TOE10G-IP 標準(CPU 制御)デモ手順書(Intel 版)

Rev1.2J 2018/04/09

［目次］

1 概要

本ドキュメントは TOE1G-IP コアを使用し 10Gb イーサネット経由で TCP/IP プロトコルを通信する実機デモについて説明したものです。接続相手は図 1-1 に示す通り、テスト PC または TOE10G-IP コアを搭載した別の FPGA 評価ボードの 2 種類です。

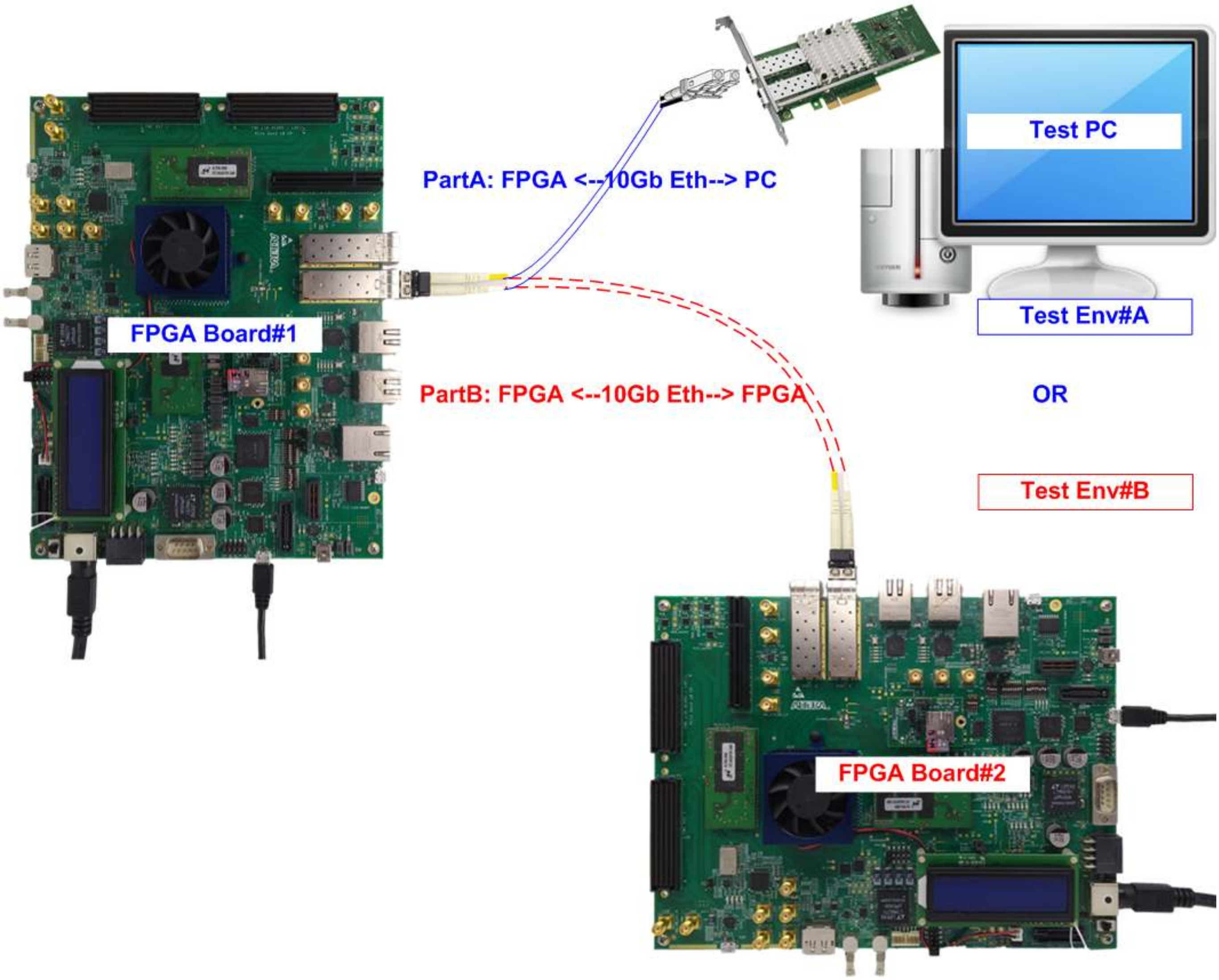

図 1-1: TOE10G-IP コアの 2 種類のデモ環境

本ドキュメントは 2 章に分かれております、すなわち前半は" Part A: FPGA と PC を使った TOE10G-IP コアのデモ"で、PC-FPGA 間の接続環境で評価するデモを説明し、後半は"PartB: 2 枚の FPGA ボードを使った TOE10G-IP コアのデモ"で、2 枚の Arria10SoC ボードを使ってのデモ手順を説明します。いずれのデモにおいても、FPGA ボードのテスト・パラメータは NiosII コマンド・シェルを使って JTAG UART 経由にて設定します。

Part A: FPGA と PC を使った TOE10G-IP コアのデモ

TOE10G-IP コアとテスト PC 間のデータ転送を実行するにあたって、ユーザは半二重通信か全二重通信かのいずれかのテストを選択できます。半二重通信テストすなわち TOE10G-IP コアから PC へのデータ送信または PC から TOE10G-IP コアへのデータ受信のいずれか片方では、"tcpdatatest.exe"アプリケーションをテスト PC 側で使用します。一方全二重通信テストすなわち PC と FPGA 間でデータの送信と受信を同時に実行する場合は、"tcp_client_txrx_10G.exe"アプリケーションをテスト PC 側で使用します。より詳細については以下を参照してください。

2 動作環境

TOE10G-IP コアの CPU 制御デモを実施するには図 2-1 に示した以下のテスト環境を用意してください。

1) FPGA 評価ボード、本手順書では Arria10SoC ボードで説明します。
2) 10Gbit イーサネット対応 PC(または 10Gbit イーサネット・カードを装着)
3) 10Gbit SFP+トランシーバおよび光ケーブル、あるいは SFP+ダイレクト・アタッチ・ケーブル
4) FPGA プログラム用の micro USB ケーブル
5) 評価ボードに対応するテスト・アプリケーションの"tcpdatatest.exe"および"tcp_client_txrx_10G.exe"(DesignGateway 社 Web ページからダウンロード)
6) PC にインストールした QuartusII プログラム(FPGA コンフィグレーション用)、FPGA 評価ボード上の発振器周波数設定用クロック・コントロールおよび NiosII コマンド・シェル

* 参考情報として、DesignGateway 社で実機確認した環境は以下の通りです

[1] 10G Network Adapter: Intel X520-DA2
http://www.intel.com/content/www/us/en/network-adapters/converged-network-adapters/ethernet-x520-server-adapters-brief.html
[2] a) 10-Gigabit SFP+ DAC cable (1m length)
http://www.netgear.com/business/products/switches/modules-accessories/axc761.aspx
b) 2x10-Gigabit SFP+ transceiver with optical cable
SFP+ transceiver (850nm)
http://www.fit-foxconn.com/Product/ProductDetail?topClassID=Electronic%20Module&&PN=AFBR-709SMZ
Optical cable 2105027-3 (LC to LC 1.8mm OM3 DPX LSZH&OFNR 3M)
[3] PC: Motherboard ASUS Z170-K, 32 GB RAM, 64-bit Windows7 OS

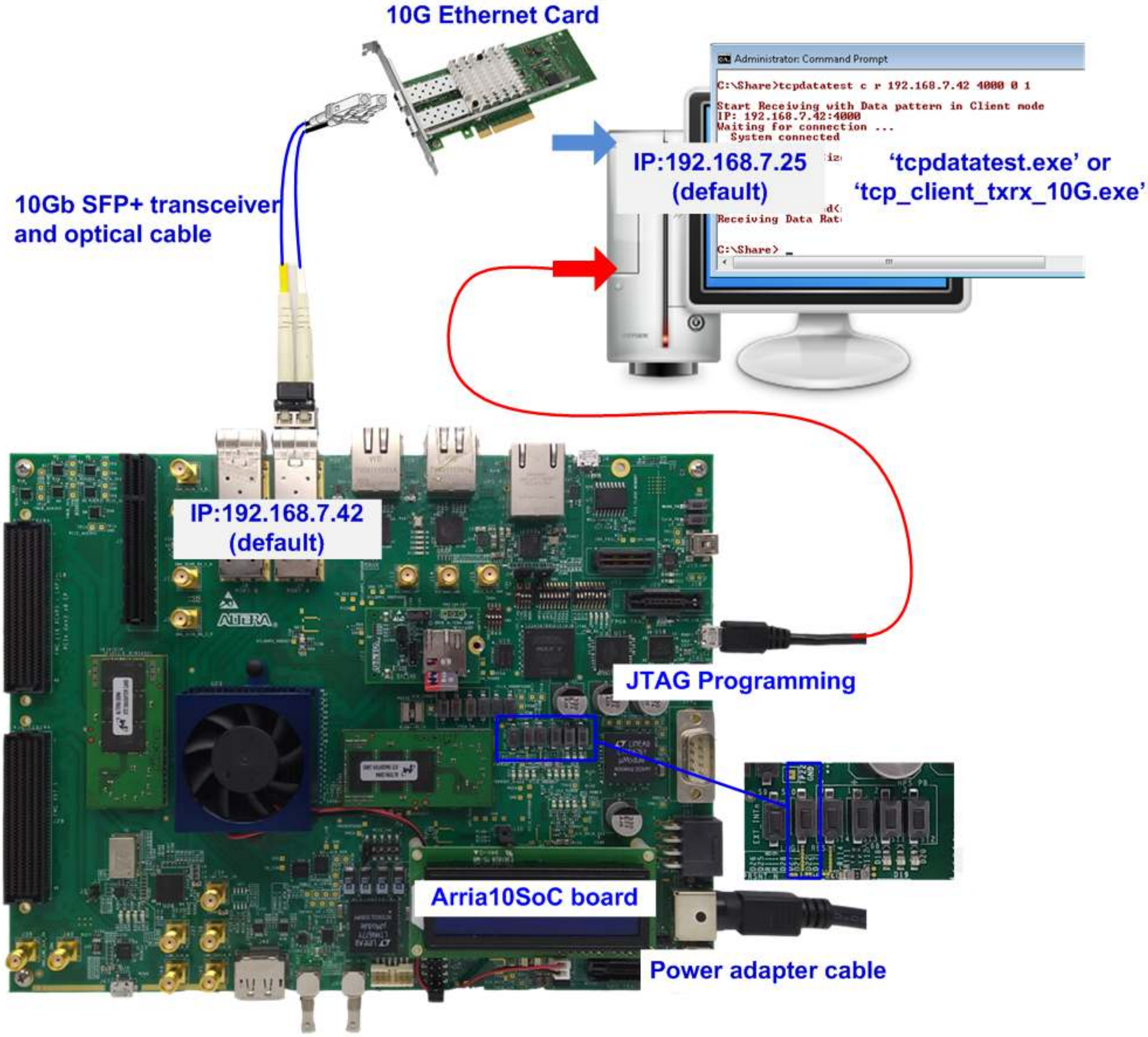

図 2-1: Arria10SoC ボードを使った TOE10G-IP コアの CPU 制御デモ環境

3 PC 設定

デモを実行する前にテスト PC 側のネットワークを設定する必要があります。ネットワークの設定例を以下に説明します。

3.1 IP アドレスの設定

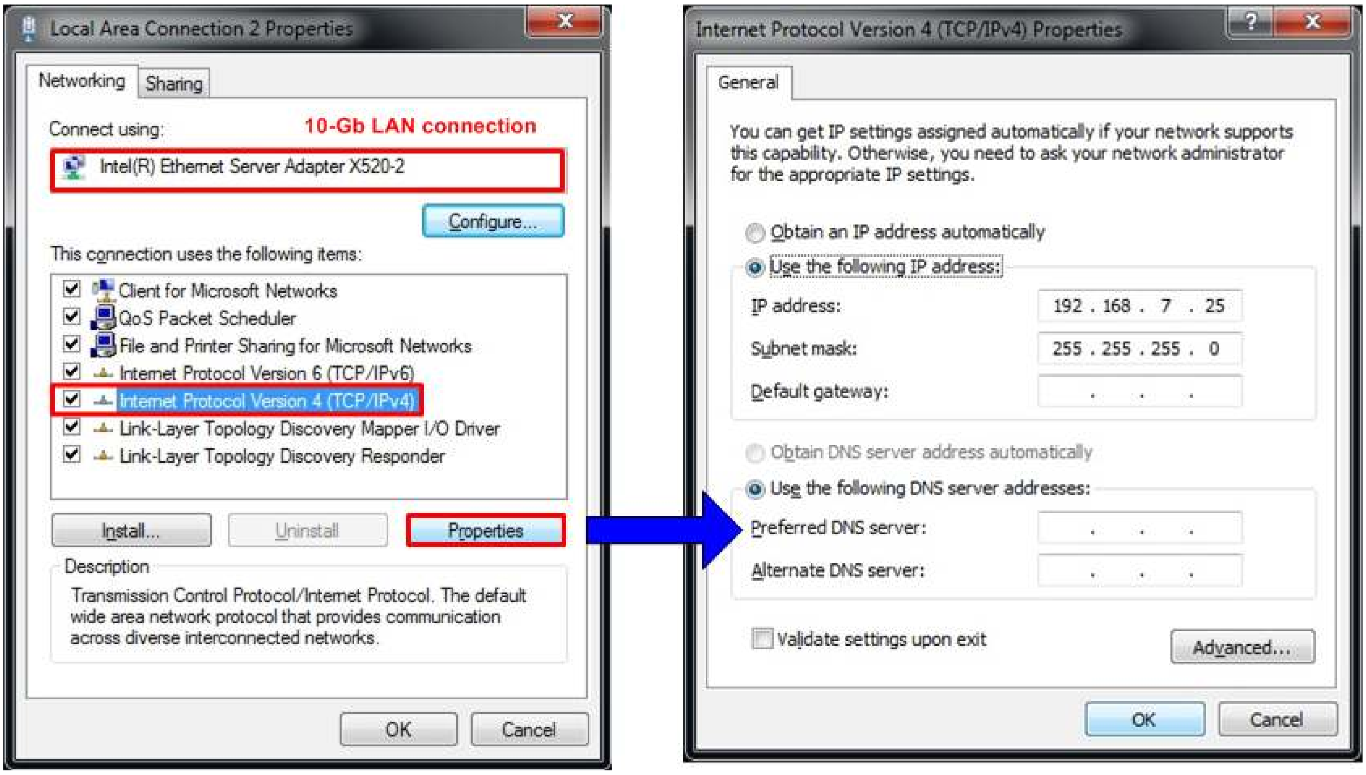

図 3-1: IPv4 の設定

1) 図 3-1 に示すように 10Gb 接続のローカル・エリア接続プロパティを開きます。
2) "TCP/Ipv4"を選択しプロパティのボタンをクリックします。
3) IP アドレスを 192.168.7.25 にセットし、サブネット・マスクを 255.255.255.0 に設定します。

3.2　速度とフレームの設定

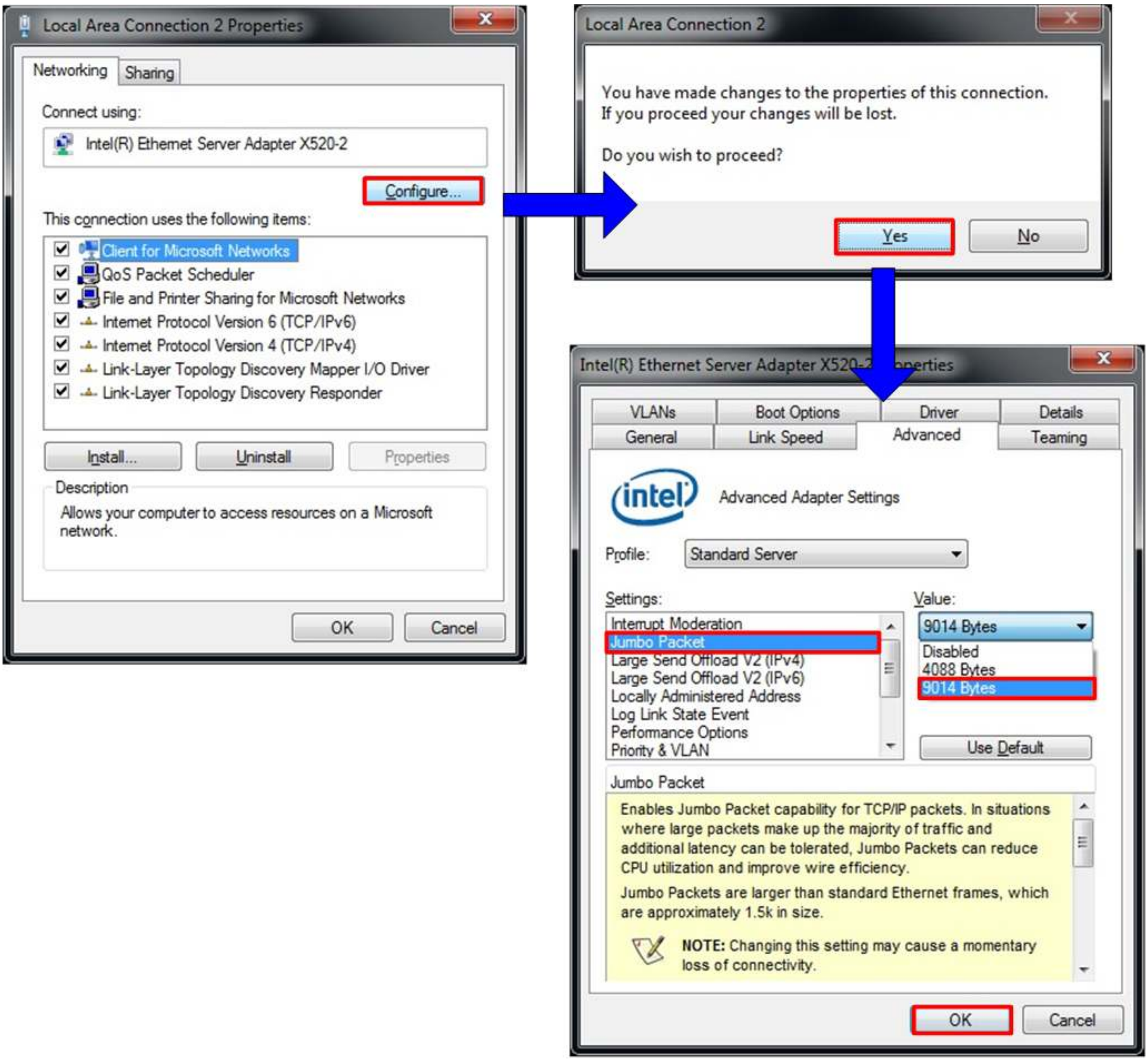

図 3-2: フレーム・サイズをジャンボ・フレームに設定

1) 図 3-2 に示すようにローカル・エリア接続プロパティの"Configure"ボタンをクリックします。
2) Advanced タブにて、"Jumbo Packet"を選択し、ジャンボ・フレームを検証する場合はこの値を "9014 Bytes"に設定し、非ジャンボ・フレームを検証する場合はこの値を"Disabled"にセットします。

