TOE1G-IP 2 ポート・リファレンス・デザイン説明書(Xilinx 版)

Rev1.2J　2016/09/01

1　概要

GbE アプリケーションにおいては、TOE1G-IP コア(旧製品名：TOE2-IP コア)を使った高速 TCP/IP データ送受信と並行して、CPU 等により例えば ICMP や UDP のような TCP/IP 以外のプロトコルや制御/ステータス情報の低速 TCP/IP 転送が必要となる場合があります。TOE1G-IP コアの参照デモ・デザインとして TOE1G-IP コアの高速ポートと CPU ファームウエアと簡単な追加ロジックによる低速ポートの2ポートを実装したリファレンス・デザインを提供しています。本 2 ポート・デザインでは低速ポートにおいて ICMP による Ping コマンドをサポートします。

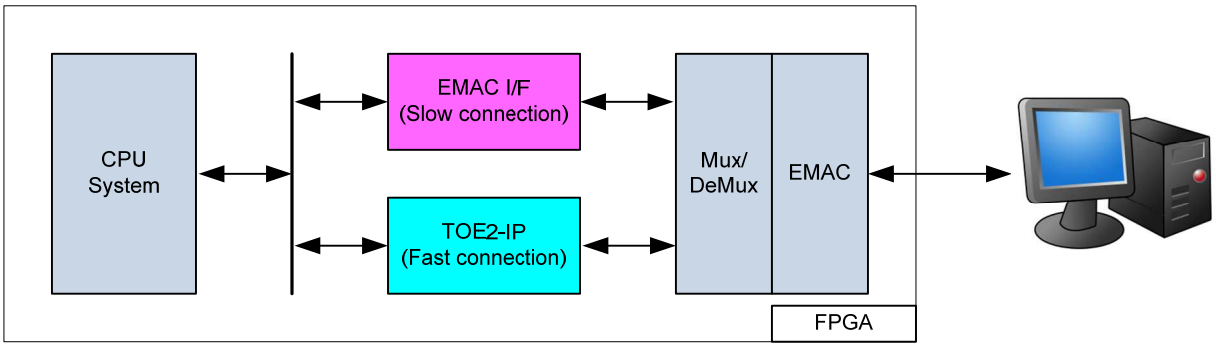

図 1: 2 ポート・デザインのハードウエア構成

本デモ・システムにおいてユーザからはシリアル・コンソールを介してテスト動作を選択できます。本デモには3種類のテスト・モードがあり、高速受信コネクション、高速送信コネクション、低速送受信コネクションです。高速コネクションのテストでは DesignGateway 社から提供される専用のテスト・アプリケーションを接続 PC で使います。一方低速コネクションのテストは DOS 窓で実行する ping コマンドを使います。Ping コマンドではエコー応答とリクエストの送信および応答待機に ICMP プロトコルが使われます。

それ以外には本デモと同じハードウエア・デザインにて FTP サーバー・デモが DesignGateway 社から提供されますが、そちらは CPU のファームウエアが編集されています。従って2ポート用のハードウエア・システムを適用することで、ユーザは CPU ファームウエアの更新だけで様々なアプリケーションの開発が可能です。ハードウエアの詳細については次の章で説明します。

2　ハードウエアの構成

図 2 に示すように本システムのハードウエアは大きく2つのブロックに分けられます、ひとつは高速コネクション用の AXITOE1G ブロック(図 2 水色)でもうひとつは低速コネクション用の AXIEMAC(図 2 紫色)ブロックです。それぞれのブロックにて EMAC からのパケットは異なるポート番号により区別して動作します。 そのポート番号は CPU ファームウエアによりレジスタ・アクセス経由で設定します。

