

UDP10G-IP リファレンス・デザイン説明書 (Intel 版)

Rev1.1J 2017/11/20

1. UDP プロトコル概要

TCPと比較するとUDPプロトコルは最小限のプロトコル・メカニズムでメッセージの送信を可能としています、ただしハンドシェイクがないためデータは保証されません。TCPと同様UDPにおいてもデータ内容の信頼性確認のためチェックサム機能を持ち、データ送受信ダイアグラムで異なる機能をアドレッシングするためポート番号が用いられます。

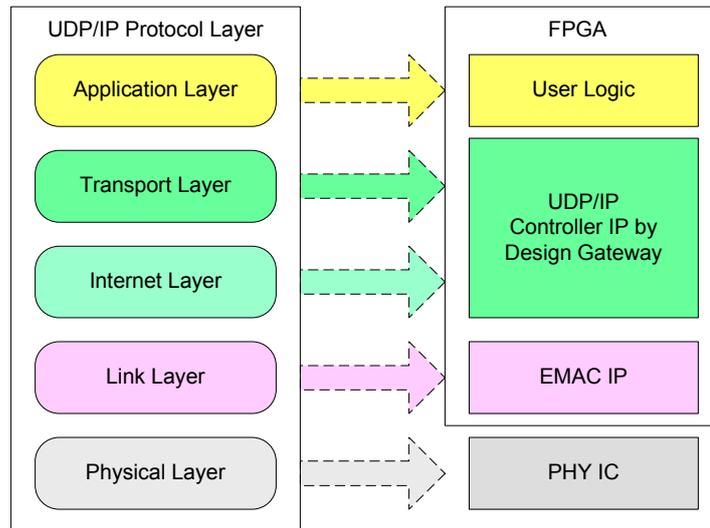


図 1-1: UDP プロトコルのレイヤ図

UDP10G-IP コア(以下本 IP コア)は UDP/IP プロトコルにおいてトランスポート層とインターネット層を実装します。送信機能において本 IP コアはユーザ・ロジックからの UDP データを用意し UDP/IP ヘッダを付加して EMAC を通して送信するイーサネット・パケットを生成します。受信機能においては、本 IP コアはイーサネット・パケットから UDP データを抽出します。このとき UDP/IP ヘッダは有効なパケットであるかチェックされます。パケットが有効であった場合 UDP データが抽出されユーザ・ロジックから読み出せるようデータ・バッファに格納されます。

より低いプロトコル層は Intel の 10Gb イーサネット MAC および 10Gb BASE-R PHY で実装します。

本リファレンス・デザインは簡単なユーザ・ロジックと接続した本 IP コアを使ってデータの送信・受信を評価するシステムを提供します。本システムは Intel 評価ボードおよびテスト・アプリケーションを搭載したテスト PC で高い転送パフォーマンスが得られることを検証します。より詳細については以下で説明します。

本デザインにて CPU ファームウェアは OS を使わないベア・メタルで実装されます。ユーザは NiosII ターミナル経由で各パラメータや転送方向を指定しテストを開始します。PC 側のテスト・アプリケーションは"UDPDataTest.exe"です。