3) 図 3-3 に示すように Link Speed タブにて、“10 Gbps Full Duplex”を選択し 10Gb 転送テストが行えるようにします。

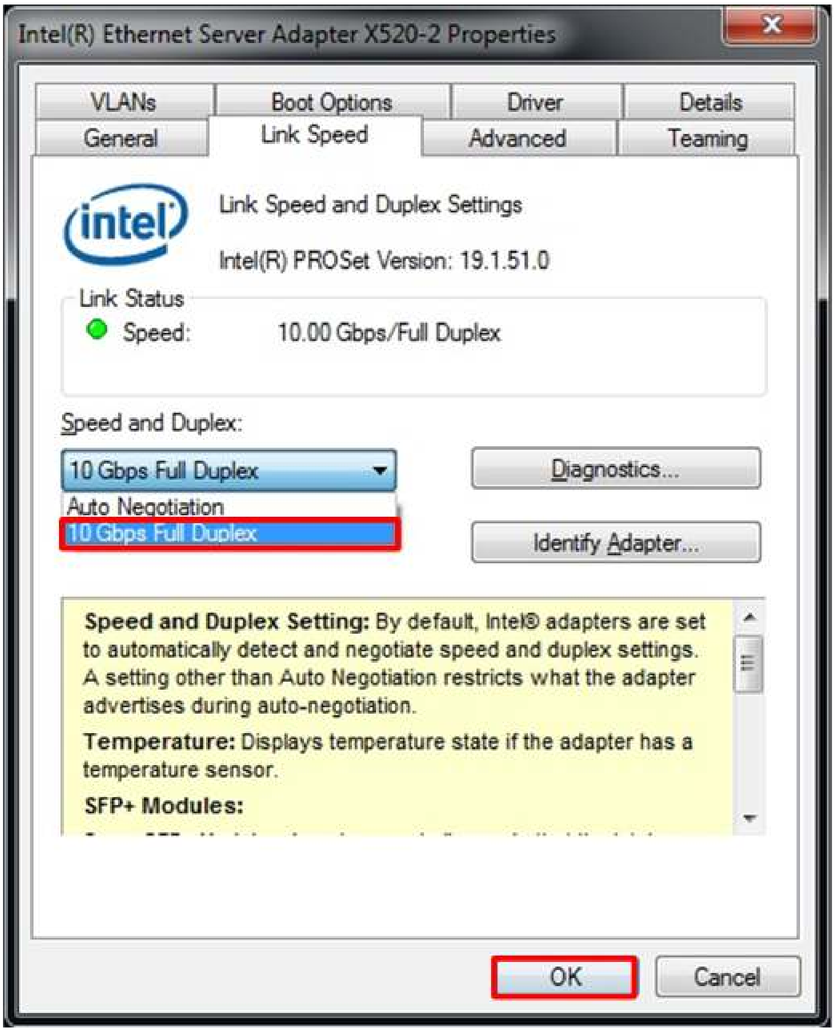

図 3-3: リンク速度=10Gbps に設定

4) Advanced タブにて、“Performance Options”を選択し “Properties”ボタンをクリックします。
5) “Performance Options”ウインドウにおいて、“Low Latency Interrupts”を選択し“Properties”ボタンをクリックします。
6) “Low Latency Interrupts”ウインドウにおいて、“Use Low Latency Interrupts”を選択し“OK”ボタンをクリックします。
7) 全部のウインドウで“OK”ボタンを押して設定をセーブします。

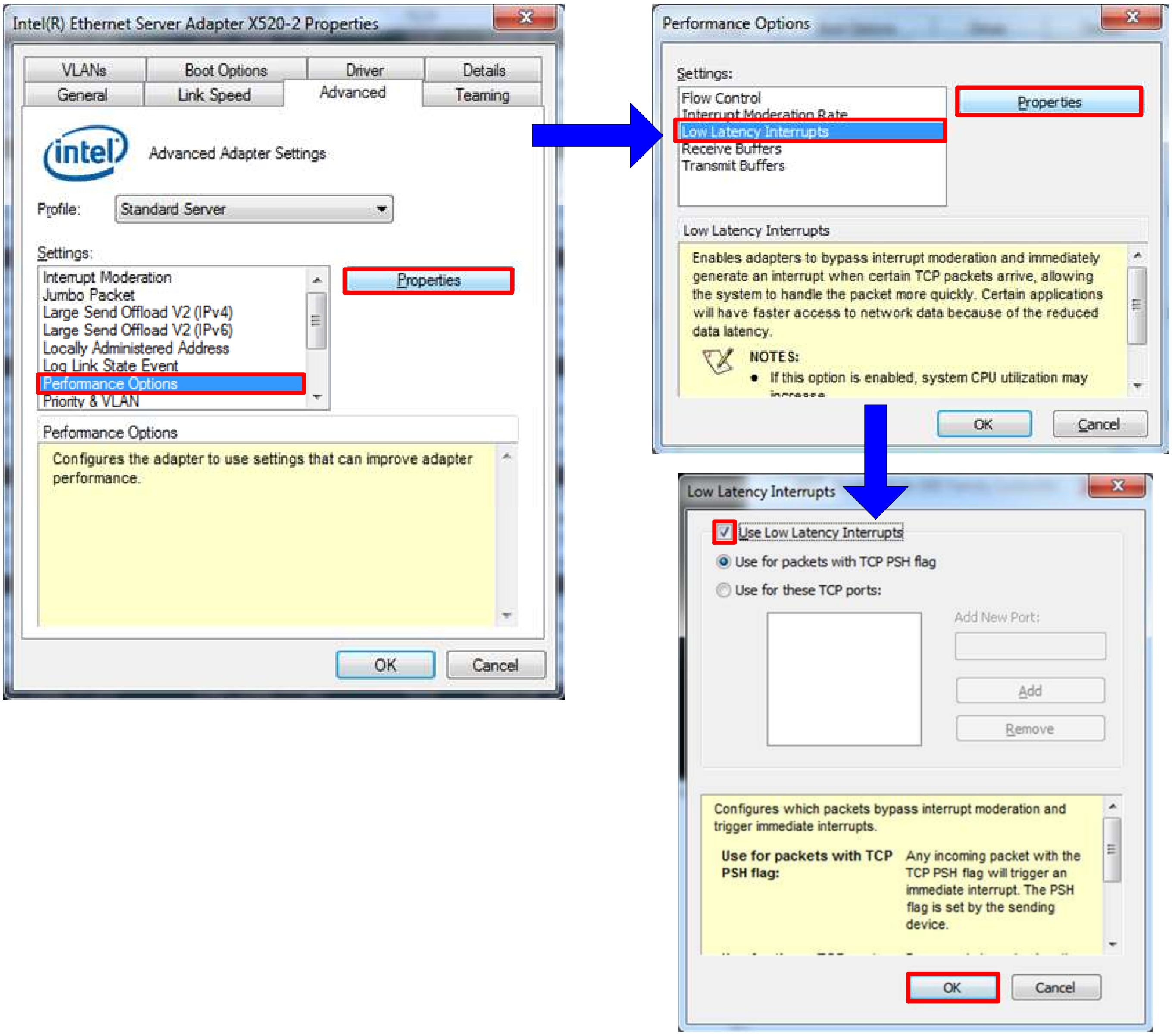

図 3-4: パフォーマンス・オプションの設定

3.3 電源オプションの設定

1) コントロール・パネルを開き図 3-5 左側のように電源オプションを選択します。
2) 設定を図 3-5 右側のように高パフォーマンスに変更します。

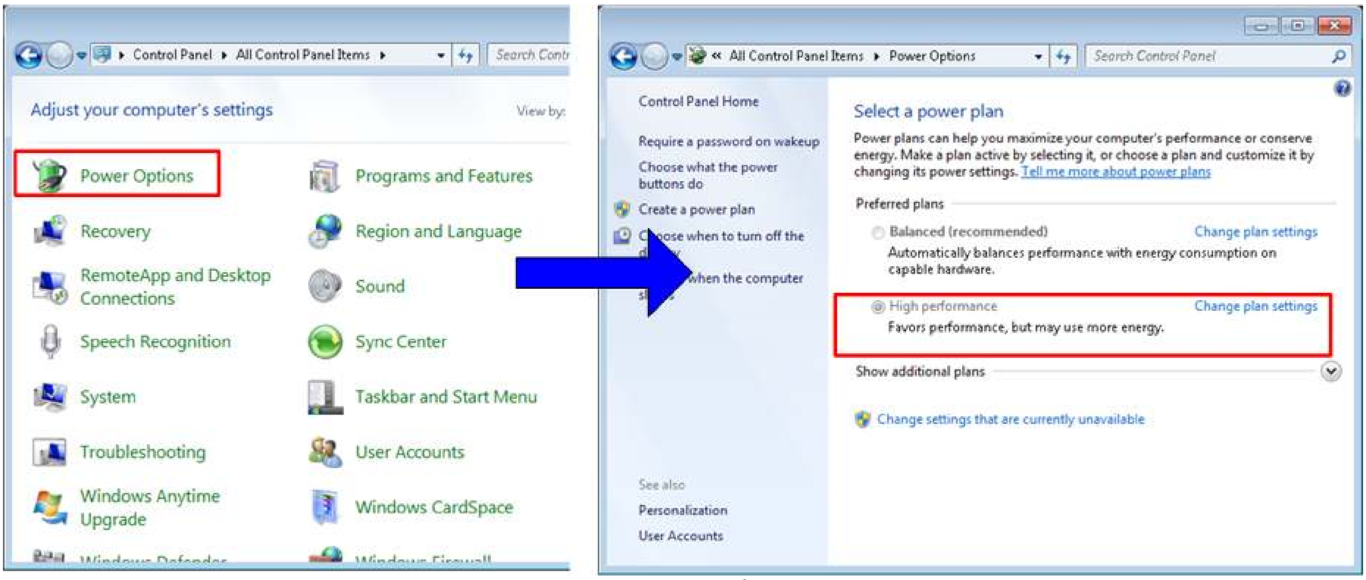

図 3-5: 電源オプション

4 FPGA ボードの設定

1) 電源が切断されていることを確認し FPGA ボードの電源ケーブルを接続します。
2) microUSB ケーブルで FPGA ボードと PC 間を接続します。
3) 10 Gb SFP+ DAC ケーブルまたは光ケーブルと接続した SFP+光トランシーバを SFP+ケージに挿入し PC の 10Gb ポートと接続します。
4) FPGA ボードの電源を投入します。

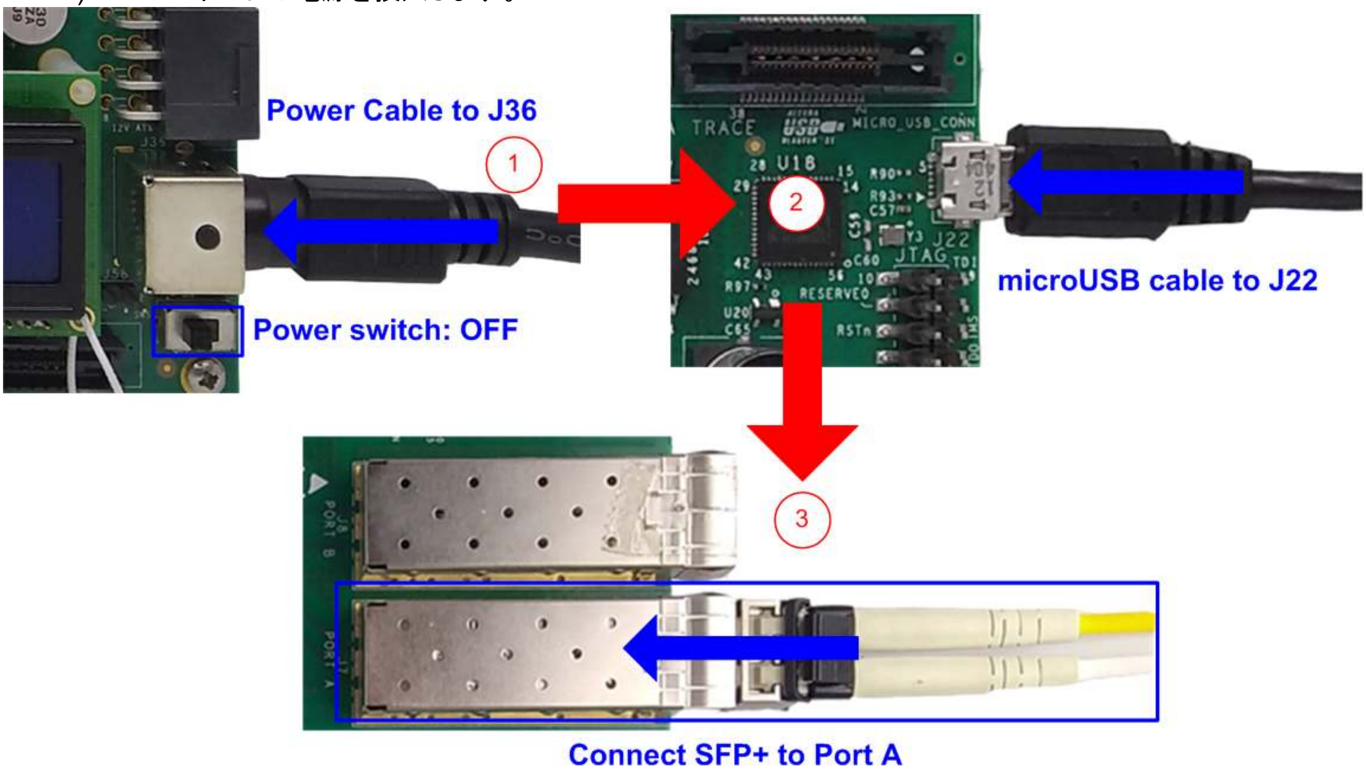

図 4-1: 電源,microUSB および SFP+の接続

5) Intel 製 FPGA ボード付属の”Clock Control”ソフトウエアを起動し、ボード上のプログラマブル・オシレータの周波数を 322.265625 MHz にセットします。
   a. “Clock Control”アプリケーションを起動します。
   b. 3 版目のタブ(U50)を選択し、CLK3 の周波数 = 322.265625 MHz に設定します。
   c. “Set”ボタンを押してアプリケーションで正しく周波数が設定されたことを確認します。
   d. “Clock Control”アプリケーションを終了します。

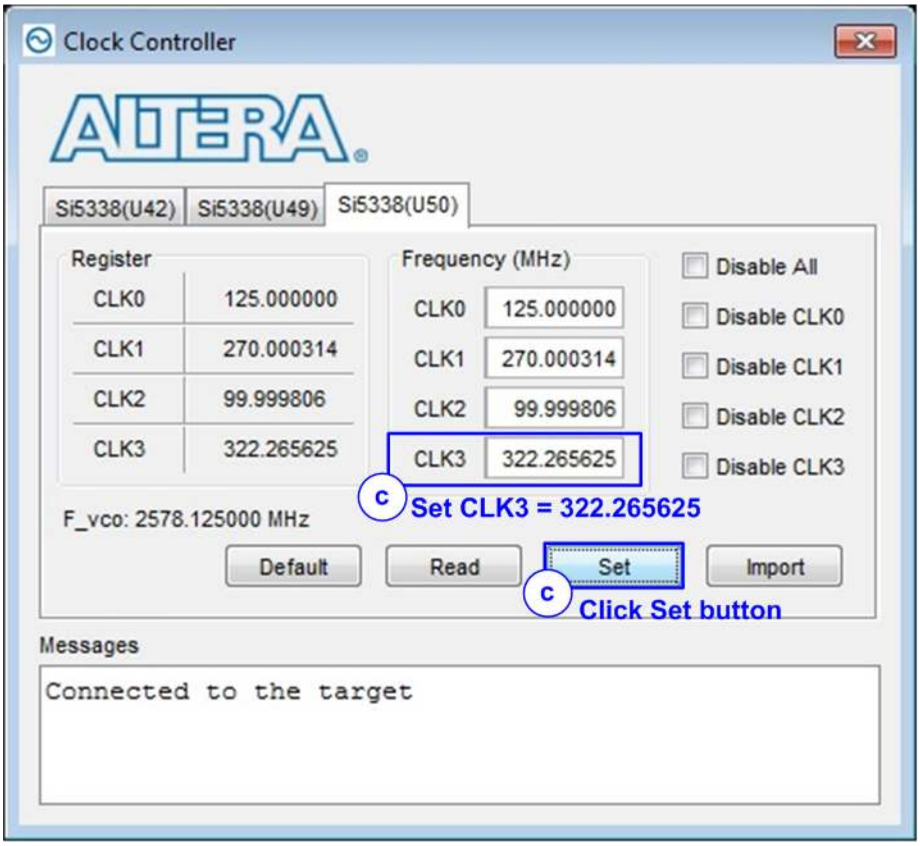

図 4-2: “Clock Control”アプリケーションの U50 周波数設定

6) QuartusII プログラマを起動し以下の手順で USB-1 経由での FPGA コンフィグレーションを実行します。
   a. “Hardware Setup…”をクリックし USB-BlasterII[USB-1] を選択します。
   b. “Auto Detect”をクリックし FPGA (10AS066N3) を選択します。
   c. A10 デバイスのアイコンを選択します。
   d. “Change File”ボタンを押して評価用 SOF ファイルを選択します。
   e. “program/Configure”を選択します。
   f. “Start”ボタンを押して FPGA のコンフィグレーションを行います。
   g. プログレスの進捗が 100%となるのを待ちます。

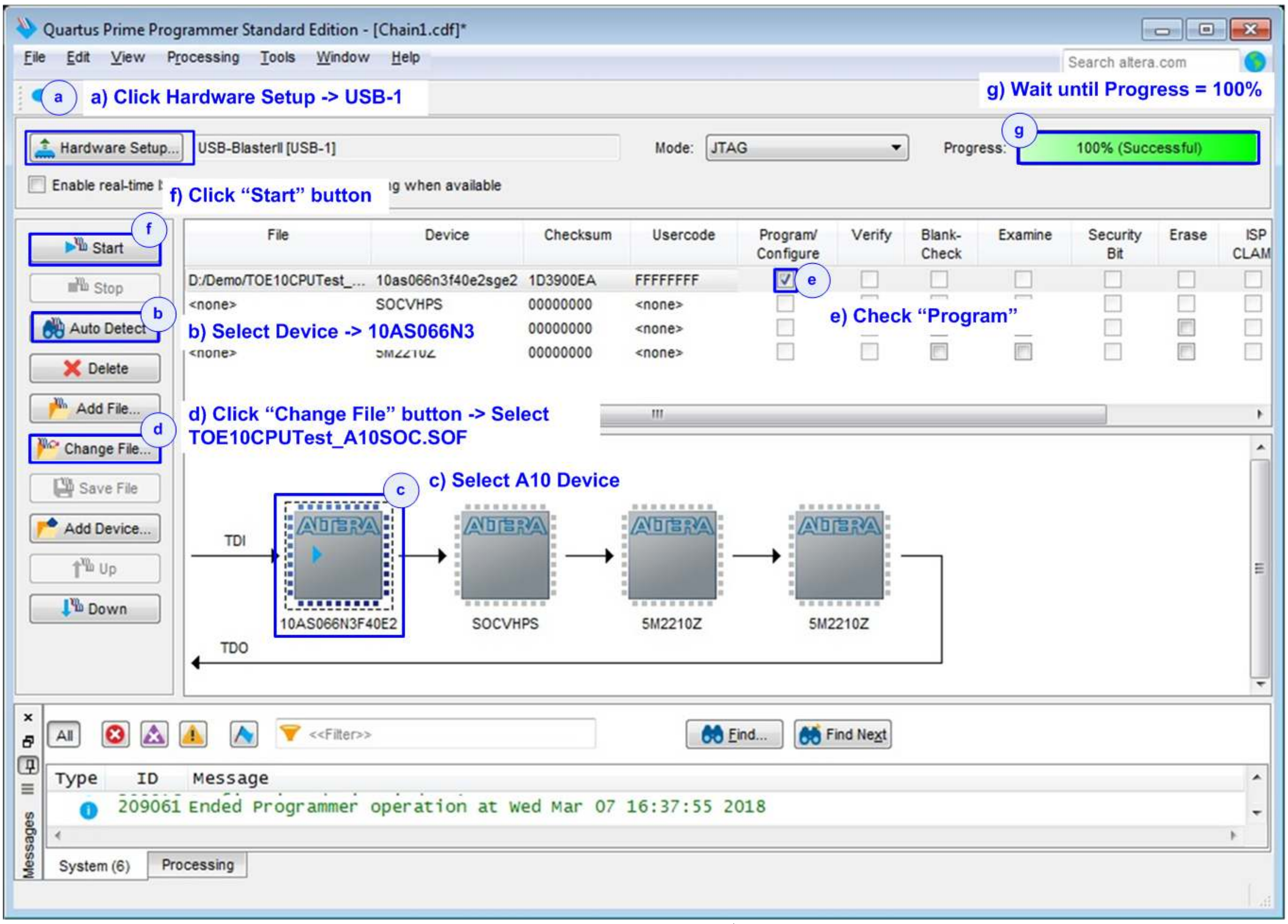

図 4-3: FPGA のコンフィグレーション

7) NiosII コマンド・シェルを開きます。
    a. コマンド・シェルにて“nios2-terminal”とタイプします。
    b. ‘0’を入力しTOE10G-IPコアをクライアント・モード(ARPを送信しPC側のMACアドレスを問い合わせる)で初期化します。
    c. クライアント・モードの初期パラメータがコンソール上に表示されます。

