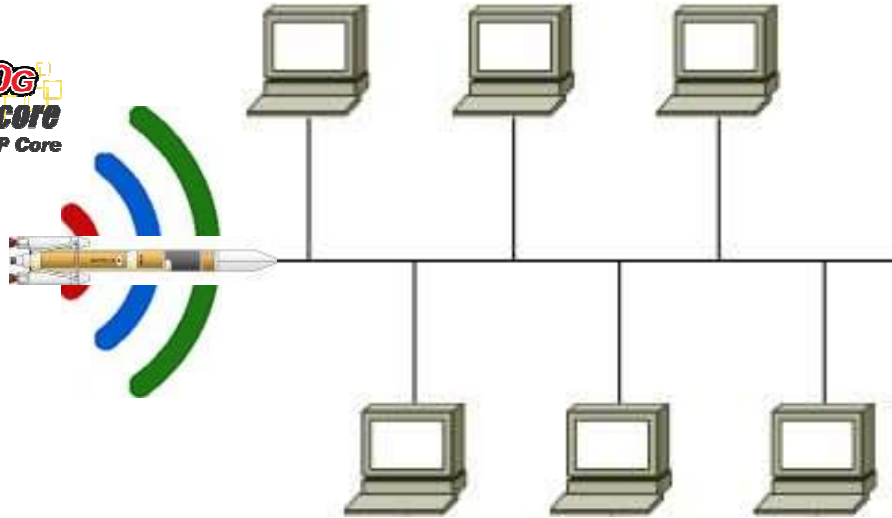


Xilinx版UDP10G-IPコアのご紹介

Ver1.1J

UDP 10G
IP core
User Datagram Protocol IP Core



純ロジックのIPコアで超高速UDP実装

2019/3/26

Design Gateway

Page 1

アジェンダ

- ・ UDPプロトコルの特徴や実装の課題点
- ・ UDP10G-IPコアの概要
- ・ コアの動作
 - 初期化
 - 高速送信
 - 高速受信
- ・ ユーザI/F・バッファ容量のパラメタライズ
- ・ リファレンス・デザイン
- ・ コア消費リソース・実機パフォーマンス
- ・ アプリケーション例



2019/3/26

Design Gateway

Page 2

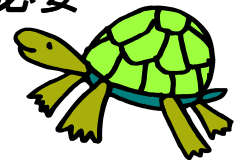
UDPプロトコルの特徴

- ・ 長所
 - 最小のオーバーヘッドにより高速かつ低レイテンシ
 - 1対複数のマルチ/ブロード・キャストが可能
 - 動画配信などリアルタイム重視のアプリに最適
- ・ 短所
 - 受信確認や再送がないので信頼性が保証されない
 - 信頼性維持のためにはアプリ側で対応する必要がある



CPUでのUDP実装課題

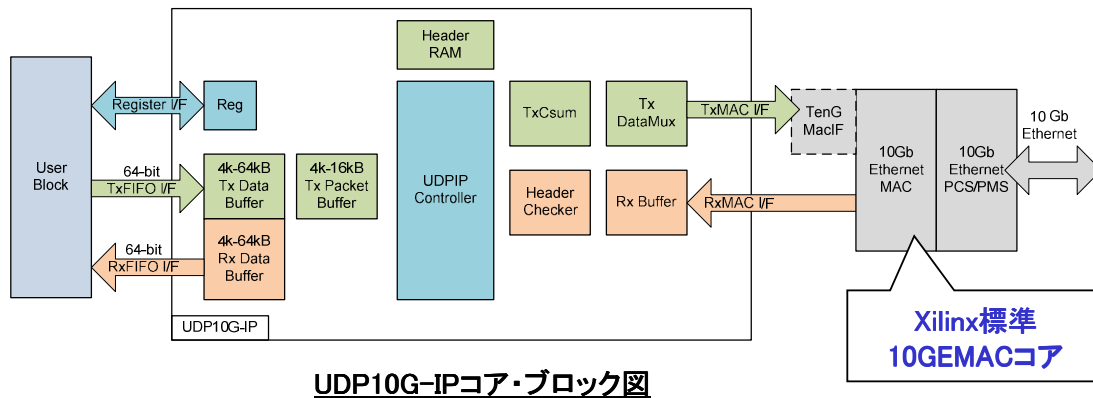
- ・ パフォーマンスやレイテンシに問題
 - チェックサム計算やヘッダ付加処理がファームで必要
 - CPUリソースを消費し他のタスクへ影響
 - ファーム処理なので転送性能が安定しない
- ・ Full Duplexでさらに問題が顕著化
 - 同時送受信の場合CPUが時分割で処理する必要がある
 - 送受信同時ファーム処理により転送性能が更に悪化
 - リアルタイム性が必要なアプリには致命的