2. ハードウェアの説明

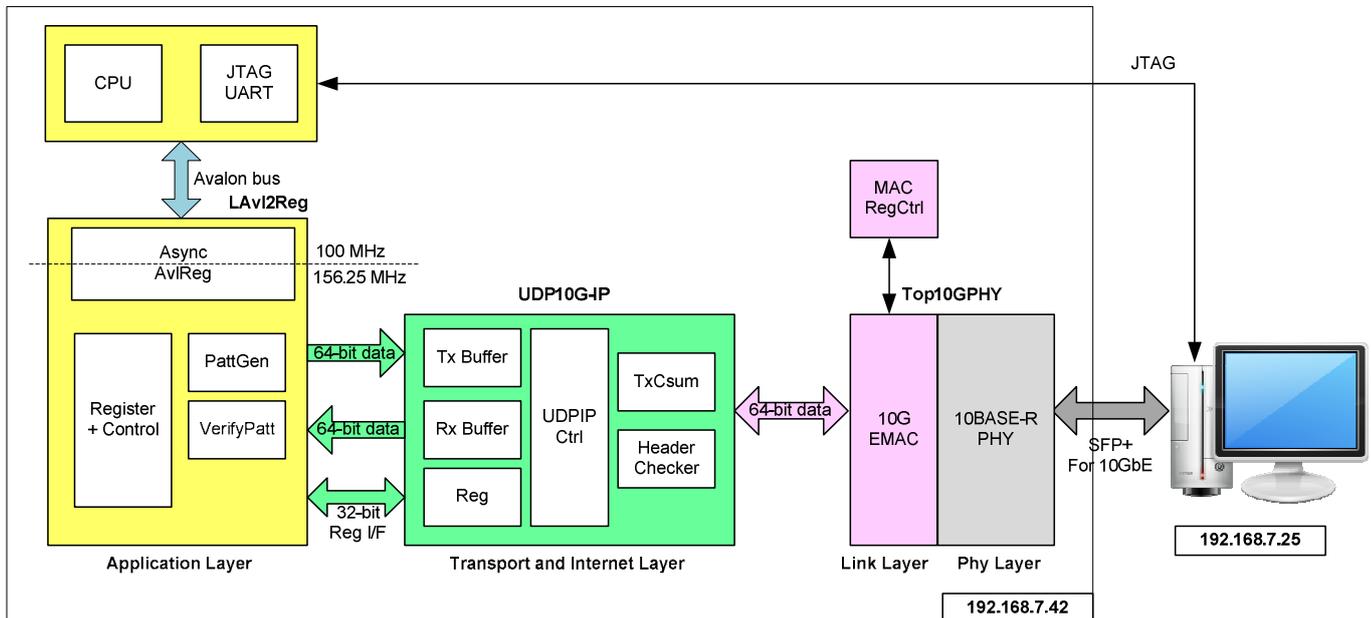


図 2-1: リファレンス・デザインのハードウェア構成

図 2-1 に示すようにハードウェア全体は大きく 4 つのモジュールに分かれており、UDP 各々のプロトコル層をサポートします。本 IP コアは EMAC および 10Gb BASE-R PHY とあわせて UDP プロトコルの低層を実装します。UDP/IP はハンドシェイク・パケットを使わないため、本 IP コアは送信と受信が同時に実行できます。ただし PC 側の CPU で 10Gb 速度ものデータを送受信同時に処理するのは現実的ではなくデータ・ロスが発生するため本リファレンス・デザインでは方向のみサポートします。

FPGA から PC への送信方向において UDP データは LAvl2Reg モジュール内のパターン発生器で生成されます。PC 側では受信したデータをテスト・アプリケーション内でベリファイします。PC から FPGA への受信方向においては、PC 上のテスト・アプリケーションで生成されたテスト・パターンは LAvl2Reg モジュール内でベリファイします。LAvl2Reg モジュールはアダプタ・ロジックを内蔵し CPU システムの Avalon バス・インターフェイスを本 IP コアのレジスタ・インターフェイスに変換します。このためユーザはシリアル・コンソールを介して本 IP コアのパラメータをセットしステータスをモニタできます。より詳細については以下に説明します。

- 10Gb EMAC コアおよび 10Gb BASE-R PHY

10Gb イーサネットのリンク層と物理層はどちらも Intel 製 IP コアで実装されます。10Gb EMAC はリンク層を、10Gb BASE-R は物理層をそれぞれ担当します。10Gb EMAC のデータ・バスは 64 ビット Avalon ストリーム・インターフェイスです。10Gb EMAC と 10Gb BASE-R PHY の詳細については以下の Intel サイトを参照してください。

<https://www.altera.com/products/intellectual-property/ip/interface-protocols/m-alt-10gbps-ethernet-mac.html>
<https://www.altera.com/products/intellectual-property/ip/interface-protocols/m-alt-10gbase-r-pcs.html>

● UDP10G-IP

本 UDP10G-IP コアの詳細仕様については以下の UDP10G-IP コア・データシートを参照してください。
http://www.dgway.com/products/IP/UDP10G-IP/dg_udp10gip_data_sheet_intel_jp.pdf