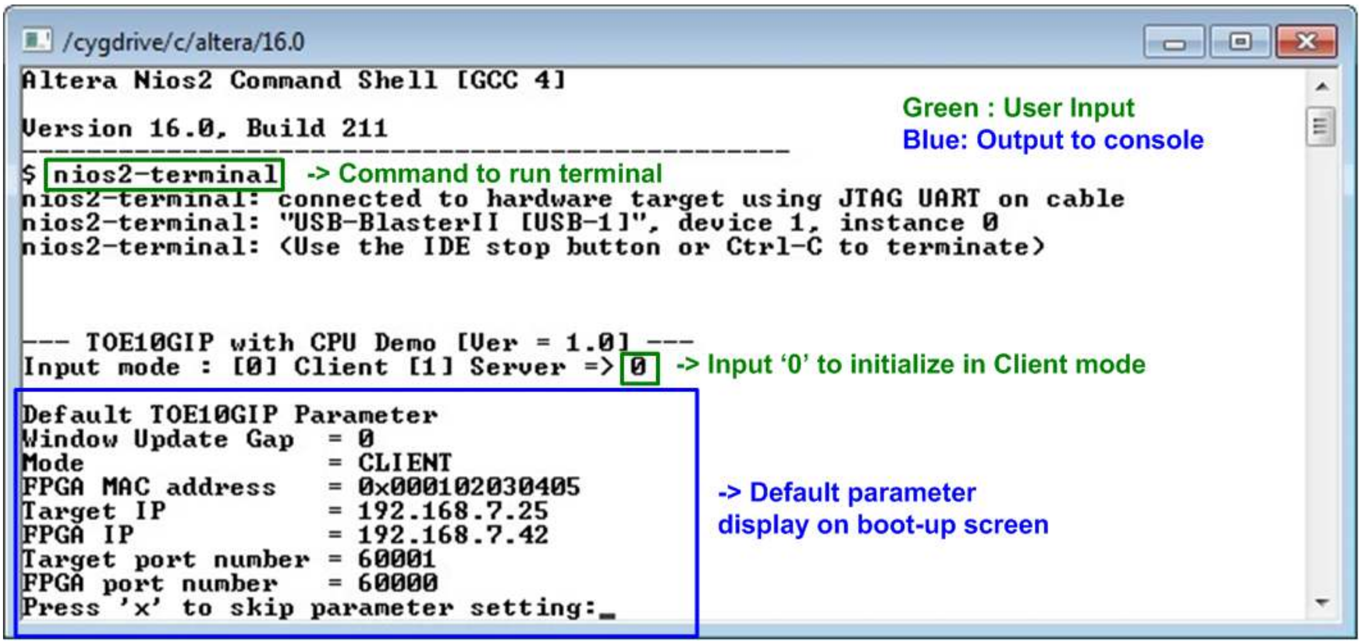

図 4-4: ブート起動後の NiosII ターミナル画面

8) 図 4-5 に示すように‘x’を入力することでパラメータの設定をスキップしデフォルトのパラメータでメイン・メニューに移動することができます。ここで他のキーを入力するとパラメータ変更のメニューに移動します。パラメータの変更手順はメニュー[1]と同じですので、次のトピックで説明します。

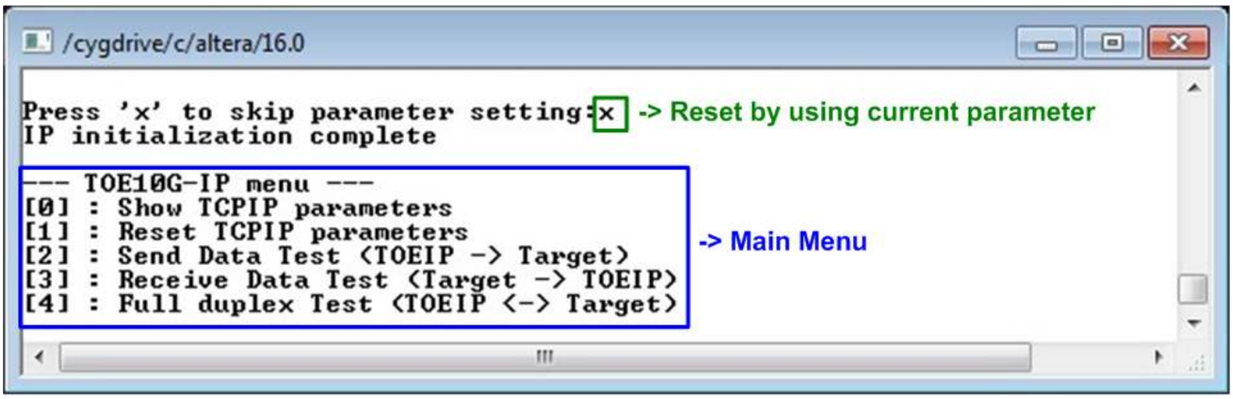

図 4-5: 初期化が完了した時点でのメイン画面

※ 注意: 10GbE の転送パフォーマンスはテストで使用する PC のパフォーマンスに依存します。このため、テスト PC には 10Gb イーサネット・データの送受信で十分なパフォーマンスを得られる高い能力を持つ PC を選んでください。

5 メイン・メニュー

5.1 パラメータ現在値の表示

‘0’を入力することで本デモの各パラメータ現在値を表示します。 全部で 7 種類のパラメータが NiosII コマンド・シェル上に表示されます。

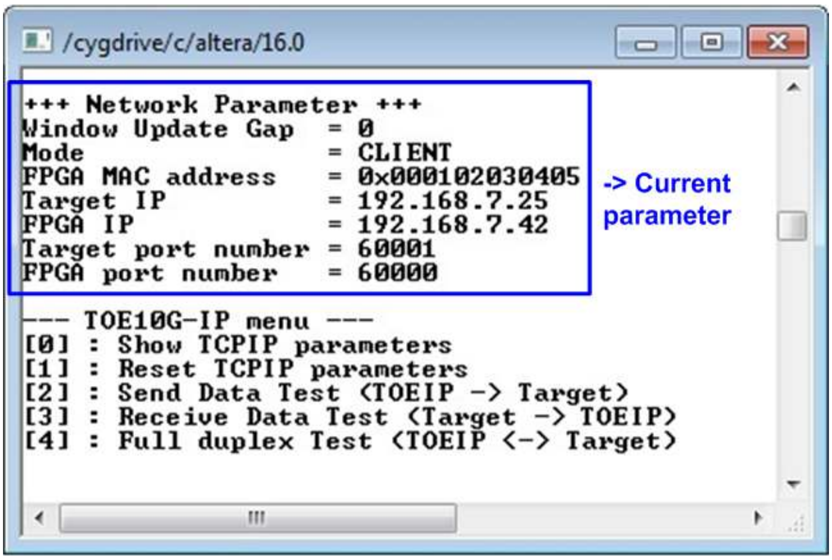

図 5-1: パラメータ現在値の表示例

1) Window Update Gap: ウインドウ・アップデート・パケットを送信する閾値を設定します。有効な値は 0x00 – 0x3F (0-63)です。ここで設定する敷居値の単位は 1K バイトです。 初期値は 0 (ウインドウ・アップデート機能を使わない)です。
2) Mode: TOE10G-IP コアがサーバーまたはクライアントのどちらのモードとするかをセットします。 FPGA が PC と接続する場合、’0’を指定しクライアント・モードで初期化する必要があります。
3) FPGA MAC address: FPGA ボード側の MAC アドレスを 48 ビットの 16 進数で設定します。 初期値は 0x000102030405 です。
4) Target IP: 10Gb イーサネット・データを転送する PC 側の IP アドレスを 10 進数で設定します。 初期値は 192.168.7.25 です。
5) FPGA IP: 10Gb イーサネット・データを転送する FPGA 側の IP アドレスを 10 進数で設定します。初期値は 192.168.7.42.です。
6) Target port number: PC 側の 10Gb イーサネット・データの転送ポート番号を 10 進数で指定します。初期値は 60001 です。
7) FPGA port number: FPGA 側の 10Gb イーサネット・データの転送ポート番号を 10 進数で指定します。初期値は 60000 です。

パラメータを変更する場合、メニュー[1]を選んでください。

5.2 TOE10G-IP コアのリセット

‘1”で選択されるこのメニューでは TOE10G-IP コアへ設定する各パラメータを変更しコアをリセットすることができます。その後更新したパラメータ情報が表示されます。ユーザは’x’を入力することで現在の設定と同じパラメータを使いコアをリセットするか、あるいは他(‘x’以外)キーを入力することでパラメータを変更できます。入力値が無効な値（たとえば’n’など）の場合パラメータは変更されません。より詳細については以下に説明します。

1) Mode: ‘0’をセットして IP コアをクライアント・モードで初期化する必要があります。
2) FPGA MAC address: 12 桁の 16 進数で入力します。先頭に”0x”のサフィックスをつけて 16 進数であることを示してください。
3) FPGA IP address: “.”(ドット)をセパレータとした 4 つの 10 進数で入力します。各入力値にて有効な範囲は 0-255 です。
4) FPGA port number: 有効な範囲は 0-65535 です。
5) Target IP address: 上記 3)と同様に 10 進数で PC 側 IP アドレスを指定 します。
6) Target port number: 有効な範囲は 0-65535 です。

パラメータの入力が終わったら更新されたパラメータが NiosII コマンド・シェル上に表示されます。次に IP コアに対してリセットが発行され、新たなパラメータがコアにロードされます。最後に “IP initialization complete”のメッセージが図 5-2 のように表示され初期化プロセスが完了したことを示します。

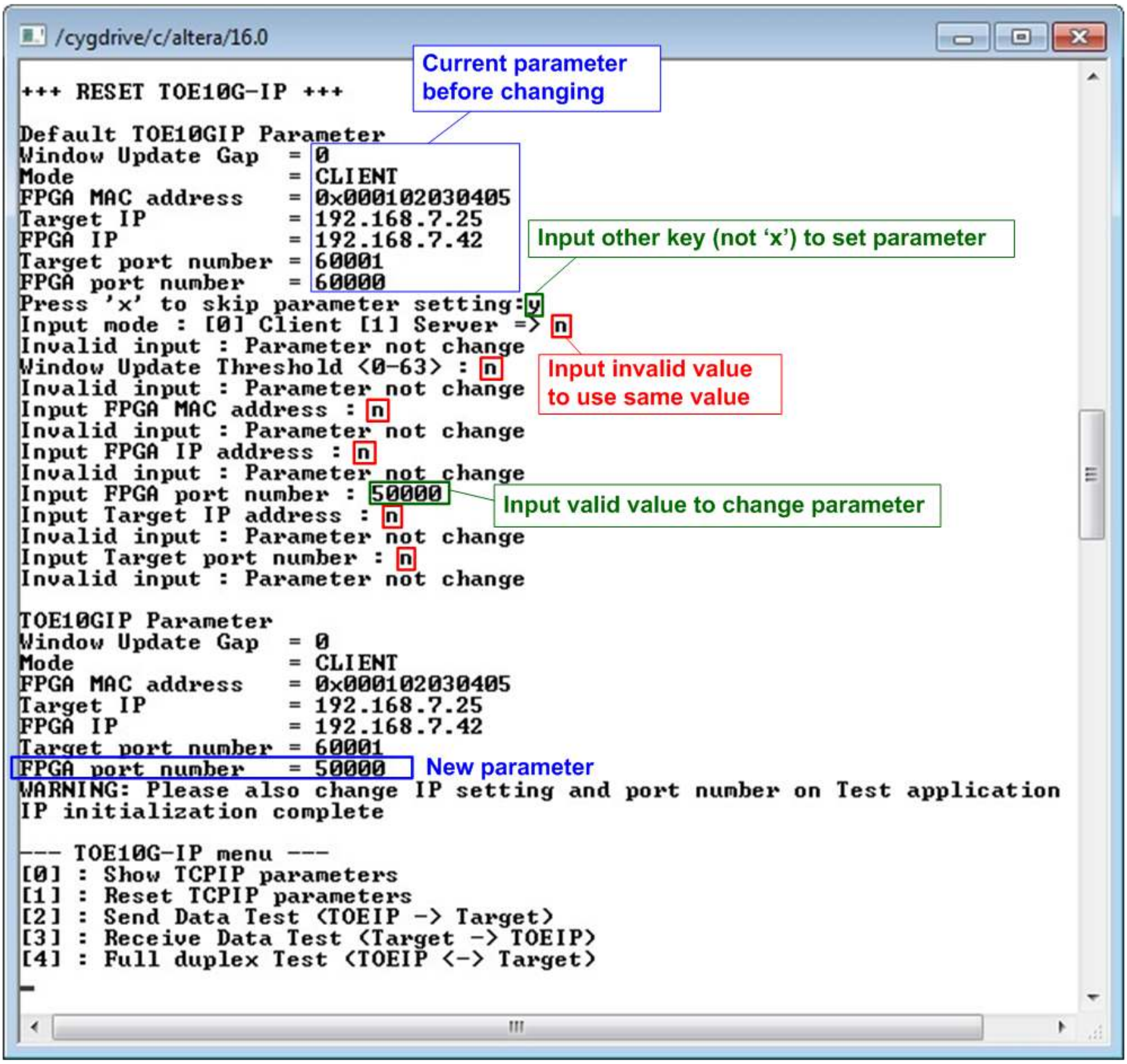

図 5-2: パラメータの変更結果例

5.3 データ送信テスト

'2'で選択されるこのメニューでは FPGA から PC へのデータ送信をテストできます。ユーザは NiosII コマンド・シェルを通して送信するデータのテスト・パラメータを設定できます。 一方接続先のテスト PC 側では"tcpdatatest"アプリケーションをコマンド・プロンプトから実行することでデータをベリファイできます。本テストの実施手順を以下に説明します。

1) NiosII コマンド・シェルにて 3 つのパラメータを入力します。
   a) 転送サイズを入力します。転送サイズの単位はバイト数です。有効な値は 0x8 - 0x7FFFFFFF8 です。ただし入力数は必ず 8 の倍数でなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に"0x"のプレフィックスを付けてください。
   b) パケット・サイズを入力します。パケット・サイズの単位はバイト数です。有効な値は 8 – 8960 です。ただし入力数は必ず 8 の倍数でなくてはなりません。 ユーザは 10 進数か 16 進数で入力できます。 16 進数で入力する場合、先頭に"0x"のプレフィックスを付けてください。 パケット・サイズが 1456 より大きい場合、テスト PC 側の 10Gb イーサネットはジャンボ・フレームに対応しなくてはなりません。
   c) モードを入力します。本テストでは'1'をセットし FPGA をサーバー・モードとしてください。
2) 全ての入力が有効であった場合、"Wait Open connection …"のメッセージとあわせて、テスト PC 側のテスト・アプリケーションで推奨されるパラメータ情報が表示されます。
3) テスト PC でコマンド・プロンプト(DOS 窓)を起動し、"tcpdatatest"を実行します、このときシリアル・コンソールで推奨されたパラメータを使って起動してください。この tcpdatatest では以下 6 個のパラメータがあります。
   >> tcpdatatest <mode> <dir> <server IP> <server port> <bytelen> <pattern>

説明:	mode	- 'c'を指定しテスト PC はクライアント・モードで起動します。
	dir	- 'r'を指定しテスト PC は FPGA からのデータ受信とベリファイを指定します。
	server IP	- FPGA 側の IP アドレスを入力します。
	server port	- FPGA 側のポート番号を入力します。
	bytelen	- 上記ステップ 1 a)で設定した転送サイズを指定します。
	pattern	- '1'を指定し FPGA からのデータ・ベリファイを行います。

4) tcpdatatest アプリケーションを起動するとポートが生成され、コマンド・プロンプトで "System connected" が表示されます。また、NiosII コマンド・シェルとコマンド・プロンプトの両方で 1 秒毎に転送サイズがリアルタイムで表示されます。 FPGA から全データの送信が完了すると NiosII コマンド・シェル上に"Send data complete"が表示されます。
5) FPGA が全データの送信を完了しコネクションをクローズした後に NiosII コマンド・シェル上に"Connection closed"と表示されます。 総転送サイズとパフォーマンス実測値が NiosII コマンド・シェルとコマンド・プロンプトの両方にテスト結果として表示されます。

図 5-3 にデータ送信テストを非ジャンボ・フレーム・サイズで実行した時の結果例を示します。 図左側は NiosII コマンド・シェル画面で FPGA はサーバー・モードで動作したものです、図右側はコマンド・プロンプト画面で PC はクライアント・モードで動作したものです。
図 5-4 にジャンボ・フレームで実行したデータ送信テスト例を示します。ジャンボ・フレームの場合非ジャンボ・フレームより良好なパフォーマンスが得られます。