⇒UDP10G-IPがこの問題を解決します！

UDP10G-IPコアの概要

- ・ 完全ハード・ワイヤード化した純ロジック・コア
- ・ ユーザ回路とXilinx製10GEMACコアの間に挿入
- ・ Full Duplex(送受信同時)通信をサポート



UDP10G-IPコアの特長1

- ・ UDP送受信処理を完全ロジック・ハードウェア化
 - CPUなしでの組込みシステム実装が可能
 - CPUシステムではCPU負荷がゼロ
- ・ 送信のみ/受信のみ/同時送受信を高速転送
 - 1200MByte/secを超える実パフォーマンス
- ・ 転送データの信頼性が維持できる
 - 送信時:チェックサムを自動計算・ヘッダに付加
 - 受信時:チェックサム結果の不一致でパケット自動破棄



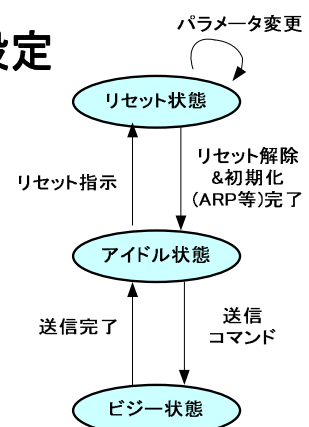
UDP10G-IPコアの特長2

- ・ データバッファ容量を選択可能
 - FPGAメモリ・リソースとパフォーマンスからユーザが選択
- ・ IPフラグメント受信に対応
 - 正しい順番のIPフラグメント・パケット受信が可能
- ・ 実機動作リファレンス・デザイン
 - Xilinx評価ボードで動作するプロジェクト
 - 購入前に実動作や実パフォーマンスの評価を検証可能
 - 製品のリファレンスはコア以外の全回路をソースで添付
- ・ マルチキャスト/ブロードキャスト送信に対応可能
 - カスタマイズにより複数ターゲットに同時送信



UDP10G-IPコアの動作概要

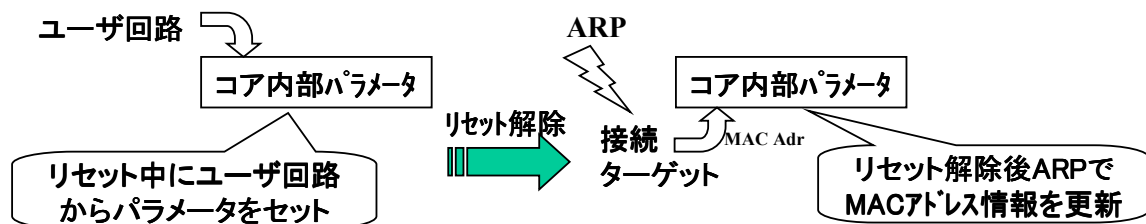
- ・ リセット状態でパラメータ(IP&MACアドレス等)を設定
- ・ リセット解除で初期化(ARP等)を実行
- ・ 初期化完了でアイドル(コマンド待ち)状態
- ・ ユーザ・コマンドにより送信動作
- ・ 受信は常時可能
(設定パラメータに合致するパケットは常時受信)
- ・ 送信と受信は独立して動作(**同時送受信可**)
- ・ パラメータ変更はリセット状態で実施
(転送長/パケット長はビジュー状態以外で変更可)