● MACRegCtrl

このモジュールはパラメータを 10Gb イーサネット MAC に設定し EMAC ステータスを Avalon-MM バスを介してモニターするためのものです。内部ロジックはシステム起動時にイーサネット MAC を初期化するため 1 度だけ実行するシンプルなステート・マシンでデザインされています。

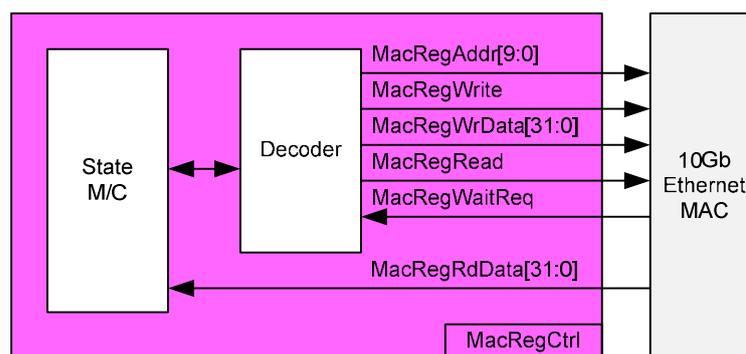


図 2-2: MACRegCtrl ブロック図

初期化シーケンスで以下を実行します。

- 1) EMAC の受信パスをディスエーブルにします
- 2) EMAC の送信パスをディスエーブルにします
- 3) 受信パスがアイドル状態であることを確認します
- 4) 送信パスがアイドル状態であることを確認します
- 5) 受信パスにて CRC とパディング削除を指定します
- 6) 送信パスにてフレーム送信の一時停止をディスエーブルにします
- 7) EMAC の受信パスをイネーブルにします
- 8) EMAC の送信パスをイネーブルにします