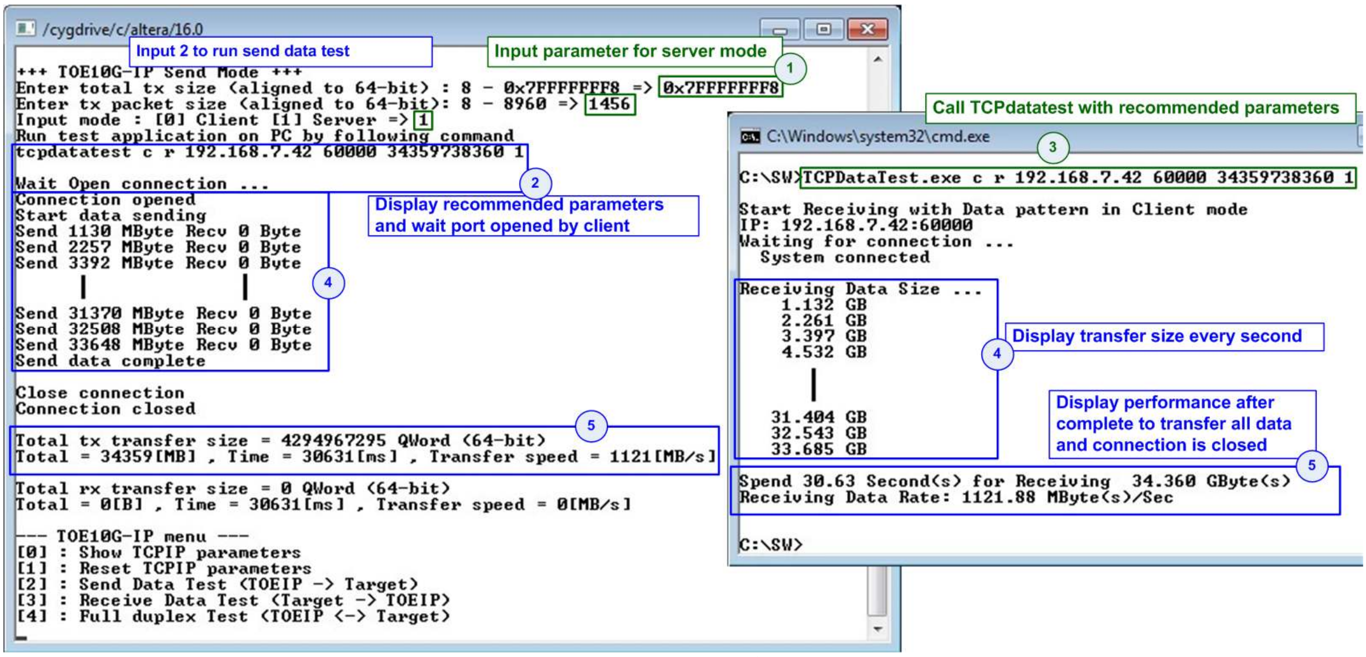

図 5-3: 非ジャンボ・フレームでのデータ送信テスト結果例

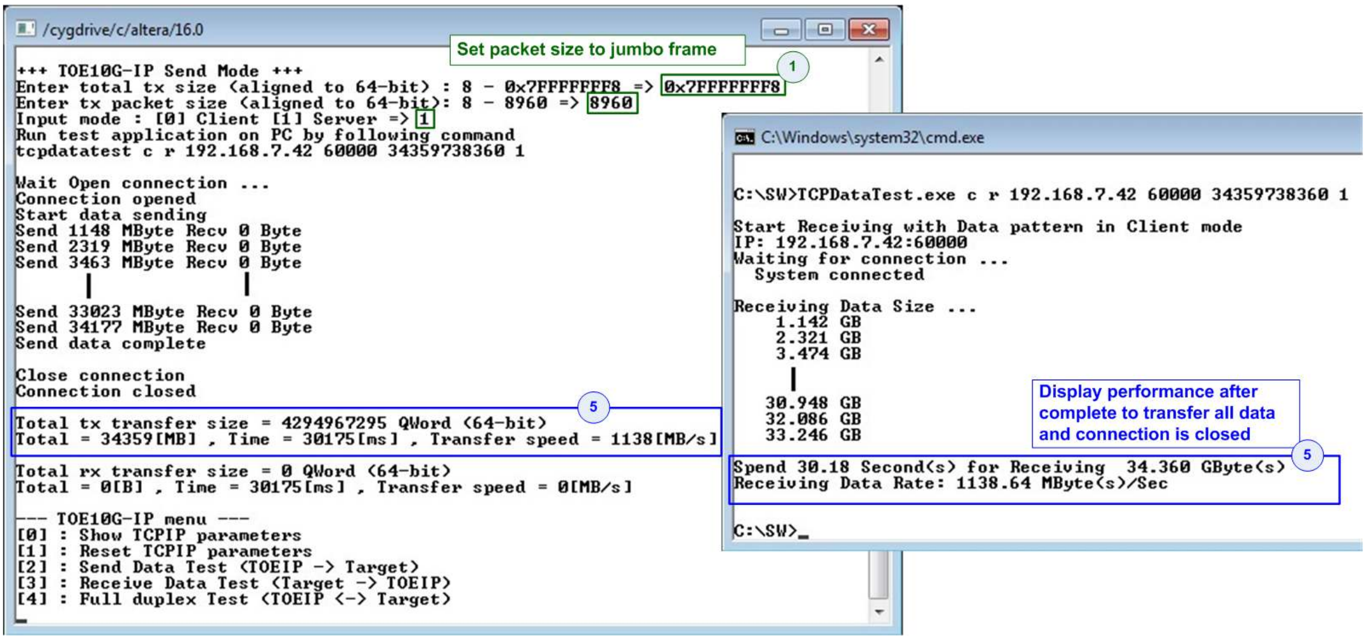

図 5-4: ジャンボ・フレームでのデータ送信テスト結果例

入力したパラメータが無効だった場合、図 5-5 ～ 図 5-7 のように“Out-of-range input”または”Invalid input”と表示され、データ送信テスト動作はキャンセルされます。

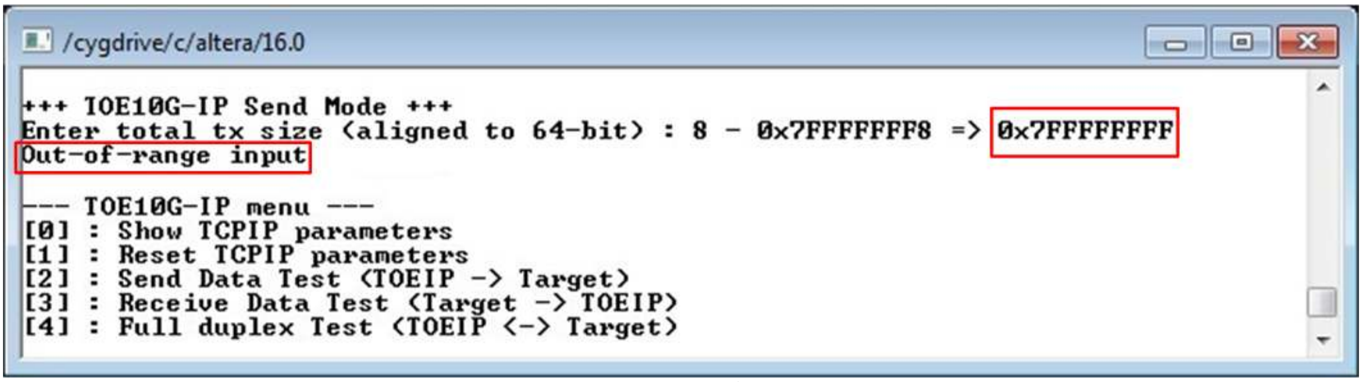

図 5-5: 無効な転送サイズを入力した場合

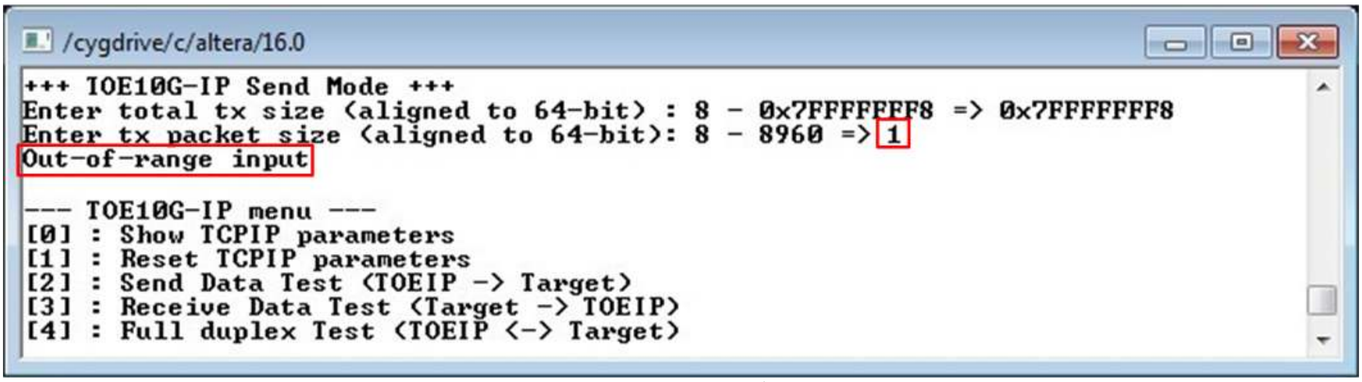

図 5-6: 無効なパケット・サイズを入力した場合

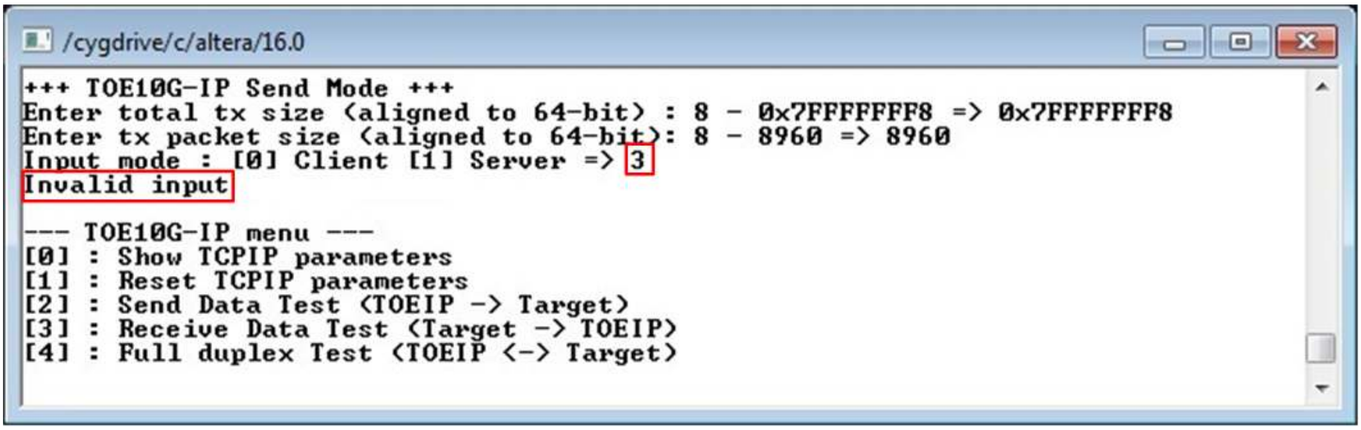

図 5-7: 無効なモードを入力した場合

5.4 データ受信テスト

‘3’で選択されるこのメニューでは PC から FPGA へのデータ受信をテストできます。ユーザは NiosII コマンド・シェルを通して受信するデータのテスト・パラメータを設定できます。 一方接続先のテスト PC 側では”tcpdatatest”アプリケーションをコマンド・プロンプトから実行することでデータを送信できます。 本テストの実施手順を以下に説明します。

1) NiosII コマンド・シェルにて 3 つのパラメータを入力します。
   a) 転送サイズを入力します。転送サイズの単位はバイト数です。有効な値は 0x8 - 0x7FFFFFFF8 です。ただし入力数は必ず 8 の倍数でなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に”0x”のプレフィックスを付けてください。
   b) データ・ベリファイのモードを入力します。パフォーマンスの最高値を測定する場合ダミー・データを使うため’0’をセットしベリファイ機能を休止します。一方データの信頼性を評価する場合は、テスト PC から送信するテスト・パターン・データを照合するため’1’としてベリファイ機能を有効とします。
   c) モードを入力します。本テストでは’1’をセットし FPGA をサーバー・モードとしてください。
2) 全ての入力が有効であった場合、“Wait Open connection …”のメッセージとあわせて、テスト PC 側のテスト・アプリケーションで推奨されるパラメータ情報が表示されます。
3) テスト PC でコマンド・プロンプト(DOS 窓)を起動し、“tcpdatatest”を実行します、このとき NiosII コマンド・シェルで推奨されたパラメータを使って起動してください。この tcpdatatest では以下 6 個のパラメータがあります。
   >> tcpdatatest <mode> <dir> <server IP> <server port> <bytelen> <pattern>
   説明:

mode	- ‘c’を指定しテスト PC はクライアント・モードで起動します。
dir	- ‘t’を指定しテスト PC は FPGA へのデータ送信を指定します。
server IP	- FPGA 側の IP アドレスを入力します。
server port	- FPGA 側のポート番号を入力します。
bytelen	- 上記ステップ 1 a)で設定した転送サイズを指定します。
pattern	- パフォーマンス最高値を測定する場合’0’を指定しダミー・データとします。 データ信頼性を評価する場合’1’を指定しテスト・パターン・データとします。
4) tcpdatatest アプリケーションを起動するとポートが生成されます。また、NiosII コマンド・シェルとコマンド・プロンプトの両方で 1 秒毎に転送サイズがリアルタイムで表示されます。
5) PC の tcpdatatest アプリケーションで全データの送信が完了しコネクションがクローズされると NiosII コマンド・シェル上に“Connection closed”と“Received data completed”が表示されます。

図 5-8 にデータ受信テストをデータ・ベリファイなしで動作した場合のテスト結果例を示します。FPGA 側では受信データのベリファイは行わず、PC からはダミー・データが送信されます。 図左側は NiosII コマンド・シェルの表示例で図右側はコマンド・プロンプトの表示例です。

図 5-9 にデータ受信テストをデータ・ベリファイありで動作した場合のテスト結果例を示します。FPGA 側では受信データのベリファイを実行し、PC からはテスト・パターンとしてインクリメンタル・データが送信されます。 図 5-8 のベリファイなしと比べるとベリファイありのテスト結果例ではパフォーマンスが低下しますがこれはテスト PC 側でインクリメンタル・パターンのテスト・データを生成するため PC リソースを消費してしまうことが原因です。

図 5-10 にデータ・ベリファイでのデータ不一致エラー結果例を示します。 FPGA はベリファイをイネーブルしていますが tcpdatatest はダミー・データを送った場合の例です。エラー・メッセージが NiosII コマンド・シェル上に表示されます。

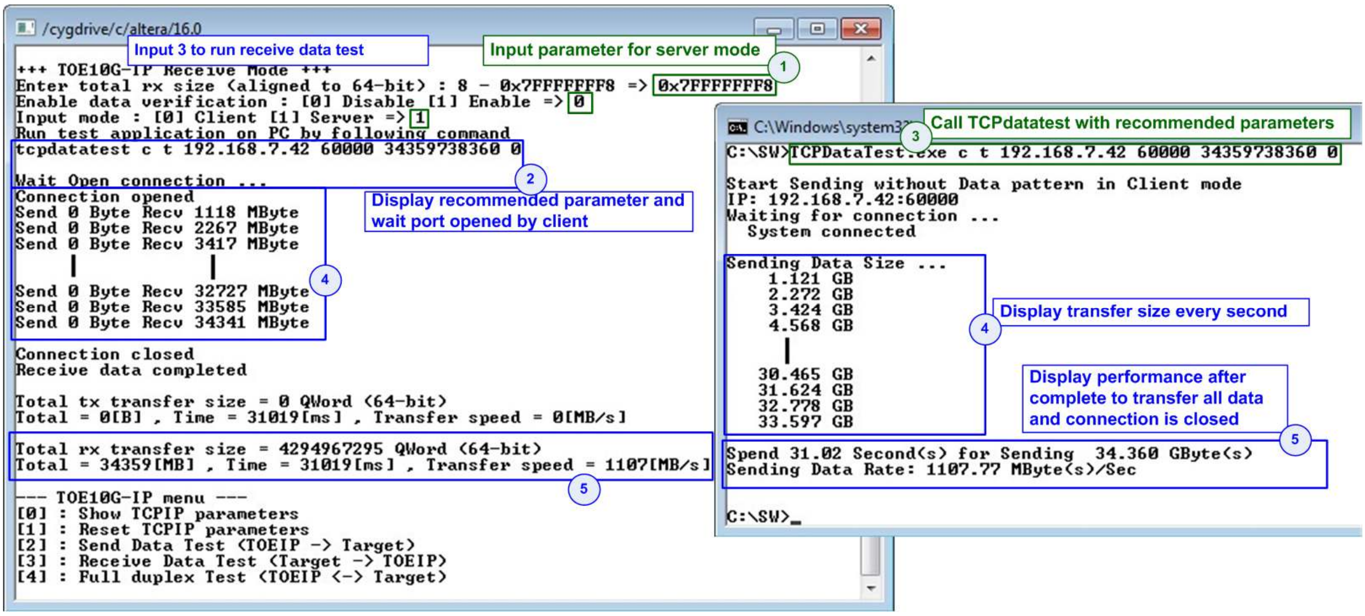

図 5-8: データ・ベリファイなしのデータ受信テスト結果例

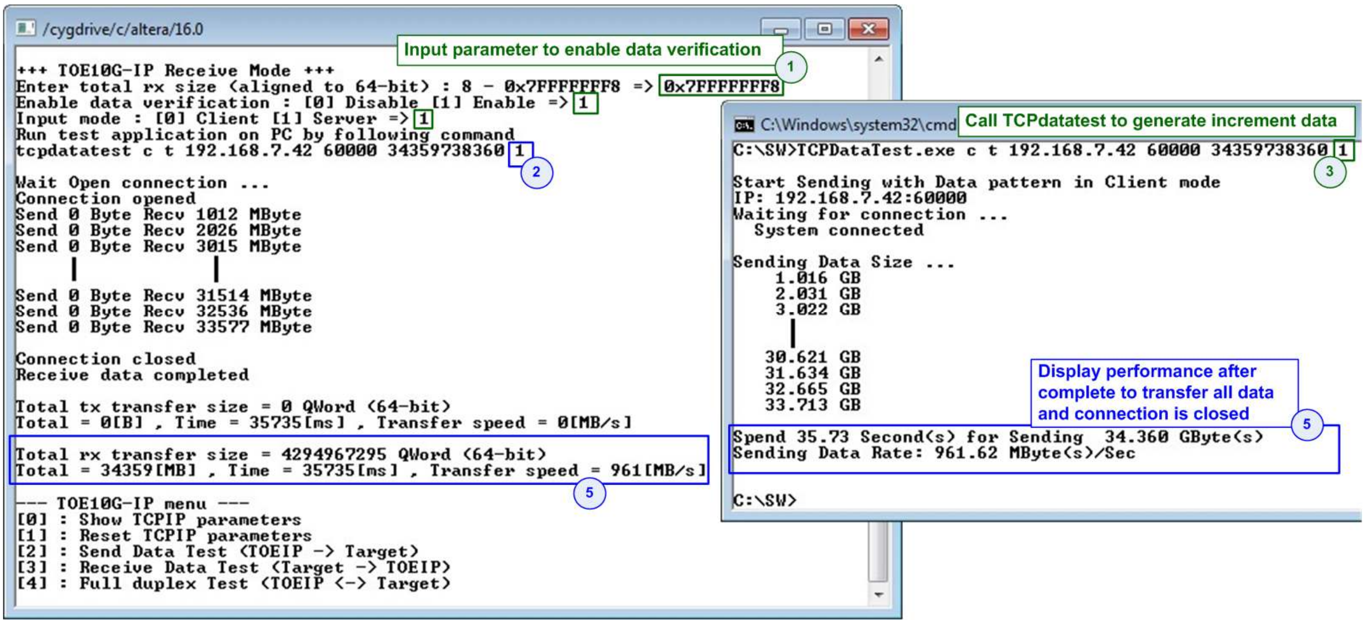

図 5-9: データ・ベリファイありのデータ受信テスト結果例

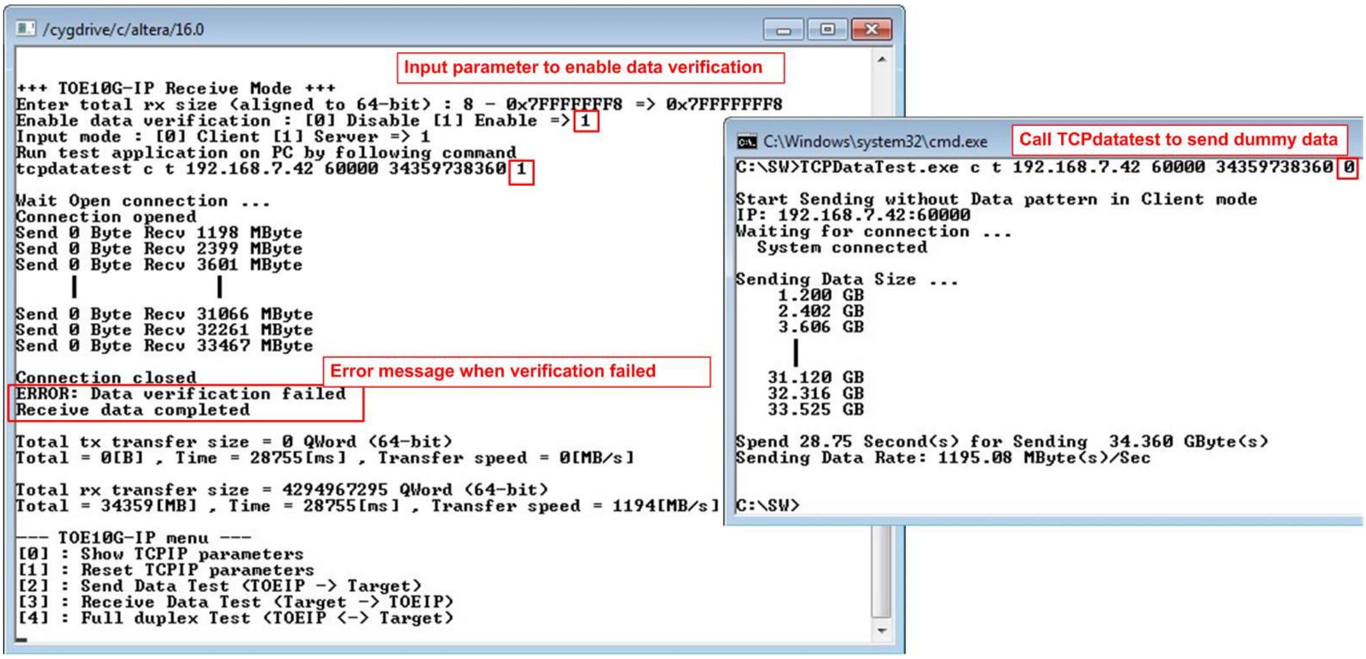

図 5-10: データ・ベリファイありで不一致エラーの結果例

5.5 全二重通信テスト

‘4’で選択されるこのメニューでは FPGA と PC 間で同時に送受信する全二重通信をテストできます。ユーザは NiosII コマンド・シェルを通して通信するデータのテスト・パラメータを設定できます。一方接続先のテスト PC 側では”tcp_client_txrx_10G”アプリケーションをコマンド・プロンプトから実行することで 10Gb データを同時送受信できます。本テストの実施手順を以下に説明します。