コアの状態遷移図

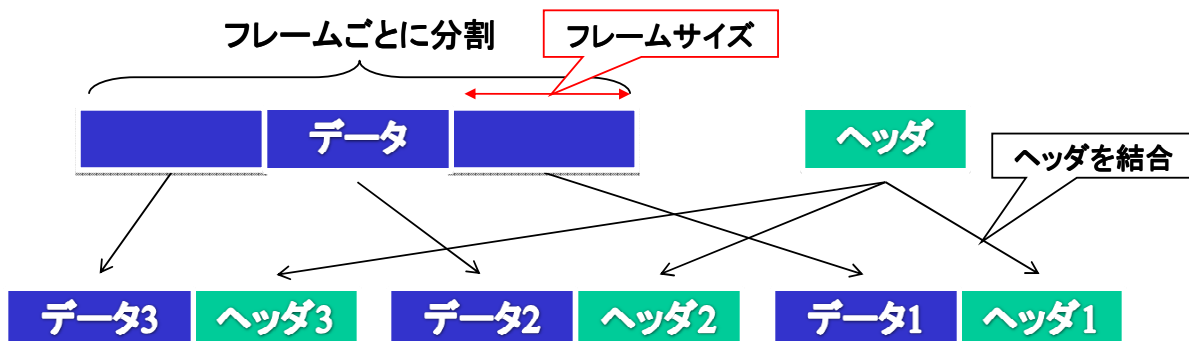
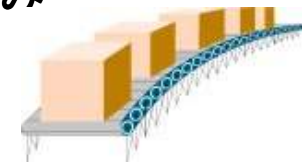
初期化動作

- ・ パラメータの初期値設定
 - コアのリセット維持中にユーザ回路より設定
 - IPおよびMACアドレス・ポート番号を指定
 - 設定を完了するとリセットを解除
- ・ リセット解除後ARP実行で相手側MACアドレス取得
 - クライアントの場合接続ターゲットに対してARPを発行
 - サーバーの場合ARP受信を待機



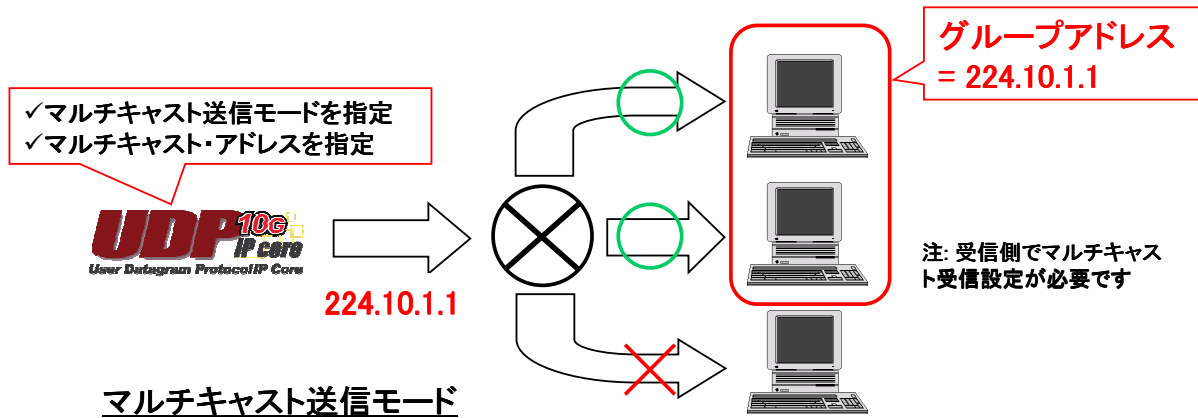
高速送信

- ・ 送信パケットの生成
 - ユーザ回路は送信データをFIFO I/Fで書込み
 - 送信データをフレームサイズで分割
 - チェックサムを自動計算しヘッダへ付加
 - ヘッダと送信データを結合し10GEMACへ出力