● LAvl2Reg

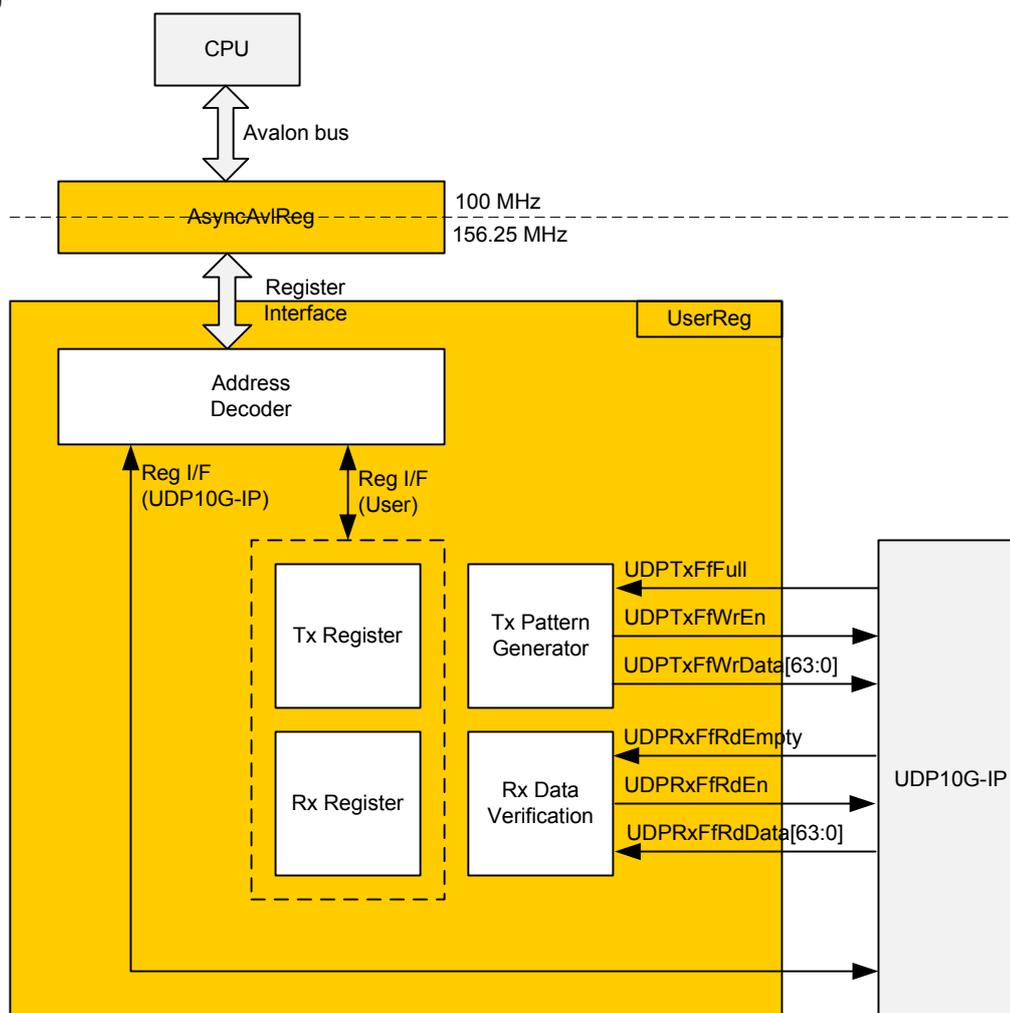


図 2-3: LAvl2Reg ブロック図

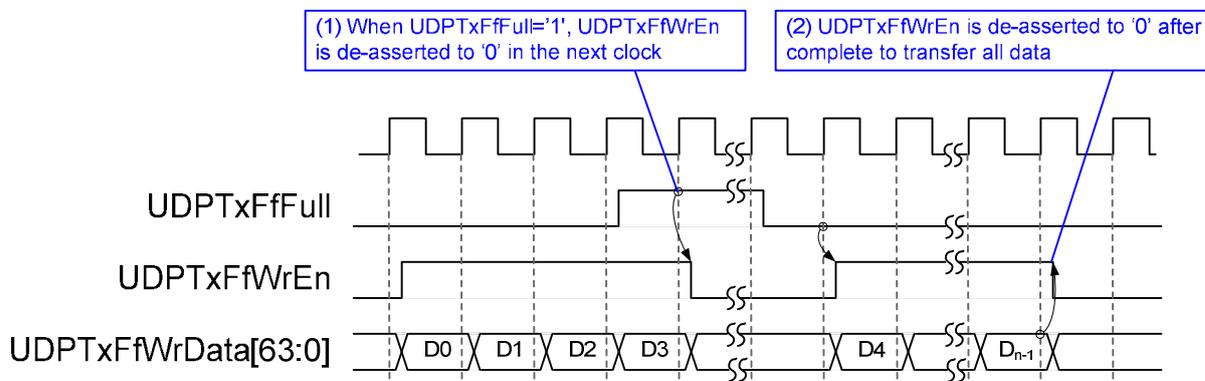
このモジュールは AsyncAvlReg と UserReg の 2 つのサブ・モジュールから構成されています。AsyncAvlReg モジュールは非同期クロック・ドメイン間の通信をサポートします。また、CPU バスとなる Avalon-MM バスをレジスタ・インターフェイスに変換する機能も含まれます。UserReg モジュールは図 2-3 に示すように、アドレス・デコーダ、送受信レジスタ、パターン発生器、受信データ・ベリファイ、の 4 モジュールから構成されます。

- アドレス・デコーダはレジスタ・マップを 2 エリアに分割し表 2-1 に示すように、0x0000~0x00FF を UDP10G-IP コアの内部レジスタにアサインし、0x1000~0x10FF を送信パターン発生器と受信データ・ベリファイ回路への制御用内部レジスタにアサインします。
- ライト・レジスタは総転送サイズ設定、データ・ベリファイ機能イネーブル、ステータス・フラグのクリア、内部ロジックのリセットに使われます。リード・レジスタは現在の転送サイズ、エラー・テスト・パターン、割り込みフラグ、エラー・フラグに使われます。

