (SRV[0]='1')

サーバー・モードとクライアント・モードの違いは初期化プロセスでのターゲットのMACアドレス取得方法です。クライアント・モードではMACアドレスはIPコアがARP要求パケットを送信した後のARP応答パケットで受信します。一方サーバー・モードにおいてはMACアドレスはターゲットIPアドレスと一致する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・モードでの動作シーケンス例を以下に説明します。

- 1) RSTレジスタ='1'としてIPコアをリセット状態とします。
- 2) 自身のMACアドレスをSML/SMHに、自身およびターゲットのIPアドレスをSIP/DIPに、自身およびターゲットのポート番号をSPN/DPNにセットします。
- 3) RSTレジスタをセットしリセット状態を解除します。IPコアは初期化プロセスを開始しターゲットのMACアドレスを取得するためARP要求パケットを待機し続けます。ターゲットからのARP要求パケットを受信するとIPコアはターゲットに対してARP応答パケットを送信します。初期化プロセスが完了しビジー・フラグが'0'クリアされます。
- 4) それ以降のプロセスはクライアント・モードのステップ4~6と同じです。

コアの検証方法

本 TOE10G-IP コアのユーザ I/F 部については機能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 Arria10SoC/Arria10GX/Intel PAC 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 Intel 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、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 Quartus 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・ファミリに応じて以下のコアのラインナップが用意されています。(DG-10GEMAC-IP-A は Intel 純正 EMAC IP コアとインターフェイス互換の TOE10G-IP コア専用の低価格・低レイテンシ EMAC IP コアです)

コア型番	対応ファミリ	ツール環境	検証用評価ボード	説明
TOE10G-IP-A10	Arria10	Quartus 16.0	DK-SOC-10AS066S-A 又は DK-DEV-10AX115S-A	Arria10 向け TOE10G-IP コア
TOE10G-IP-PAC	IntelPAC 専用	Quartus 16.0	Intel PAC	
DG-10GEMAC-IP-A	Arria10	Quartus 16.0		TOE10G-IP コア専用低レイテンシ EMAC IP コア

IP コアの価格やライセンス条件あるいは上記以外のファミリ対応状況につきましては、デザイン・ゲートウェイ (sales@dgway.com) または国内 Altera 各代理店までお問い合わせください。また、コアご購入後のサポートは製品添付の実機動作リファレンス・デザインと同一の環境が前提となるため、対象ファミリの Altera 評価ボードの手配が必須となります。

更新履歴

リビジョン	日付	更新内容
1.0	May-16-2016	New release
1.0J	2016/09/26	日本語版の初版発行
1.2J	2019/07/29	英語版最新の Rev1.2 に対応してアップデート