1) NiosII コマンド・シェルにて 4 つの全二重通信テスト用パラメータを入力します。
   a) 転送サイズを入力します。転送サイズの単位はバイト数です。ここでは PC 側テスト・アプリケーションと同じ値となる 0x7FFFFFC0 と入力してください。
   b) パケット・サイズを入力します。パケット・サイズの単位はバイト数です。有効な値は 8 – 8960 です。ただし入力数は必ず 8 の倍数でなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に”0x”のプレフィックスを付けてください。
   c) データ・ベリファイのモードを入力します。パフォーマンスの最高値を測定する場合ダミー・データを使うため’0’をセットしベリファイ機能を休止します。一方データの信頼性を評価する場合は、テスト PC から送信するテスト・パターン・データを照合するため’1’としてベリファイ機能を有効とします。
   d) モードを入力します。本テストでは’1’をセットし FPGA をサーバー・モードとしてください。
2) 全ての入力が有効であった場合、“Wait Open connection …”のメッセージとあわせて、テスト PC 側のテスト・アプリケーションで推奨されるパラメータ情報が表示されます。
3) テスト PC でコマンド・プロンプト(DOS 窓)を起動し、“tcp_client_txrx_10G”を実行します、このとき Nioiss コマンド・シェルで推奨されたパラメータを使って起動してください。この tcp_client_txrx_10G では以下 3 個のパラメータがあります。

   >> tcp_client_txrx_10G <server IP> <server port> <pattern>

   説明:
   server IP - FPGA(サーバー)側の IP アドレスを入力します。
   server port - FPGA(サーバー)側のポート番号を入力します。
   pattern - パフォーマンス最高値を測定する場合’0’を指定しダミー・データを送信して受信データのベリファイを行いません。
   データ信頼性を評価する場合’1’を指定してテスト・パターン・データを送信します、また FPGA からの受信データではベリファイを実行します。
4) tcp_client_txrx_10G アプリケーションが起動するとポートが生成さえrます。また、NiosII コマンド・シェルとコマンド・プロンプトの両方で 1 秒毎に転送サイズがリアルタイムで表示されます。
5) 全データの送信と受信が完了しコネクションがクローズされると NiosII コマンド・シェル上に”Send data complete”と表示されます。総転送サイズとパフォーマンス実測値が NiosII コマンド・シェルとコマンド・プロンプトの両方にテスト結果として表示されます。

そしてステップ 4)～ステップ 5)が永久に繰り返されます。テストを終了する場合はテスト PC のコマンド・プロンプトで”Ctrl+C”キーを入力し、FPGA の NiosII コマンド・シェルで何かキー入力を行ってください。

図 5-11 にデータ・ベリファイなし（パフォーマンス測定）の、図 5-12 にデータ・ベリファイあり(信頼性評価)の全二重通信テスト結果例をそれぞれ示します。図左側は FPGA からの NiosII コマンド・シェル出力例で右側はテスト PC 側のコマンド・プロンプト出力例です。図 5-12 に示すデータ・ベリファイあり（信頼性評価）はダミー・データを使った図 5-11 の結果よりもパフォーマンスは大きく低下します。これはテスト PC 側にてテスト・パターンのデータ発生や受信データのベリファイを送信・受信と同時に行う必要があることから CPU リソースを消費するためです。

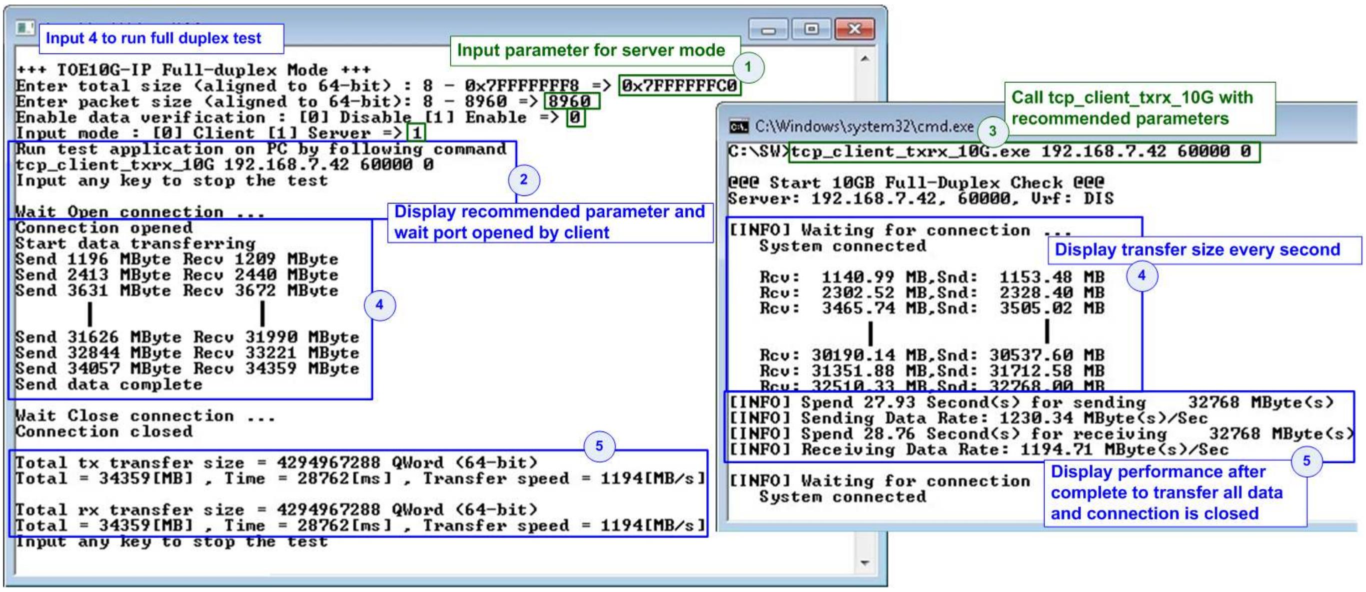

図 5-11: データ・ベリファイなし(パフォーマンス測定)の全二重通信テスト結果例

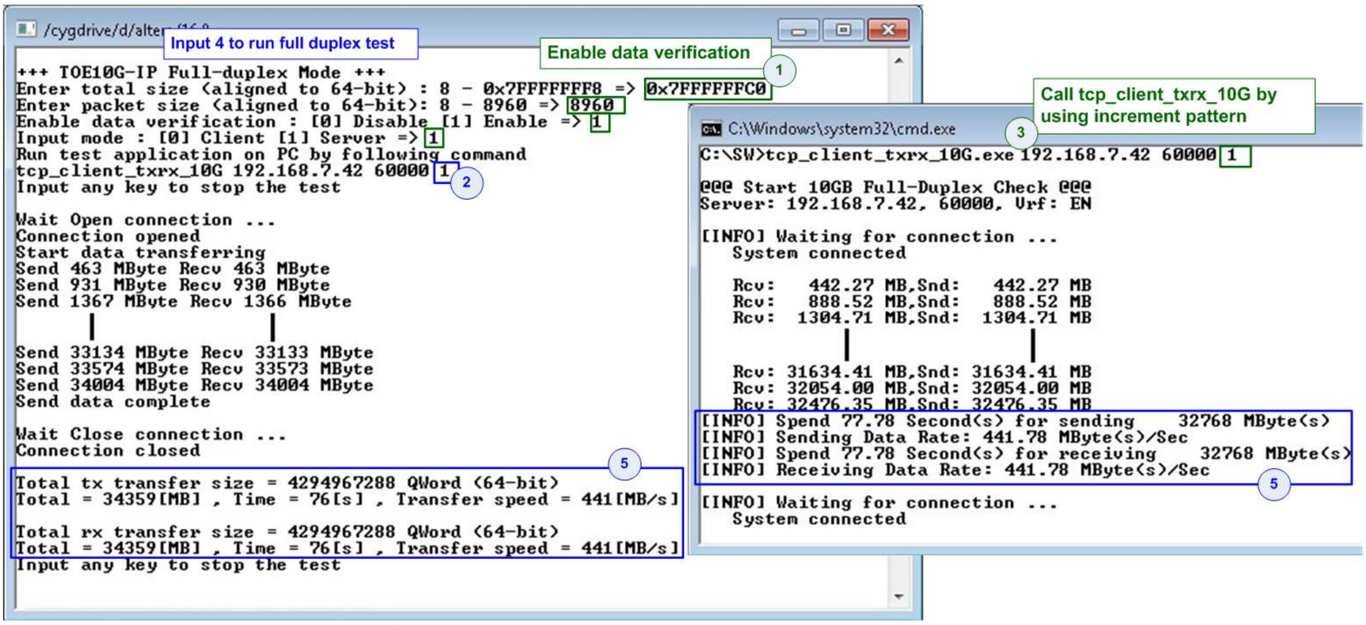

図 5-12: データ・ベリファイあり(信頼性評価)の全二重通信テスト結果例

Part B: 2 枚の FPGA ボードを使った TOE10G-IP コアのデモ

6 動作環境

図 6-1 に示すように 2 枚の FPGA ボードを相互接続して TOE10G-IP コアのデモを実施するためには以下のテスト環境を用意してください。

1) FPGA ボード 2 枚(Arria10SoC ボード)
2) 2 枚の FPGA ボード間を接続する 10Gbit SFP+トランシーバ 2 個および光ケーブル 2 本
3) 2 枚の FPGA ボードと PC 間をそれぞれ接続する FPGA コンフィグレーション用兼 JTAG UART 用の microUSB ケーブル 2 本
4) PC にインストールした QuartusII プログラム(FPGA コンフィグレーション用)、FPGA 評価ボード上の発振器周波数設定用クロック・コントロールおよび NiosII コマンド・シェル

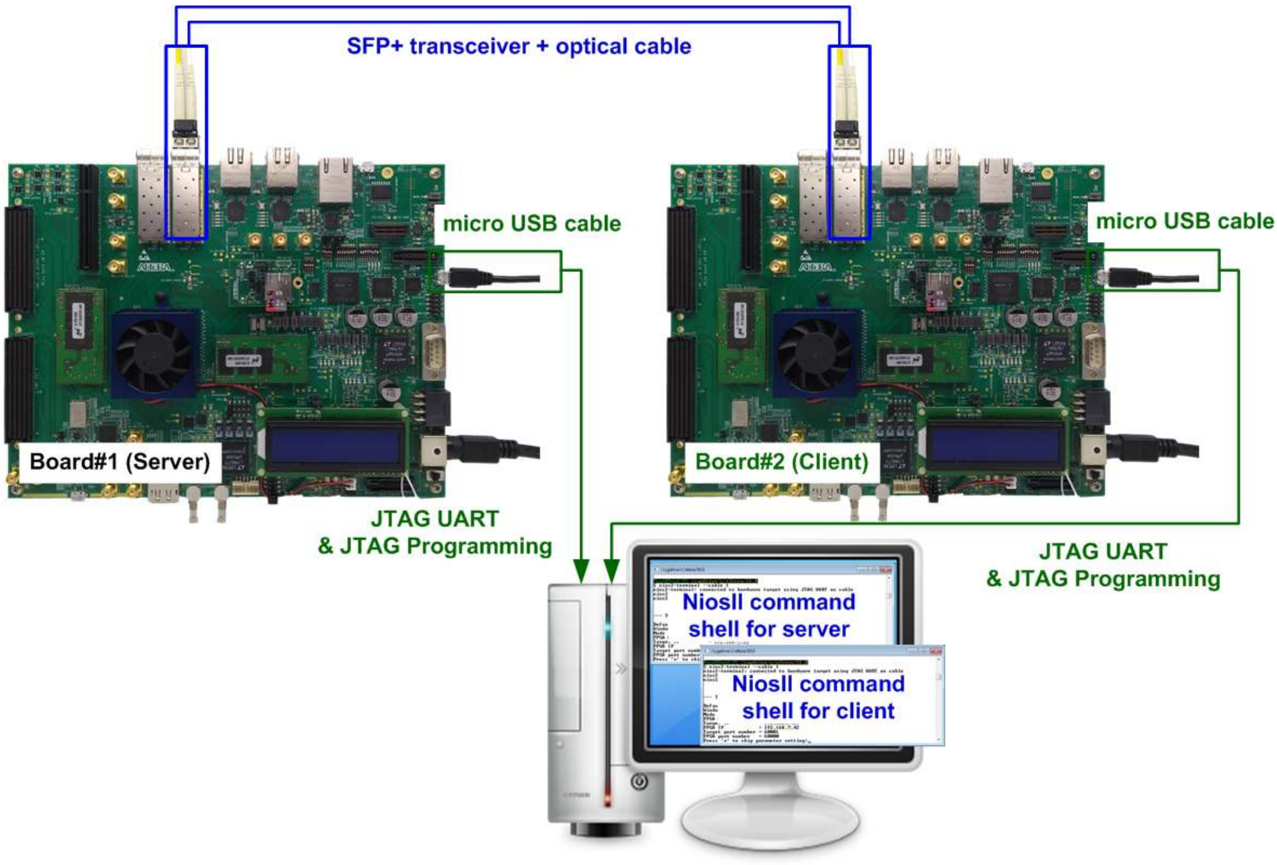

図 6-1: 2 枚の Arria10SoC ボード間の TOE10G-IP 相互接続テスト環境

7 FPGA ボードの設定

1) 2 枚の FPGA ボード(サーバーおよびクライアント)とも電源が切断されていることを確認し FPGA ボードの電源ケーブルをそれぞれ接続します。

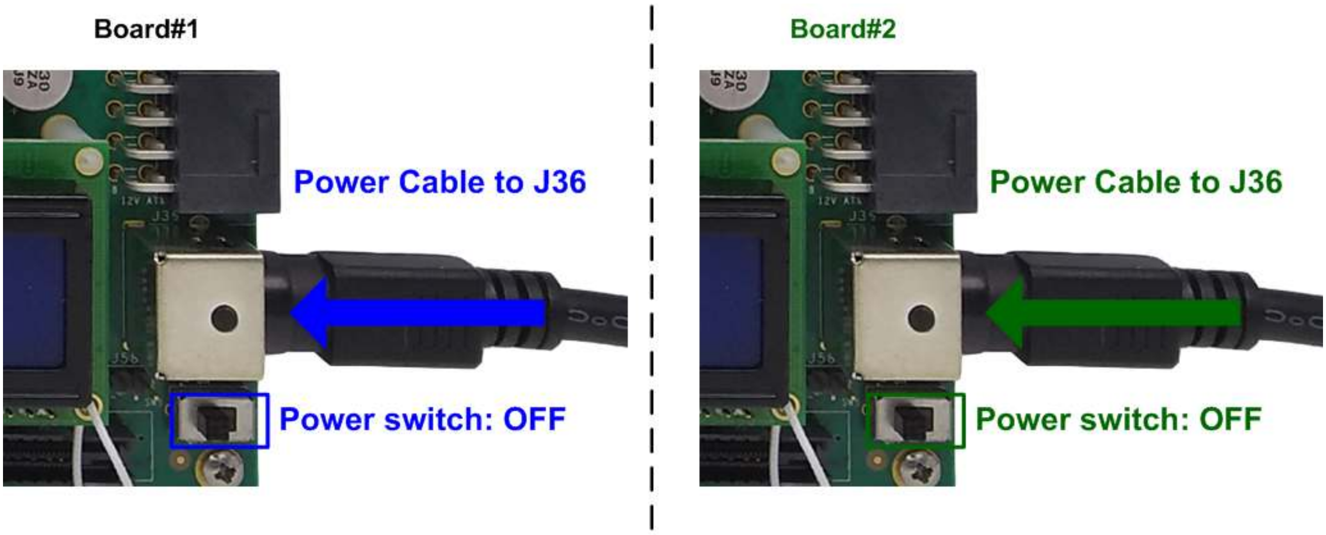

図 7-1: 電源ケーブルの接続

2) FPGA ボード#1 に microUSB ケーブルを接続し#1 ボードの電源を投入します。

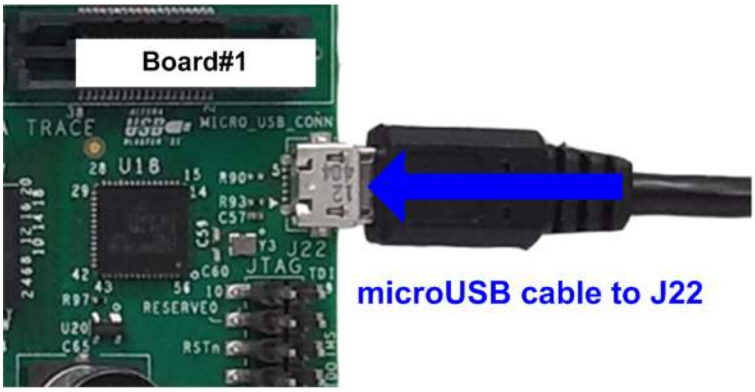

図 7-2: microUSB ケーブルの接続

3) Intel 製 FPGA 評価キット付属の"Clock Control"ソフトウエアを起動し、ボード上のプログラマブル・オシレータの周波数を 322.265625 MHz にセットします。 3 番目のタブ(U50)を選択し、CLK3 の周波数 = 322.265625 MHz に設定し"Set"ボタンを押します。

