



この翻訳版ドキュメントのメンテナンスは終了しております。

この文書には、古いコンテンツや商標が含まれている場合があります。

最新情報につきましては、次のリンクから英語版の最新資料をご確認ください。

<https://www.intel.com/content/www/us/en/programmable/documentation/lit-index.html>

Please take note that this document is no longer being maintained. It may contain legacy content and trademarks which may be outdated.

Please refer to English version for latest update at

<https://www.intel.com/content/www/us/en/programmable/documentation/lit-index.html>

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。こちらの日本語版は参考用としてご利用ください。設計の際には、最新の英語版で内容をご確認ください。

AN-617-1.0

RapidIO のダイナミック・データ・レートのリコンフィギュレーションのリファレンス・デザインは、Stratix® IV GX EP4SGX230KF40C3ES シグナル・インテグリティ・トランシーバの開発ボードを使用して RapidIO® MegaCore® ファンクションのデータ・レートをリコンフィギュレーションする ALTGX_RECONFIG メガファンクションを使用する方法を示します。ALTGX_RECONFIG メガファンクションは、デザインを再コンパイルせずに実行時にデータ・レートを変更することができます。

このアプリケーション・ノートには以下の項が含まれています。

- 「機能の説明」
- 1-12 ページの「リファレンス・デザインの使用」
- 1-17 ページの「検証用のテスト・ケース (デュアル・ボードの接続)」



RapidIO MegaCore ファンクションについて詳しくは、「[RapidIO MegaCore Function User Guide](#)」を参照してください。

機能の説明

リファレンス・デザインは、ダイナミック・リコンフィギュレーション・プロセスをコントロールするために、Nios® II エンベデッド・プロセッサを使用します。データ・レートのリコンフィギュレーションを処理と実行するために Nios II プロセッサへのソフトウェア・コマンドを提供することができます。リコンフィギュレーション・プロセスは、希望のデータ・レートと RapidIO MegaCore ファンクションのトランシーバをリコンフィギュレーションする ALTGX_RECONFIG メガファンクションをトリガします。

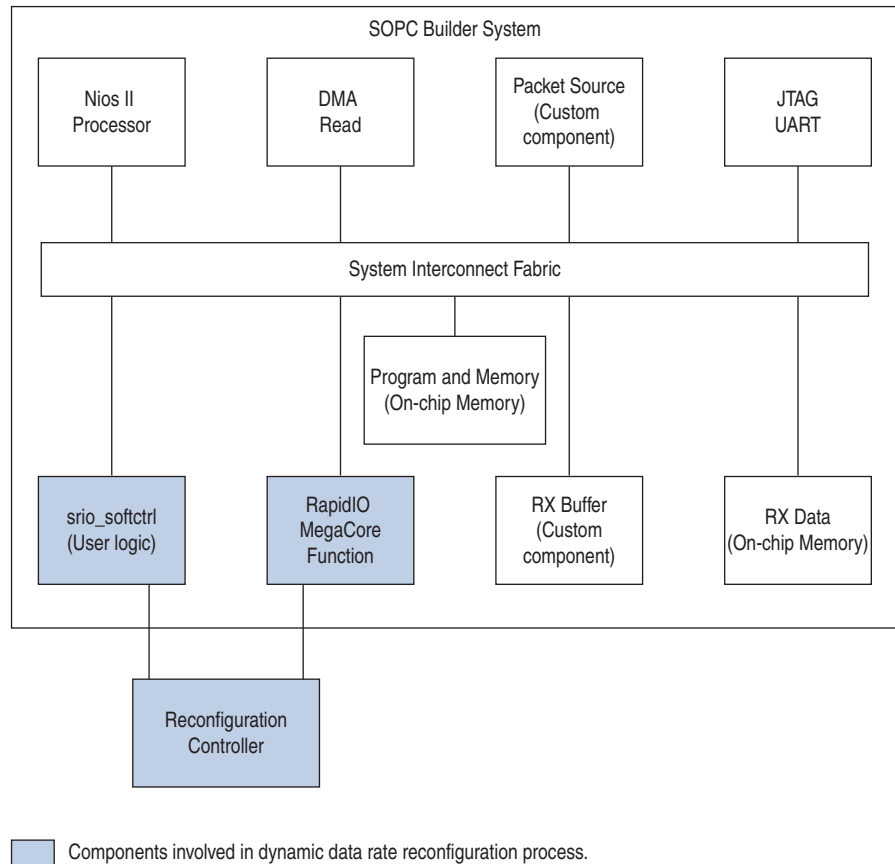
リファレンス・デザインは、チャンネルとクロック・マルチプライヤ・ユニット (CMU) の PLL (Phase-Locked Loop) をリコンフィギュレーションするリコンフィギュレーションモードの RapidIO MegaCore ファンクションのトランシーバを使用しています。チャンネルと CMU PLL のリコンフィギュレーション・モードでは、データ・レートなどのトランシーバの設定を変更するには、メモリ初期化ファイル (.mif) を使用します。リファレンス・デザインは、RapidIO MegaCore ファンクションの送信 (TX) および受信 (RX) チャンネルをリコンフィギュレーションします。



システム・アーキテクチャ

図 1 に、システム・アーキテクチャの概要を示します。

図 1. システム・アーキテクチャの概要



以下の項では、リファレンス・デザインにおけるメイン・システム・コンポーネントの役割について説明します。

Nios II プロセッサ

Nios II プロセッサは、プロセスやデータ・レートのリコンフィギュレーション・コマンドを発行すると、ダイナミック・データ・レートのリコンフィギュレーション・プロセスを処理および実行します。

DMA 読み出し

この DMA 読み出しコンポーネントは、RapidIO MegaCore ファンクションの I/O スレーブ・リード・ポートから RX データ・メモリにデータを転送します。

パケット・ソース (カスタム・コンポーネント)

パケット・ソースのカスタム・コンポーネントは、RapidIO MegaCore ファンクションのライト・スレーブ・ポートへの I/O バーストの転送パケットを生成します。