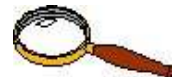
マルチ/ブロードキャスト高速送信(オプション)

- ・ カスタマイズでマルチ/ブロードキャスト送信をサポート
 - コア初期化時のARP自動実行を抑制
 - マルチ(ブロード)キャストIP/MACアドレスをコアに設定



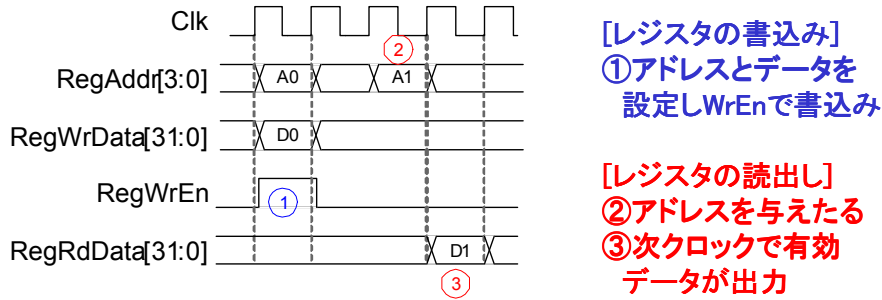
高速受信

- ・ 受信ヘッダのフィルタリング
 - MACヘッダ、IPヘッダ、UDPヘッダの全てを評価
 - 正しい順番のIPフラグメント・パケットを受信
- ・ チェックサムの自動計算と評価
 - 受信パケットからチェックサムをコア内で自動計算
 - 計算結果と受信パケット内チェックサムを評価
 - 不一致の場合パケットを破棄(無視)



ユーザ・インターフェース(制御)

- レジスタI/F、送信FIFO I/F、受信FIFO I/Fの3種類
 - レジスタI/Fは初期パラメータの設定、送信命令
 - 送信データ・受信データ用I/Fは標準的なFIFO I/F

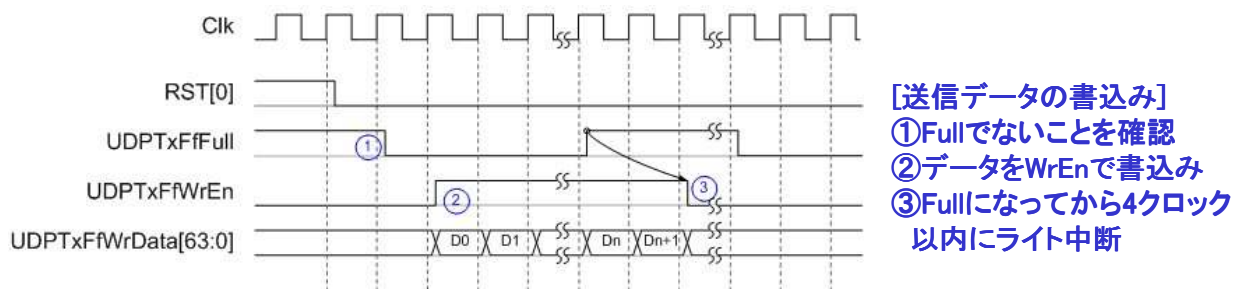


[レジスタの書込み]
①アドレスとデータを
設定しWrEnで書込み

[レジスタの読出し]
②アドレスを与えたる
③次クロックで有効
データが出力

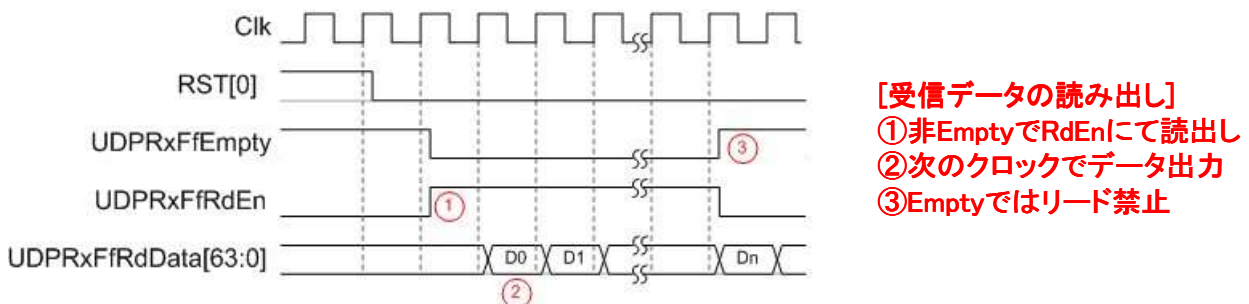
レジスタI/Fのタイムチャート

ユーザ・インターフェース(データ)