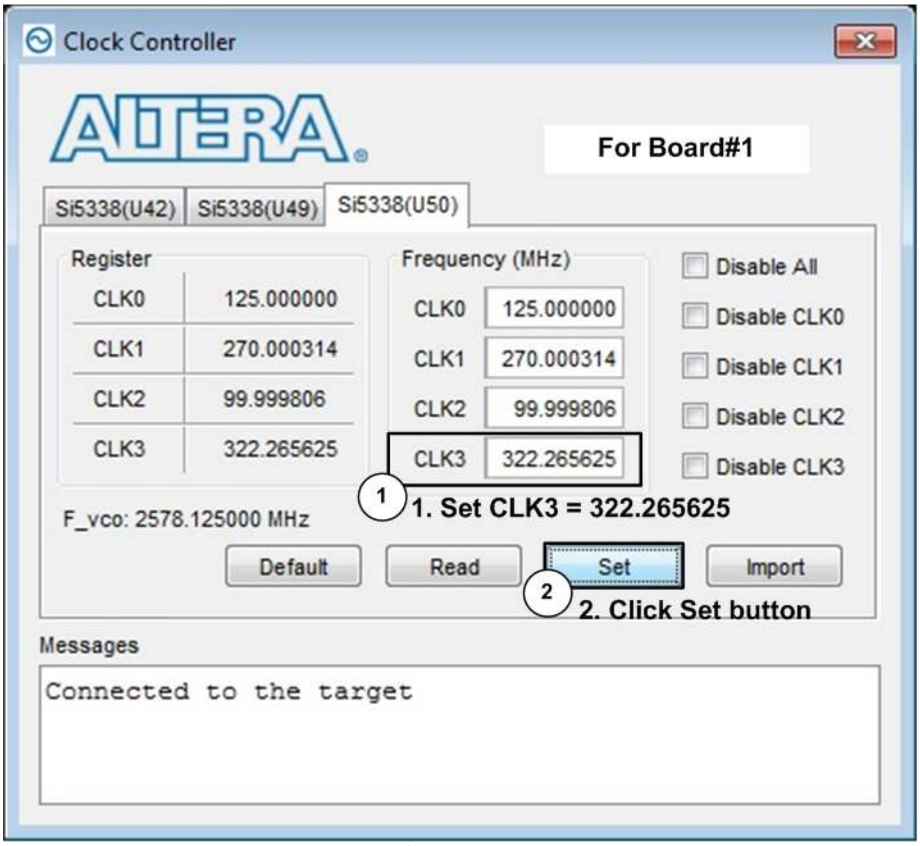

図 7-3: "Clock Control"アプリケーションで U50 の周波数を設定

4) "Clock control"アプリケーションを一旦終了します。
5) FPGA ボード#1 から micro USB ケーブルを外しもう片方の FPGA ボード#2 に接続します。そして FPGA ボード#2 の電源を投入します。

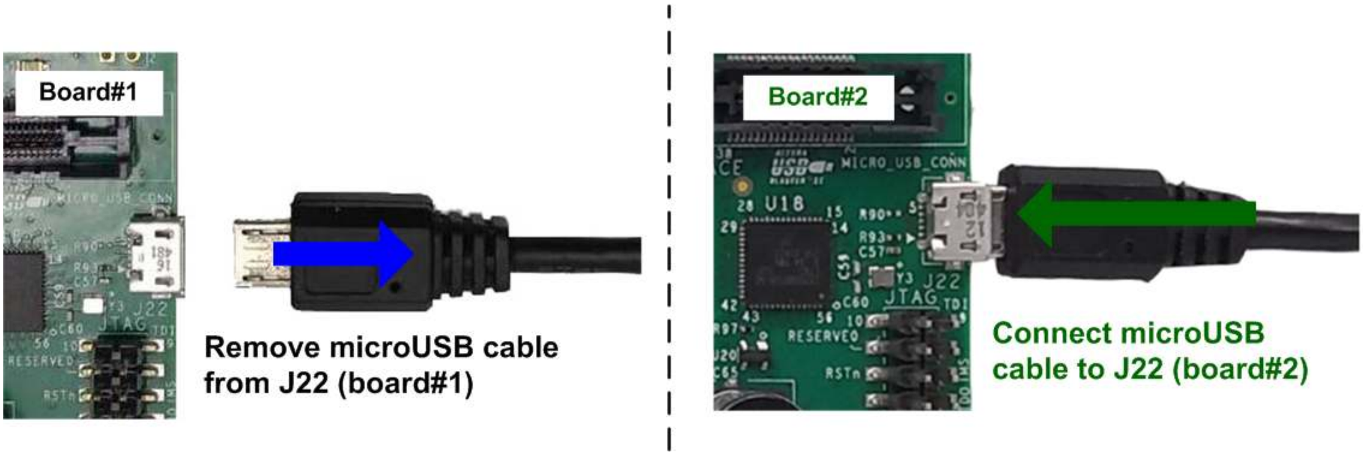

図 7-4: microUSB ケーブルの繋ぎ替え

6) 再度 “Clock Control”アプリケーションを起動し、ステップ 3)と同様に今度は FPGA ボード#2 の U50 周波数を 322.265625 MHz にセットします。ボード#2 のクロック設定が完了したらアプリケーションも終了します。

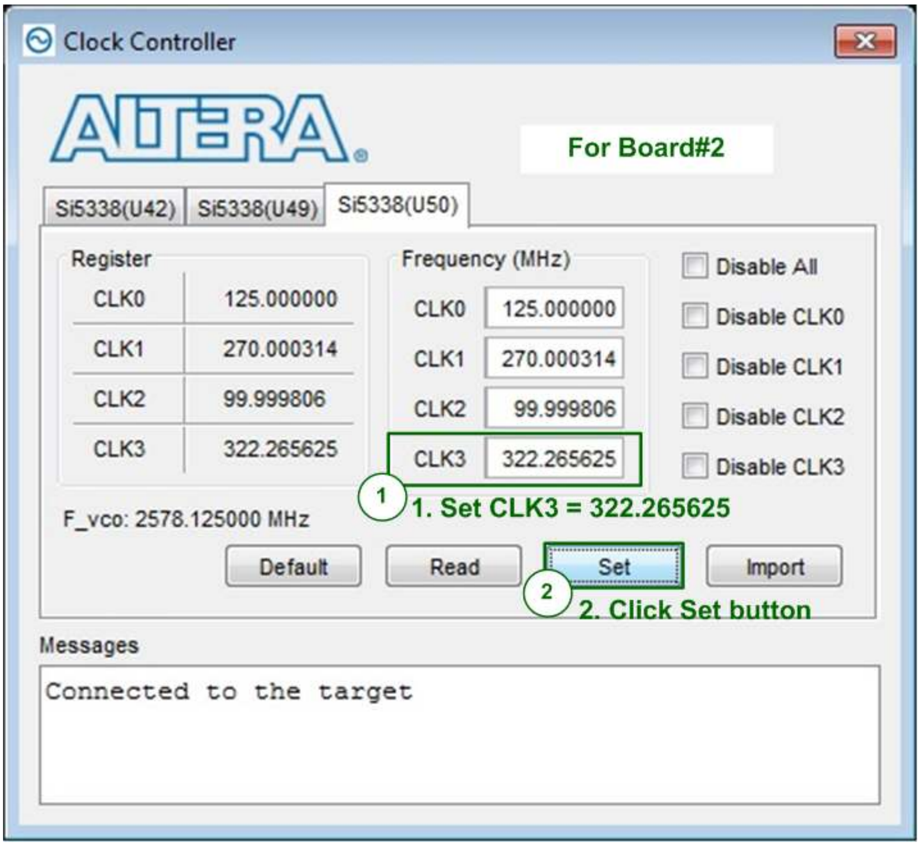

図 7-5: “Clock Control”アプリケーションでボード#2 の U50 も周波数を設定

7) (FPGA ボード#2 の micro USB ケーブルを繋いだまま)別の micro USB ケーブルを PC と FPGA ボード#1 間に追加接続します。すると PC は 2 つの FPGA ボードを USB-1 および USB-2 として認識します。

8) 10G ビット SFP+ 光トランシーバと光ケーブルを 2 つの FPGA ボード間で相互接続します。

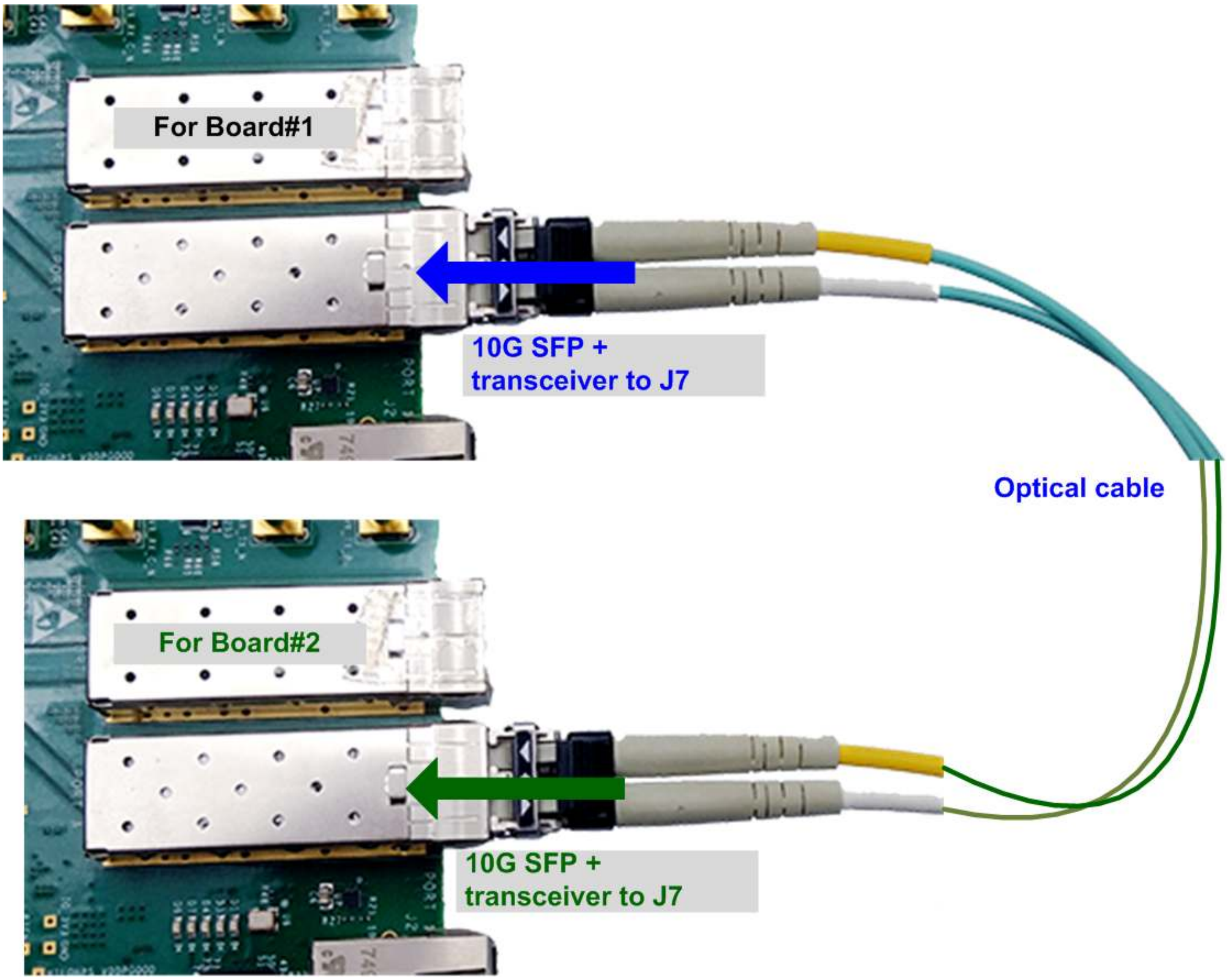

図 7-6: SFP+光トランシーバの接続

9) QuartusII プログラマを起動し以下の手順で USB-1 経由での FPGA コンフィグレーションを実行します。
   a. “Hardware Setup…”をクリックし USB-BlasterII[USB-1] を選択します。
   b. “Auto Detect”をクリックし FPGA (10AS066N3) を選択します。

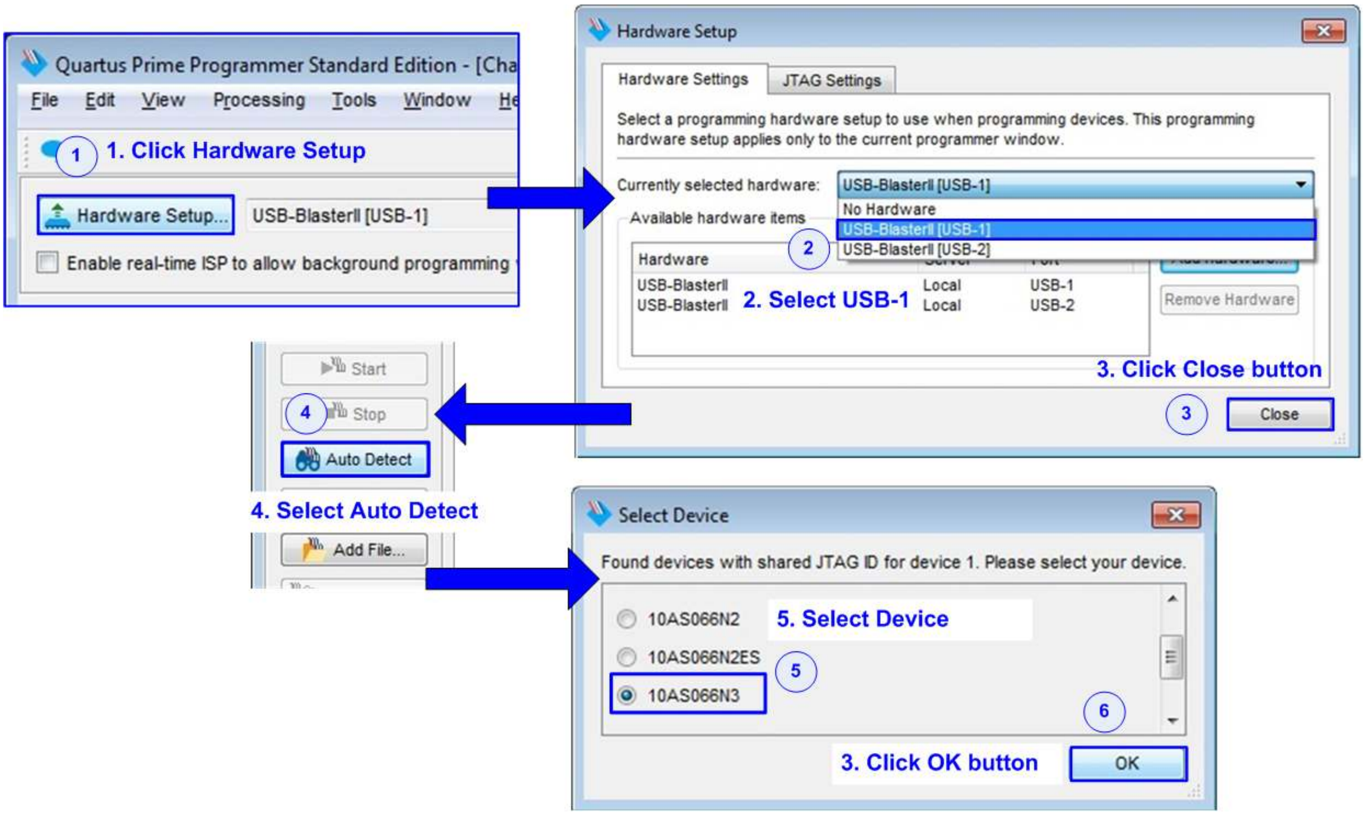

図 7-7: USB-BlasterII の選択 (USB-1)

10) USB-1 を通して FPGA に SOF ファイルでコンフィグレーションします。
  a. A10 デバイスのアイコンを選択します。
  b. “Change File”ボタンを押します。
  c. ポップアップ・ウインドウから評価用 SOF ファイルを選択し、”open”ボタンをクリックします
  d. “program/Configure”を選択し”Start”ボタンを押してコンフィグレーションを実行します。
  e. プログレスの進捗が 100%となるのを待ちます。

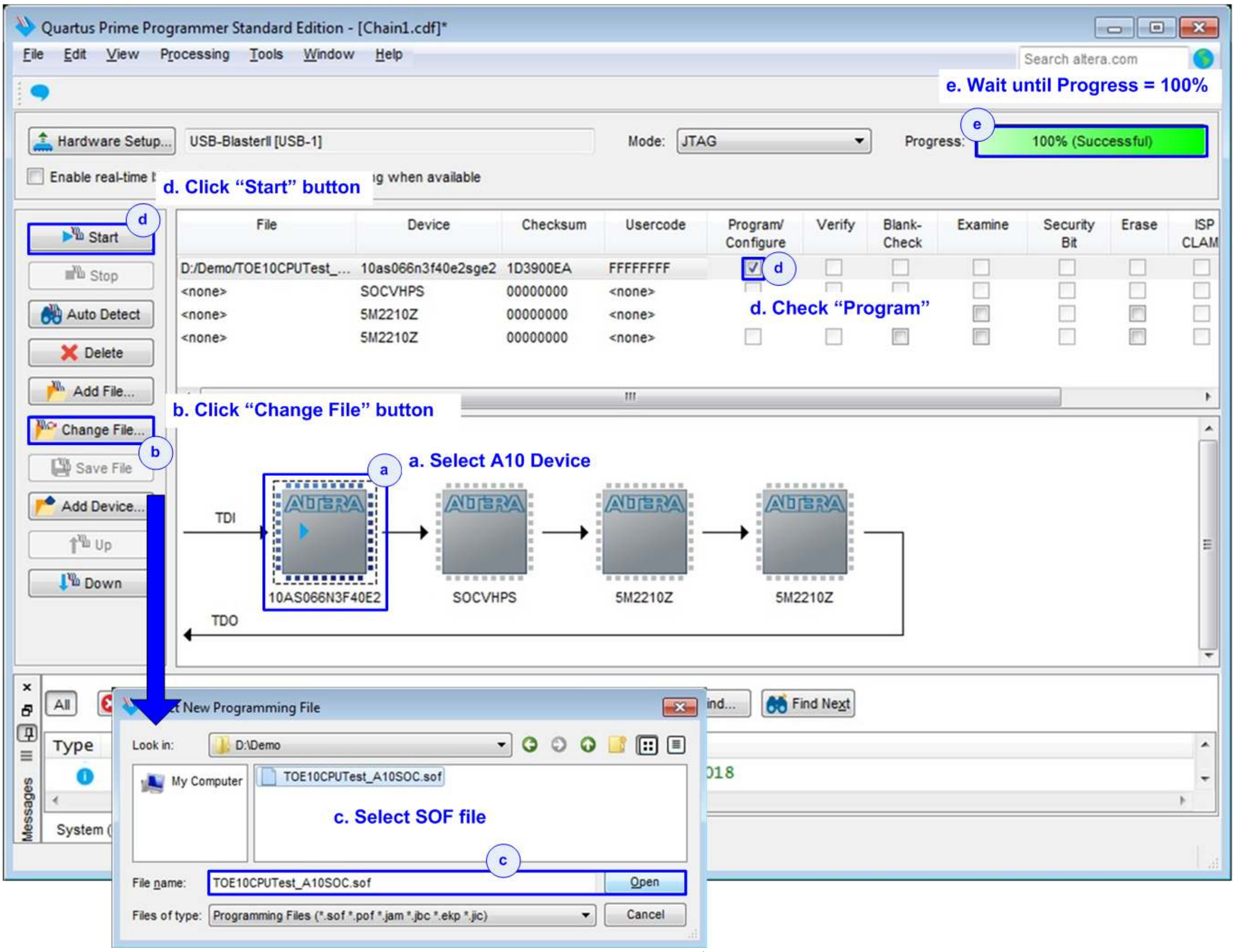

図 7-8: FPGA のコンフィグレーション

11) 上記のステップ 9) – 10)を、今度は USB-1 のかわりに USB-2 にて実行します。

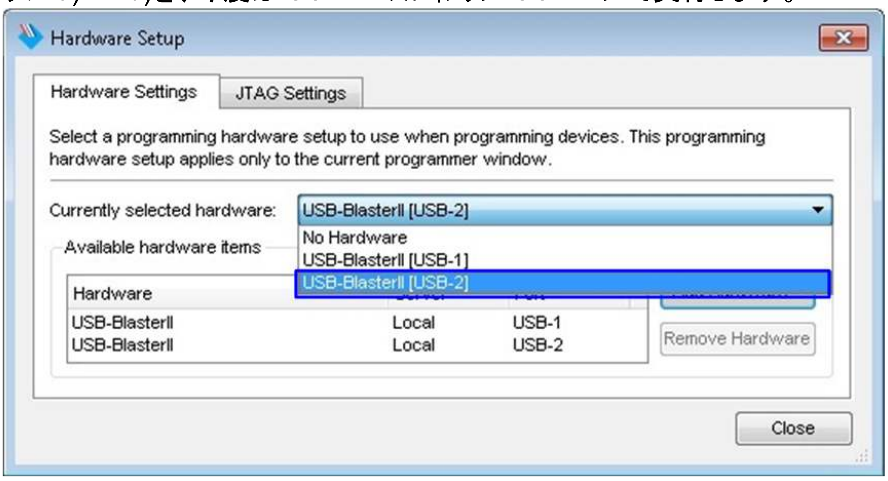

図 7-9: もう一方の FPGA をコンフィグレーションするため USB-1 から USB-2 へ切り替え

12) NiosII コマンド･シェルを開きます。
   a. FPGA#1 ボード向けに、”nios2-terminal --cable 1”コマンドを実行します。
   b. FPGA#2 ボード向けに、”nios2-terminal --cable 2”コマンドを実行します。
   c. 各FPGAボードで図7-10に示すようそれぞれモード(片方がサーバーでもう片方がクライアント)を選択します。