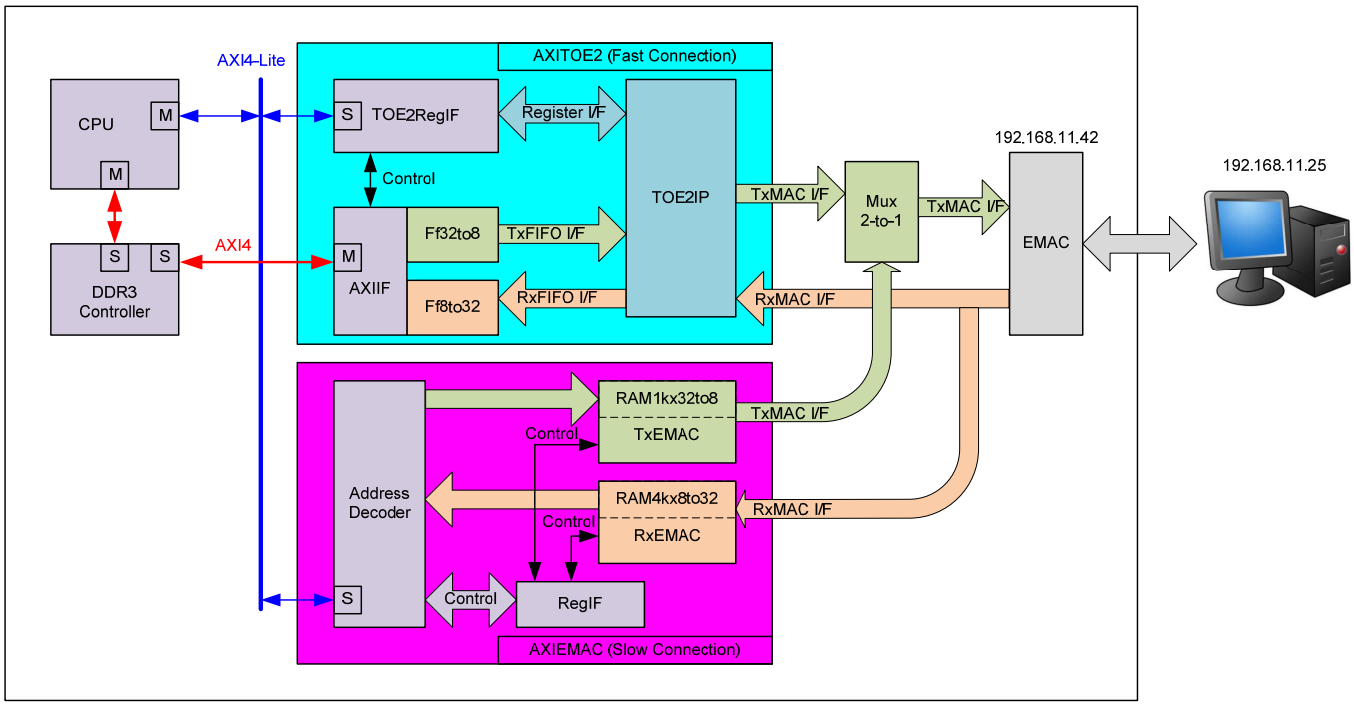

図 2: 2 ポート・デザインのハードウエア・ブロック図

TOE1G-IP コアを搭載した高速コネクション側において転送データは、AXI4 バスを通して DDR3 メモリに直接バースト転送されることで高い転送帯域を実現します。AXIIF モジュール内の DMA エンジンは転送方向に応じて DDR3 メモリと Ff32to8 間のバースト転送を実行します CPU は高速コネクションのデータ転送を AXI4-Lite バス・インターフェイスのレジスタ・アクセスを通して制御します。つまり TOE1GRegIF は AXI4-Lite バス変換と各種制御/ステータス・レジスタのアドレス・デコードを実装しています。このレジスタは TOE1G-IP コアの動作と AXIIF の DMA エンジンの両方を制御するために使われます。

CPU による低速コネクション側においてのハードウエアは、EMAC からのイーサネット・パケットをデータ・バッファとして実装した FPGA 内部 RAM を通して接続する簡単な回路となります。 送信側においては TxEMAC により RAM から EMAC へとデータが転送されます、受信側においては CPU で設定可能なパケット･ヘッダのフィルタリング機能を実装します。 従って有効なヘッダを持つパケットのみが RAM に格納され CPU によって処理されます。ただし低速コネクション側のパケットはヘッダを自動的に付加や削除する機能は実装されないため、CPU はヘッダを含めた全てのパケットを自身で作成する必要があります。 低速コネクション側においては高いパフォーマンスは必要としないため、CPU からのパケット・データ格納用 RAM へのアクセスはレジスタと同じく AXI4-Lite バス経由で行います。ブロック内のアドレス・デコーダによって AXI4-Lite のアドレス・マップは、TxEMAC 向け RAM、RxEMAC 向け RAM、そして他の制御/ステータス・レジスタの3つの空間に分割されます