JTAG UART

JTAG UART コンポーネントは、ユーザーのデバイスに Nios II ターミナルと通信するメカニズムを提供します。Nios II ターミナルは RapidIO MegaCore ファンクション・ドライバ用のユーザー・インタフェースです。

プログラムおよびメモリ (オンチップ・メモリ)

プログラム・コンポーネントおよびメモリ・コンポーネントは、Nios II プロセッサのソフトウェアのプログラム・コードを格納します。ソフトウェアのプログラム・コードをダウンロードすると、このコンポーネントは、コードを格納し、Nios II プロセッサは、コードを実行します。

RX バッファ (カスタム・コンポーネント)

RX バッファのカスタム・コンポーネントは、それぞれ RapidIO MegaCore ファンクションのマスタ・ポートに書き込まれたデータ、およびマスタ・ポートから読み出されるデータを格納します。

RX データ オンチップ・メモリ)

RX データは、DMA リード・コンポーネントからの読み出しデータを格納します。

ダイナミック・データ・レートのリコンフィギュレーションに役割を持っています。それは、次のセクションで説明します。

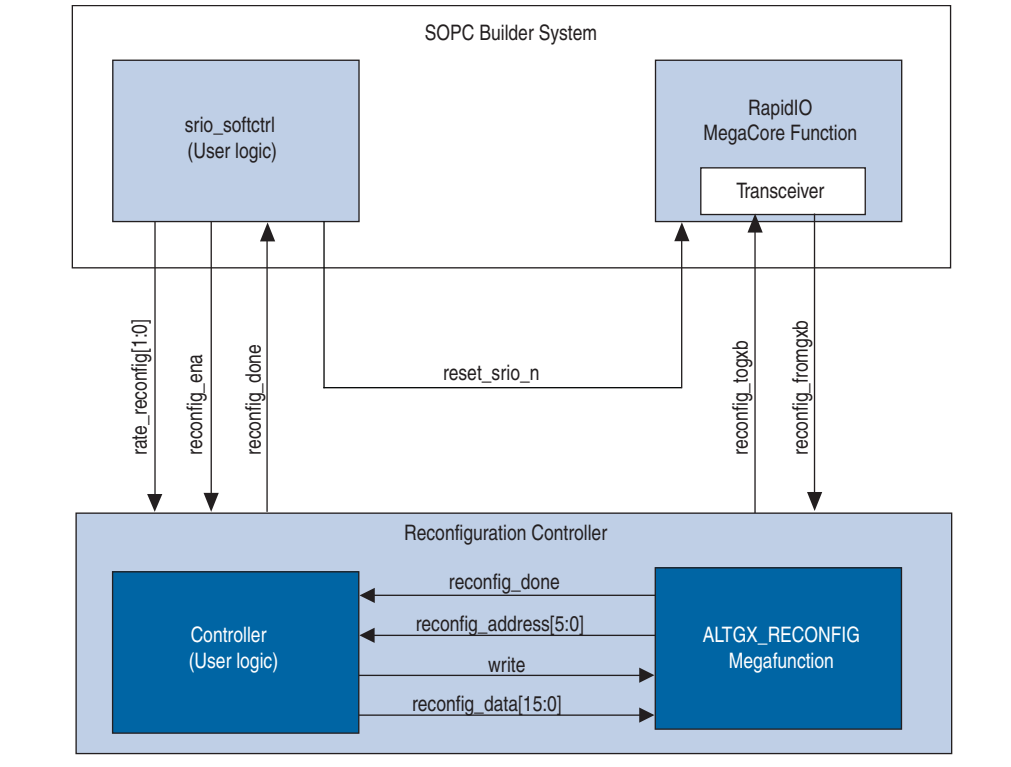
ダイナミック・データ・レートのリコンフィギュレーション・コンポーネント

この項では、以下のダイナミック・データ・レートのリコンフィギュレーション・コンポーネントについて説明します。

- [srio_softctrl \(ユーザー・ロジック\)](#)
- [RapidIO MegaCore ファンクション](#)
- [リコンフィギュレーション・コントローラ](#)


には `srio_softctrl`、リコンフィギュレーション・コントローラ、および RapidIO MegaCore ファンクションのコンポーネントのための信号を示します。


図 2. `srio_softctrl`、リコンフィギュレーション・コントローラ、および RapidIO MegaCore ファンクションの信号



`srio_softctrl` (ユーザー・ロジック)

`srio_softctrl` のユーザー・ロジックは、リコンフィギュレーション・コントローラにリコンフィギュレーション命令を送信し、同時に RapidIO MegaCore ファンクションをリセットするリセット信号を送信します。このユーザー・ロジックは、Nios II プロセッサからリコンフィギュレーション・コントローラにリコンフィギュレーション・ソフトウェアのコマンドを提供するための Avalon® Memory-Mapped (Avalon-MM) インタフェースに接続します。

 SOPC Builder システム内のカスタム・コンポーネントを作成するには、`srio_softctrl` のユーザー・ロジックは、他のコンポーネントと通信するためのコンポーネント・エディタによって認識される Avalon-MM インタフェース信号を提供します。コンジットでのエクスポート信号 (`reset_srio_n` を除く) はリコンフィギュレーション・コントローラに接続します。`reset_srio_n` 信号は、RapidIO MegaCore ファンクションのメイン・リセットに接続します。

 SOPC Builder システムの Verilog HDL ファイルに基づいて新しいコンポーネントを作成する方法について詳しくは、「*Quartus II* ハンドブック」の「*Volume 4: SOPC Builder*」を参照してください。

RapidIO MegaCore ファンクション

このコンポーネントはリンク・パートナーに RapidIO リンクを確立します。また、RapidIO MegaCore ファンクションは、Avalon-MM インタフェース上に渡したトランザクションに対応する RapidIO トランザクションに変換して、RapidIO シリアル・リンクを介して転送します。RapidIO MegaCore ファンクションは、RapidIO シリアル・リンクからの RapidIO トランザクションを I/O バースト転送に変換し、これらのバースト転送に対応する Avalon-MM スレーブまたはマスタ・ポートに渡します。