表 2-1: レジスタ・マップ定義

アドレス Wr/Rd 種類	レジスタ名 (“udp10cpu_demo.c” のラベル)	説明 (Wr:ライト定義, Rd:リード定義)
BA+0x00 – BA+0xFF: UDP10G-IP コア内部レジスタ・エリア (BA=0x0030_0000) ※各レジスタの詳細については UDP10G-IP コア・データシートの表 2 を参照してください。		
BA+0x00	UDP10_RST_REG	UDP10G-IP コアの RST レジスタ
BA+0x04	UDP10_CMD_REG	UDP10G-IP コアの CMD レジスタ
BA+0x08	UDP10_SML_REG	UDP10G-IP コアの SML レジスタ
BA+0x0C	UDP10_SMH_REG	UDP10G-IP コアの SMH レジスタ
BA+0x10	UDP10_DIP_REG	UDP10G-IP コアの DIP レジスタ
BA+0x14	UDP10_SIP_REG	UDP10G-IP コアの SIP レジスタ
BA+0x18	UDP10_DPN_REG	UDP10G-IP コアの DPN レジスタ
BA+0x1C	UDP10_SPN_REG	UDP10G-IP コアの SPN レジスタ
BA+0x20	UDP10_TDL_REG	UDP10G-IP コアの TDL レジスタ
BA+0x24	UDP10_TMO_REG	UDP10G-IP コアの TMO レジスタ
BA+0x28	UDP10_PKL_REG	UDP10G-IP コアの PKL レジスタ
BA+0x38	UDP10_SRV_REG	UDP10G-IP コアの SRV レジスタ
BA+0x1000 – BA+0x10FF: 制御/ステータス UserReg エリア (BA=0x0030_0000)		
BA+0x1000 Wr/Rd	総送信サイズ (USER_TXTDL_REG)	Wr [31:0] – 総転送サイズをバイト単位で指定、この値は 8 の倍数とする必要がある、このため bit[2:0]は必ず“000”とすること。 Rd [31:0] – 現在の送信済みデータ数をバイト単位で示す。
BA+0x1004 Rd	総受信サイズ (USER_RXTDL_REG)	Rd [31:0] – 現在の受信済みデータ数をバイト単位で示す。
BA+0x100C Wr/Rd	ユーザ制御 (USER_CTRL_REG)	Wr [0] – 送信開始、'1'をライトして送信動作を開始する。このビットはセルフ・クリーニングのため'0'をライトする必要はない。 [1] – データ・ベリファイのイネーブル (‘0’: ベリファイ機能イネーブル, ‘1’: ベリファイ機能ディスエーブル) Rd [0] – 送信ビジー (‘0’: アイドル, ‘1’: 送信中)
BA+0x1010 Wr/Rd	ユーザ・エラー状態 (USER_ERR_REG)	Wr [0] – エラー・フラグのクリア、'1'をライトして全エラー・フラグをクリアする。このビットはセルフ・クリーニングのため'0'をライトする必要はない。 Rd [0] – データ・ベリファイ・エラー (‘0’: 通常, ‘1’: エラー). このビットはユーザが次の動作を開始するかリセットすることで自動的にクリアされる。 [1] – UDP10G-IP コアの TimerInt 出力をラッチ (‘0’: TimerInt の発生なし, ‘1’: TimerInt='1'が検出された) このビットはシステム・リセットかユーザ回路が USER_ERR_REG[0]='1'をライトするとクリアされる。

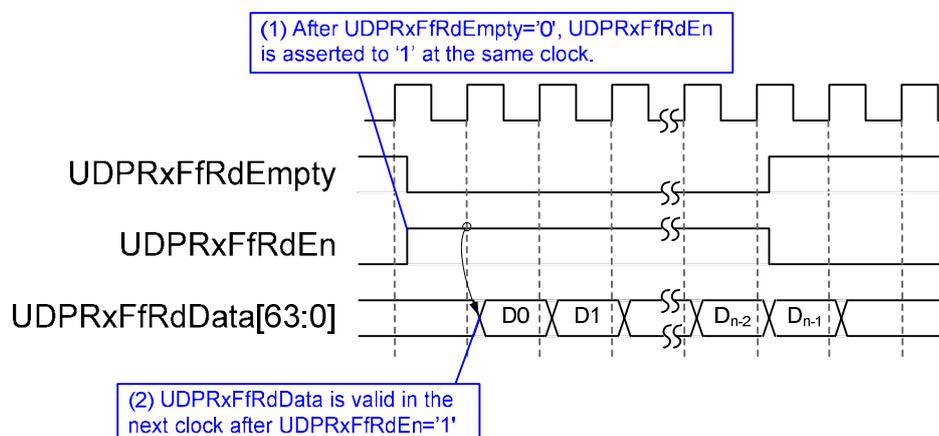
- c) 送信パターン発生回路は32ビットのインクリメンタル・テスト・パターンを発生しUDPTxFf(UDP10G-IPコアの送信データ)インターフェイスへ出力します。コアの送信 FIFO が一杯になり UDPTxFfFull が検出されるとデータ発生は一時停止します。図 2-4 に示すように UDPTxFfFull='1'または全データの送信完了で UDPTxFfWrEn は'0'ネゲートされます。