[送信データの書込み]
①Fullでないことを確認
②データをWrEnで書込み
③Fullになってから4クロック
以内にライト中断

送信FIFO I/Fのタイムチャート



[受信データの読み出し]
①非EmptyでRdEnにて読出し
②次のクロックでデータ出力
③Emptyではリード禁止

受信FIFO I/Fのタイムチャート

データ・バッファ容量の設定

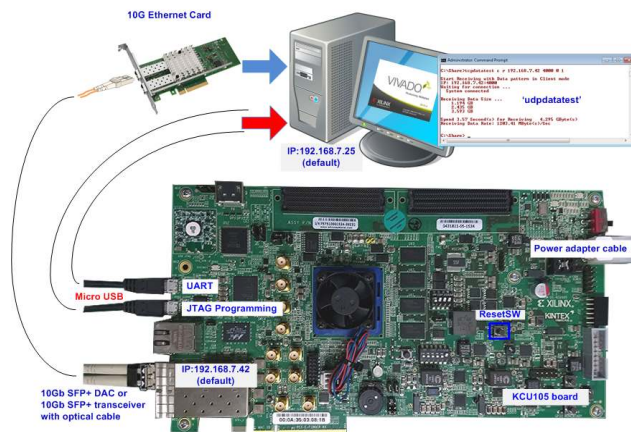
- ・ 3種類のデータ・バッファをパラメタライズで設定可能
 - ① 送信データ・バッファ: 4KByte~64KByte
 - ② 送信パケット・バッファ: 4KByte~16KByte
 - ③ 受信データ・バッファ: 4KByte~64KByte
- ・ リソースとパフォーマンスの最適点を調整できる

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・サイズをアドレス・ビット幅で設定します。例えば9の場合4Kバイト、13の場合64Kバイトとなります。
TxPacBitWidth	9-11	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9の場合4Kバイト、11の場合16Kバイトとなります。
RxBufBitWidth	9-13	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9の場合4Kバイト、13の場合64Kバイトとなります。

各データ・バッファはパラメタライズで設定できる

評価用BITファイル

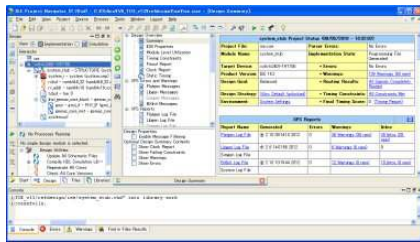
- ・ Xilinx評価ボードで動作するbitファイル
 - PC-評価ボード間での送信/受信の実機評価
 - 転送パフォーマンス測定・データベリファイ確認



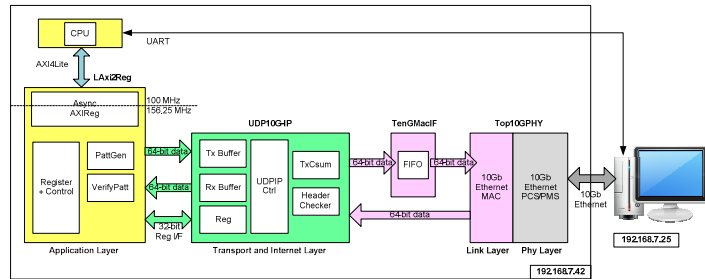
Xilinx評価ボード(KCU105)を使った実機検証環境

リファレンス・デザイン概要

- ・ 実機動作するVivadoデザイン・プロジェクト
 - 各デバイス・ファミリ標準のXilinx評価ボードで実装
 - 評価用BITファイルのプロジェクトをコア製品に添付
 - コア(ネットリスト)部以外の全回路をソースコードで提供