このリファレンス・デザインでは、RapidIO MegaCore ファンクションは、次のデフォルトのパラメータ設定があります。

- 最初のデータ・レート : 2.5 GBaud
- 基準クロック周波数 : 156.25 MHz
- モード選択 : 1x シリアル

リコンフィギュレーション・コントローラ

リコンフィギュレーション・コントローラ・ブロックは、ALTGX_RECONFIG メガファンクションおよびコントローラ（ユーザー・ロジック）で構成されています。リコンフィギュレーション・コントローラ・ブロックは、srio_softctrl ユーザー・ロジックから命令を受信し、トランシーバをリコンフィギュレーションします。

ALTGX_RECONFIG メガファンクション

ALTGX_RECONFIG メガファンクションは、RapidIO MegaCore ファンクションのトランシーバをリコンフィギュレーションします。リコンフィギュレーションするトランシーバに、ALTGX_RECONFIG メガファンクションは、RapidIO MegaCore ファンクションに送信するために正しいコンフィギュレーションの設定を持つ必要があります。**.mif** ファイルは、これらのコンフィギュレーションの設定が含まれています。リファレンス・デザインでは、チャンネルと CMU PLL のリコンフィギュレーション・モードが使用されています。reconfig_mode_sel ポートの値が 101 に設定されています。

コントローラ（ユーザー・ロジック）

コントローラは、割り当てられた RAM から **.mif** ファイルの内容を取得するためにユーザー定義のロジックを含みます。リコンフィギュレーション時に、コントローラは srio_softctrl ユーザー・ロジックから rate_reconfig[1:0] 入力信号に基づいて **.mif** ファイルを選択します。コントローラは、ALTGX_RECONFIG メガファンクションで選択する **.mif** ファイルの内容をロードします。

リファレンス・デザインは、*RapidIO Specification, 2.1* で要求されるさまざまなデータ・レート（5.000、3.125、2.500、および 1.250 GBaud）でトランシーバのコンフィギュレーションの設定を指定する 4 つの **.mif** が提供されています。このリファレンス・デザイン用の **.mif** ファイルは reconfig_mif のフォルダに格納されています。

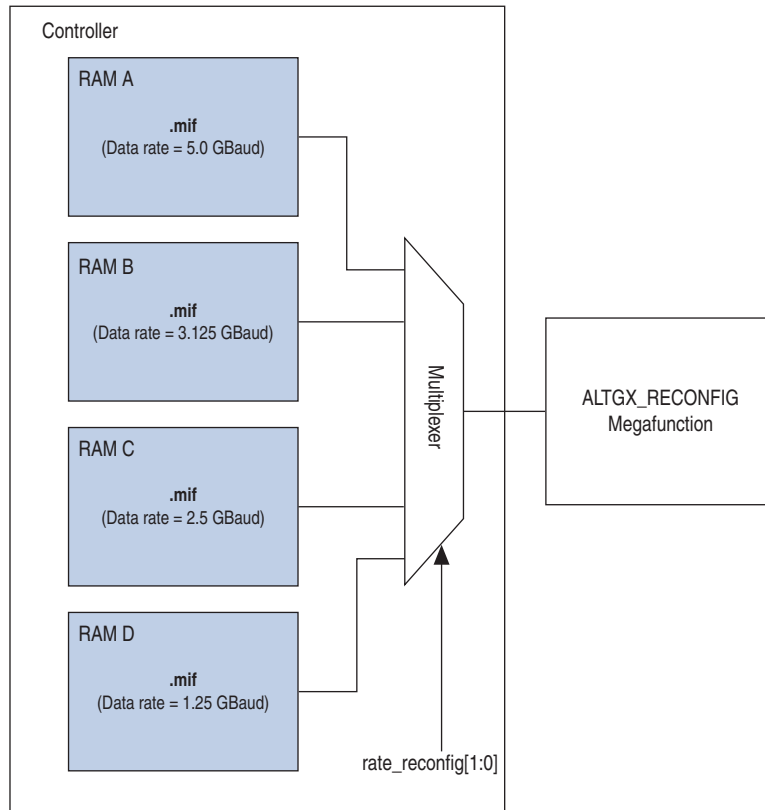
4 つの RAM はありますが、各 RAM は 1 つの **.mif** が格納されます。ボード上でデザインを実行すると、RAM からの **.mif** ファイルの内容にアクセスできます。



複数の **.mif** ファイルのインスタンス化について詳しくは、「Stratix IV デバイス・ハンドブック Volume 2」の「Stratix IV ダイナミック・リコンフィギュレーション」の章を参照してください。