- (1) UDPTxFfFull='1'アサートされると次クロックで UDPTxFfWrEn は'0'ネゲートします。
 (2) 全データの送信が完了すると UDPTxFfWrEn は'0'ネゲートします。

図 2-4: 送信パターン発生回路のタイムチャート

- d) 受信データ・ベリファイ回路は UDPRxFfRdEmpty の状態をモニタします。この信号が'0'ネゲートされている場合、UDPRxFfRdEn'1'にアサートされ本 IP コアの受信 FIFO からデータを読み出します。受信データは次のクロック期間で有効な値が出力されベリファイ機能がイネーブルとなっていた場合に期待パターンと比較されます。送信データ・パスと同様、期待パターンは 32ビットのインクリメンタル・パターンです。リード・データが期待パターンと異なる場合ベリファイ不一致のフラグが'1'にアサートされます。この不一致フラグシステムのリセットまたは USER_ERR_REG[0]によりユーザ・ロジックのリセットでクリアされます。



- (1) UDPRxFfRdEmpty='0'の場合同じクロック期間で UDPRxFfRdEn='1'アサートされます。
 (2) UDPRxFfRdData は UDPRxFfRdEn='1'の次クロック期間で有効となります。

図 2-5: 受信データ・ベリファイのタイムチャート

3. CPU ファームウェアの動作シーケンス

CPU ファームウェアのメインメニューは以下 3 つの動作があります。

- 1) IP コアのリセット: このメニューは本 IP コア内パラメータすなわち FPGA 側 MAC アドレス/IP アドレス/FPGA 側ポート番号/ターゲット IP アドレス/ターゲット側ポート番号を変更するために使われます。全てのレジスタを設定した後本 IP コアはリセットを解除し初期化プロセスを開始します。CPU は本 IP コアからのビジー・フラグを関し初期化プロセスが完了したことを確認します。
- 2) 送信テスト: このメニューが選択されると 2 つのユーザ入力を求めます、すなわち最大 4G バイトの総転送サイズと 8 の倍数で指定するパケット・サイズです。その後データは PC へ転送されます。PC ではテスト・アプリケーションが実行され、FPGA からのデータを受信・ベリファイします。コンソール上では 1 秒毎に総受信データ数が表示されます。転送が完了すると総転送サイズとパフォーマンス結果が表示されます。
- 3) 受信テスト: このメニューが選択されると UserReg モジュールにてベリファイ機能を使うか使わないかを指定できます。UDP プロトコルのためデータがロスしベリファイでエラーが発生することがあります。送信テストと同様、転送完了後シリアル・コンソール上に総転送サイズとパフォーマンス結果が表示されます。

3.1 IP コアのリセット

このメニューの動作シーケンスは以下となります。

- 1) コンソール上に現在のパラメータ設定値を表示します。
- 2) ユーザからパラメータが入力されると有効な値かを評価します。不適当な値の場合現在のパラメータ値は変更しません。
- 3) UDP10_RST_REG[0]='1' にセットし本 IP コアをリセットします、そしてパラメータ設定が全て完了するまでリセット状態を継続します。
- 4) リセット状態を維持したままユーザからの入力パラメータを本 IP コアのレジスタすなわち UDP10_SML_REG, UDP10_DIP_REG, UDP10_SPN_REG, UDP10_TMO_REG にセットします。
- 5) UDP10_RESET_REG[0]='0' としてコアのリセット状態を解除します。
- 6) 本 IP コアのビジー・フラグ (UDP10_CMD_REG[0]) をモニタし、このフラグが '0' にネゲートされ初期化シーケンスが完了するまで待機します