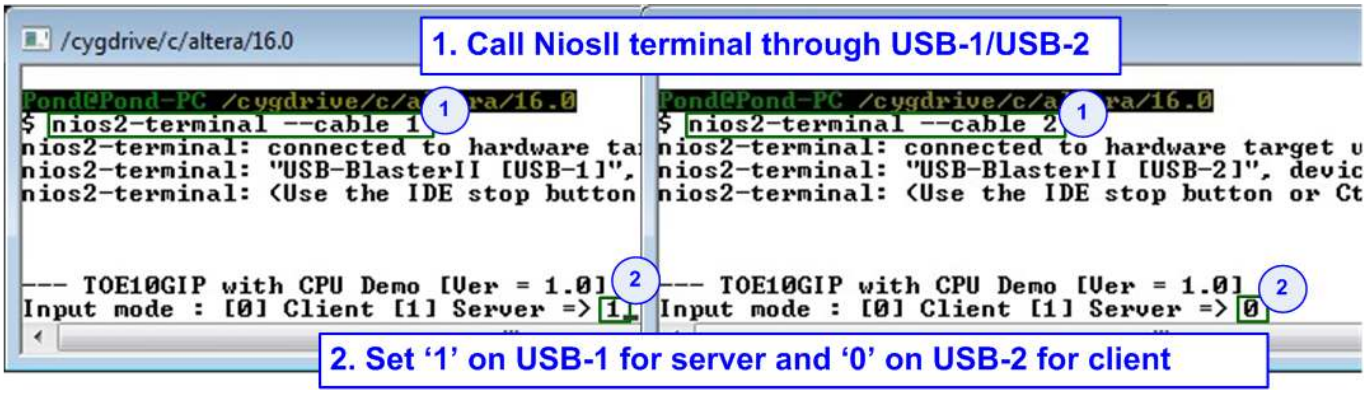

図 7-10: モードの入力

13) NiosII コマンド･シェル上にサーバーおよびクライアントのモードに従ってデフォルト･パラメータが表示されます。ここでユーザは’x’を入力することでパラメータの設定をスキップしメイン･メニューに入ることができます。
ご注意: 必ずクライアントよりもサーバの方を先にパラメータの設定を完了しリセット処理を行ってください。

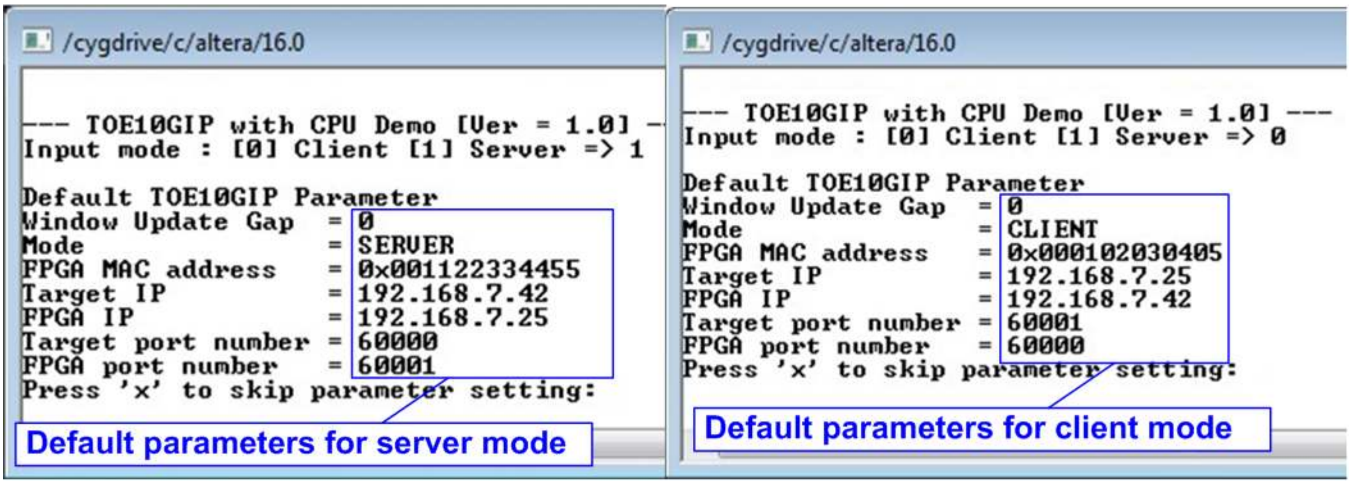

図 7-11: デフォルトのパラメータを表示した NiosII ターミナル画面 (‘x’の入力は必ずサーバー側から行ってください)

14) パラメータの設定が完了すると IP コアは初期化プロセスを開始します。 その処理が終わると “IP initialization complete”が表示されます。その後メイン・メニューが NiosII コマンド・シェル上に表示されます。

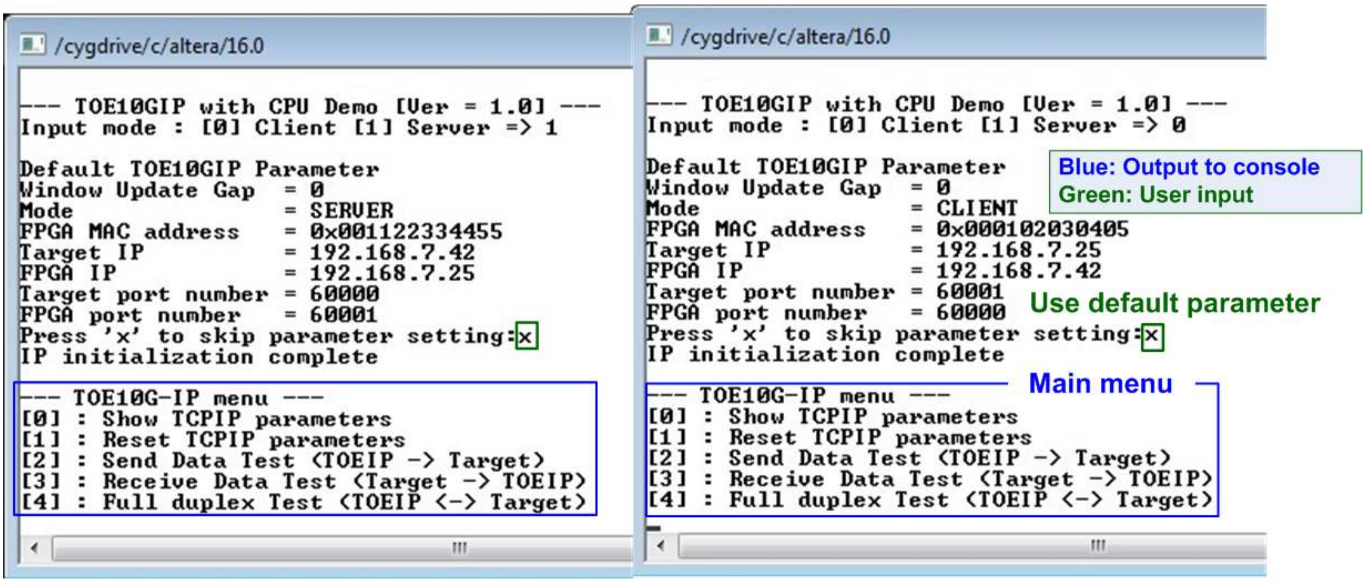

図 7-12: メイン・メニュー

8 メイン・メニュー

8.1 パラメータ現在値の表示

‘0’を入力することで本デモの各パラメータ現在値を表示します。 全部で 7 種類のパラメータが NiosII コマンド・シェル上に表示されます。

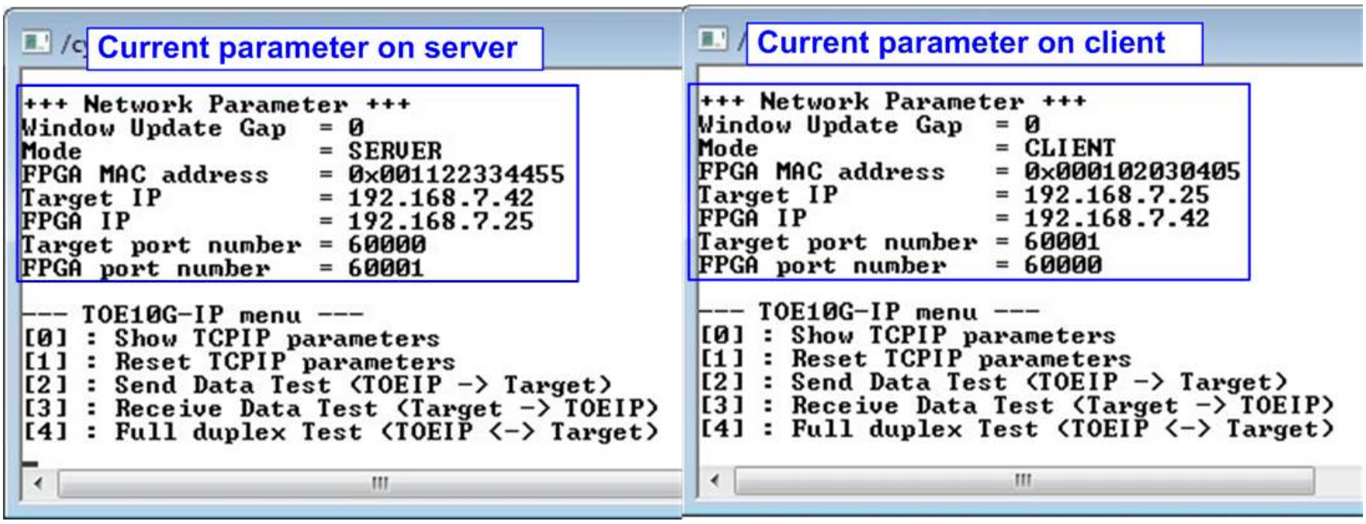

図 8-1: パラメータ現在値の表示例

1) Window Update Gap: ウインドウ・アップデート・パケットを送信する閾値を設定します。有効な値は 0x00 – 0x3F (0-63)です。ここで設定する敷居値の単位は 1K バイトです。 初期値は 0 (ウインドウ・アップデート機能を使わない)です。
2) Mode: TOE10G-IP コアがサーバーまたはクライアントのどちらのモードとするかをセットします。‘0’がクライアントで’1’がサーバーです。
3) FPGA MAC address: FPGA(自分側)の MAC アドレスを 48 ビットの 16 進数で設定します。初期値はクライアントの場合 0x000102030405 で、サーバーの場合 0x001122334455 です。
4) Target IP: 転送相手側の IP アドレスを 10 進数で設定します。 初期値はクライアントの場合 192.168.7.25 で、サーバーの場合 192.168.7.42 です。
5) FPGA IP: FPGA(自分側)の IP アドレスを 10 進数で設定します。 初期値はクライアントの場合 192.168.7.42 で、サーバーの場合 192.168.7.25 です。
6) Target port number: 転送相手側のポート番号を 10 進数で指定します。 初期値はクライアントの場合 60001 で、サーバーの場合 60000 です。
7) FPGA port number: FPGA(自分側)のポート番号を 10 進数で指定します。 初期値はクライアントの場合 60000 で、サーバーの場合 60001 です。

パラメータを変更する場合、メニュー[1]を選んでください。

8.2 TOE10G-IP コアのリセット

‘1”で選択されるこのメニューでは TOE10G-IP コアへ設定する各パラメータを変更しコアをリセットすることができます。その後更新したパラメータ情報が表示されます。ユーザは’x’を入力することで現在の設定と同じパラメータを使いコアをリセットするか、あるいは他(‘x’以外)キーを入力することでパラメータを変更できます。入力値が無効な値(たとえば’n’など)の場合パラメータは変更されません。より詳細については以下に説明します。

ご注意:

1. 本メニューでサーバー側のパラメータをリセットする場合、クライアント側の FPGA も同じメニューでリセットしてください。さらにリセットはサーバー側の方を先に行うことでクライアントからの ARP 要求を待機してください。

2. クライアントとサーバーのパラメータはお互いに合致しなくてはなりません。
   - **a. サーバーの Target IP = クライアントの FPGA IP**
   - **b. サーバーの FPGA IP = クライアントの Target IP**
   - **c. サーバーの Target port number = クライアントの FPGA port number**
   - **d. サーバーの FPGA port number = クライアントの Target port number**

1) Mode: FPGA の初期化モードを指定します。クライアントの場合‘0’を、サーバーの場合‘1’をセットしてください。2 枚の FPGA ボードにて片方がサーバーもう片方がクライアントと、必ず異なるモードとする必要があります。
2) FPGA MAC address: 12 桁の 16 進数で入力します。先頭に”0x”のサフィックスをつけて 16 進数であることを示してください。
3) FPGA IP address: “.”(ドット)をセパレータとした 4 つの 10 進数で入力します。各入力値にて有効な範囲は 0-255 です。
4) FPGA port number: 有効な範囲は 0-65535 です。
5) Target IP address: 上記 3)と同様に 10 進数で入力 します。
6) Target port number: 有効な範囲は 0-65535 です。

パラメータの入力が終わったら更新されたパラメータが NiosII コマンド・シェル上に表示されます。次に IP コアに対してリセットが発行され、新たなパラメータがコアにロードされます。最後に “IP initialization complete”のメッセージが図 8-2 のように表示され初期化プロセスが完了したことを示します。

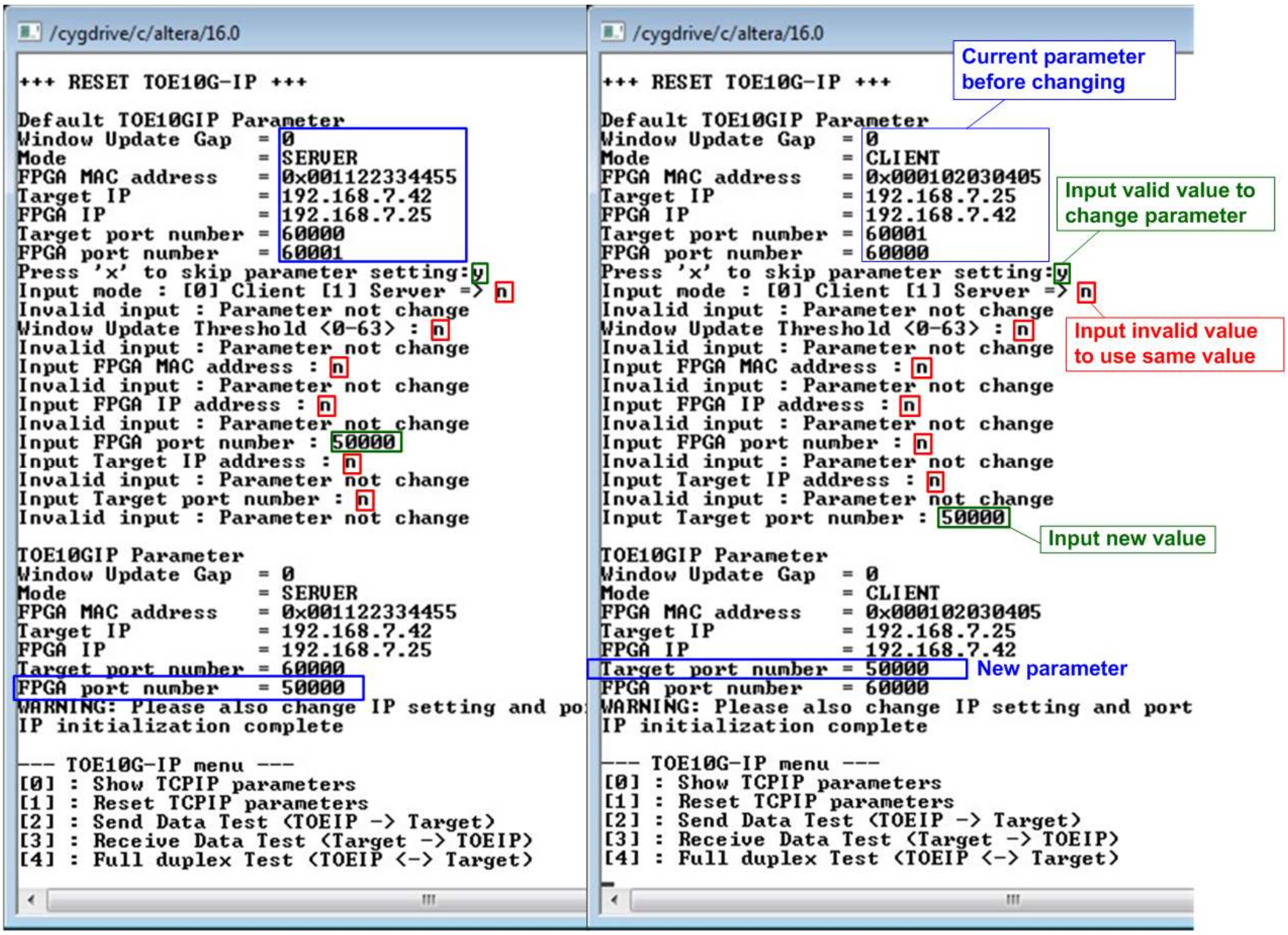

図 8-2: パラメータの変更結果例

8.3 データ送信テスト(サーバーからクライアントへ)