図 3 にコントローラのアーキテクチャを示します。

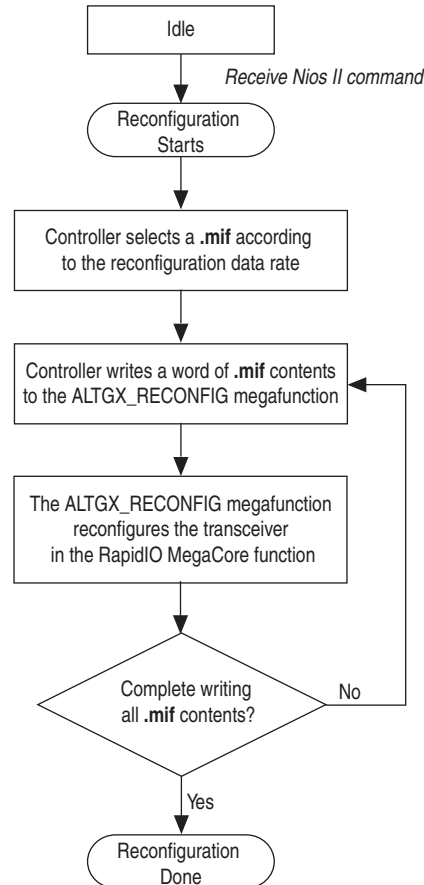
コントローラのコンポーネント・アーキテクチャ



ダイナミック・リコンフィギュレーション・プロセス

図 4 に、RapidIO MegaCore ファンクションのダイナミック・データ・レートのリコンフィギュレーションのためのプロセス・フローを示します。

ダイナミック・データ・レートのリコンフィギュレーション・プロセスのフロー



Nios2-terminal で `rate_reconfig <desired data rate>` のコマンドを入力すると、Nios II プロセッサは、リコンフィギュレーション・プロセスを開始するための Avalon-MM インタフェースを介して `srio_softctrl` ユーザー・ロジックにコマンドを送信します。Nios2-terminal でのソフトウェア・コマンドの `rate_reconfig<desired data rate>` を入力すると、`srio_softctrl` のユーザー・ロジックは `reconfig_ena` 信号をアサートします。`reconfig_ena` 信号がリコンフィギュレーション・プロセスを開始し、`rate_reconfig[1:0]` 信号で希望のデータ・レートを符号化します。`rate_reconfig[1:0]` 信号は、選択される `.mif` を決定するマルチプレクサ選択信号です。

コントローラは、適切な `.mif` ファイルを選択した後、それは ALTGX_RECONFIG メガファンクションに `.mif` ファイルの内容をロードします。ALTGX_RECONFIG メガファンクションは、`reconfig_togxb` 信号を介して新しいトランシーバの設定での RapidIO MegaCore ファンクションのトランシーバをリコンフィギュレーションします。

コントローラが **.mif** ファイルをロードするとき、それは **ALTGX_RECONFIG** メガファンクションにライト信号をアサートします。各ライト・サイクルでは、コントローラは **ALTGX_RECONFIG** メガファンクションに **.mif** ファイルの内容の 1 ワードを書き込みます。コントローラは **ALTGX_RECONFIG** メガファンクションへのすべての内容の書き込みを完了するまでこのプロセスが繰り返されます。

コントローラが **.mif** ファイルの内容の書き込みを完了すると、**ALTGX_RECONFIG** メガファンクションはリコンフィギュレーション・プロセスが正常に完了したことを示すために **reconfig_done** 信号をアサートします。

内部信号

表 1 に、4 ページの図 2 に示すダイナミック・データ・レートのリコンフィギュレーション・コンポーネント用の内部リコンフィギュレーション信号をリストします。

表 1. ダイナミック・データ・レートのリコンフィギュレーション・コンポーネントの内部リコンフィギュレーション信号 (その 1)

信号	説明
rate_reconfig[1:0]	RapidIO MegaCore ファンクションのトランシーバにどのデータレートをリコンフィギュレーションする 2 ビットのコントロール信号です。 2'b00 = 5.000 GBaud 2'b01 = 3.125 GBaud 2'b10 = 2.500 GBaud 2'b11 = 1.250 GBaud
reconfig_ena	リコンフィギュレーションのプロセスをイネーブルします。srio_softctrl ユーザー・ロジックはこの信号をアサートすると、rate_reconfig 信号の値が有効です。
reconfig_done	リコンフィギュレーション・プロセスが完了するとリコンフィギュレーション・コントローラはアイドル状態に戻ったことを示すステータス信号。
reconfig_address[5:0]	.mif ファイルの内容の特定のワードを取得するためにコントローラに .mif ファイルの内容のアドレスを提供します。
write	各パルスの ALTGX_RECONFIG メガファンクションに書き込まれる .mif の内容のワードをイネーブルする連続的なパルス信号です。
reconfig_data[15:0]	コントローラから ALTGX_RECONFIG メガファンクションにリコンフィギュレーション .mif のコンテンツ・データをロードします。
reconfig_togxb	RapidIO MegaCore ファンクションのトランシーバをリコンフィギュレーションする ALTGX_RECONFIG メガファンクションからのリコンフィギュレーション信号バスです。

表 1. ダイナミック・データ・レートのリコンフィギュレーション・コンポーネントの内部リコンフィギュレーション信号 (その 2)

信号	説明
reconfig_fromgxb	ALTGX_RECONFIG メガファンクションにリコンフィギュレーション・データを提供するの RapidIO MegaCore ファンクションのトランシーバからのリコンフィギュレーション信号のバス出力です。
reset_srio_n	以下のいずれかの条件によってトリガリセットされる時に RapidIO MegaCore ファンクションをリセットするアクティブ Low 信号です。 <ul style="list-style-type: none"> ■ ソフトウェア・リセット (16 Avalon-MM システム・クロック・サイクル) ■ ハードウェア・リセット ■ リコンフィギュレーション・プロセス時のリセット (プロセスが完了するまでリセットを保持する)

Nios II エンベデッド・プロセッサのソフトウェア・アプリケーション

C プログラミング・ファイルの **srio_main_full.c** は、Nios II エンベデッド・プロセッサ用のソフトウェア・アプリケーションが含まれています。ソフトウェア・アプリケーション・ファイルでは、Nios II エンベデッド・プロセッサへのソフトウェアのコマンドを発行するによって、ダイナミック・データ・レートのリコンフィギュレーションを実行するなどのさまざまなタスクを実行することができます。

srio_main_full.c ファイルは **software_app** のフォルダに格納されており、リファレンス・デザインを実行すると、Nios II エンベデッド・プロセッサにダウンロードされます。

表 2 に、このリファレンス・デザインで使用されるソフトウェア・コマンドをリストします。Nios II エンベデッド・プロセッサおよび他のサブシステム・モジュールを指示するには、nios2-terminal におけるソフトウェア・コマンドを入力します。

表 2. Nios II エンベデッド・プロセッサ用のソフトウェア・コマンド

ソフトウェア・コマンド	説明
h	すべての使用可能な命令をリストします。
start	<p>パケットのソース・モジュールは、RapidIO MegaCore ファンクションへのデータ・パケットの送信を開始します。</p> <p>RapidIO の内部レジスタの初期化。設定されている特定のレジスタの詳細については、srio_main_full.c ファイルを参照してください。</p> <p>RapidIO リンクのステータスをチェックします。</p> <p>リンクがアップしている場合、コマンド = リンクがアップ状態に戻った。</p> <p>リンクがダウンしている場合、= リンクダウンコマンドを返しました</p> <p>ソフトウェアリセット。リセットされますの RapidIO MegaCore ファンクション。</p> <p>データ・レートのリコンフィギュレーション・プロセスを開始します。リコンフィギュレーション後、RapidIO MegaCore ファンクションは、所望のデータ・レートで動作します。</p>