3.2 送信テスト

このメニューの動作シーケンスは以下となります。

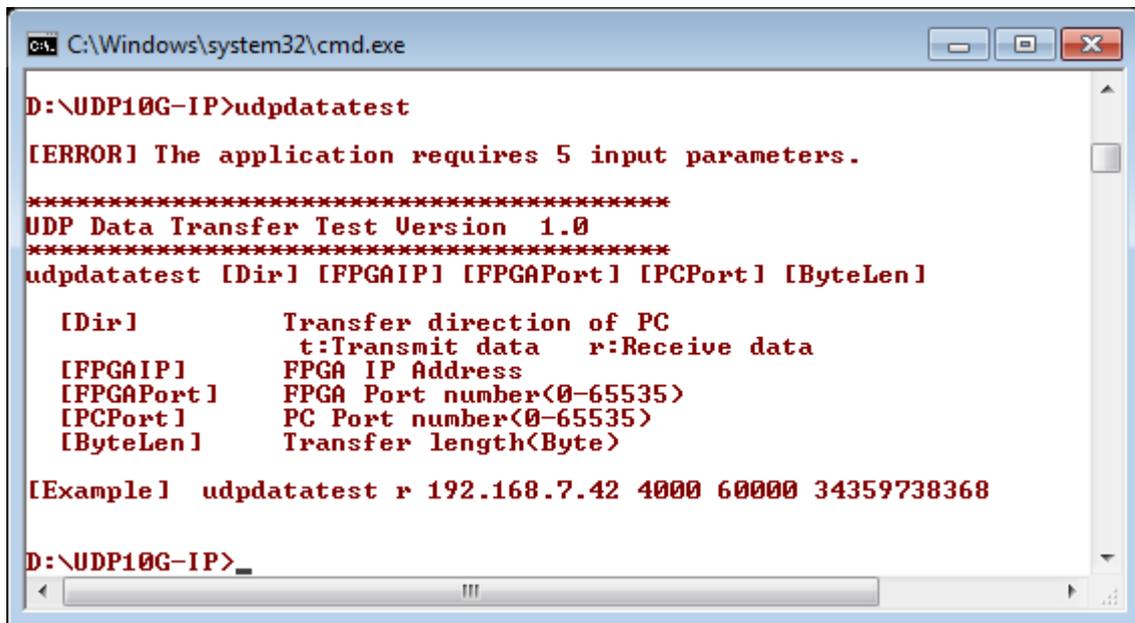
- 1) ユーザから入力された送信転送サイズとパケット・サイズが有効値であることを確認します。
- 2) UserReg モジュール内レジスタすなわち総送信サイズ (USER_TXTDL_REG) をセットし、テスト・パターンの初期値や各フラグを (USER_ERR_REG) でリセットし、USER_CTRL_REG[0]='1' としてテスト・パターン発生を開始します。
- 3) CPU は UDP レジスタすなわち UDP パケット長 (UDP10_PKL_REG), UDP 総転送長 (UDP10_TDL_REG) をセットし、UDP10_CMD_REG[0]='1' としてデータ送信を開始します。
- 4) IP コアのビジー・フラグ (UDP10_CMD_REG[0]) がネゲートされ送信プロセスが完了するまで待ちます。
- 5) 送信実行中は USER_TXTDL_REG[31:0] で転送済みサイズをモニタし 1 秒毎にシリアル・コンソールに表示します。
- 6) 送信が完了したらタイマーを停止しパフォーマンス結果をコンソールに表示します。