2.1 AXITOE1G

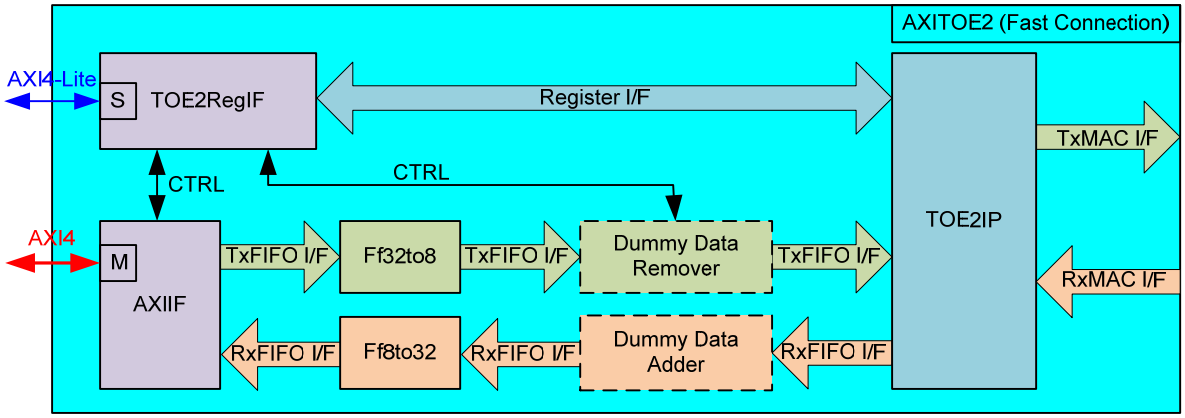

図 3: AXITOE1G ブロック図

AXITOE1G モジュールは TOE1G-IP コアを含み高速コネクションの TCP/IP パケットを処理します。 高い転送パフォーマンスをシンプルなデザインで実現するため、AXIIF 内の DMA エンジンでは 128 ビート×32 ビットを使った 512 バイトの固定バースト転送長に設定されています。CPU で設定する転送データ長を 512 バイト単位に揃える制限をなくすため、FIFO と TOE1G-IP コア間にて送信データでは余分なダミーデータを除去し、受信データでは不足分のダミーデータを付加するロジックが挿入されています。Ff32to8 と Ff8to32 はデータ・バッファとして使われますが、同時に 32 ビット幅の AXI4 バスと 8 ビット幅の TOE1G-IP コアのバス幅変換としても使われます。 AXITOE1G モジュール内のサブ・モジュール詳細について以下に説明します。

2.1.1 TOE1G-IP

この IP コアは TCP/IP スタックを含み、さらに送信と受信の両側で TCP パケットのチェックサムを計算するオフロード・エンジンを含みます。TOE1G-IP コアの詳細についてはデータシートを参照してください。

2.1.2 AXIIF

DMA エンジンおよび AXIIF モジュールはステート・マシンが内蔵されます。 AXIr インターフェイス(AXI4→TxFIFO)および AXIw インターフェイス(RxFIFO→AXI4)はそれぞれ独立して動作し、それぞれのインターフェイスを独立して制御するため2つのステート・マシンが使われます。

送信側(AXIr インターフェイス)では CPU からの開始信号を受けるとステート・マシンはまず送信 FIFO である Ff32to8 の空きスペースが 512 バイト以上であることを確認します。 送信 FIFO の空きが十分にある場合、AXI4 バスに対して 512 バイトのリード要求を出します。すると AXI4 バスから Ff32to8 に転送する 512 バイト・データが送られてきます。ステート･マシンは 512 バイトごとの各データ転送完了を FIFO ステータスで確認し、全データ転送が完了するまでループ動作します。