表 2. Nios II エンベデッド・プロセッサ用のソフトウェア・コマンド

ソフトウェア・コマンド	説明
stop	パケットのソース・モジュールは、RapidIO MegaCore ファンクションへのデータ・パケットの送信を停止します。
init	RapidIO の内部レジスタの初期化。設定されている特定のレジスタの詳細については、 srio_main_full.c ファイルを参照してください。
link	RapidIO リンクの状態を確認します。 リンクはアップの場合、コマンド = Link Up リンクはダウンの場合、= Link Down
r	ソフトウェア・リセット。RapidIO MegaCore ファンクションをリセットします。
rate_reconfig<desired data rate>	データ・レートのリコンフィギュレーション・プロセスを開始します。リコンフィギュレーション後、RapidIO MegaCore ファンクションは、所望のデータ・レートで動作します。

クロッキングおよびリセット

リファレンス・デザイン内のクロッキングおよびリセット・システムには、156.25 MHz のトランシーバ・クロックおよび 50 MHz の周波数のクロックが含まれています。Stratix IV GX シグナル・インテグリティ開発ボードからこれらの 2 つのクロック周波数のクロック・ソースを得ることができます。156.25 MHz のトランシーバ・クロックは、ボード上に専用の 156.25 MHz のトランシーバ・クロック (G38) に接続し、50 MHz のクロックは、専用の 50 MHz のコア・クロック (AR22) に接続します。

高速データ伝送のための RapidIO MegaCore ファンクションのトランシーバに 156.25 MHz のトランシーバ・クロックを接続してください。Avalon-MM システム・クロックは、[図 5](#) に示すように、50 MHz の周波数のクロック (clk_50) で動作します。clk_50 クロックは reconfig_clk と clk 信号に接続します。

図 5 には、RapidIO MegaCore ファンクションのダイナミック・データ・レートのリコンフィギュレーションをサポートするクロッキングおよびリセット・システムを示しています。

図 5. ダイナミック・データ・レートのリコンフィギュレーションのクロッキングおよびリセット・システム

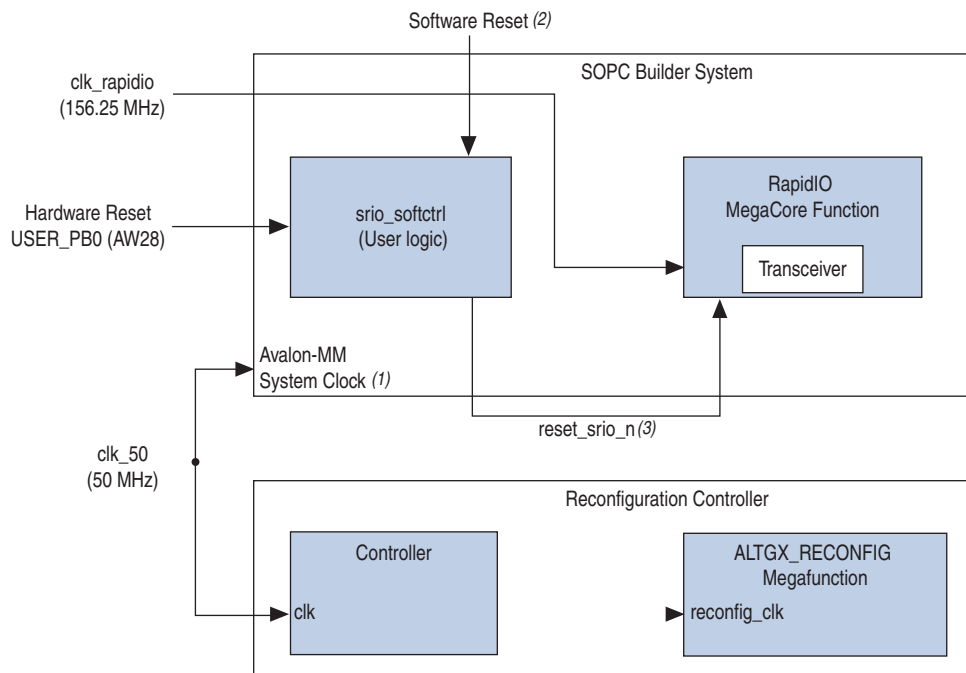


図 5 の注:

- (1) Avalon-MM システム・クロックは、SOPC Builder システム内のコンポーネントに接続されています。
- (2) ソフトウェア・リセットは、nios2-terminal からのソフトウェア・コマンドです。
- (3) これは、アクティブ Low の信号です。次のいずれかの条件は、このリセット信号をトリガします：
ハードウェア・リセット、ソフトウェア・リセット、またはリセット・リコンフィギュレーション。

リファレンス・デザインは、clk_50 にクロック入力を接続します。また、デザインのためのさまざまなソースに独立してこれらのクロックを接続することができます。

リコンフィギュレーション・プロセスが完了するには 7800 クロック・サイクルを必要とします。リコンフィギュレーション・プロセスが開始されると、RapidIO MegaCore ファンクションは、自動的にリセット状態になり、その RapidIO リンクはダウンします。リセット時には、RapidIO MegaCore ファンクションの FIFO バッファも同様にクリアされます。RapidIO MegaCore ファンクションに（送信前または受信後）、すべてのパケットが処理を保留する場合、FIFO バッファのクリアは、データ損失が発生する可能性があります。




アルテラは、データ損失を防止するために、RapidIO MegaCore ファンクションへのデータ送信を控えること、および RapidIO MegaCore ファンクションからすべての応答待ちのパケットを受信するまで待つことを推奨します。