実動作するVivado/EDKプロジェクト



リファレンス・デザイン・ブロック図

リファレンスと実機評価による開発

- ・ リファレンス+評価ボードによる確実な開発
 - まず最初に製品添付のリファレンスで実機動作を確認
 - そこからユーザ製品に向け少しずつ編集
 - 編集ごとに実機動作をStep by Stepで確認
 - 問題があれば1ステップ前に戻るだけで動く状態にすぐ復帰できる



大きな後戻りがなく確実に短期間での製品開発が可能!

消費リソース

- コアの消費リソース

- 送信データ・バッファ=64KByte、送信パケット・バッファ=16KByte、受信データ・バッファ=64KByteの最大設定時



コンパイル結果例 (7シリーズ)

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices ¹	BRAMTile ²
Kintex-7	XC7K325TFFG900-2	156.25	1874	2098	772	36
Virtex-7	XC7VX485TFFG1761-2	156.25	1874	2093	760	36
Zynq-7000	XC7Z045FFG900-2	156.25	1874	2099	784	36

コンパイル結果例 (UltraScale シリーズ)

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	CLB ¹	BRAMTile ²
Kintex-UltraScale	XCKU040FFVA1156-2E	156.25	1862	2147	455	34.5

UDP10G-IPコア単体コンパイル結果

メモリ消費量は送受信ともバッファを最大に設定した場合です。
バッファ容量を削減すれば内部メモリ消費リソースを節約できます。

転送パフォーマンス

- データ送信/受信での実機パフォーマンス

```

COM5 - Tera Term VT
File Edit Setup Control Window Help
+++ UDP10G-IP Send Mode +++
Enter total tx size (aligned to 64-bit) : 8 - 0xFFFFFFFF8 => 0xFFFFFFFF8
Enter tx packet size : 8 - 8968 => 8968
Run updatatest application on PC by following command
updatatest r 192.168.7.42 4000 60000 4294967288
Press any key to start data sending
Send 1240.101 MB Recv 0.000 MB
Send 2480.152 MB Recv 0.000 MB
Send 3720.204 MB Recv 0.000 MB
Total = 42941MB, Time = 3463[ms], Transfer speed = 1240[MB/s]

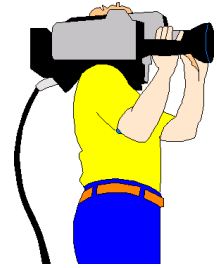
+++ UDP10G-IP Receive Mode +++
Input data verification mode : [0]-Disable [1]-Enable => 0
Wait data from PC
Run updatatest application on PC by following command
updatatest t 192.168.7.42 4000 60001 4294967288
Send 0.000 MB Recv 1239.399 MB
Send 0.000 MB Recv 2354.318 MB
Send 0.000 MB Recv 3468.121 MB
Total = 4291[MB], Time = 3600[ms], Transfer speed = 1192[MB/s]
    
```

送信(FPGA->PC)
1240MByte/secの高速転送

受信(PC->FPGA) 1192MByte/Sec
(PCの10GbE送信性能に依存)

UDP10G-IPのアプリケーション

- ・ ブロードキャストによる動画配信
 - リアルタイム性重視のストリーム・データ配信
 - 最小のオーバーヘッド/レイテンシが重要
 - 純ロジックによるUDP10G-IPで最大限のメリット
- ・ リアルタイム性の高いオンライン・ゲーム
 - ゲーム・データとユーザ操作情報の双方向通信
 - ユーザ操作性の維持に低レイテンシが必須
 - 同時双方向に対応するUDP10G-IPが最適



問い合わせ

- ・ ホームページに詳細な技術資料を用意
 - http://www.dgway.com/UDP10G-IP_X.html
- ・ 問い合わせ
 - 株式会社Design Gateway
 - E-mail : info@dgway.com
 - FAX : 050-3588-7915