受信側(AXIw インターフェイス)では CPU からの開始信号を受けるとステート・マシンはまず受信 FIFO である Ff8to32 に格納済みのデータ量が少なくとも 512 バイト以上となるまで待機します。 次に AXI4 バスに対して 512 バイトのライト要求を出します。AXI4 から要求のアクノリッジを受け取ると Ff8to32 から AXI4 に対して 512 バイトをバースト転送します。 AXIr と同様ステート・マシンはデータ送信を FIFO ステータスで確認しながら全データ転送が完了するまでループ動作します。

2.1.3 TOE1GRegIF

AXITOE1G のメモリ・マップは表 1 に示すように2つの空間に分類されます、すなわちひとつは TOE1G-IP コアのレジスタ空間でありもう一つは AXITOE1G モジュールの制御/ステータス・レジスタ空間です。TOE1GRegIF は AXI4-Lite バスのアドレスをデコードし、TOE1G-IP コアへの信号や制御/ステータス・レジスタのアクセスに変換します。 TOE1G-IP コアの全レジスタは CPU ファームウエアによって TOE1G-IP コア機能を直接制御できるようマップされます。制御/ステータス・レジスタは AXIIF 内 DMA エンジンの制御や TOE1G-IP コアからの出力ピンの状態をモニタするために使います。

アドレス Rd/Wr	レジスタ名 (“ftp_demo.c”内のラベル名)	説明
BA+0x80000 ～ BA+0x8002B: TOE1G-IP レジスタ空間		
BA+0x80000 Wr	RST Reg of TOE1G-IP (TOE1G_RST_REG)	TOE1G-IP コア内部レジスタの詳細については TOE1G-IP コアのデータシート表 2 を参照してください。ただし本デザインにおけるアドレス表記はバイト(8bit)単位ですがデータシートではダブルワード(32bit)単位となっております。
BA+0x80004 Wr/Rd	CMD Reg of TOE1G-IP (TOE1G_CMD_REG)	
BA+0x80008 Wr	SML Reg of TOE1G-IP (TOE1G_SML_REG)	
BA+0x8000C Wr	SMH Reg of TOE1G-IP (TOE1G_SMH_REG)	
BA+0x80010 Wr	DIP Reg of TOE1G-IP (TOE1G_DIP_REG)	
BA+0x80014 Wr	SIP Reg of TOE1G-IP (TOE1G_SIP_REG)	
BA+0x80018 Wr	DPN Reg of TOE1G-IP (TOE1G_DPN_REG)	
BA+0x8001C Wr	SPN Reg of TOE1G-IP (TOE1G_SPN_REG)	
BA+0x80020 Wr/Rd	TDL Reg of TOE1G-IP (TOE1G_TDL_REG)	
BA+0x80024 Wr/Rd	TMO Reg of TOE1G-IP (TOE1G_TMO_REG)	
BA+0x80028 Wr	PKL Reg of TOE1G-IP (TOE1G_PKL_REG)	
BA+0x80100 ～ BA+0x8011B: その他 AXITOE1G モジュール内の制御/ステータス・レジスタ		
BA+0x80100 Wr/Rd	Start Transmit DDR Address (TX_DDR_ADDR)	Wr [31:0] – DDR から TOE1G-IP へのデータ転送における DDR 開始アドレス (32 ビットのアライメント維持のため bit[1:0]は”00”とする必要がある) Rd [31:0] – DDR から TOE1G-IP へのデータ転送における現在の DDR アドレス
BA+0x80104 Wr	Transmit Length (TX_DDR_LEN)	Wr [31:0] – DDR から TOE1G-IP への総データ転送サイズをバイト単位で設定
BA+0x80108 Wr/Rd	Start Receive DDR Address (RX_DDR_ADDR)	Wr [31:0] – TOE1G-IP から DDR へのデータ転送における DDR 開始アドレス (32 ビットのアライメント維持のため bit[1:0]は”00”とする必要がある) Rd [31:0] –TOE1G-IP から DDR へのデータ転送における現在の DDR アドレス
BA+0x8010C Wr	Receive Length (RX_DDR_LEN)	Wr [31:0] –TOE1G-IP から DDR への総データ転送サイズをバイト単位で設定
BA+0x80110 Wr/Rd	AXITOE1G Control (AXITOE1G_CTRL)	Wr [0] – DDR から TOE1G-IP へのデータ送信を開始する [1] – TOE1G-IP から DDR へのデータ受信を開始する Rd [0] – DDR から TOE1G-IP へのデータ送信実行中を示すビジー・フラグ [1] – TOE1G-IP から DDR へのデータ受信実行中を示すビジー・フラグ
BA+0x80114 Rd	ConnOn of TOE1G-IP CONNON_REG	Rd [0] –TOE1G-IP の ConnOn 信号出力をモニタ (ポート状態確認用)
BA+0x80118 Rd	FIFO Read Count of TOE1G-IP FFRDCNT_REG	Rd [15:0] – TOE1G-IP の TCPRxFfRdCnt 信号出力をモニタ (受信データ数をバイト単位でモニタ)