サーバーからクライアントへのデータ送信をテストするためには、サーバー側 FPGA にてメニュー'2'を選択することで送信し、クライアント側 FPGA にてメニュー'3'を選択することで受信します。ユーザは NiosII コマンド・シェルを通してテスト・パラメータを設定できます。本テストの実施手順を以下に説明します。

1) サーバー側の NiosII コマンド・シェルにて送信テストとして 3 つのパラメータを入力します。
   a) 転送サイズを入力します。転送サイズの単位はバイト数です。有効な値は 0x8 - 0x7FFFFFF8 です。ただし入力数は必ず 8 の倍数でなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に"0x"のプレフィックスを付けてください。
   b) パケット・サイズを入力します。パケット・サイズの単位はバイト数です。有効な値は 8 – 8960 です。ただし入力数は必ず 8 の倍数でなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に"0x"のプレフィックスを付けてください。
   c) モードを入力します。本サーバーからの送信テストでは'1'をセットし FPGA をサーバー・モードとしてください。
2) 全ての入力が有効であった場合、"Wait Open connection …"のメッセージが表示されます。
3) クライアント側の NiosII コマンド・シェルにて受信テストとして 3 つのパラメータを入力します。
   a) 転送サイズを入力します。転送サイズの単位はバイト数です。有効な値は 0x8 - 0x7FFFFFF8 です。ただし入力数は必ず 8 の倍数でなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に"0x"のプレフィックスを付けてください。この値はサーバー側 FPGA の転送サイズ(上記 1a で入力する値)と等しくする必要があります。
   b) データ・ベリファイのモードを入力します。 ベリファイを行わない場合は'0'を、サーバーから受信したデータでベリファイを行う場合は'1'を入力してください。
   c) モードを入力します。本サーバーからの送信テストではクライアント側は'0'をセットしてください。

   入力されたパラメータが有効であった場合、テスト動作が開始されます。
4) クライアント側の動作が開始すると、サーバー・クライアント両方の NiosII コマンド・シェルにて 1 秒毎に転送サイズがリアルタイムで表示されます。 サーバー側では全データの送信が完了すると "Send data complete"のメッセージが NiosII コマンド・シェル上に表示されます。
5) サーバー側はコネクションをクローズします。 そしてサーバー・クライアントの両方の NiosII コマンド・シェルにて総転送サイズとパフォーマンス結果が表示されます。

図 8-3 にサーバーからのデータ送信テストを非ジャンボ･フレーム･サイズで実行した時の結果例を示します。図左側はサーバー側の NiosII コマンド･シェル画面で図右側はクライアント側の NiosII コマンド･シェル画面です。図 8-4 にジャンボ･フレームで実行したテスト例を示します。ジャンボ･フレームの場合非ジャンボ･フレームより良好なパフォーマンスが得られます。

入力したパラメータが無効であった場合、“Out-of-range input”または”Invalid input”のメッセージが表示され、テスト動作は図 5-5～図 5-7(FPGA～PC 間でのテストにおける無効な入力があった場合のメッセージ)と同じくキャンセルされます。

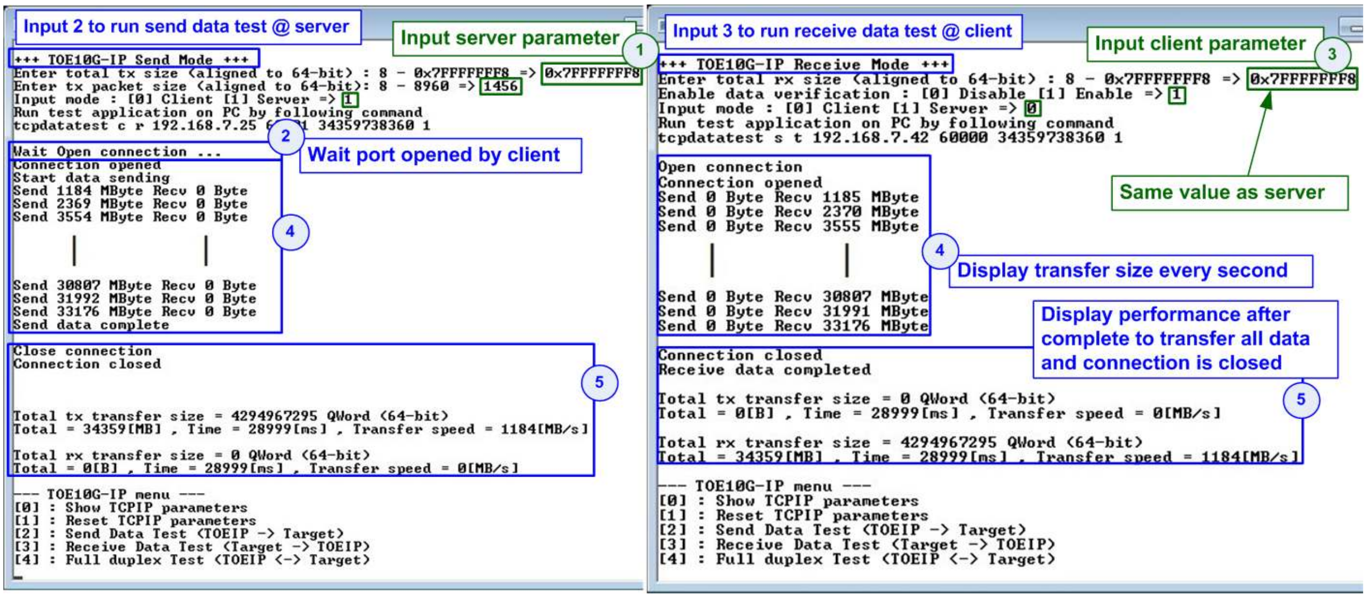

図 8-3: 非ジャンボ･フレームでのデータ送信テスト結果例

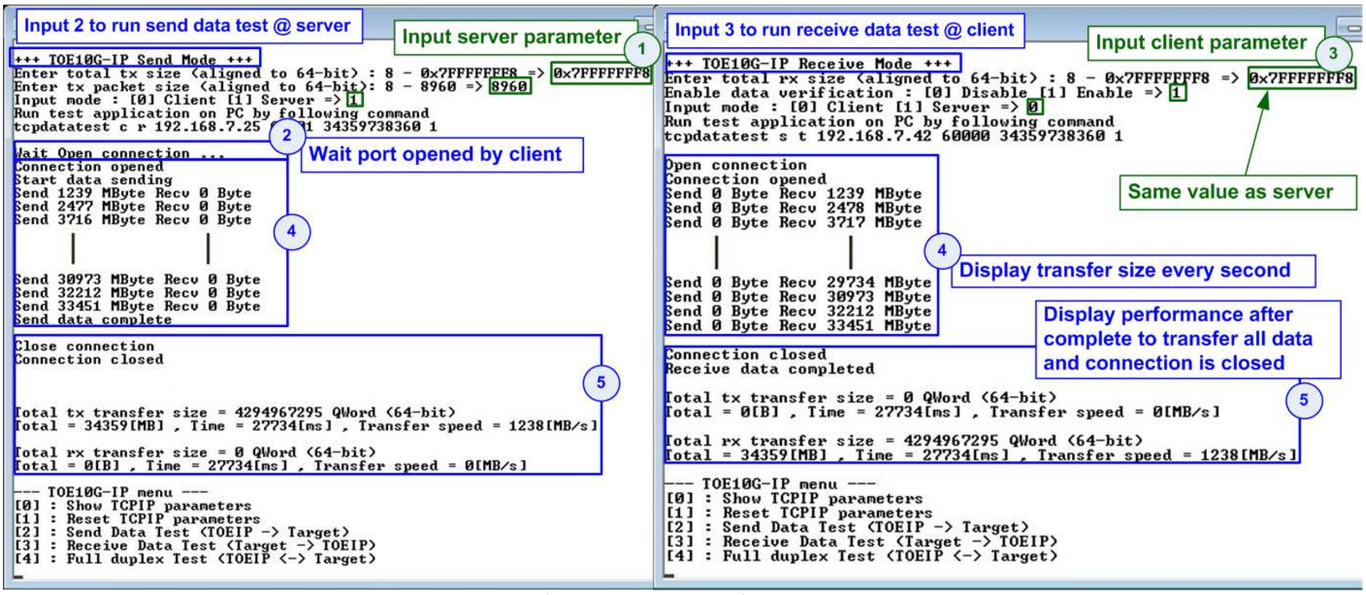

図 8-4: ジャンボ･フレームでのデータ送信テスト結果例

8.4 データ受信テスト(クライアントからサーバーへ)

クライアントからサーバーへのデータ受信をテストするためには、サーバー側 FPGA にてメニュー'3'を選択することで受信し、クライアント側 FPGA にてメニュー'2'を選択することで送信します。ユーザは NiosII コマンド・シェルを通してテスト・パラメータを設定できます。本テストの実施手順を以下に説明します。

1) サーバー側の NiosII コマンド・シェルにて受信テストとして 3 つのパラメータを入力します。
   a) 転送サイズを入力します。転送サイズの単位はバイト数です。有効な値は 0x8 - 0x7FFFFFF8 です。ただし入力数は必ず 8 の倍数でなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に"0x"のプレフィックスを付けてください。
   b) データ・ベリファイのモードを入力します。 ベリファイを行わない場合は'0'を、クライアントから受信したデータでベリファイを行う場合は'1'を入力してください。
   c) モードを入力します。本クライアントからの受信テストでは'1'をセットし FPGA をサーバー・モードとしてください。
2) 全ての入力が有効であった場合、"Wait Open connection …"のメッセージが表示されます。
3) クライアント側の NiosII コマンド・シェルにて送信テストとして 3 つのパラメータを入力します。
   a) 転送サイズを入力します。転送サイズの単位はバイト数です。有効な値は 0x8 - 0x7FFFFFF8 です。ただし入力数は必ず 8 の倍数でなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に"0x"のプレフィックスを付けてください。この値はサーバー側 FPGA の転送サイズ(上記 1a で入力する値)と等しくする必要があります。
   b) パケット・サイズを入力します。パケット・サイズの単位はバイト数です。有効な値は 8 – 8960 です。ただし入力数は必ず 8 の倍数でなくてはなりません。 ユーザは 10 進数か 16 進数で入力できます。 16 進数で入力する場合、先頭に"0x"のプレフィックスを付けてください。
   c) モードを入力します。本サーバーでの受信テストではクライアント側は'0'をセットしてください。

   入力されたパラメータが有効であった場合、テスト動作が開始されます。
4) クライアント側の動作が開始すると、サーバー・クライアント両方の NiosII コマンド・シェルにて 1 秒毎に転送サイズがリアルタイムで表示されます。
5) クライアント側が全データの送信を完了しコネクションをクローズすると、サーバー側の NiosII コマンド・シェル上に"Connection closed"と"Received data completed"のメッセージが表示されます。そしてサーバー・クライアントの両方の NiosII コマンド・シェルにて総転送サイズとパフォーマンス結果が表示されます。

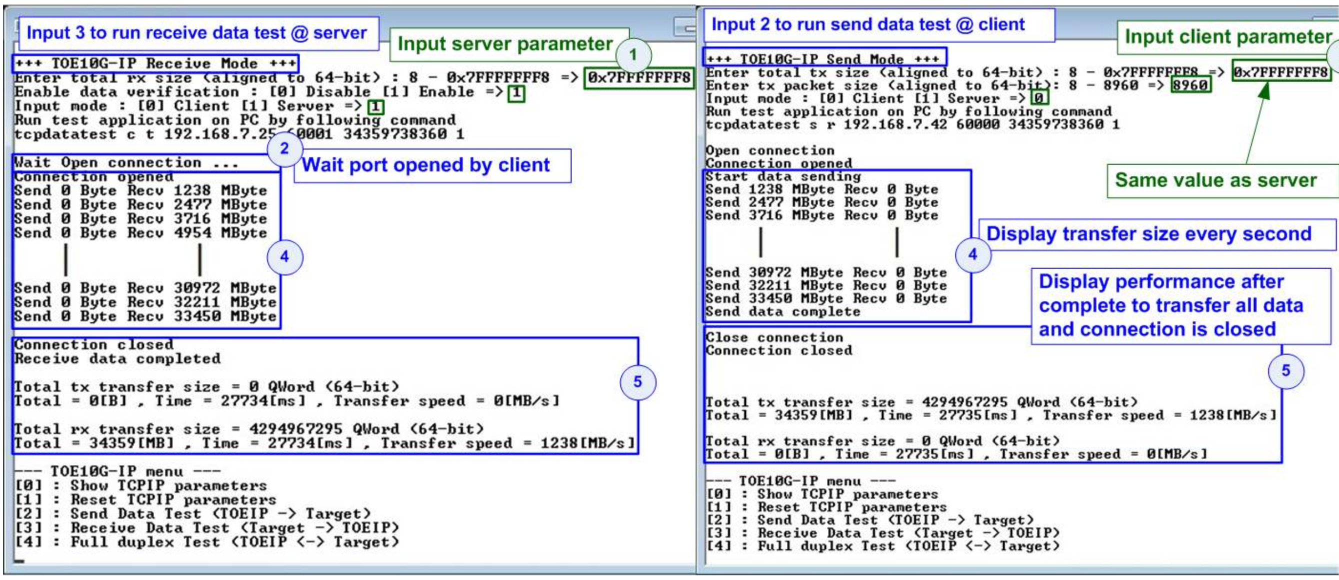

図 8-5: データ・ベリファイありのデータ受信テスト結果例

8.5 全二重通信テスト

‘4’で選択されるこのメニューではサーバーおよびクライアントの 2 枚の FPGA 間で同時に送受信する全二重通信をテストできます。ユーザは NiosII コマンド･シェルを通して通信するデータのテスト･パラメータを設定できます。本テストの実施手順を以下に説明します。

1) サーバー側の NiosII コマンド･シェルにて全二重通信テストとして 4 つのパラメータを入力します。
   a) 転送サイズを入力します。転送サイズの単位はバイト数です。有効な値は 0x8 - 0x7FFFFFFF8 です。ただし入力数は必ず 8 の倍数であり、かつクライアント側で設定する転送サイズと一致しなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に”0x”のプレフィックスを付けてください。
   b) パケット･サイズを入力します。パケット･サイズの単位はバイト数です。有効な値は 8 – 8960 です。ただし入力数は必ず 8 の倍数でなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に”0x”のプレフィックスを付けてください。
   c) データ･ベリファイのモードを入力します。 ベリファイを行わない場合は’0’を、クライアントから受信したデータでベリファイを行う場合は’1’を入力してください。
   d) サーバー側での本全二重通信テストでは’1’をセットし FPGA をサーバー･モードとしてください。
2) 全ての入力が有効であった場合、“Wait Open connection …”のメッセージが表示されます。
3) クライアント側の NiosII コマンド･シェルにて全二重通信テストとして 4 つのパラメータを入力します。
   a) 転送サイズを入力します。転送サイズの単位はバイト数です。有効な値は 0x8 - 0x7FFFFFFF8 です。ただし入力数は必ず 8 の倍数であり、かつサーバー側で設定する転送サイズと一致しなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に”0x”のプレフィックスを付けてください。
   b) パケット･サイズを入力します。パケット･サイズの単位はバイト数です。有効な値は 8 – 8960 です。ただし入力数は必ず 8 の倍数でなくてはなりません。ユーザは 10 進数か 16 進数で入力できます。16 進数で入力する場合、先頭に”0x”のプレフィックスを付けてください。
   c) データ･ベリファイのモードを入力します。 ベリファイを行わない場合は’0’を、サーバーから受信したデータでベリファイを行う場合は’1’を入力してください。
   d) クライアント側での本全二重通信テストでは’0’をセットし FPGA をクライアント･モードとしてください。

   入力されたパラメータが有効であった場合、テスト動作が開始されます。
4) クライアント側の動作が開始すると、サーバー･クライアント両方の NiosII コマンド･シェルにて 1 秒毎に転送サイズがリアルタイムで表示されます。
5) クライアント側が全データの送信および全データの受信を完了しコネクションをクローズするとクライアント側の NiosII コマンド･シェル上に “Send data completed”のメッセージが表示されます。そしてサーバー･クライアントの両方の NiosII コマンド･シェルにて総転送サイズとパフォーマンス結果が表示されます。

   そしてステップ 4)～ステップ 5)が永久に繰り返されます。テストを終了する場合はサーバーおよびクライアント両方の NiosII コマンド･シェルで何かキー入力を行ってください。

図 8-6 に全二重通信テストの結果例を示します。図左側はサーバー側の NiosII コマンド・シェル画面で図右側はクライアント側の NiosII コマンド・シェル画面です。

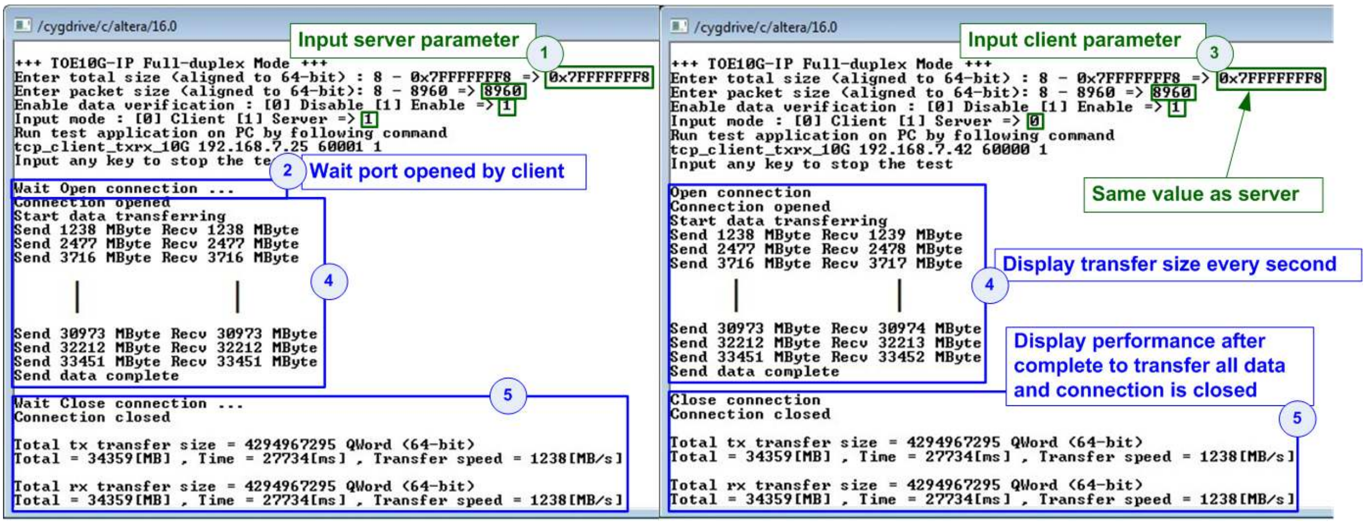

図 8-6: 全二重通信テストをベリファイありで実行した結果例

9 更新履歴

Revision	日付	内容
1.0	19-Mar-18	Initial version release
1.1	27-Mar-18	Add Part A (FPGA<->PC test)
1.1J	2018/04/03	日本語版(英語版 1.1 に対応)を作成
1.2J	2018/04/09	英語版 1.2 修正に合わせた更新、図の修正や説明の追加に対応