ALTGX_RECONFIG メガファンクションが reconfig_done の信号を再アサートした後、リコンフィギュレーション・プロセスは完了します。RapidIO MegaCore ファンクションのリセット・ステート・マシンは、リンクの回復プロセスを開始します。リセット・ステート・マシンは、リンクが確立される前にリンクの PLL (Phase-Locked Loop) と周波数がロックされることが保証されます。図 10 ~ 22 ページの図 12 での波形にこれらのリコンフィギュレーションとリセットのプロセスを観察することができます。

srio_reset_n 信号はまた、これらのイベントのいずれか、以下の RapidIO MegaCore ファンクションのリセットを制御します。

- 外部 reset_n ピン付きのハードウェア・リセット
- ソフトウェア・リセット・コマンド

 これら二つのリセットは、RapidIO MegaCore ファンクションに埋め込まれているトランシーバをリコンフィギュレーションせずに RapidIO MegaCore ファンクションをリセットします。

リファレンス・デザインの使用

以下の項では、リファレンス・デザインの設定および使用方法について説明します。

- 「ハードウェア要件」
- 1-14 ページの「ソフトウェア要件」
- 1-14 ページの「リファレンス・デザインのインストール」

ハードウェア要件

リファレンス・デザインは、ダイナミック・データ・レートのリコンフィギュレーション・プロセスを示すために Stratix® IV GX EP4SGX230KF40C3ES シグナル・インテグリティ・トランシーバの開発ボードを使用しています。次のいずれかの方法を使用してリコンフィギュレーション・プロセスを確認することができます。

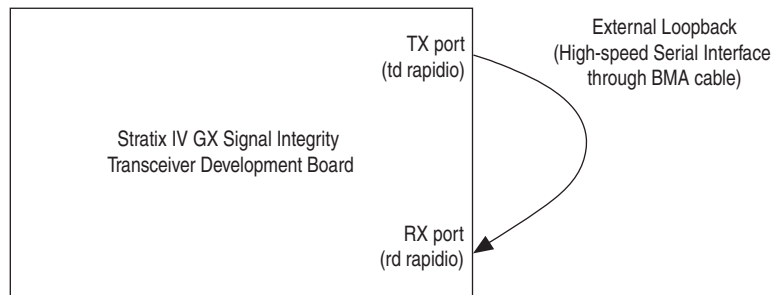
- シングル・ボードの接続
- デュアル・ボードの接続

シングル・ボードの接続

シングル・ボードの接続方法では、TX および RX ポートは、外部ループ・バックを経由して接続、および同じクロック・ソースとトランシーバ・チャンネルを共有します。ALTGX_RECONFIG メガファンクションは、ダイナミックに必要なデータ・レートでトランシーバをリコンフィギュレーションしているときに、それは TX と RX ポートのチャンネル、および PLL の設定を変更します。TX と RX ポートのデータ・レートが同時に変化します。トランシーバで、目的のデータ・レートをプログラムした後、RapidIO MegaCore ファンクションは、新しいオペレーティング・データ・レートでの RapidIO リンクを再確立します。

図 6 には、シングル・ボードの接続を示します。

図 6. シングル・ボードの接続



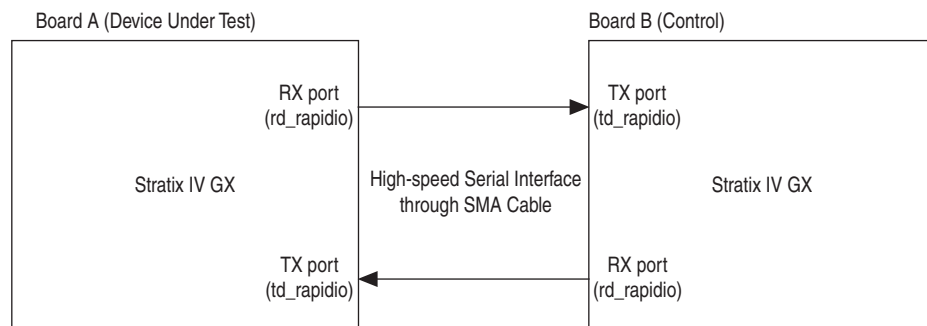
デュアル・ボードの接続

デュアル・ボードの接続方法では、デザインを説明するために、2つの Stratix IV GX EP4SGX230 シグナル・インテグリティ・トランシーバの開発ボードを使用してください。最初のボード（ボード A）はテスト・ボードであり、第二番目のボード（ボード B）は、コントロール・ボードです。

図 7 に示すようにボード A およびボード B の TX と RX ポートの接続を示します。両方のボード上のデフォルトのデータレートが 2.5 GBaud されるボード A およびボード B 上に同じデザインをプログラムしてください。ボード A およびボード B 上の RapidIO MegaCore ファンクションの初期データ・レートが同一であるとして、リンクが確立することができます。

図 7 には、デュアル・ボードの接続を示します。

図 7. デュアル・ボードの接続



JTAG ケーブルでボード B 上のリファレンス・デザインをプログラムし、ダウンロードすることができます。ボード B 上のリファレンス・デザインをプログラミングとダウンロードした後、JTAG ケーブルを外し、コントロール・ボードとしてボード B を保持します。そして、ボード A 上のリファレンス・デザインをプログラムおよびダウンロードしてください。これらの 2 つのボードの高速 RapidIO リンクとして SMA コネクタを使用します。SMA コネクタは、次のとおりです。

- TX ポート : GXB1 - TX4
 - td_rapidio(p) - AD36
 - td_rapidio(n) - AD37

- RX ポート : GXB1 - RX4
 - rd_rapidio(p) - AE38
 - rd_rapidio(n) - AE39

ソフトウェア要件

このリファレンス・デザインのアプリケーションには次のソフトウェアが必要です。

- Quartus® II ソフトウェア・バージョン 9.1
- RapidIO MegaCore バージョン 9.1
- Nios II Embedded Design Suite バージョン 9.1