表 1: TOE1GRegIF のレジスタ・マップ

注意 : BA は AXI4-Lite インターフェイスのベース・アドレスであり FPGA プラットフォームによって異なります

2.2 AXIEMAC

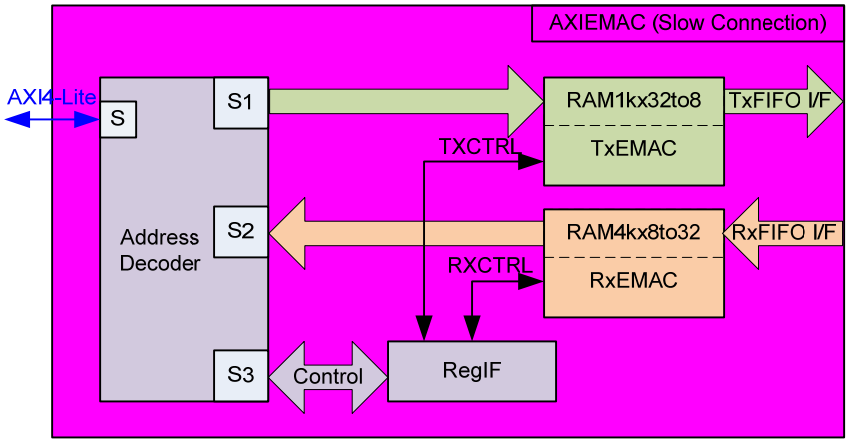

図 4: AXIEMAC ブロック図

低速コネクションは小パケットで通信の頻度が低い制御/ステータス情報の転送に使われます。 AXIEMAC のメモリ・マップは表 2 に示すように3種類のメモリ空間に分割され、それは TxEMAC(TxRAM)空間、RxEMAC(RxRAM)空間、そして一般レジスタです。TxRAM と RxRAM は CPU と EMAC 間で転送するイーサネット・パケットを格納します。