3.3 受信テスト

このメニューの動作シーケンスは以下となります。

- 1) ユーザから入力されたベリファイ・モードが有効値であることを確認します。
- 2) UserReg エリアのレジスタすなわちベリファイ機能イネーブル(USER_CTRL_REG[1])を設定し、テスト・パターンの初期値や各フラグを(USER_ERR_REG)でリセットします。
- 3) 総受信サイズ を(USER_RXTDL_REG)をモニタし、ゼロでなくなることで最初のパケット受信を検出してタイマーを開始します。
- 4) 総受信サイズ (USER_RXTDL_REG)を継続してモニタし、その値が 1msec 以上変化しない状態となるまで待機します。
- 5) コア内のタイムアウト発生を (USER_ERR_REG[1])で確認し、ベリファイ機能がイネーブルされていた場合はベリファイ・フラグ (USER_ERR_REG[0])をチェックします。
- 6) タイマーを停止しテスト・パフォーマンスをコンソールに表示します。

4. PC 側テスト・アプリケーション



```

C:\Windows\system32\cmd.exe
D:\UDP10G-IP>udpdatatest

[ERROR] The application requires 5 input parameters.

*****
UDP Data Transfer Test Version 1.0
*****
udpdatatest [Dir] [FPGAIP] [FPGAPort] [PCPort] [ByteLen]

  [Dir]          Transfer direction of PC
                  t:Transmit data   r:Receive data
  [FPGAIP]       FPGA IP Address
  [FPGAPort]     FPGA Port number(0-65535)
  [PCPort]       PC Port number(0-65535)
  [ByteLen]      Transfer length(Byte)

[Example]  udpdatatest r 192.168.7.42 4000 60000 34359738368

D:\UDP10G-IP>_

```

図 4-1: udpdatatest アプリケーションのパラメータ

“udpdatatest”は PC 側の DOS アプリケーションで UDP データの送信・受信を実行します。テスト・モードや転送方向など全部で5つのパラメータを設定します。ここで設定するパラメータはFPGA側シリアル・コンソールのパラメータと一致させる必要があります。各パラメータの詳細を以下に説明します。

- 1) Dir: t – PC が FPGA に対してデータを送信します
 r – PC が FPGA からのデータを受信します
- 2) FPGAIP: FPGA 側の IP アドレス(デフォルト値は 192.168.7.42)
- 3) FPGAPort: FPGA 側のポート番号 (デフォルト値は 4000)
- 4) PCPort: PC 側のデータ送信または受信のポート番号
(デフォルト値は PC→FPGA が 60001、FPGA→PC が 6000)
- 5) ByteLen: 送信/受信の転送数をバイト単位で指定、UDP10G-IP コアの制約から 8 の倍数とする
 必要がある。

4.1 データ受信モード

アプリケーションの動作シーケンスは以下となります。

- (1) ユーザからの入力パラメータを取得します。
- (2) ソケットを作成し受信バッファのプロパティを設定します。
- (3) ユーザ入力パラメータより PC 側 IP アドレスとポート番号を設定し接続します。
- (4) 総受信データ数が設定値に到達するかあるいは 100 ミリ秒以内に新たなデータ受信がなくなるまで、受信データのベリファイを繰り返しループします。ベリファイ・パターンは 32 ビットのインクリメンタル・パターンで 0 から始まります。パターンは 32 ビットごとにインクリメントします。アプリケーションの実行中、1 秒毎に総受信サイズが DOS プロンプト上に表示されます。
- (5) 100 ミリ秒のタイムアウトで受信動作が終了した場合、“Timeout”のメッセージが表示され全ロス・データ量と総受信データ量が表示されます。

4.2 データ送信モード

アプリケーションの動作シーケンスは以下となります。

- (1) – (3)は上記データ受信モードと同一です。
- (4) 32 ビット・インクリメンタルのテスト・データ・パターンをバッファに作成しデータを送信します。パケット・サイズは 1472 バイト固定です。
- (5) 総送信サイズが設定値に到達すると送信動作を終了し、テスト結果として総転送数とパフォーマンスが DOS プロンプト上に表示されます。

5. 更新履歴

リビジョン	日付	説明
1.0	14-Sep-17	Initial Release
1.0J	2017/11/17	日本語翻訳版の初期版作成
1.1J	2017/11/20	英語版の記載修正に合わせて日本語版も修正

Copyright: 2017 Design Gateway Co,Ltd.