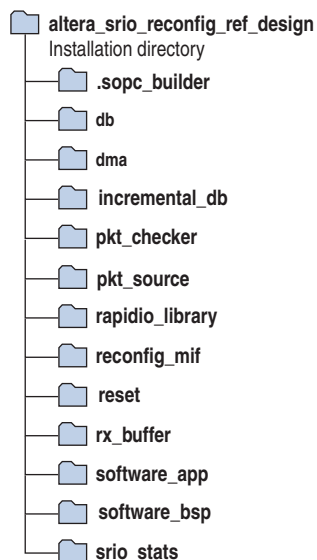
リファレンス・デザインのインストール

この項では、リファレンス・デザインのインストール方法を説明します。

altera_srio_reconfig_ref_design.zip ファイルには、このリファレンス・デザインに必要なすべてのファイルが含まれています。

このプロジェクトのために指定する作業ディレクトリに **altera_srio_reconfig_ref_design.zip** ファイルを解凍します。ファイルを解凍した後、作業ディレクトリは、[図 8](#) に示されるサブディレクトリが含まれています。

図 8. ディレクトリ構造



リファレンス・デザインの実行

リファレンス・デザインを実行するには、以下のステップに従います。

1. JTAG をデバイスに接続します。デュアル・ボードを使用する場合、JTAG ケーブルをボード B に接続します。
2. Windows Start メニューで、**All Programs > Altera > Nios II EDS <version_number>** をポイントして、NiosII コマンド・シェルを起動するには、**Nios II <version_number> Command Shell** をクリックします。
3. `/software_bsp` フォルダに移動します。
4. このリファレンス・デザインに必要な HAL (Hardware Abstraction Layer) ドライバをビルドするには、次コマンドを入力します。

```
./create-this-bsp ←
```
5. `/software_app` フォルダに移動します。このフォルダは、`srio_main_full.c`、`srio_regs.h`、および `create-this-app` ファイルを内蔵しています。
6. `srio_main_full.c` におけるドライバ・ソフトウェアをコンパイルするスクリプトを実行するには、以下のコマンドを入力します。

```
./create-this-app ←
```
7. デバイスのプログラム、ソフトウェア・イメージのダウンロード、および Nios II terminal セッションの起動を行うには、以下のステップに従います。
 - a. デバイスをプログラムするには、コマンド・シェルで以下のコマンドを入力します。

```
nios2-configure-sof -d 1 ../srio_2500_x1.sof ←
```
 - b. ソフトウェア・イメージをダウンロードするには、次のコマンドを入力します。

```
nios2-download --device=1 -g srio_test.elf ←
```
 - c. Nios II プロセッサと通信するには、次のコマンドを入力して、Nios II terminal セッションを起動します。

```
nios2-terminal --device=1 ←
```
 - d. デュアル・ボード接続を使用する場合は、ボード B からボード A に JTAG ケーブルを接続します。同じ Nios II コマンド・シェルのボード A にも、**ステップ a ~ c** を繰り返します。
8. Tools メニューで、**SignalTap II Logic Analyzer** をクリックして SignalTap™ エンベデッド・ロジック・アナライザのファイル `stp1.stp` を開きます。`stp1.stp` ファイルは、リコンフィギュレーション・プロセスを変更する内部信号のリストが含まれています。発生したリコンフィギュレーションを確認するためにこれらの信号の動作を表示します。また、**20 ページの図 10 ~ 22 ページの図 12** に、これらの信号も表示できます。
9. RapidIO MegaCore ファンクションのためのデータ・レートをリコンフィギュレーションするには、`nios2-terminal` で以下のコマンドを入力します。

```
rate_reconfig <desired data rate> ←
```



希望のデータ・レートは、1250, 2500, 3125, または 5000 である場合があります。

同じデータ・レートでの RapidIO MegaCore ファンクションをリコンフィギュレーションしようとする、Nios II プロセッサはリコンフィギュレーションし、リセット処理を実行しますが、結果のデータ・レートは同じになります。また、サポートされていないデータ・レートの RapidIO MegaCore ファンクションをリコンフィギュレーションしようとする、Nios II terminal は、例 1 に示すようにエラーが表示されます。

例 1. エラー

```
Altera RapidIO Development Platform
Serial RapidIO> rate_reconfig 4000
Command Read: rate_reconfig 4000
Dynamic rate reconfiguration
Invalid Data Rate! Supported Values : 1250 2500 3125 5000
Serial RapidIO>
```

リコンフィギュレーションの前と後の現在のリンクのステータスを確認するには、nios2-terminal で Link を入力することができます。リンクが確立されている場合は、返されるコマンドは、Link Up を示しています (例 2 を参照)。リンクがダウンした場合、返されるコマンドは、Link Down が表示されます。

例 2. 成功したリンク・アップ

```
Serial RapidIO> link
Command Read: link
Link UP!
Port Error Status, Reg 0x158 : 2
Serial RapidIO> rate_reconfig 3125
Command Read: rate_reconfig 3125
Dynamic rate reconfiguration
Serial RapidIO>
```

リコンフィギュレーションするために 1 個のみのボードを使用する場合は、TX と RX ポートは、同じデータ・レートでリコンフィギュレーションされているため、すべての場合にリンクを確立することができます。確立されたリンクを持つ二つのボード上で動作する 2 つのデザインを持っている場合、リンクは、リコンフィギュレーションされた希望のデータ・レートが他のポート上のデータ・レートと同じかどうかに応じて、リコンフィギュレーション・プロセスの後に、アップまたはダウンのいずれかになります。