アドレス Rd/Wr	レジスタ名 (“ftp_demo.c”内のラベル名)	説明
BA+0x00000- BA+0x00FFF Wr	RAM in TxEMAC (TXRAM)	低速コネクションのイーサネット送信データを格納する RAM 空間
BA+0x10000- BA+0x10FFF Rd	RAM in RxEMAC (RXRAM)	低速コネクションのイーサネット受信データを格納する RAM 空間
BA+0x20000 Wr	Destination MAC address [31:0] (DSTMACL_REG)	FPGA の MAC アドレス下位 32 ビット
BA+0x20004 Wr	Destination MAC address [47:32] (DSTMACH_REG)	FPGA の MAC アドレス上位 16 ビット
BA+0x20008 Wr	Destination IP address (DSTIP_REG)	FPGA の IP アドレス 32 ビット
BA+0x20010 Wr	Rx Pattern enable (RXPATTEN_REG)	受信パケットのヘッダ比較イネーブル(‘1’: イネーブル, ‘0’: ディスエーブル) [0] – 23 バイト目の RXPATT23_REG[7:0]比較イネーブル [1] – 36 バイト目の RXPATT23_REG[7:0]比較イネーブル [2] – 37 バイト目の RXPATT23_REG[15:8]比較イネーブル
BA+0x20014 Wr	Rx Pattern Byte 23 (RXPATT23_REG)	[7:0] 受信パケットの 23 バイト目で比較するデータ・パターン
BA+0x20018 Wr	Rx Pattern Byte 34 (RXPATT36_REG)	[15:0] 受信パケットの 36-37 バイト目で比較するデータ・パターン
BA+0x20100 Wr	Start TXRAM address (TXADDR_REG)	送信イーサネット・データの TXRAM 先頭アドレス
BA+0x20104 Wr/Rd	Transmit length (TXLEN_REG)	Wr: 送信イーサネット・データの転送長をバイト単位で指定。有効な値は 2-4095 Rd: 未送信の転送数をバイト単位で表示。本レジスタの読み出し値がゼロになるまでポーリングすることで全データの送信完了を検出する。
BA+0x20200 Rd	Received RXRAM address (RXADDR_REG)	受信パケットの現在最終アドレス。 この値と前受信パケットの値を比較することで総受信データ・サイズをバイト単位で計算する。

表 2: AXIEMAC モジュール内の RegIF メモリ・マップ

注意 : BA は AXI4-Lite インターフェイスのベース・アドレスであり FPGA プラットフォームによって異なります

2.2.1 TxEMAC

AXITOE1G 内の Ff32to8 と同様 TxRAM (RAM1kx32to8)はデータ・バッファとして使用され、また AXI4-Lite からの 32 ビット・データを 8 ビット・データに変換します。小量のデータ送信モジュールにより、RAM からのデータがリードされ EMAC に送信するためパケットとして整えます。 パケット・サイズと先頭データの RAM アドレスは CPU によって RegIF を通して設定されます。 さらに CPU はレジスタ・アクセスを介して転送データ残量をモニタすることで転送の進捗を確認できます。

RAM1kx32to8 はシンプルなデュアルポート RAM です。書き込みインターフェイスは 32 ビットのバス・サイズで、CPU からみて BA(ベース・アドレス)から、BA+0xFFF の範囲にマップされます。読み出しインターフェイスは 8 ビットのバス・サイズで TxEMAC モジュール内の内部ロジックと接続されます。

2.2.2 RxEMAC

AXITOE1G 内の Ff8to32 と同様 RxRAM (RAM4kx8to32)はデータ・バッファとして使用され、また EMAC I/F からの 8 ビット・データを CPU 向けに 32 ビット・データをに変換します。 このロジックはヘッダ・パケットのフィルタリング機能を持ち、CPU にとって有効なパケットのみを抽出します。フィルタリング機能の比較パターンは CPU からプログラム可能なロジック内に格納されます。
フィルタリングの比較パラメータにおいてロジック内の固定値は IP プロトコル・バージョンと IP ヘッダ長です。IP プロトコル・バージョンは IPv4 固定であり IP ヘッダ長は 20 バイト固定です。 CPU からプログラム可能な比較パラメータは、通信相手 MAC アドレス、通信相手 IP アドレス、プロトコル種類、通信相手ポート番号です。プロトコル種類と通信相手ポート番号は比較対象としてイネーブルするかしないかを選択できますが、それ以外の比較パラメータは常時比較対象がイネーブルとなります。

PING コマンドをサポートするため、ヘッダのフィルタリングは以下の値に設定されます。

- 通信相手 MAC アドレス= ブロードキャスト ID または CPU からセットされた値
- 通信相手 IP アドレス= CPU からセットされた値
- プロトコル種類= ICMP (RXPATT23_REG=0x01, RXPATTEN_REG[0]='1')
- 通信相手ポート番号= ディスエーブル(RXPATTEN_REG[2:1]="00")

上記の設定により、受信パケットのうち ICMP パケットのみが抽出され RxRAM 内に格納されます。

RAM4kx8to32 はシンプルなデュアルポート RAM です。 書き込み側インターフェイスは 8 ビットのバス・サイズで EMAC I/F と接続されます。読み出し側インターフェイスは 32 ビットのバス・サイズで CPU システムから見てアドレス= (BA+0x10000) – (BA+0x10FFF)にマッピングされます。

RxRAM の書き込み側アドレスは内部ロジックにより自動インクリメントし、受信パケットを連続して格納します、このため CPU は先に読み出したパケットの最終読み出しアドレスを記憶しておき、次の有効パケットを正しく読み出す必要があります。 RAM の書き込みアドレスと読み出しアドレスを比較することで、CPU は受信したパケット・サイズを計算できます。

2.2.3 RegIF

このモジュールは TxEMAC と RxEMAC の動作を制御し、あるいは現在状態のモニタを実行するため、アドレスとデータをデコードします。

3 CPU ファームウエア

本デザインにて CPU ファームウエアでは以下3種類の動作を実装します。
(1) 高速コネクションによるデータの送信テスト
(2) 高速コネクションによるデータの受信テスト
(3) 低速コネクションによる PING コマンドのテスト

ユーザはシリアル・コンソールを通して動作を選択します。 本2ポート・デザインでの CPU 動作シーケンスは以下となります。
(1) TOE1G-IP コアと AXIEMAC モジュールに対して MAC アドレス、IP アドレス、ポート番号等のネットワーク・パラメータを初期化します。
(2) TOE1G-IP コアの初期化が完了するのを待ちます。
(3) ユーザからのコンソール入力を受信します。
(4) 入力されたパラメータをハードウエアにセットし動作を開始します。
(5) FPGA ボードはサーバ・モードで動作するため、ポートは接続 PC 上のテスト・アプリケーションによりオープンされます。
(6) テスト動作が完了するまでテストを実行します。

高速コネクション用として DDR3 メモリ空間の上位側 512M バイトを送信/受信テストのデータ・バッファとして使います。このためユーザから指定された転送サイズが 512M バイトを超える場合、データ・ベリファイ機能は使えません。 また、高速コネクションのテストにて最大転送サイズは 4G バイトです。

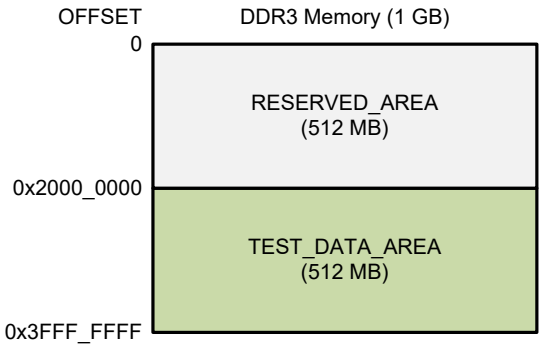

図 5: DDR3 メモリ・マップ

3.1 PING コマンド・テスト

PING コマンドは低速コネクション側において送信と受信の両機能をテストするために使います。PING コマンドで使われる ICMP プロトコルの IP ダイアグラムを下図 6 に示します。

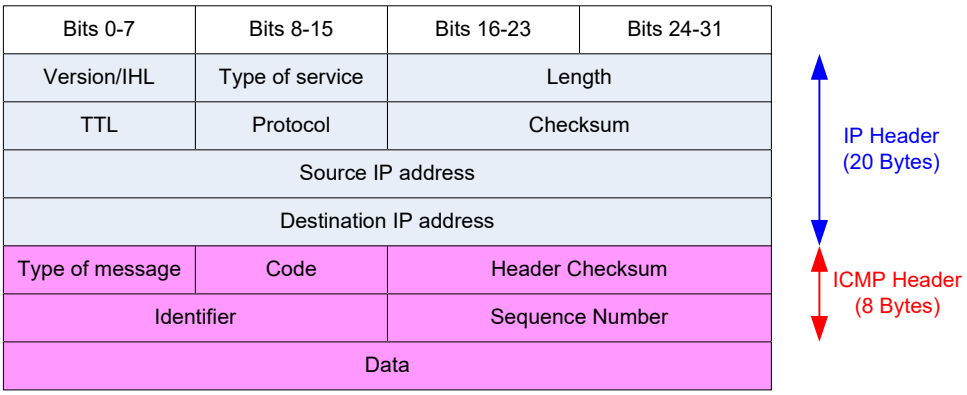

図 6: ICMP の IP ダイアグラム

PING コマンド・テストにおいて PC から FPGA に対してエコー要求（タイプ=8）を送信し、FPGA ではエコー応答(タイプ=0)を PC に返送します。エコー応答の全データはエコー要求からコピーされます。PING コマンドのより詳細な情報については以下のウェブサイトで参照できます。

http://en.wikipedia.org/wiki/Ping_(networking_utility)

ファームウエアのシーケンスは以下となります。

1) RXADDR_REG をポーリングでモニタし RXRAM 内に有効な受信データが格納されるのを待ちます。
2) ファームウエアにて受信パケットを一次バッファにダンプします。“TMPBUF_SIZE”パラメータ値により定義された 256 バイト分のみ確保されます。
3) IP と ICMP チェックサムを計算しパケットが有効であることを確認します。チェックサム値が正しくない場合はエラーを返送します。
4) エコー応答パケットを TxRAM に準備します。送信データは受信パケットからコピーしますが IP および ICMP チェックサムは再計算する必要があります。
5) 低速コネクション用の AXIEMAC モジュールにてレジスタをセットしデータの送信を開始します。

3.2 高速データ受信テスト

このテストでは“send_tcp_client”アプリケーションによりPCからFPGAに対して高速コネクションでテスト・データが転送されます。DDR3エリア内のうち512Mバイト分がデータ・バッファとして使われるので512Mバイト以上の受信テストではファームウエアでのデータ・ベリファイ機能はディスエーブルされます。ファームウエアのシーケンスは以下の通りです。

1) CONNON_REGレジスタをモニタしコネクションが確立するのを待ちます、この CONNON_REG は PC 側で“send_tcp_client”アプリケーションが実行されるとセットされます。
2) TOE1G-IP コア内受信データ FIFO に受信データが溜まるのを待ちます。
3) DDR3 の開始アドレスと転送サイズを計算し、転送開始フラグと合わせて AXITOE1G レジスタにセットします。
4) データ転送が完了するのを待ちます。
5) PC 側からコネクションがクローズされるまで上記 2)～4)のステップをループします。
6) 総受信データ数が 512M バイト以内の場合、データ・ベリファイを行うメニューを表示します。

3.3 高速データ送信テスト

このテストでは FPGA から PC へ向けてデータが転送されます。PC 側にて FPGA からの転送データを受信・ベリファイするため“recv_tcp_client_single”アプリケーションが使われます。 先に説明した高速データ受信テストと同様、転送サイズが512Mバイト以内であった場合 PC 側で受信データのベリファイが可能です。ファームウエアのシーケンスは以下の通りです。

1) DDR3 内にテスト・データを用意します。
2) ユーザが入力した転送サイズとパケット長を TOE1G-IP レジスタにセットします、そして TOE1G-IP 動作をスタートさせます。
3) 転送サイズが512Mバイト以上の場合、AXIIF の転送サイズは複数ループに分割され DDR3 アドレスがオーバーラップしないようにします。各ループ動作にて転送サイズと DDR3 のスタート番地は再計算されます。
4) 全てのデータ転送が完了するのを待ちます、そして TOE1G-IP レジスタにコネクションのクローズを指示します。
5) コネクションがクローズするのを待ちます。

4　注意事項

このデザインは簡易テストであるため高速コネクションと低速コネクションを同時に実行することはできません。ただしユーザ側にてファームウエアを編集することで高速と低速の両コネクションを並行して同時に実行することは可能です。両コネクションの同時実行例については FTP サーバー・デモを参照してください。

5　改版履歴

リビジョン	日付	説明
1.0	9-Jan-15	Initial version release
1.1	28-Oct-15	Correct Table1 description
1.1J	2016/1/5	日本語版の初期版リリース (英語版 Rev.1.1 の翻訳版)
1.2J	2016/9/1	製品名の変更(TOE2-IP → TOE1G-IP)