ソフトウェア・リセット・コマンドを使用するには、nios-2 terminal セッションで r を入力してください。

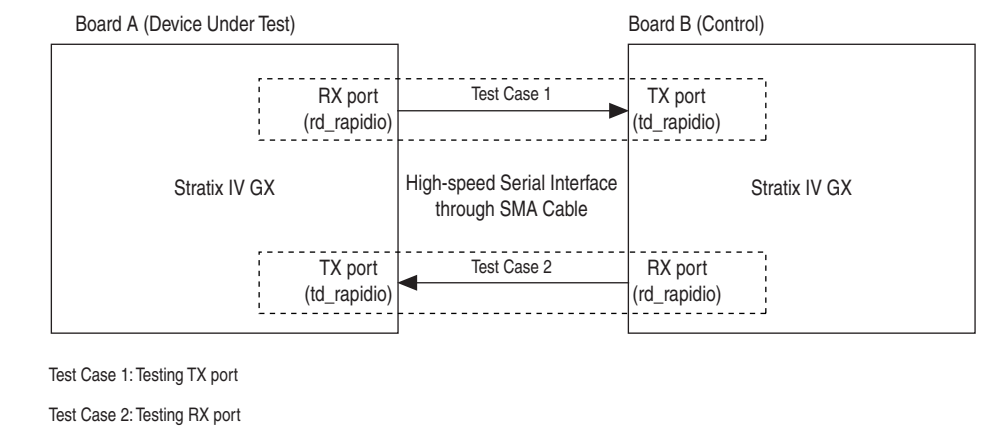
検証用のテスト・ケース（デュアル・ボードの接続）

リファレンス・デザインを実行したら、ダイナミック・データ・レートのリコンフィギュレーション・プロセスを検証するテスト・ケース 1 およびテスト・ケース 2 を実行することができます。

リンク先のデータ・レートが同一でない場合、RX ポートは、周波数をロックできません。そして、RapidIO リンクもダウンになります。しかし、リンク・パートナーのデータ・レートが同一であれば、RapidIO MegaCore ファンクションは、正常にリンクを確立します。

図 9 には、テスト・ケース 1 および 2 のデュアル・ボードの接続を示しています。

図 9. デュアル・ボードの接続用のテスト・ケース



テスト・ケース 1: TX ポートのテスト

テスト・ケース 1 の初期データ・レートは 2.500 GBaud です。このテストケースの目的は、ボード A の TX ポートをテストすることです。1.250 GBaud にボード A の初期のデータ・レートをリコンフィギュレーションした場合、ボード A と B は異なるデータ・レートで動作しているため、リンクはダウンになります。ボード A の TX ポートのデータ・レートは、希望のデータ・レートにリコンフィギュレーションされますが、ボード B の RX ポートのデータ・レートは変わりません。したがって、データ・レートは、同期不可能であり、リンクがダウンになります。

SignalTap II エンベデッド・ロジック・アナライザで port_initialized 信号でテストの結果を監視することができます。1.250 GBaud ~ 2.500 GBaud のデータ・レートにリコンフィギュレーションすると、RapidIO MegaCore ファンクションのリセット・シーケンスの監視ができます。そして、対応するリンクが再確立されることを観察できます。

表 3 には、テスト・ケース 1 のテストの状態を示します。

表 3. テスト・ケース 1 のテストの状態

ボード	ステップ	リンク (port_initialized 信号)
A および B	2.500-GBaud デザインでプログラムする	ロック (1'b1)
A	2.500 GBaud ~ 1.2500 GBaud でリコンフィギュレーションする	ロースト・ロック (1'b0)
A	1.250 GBaud ~ 2.500 GBaud でリコンフィギュレーションする	ロック (1'b1)

テスト・ケース 2: RX ポートのテスト

テスト・ケース 2 の初期のデータ・レートは 2.500 GBaud です。このテスト・ケースの目的は、RX ポートをテストすることです。ボード B の初期データ・レートを 1.250 GBaud にプログラムする場合、2.500-GBaud を持つボード A にデータ・レートのリンクを失います。ボード B が希望のデータ・レートでプログラムしますが、リコンフィギュレーションはされていないことに注意してください。1.250-GBaud データ・レートと別の .sof を準備して、その .sof をボード B にプログラムすることができます。ボード B の TX ポートは、異なるデータ・レートを持つ場合、リンクがダウンになります。

リンクを再確立するには、同じデータ・レートでボード A をリコンフィギュレーションする必要がありますが、この場合には 1.250 GBaud です。ボード A の RX ポートでの希望のデータ・レートはボード B の TX ポートでのデータ・レートと一致する場合、リンクが再確立されます。

表 4 には、テスト・ケース 2 のテストの状態を示します。

表 4. テスト・ケース 2 のテストの状態

ボード	ステップ	リンク (port_initialized 信号)
A および B	2.500-GBaud デザインでプログラムする	ロック (1'b1)
B	ボード B に 1.250 GBaud .sof をプログラムする	ロースト・ロック (1'b0)
A	2.500 GBaud ~ 1.250 GBaud でリコンフィギュレーションする	ロック (1'b1)



SignalTap II エンベデッド・ロジック・アナライザについて詳しくは、「*Quartus II Handbook Volume 3*」の「*Design Debugging Using the SignalTap II Embedded Logic Analyzer*」の章を参照してください。

波形例

この項では、ダイナミック・データ・レートのリコンフィギュレーション・プロセスのための SignalTap II 信号と波形を提供します。

表 5 には、図 10 ~ 22 ページの図 12 に示されている信号について説明します。

表 5. SignalTap II エンベデッド・ロジック・アナライザで表示される選択信号 (その 1)

信号	説明
port_initialized	シリアル RapidIO の初期化シーケンスが正常に完了したことを示します。リンクが確立されています。
reset_state_machine	リセット後のトランシーバ PLL および周波数のロック状態を監視します。RapidIO MegaCore ファンクションは、トランシーバ PLL と周波数にロックすると、RapidIO リンクが確立されています。
softctrl reset_srio_n(1)	次のいずれかの条件によってトリガされるときにをリセットします。 <ul style="list-style-type: none"> ■ ソフトウェア・リセット ■ ハードウェア・リセット ■ リコンフィギュレーション・プロセス

表 5. SignalTap II エンベデッド・ロジック・アナライザで表示される選択信号 (その 2)

信号	説明
softctrl sys_mnt_s_.....(1)	SOPC Builder システム (Nios II プロセッサ) からの srio_softctrl ユーザー・ロジックでの Avalon-MM 信号です。
reconfig reconfig_ena(2)	ダイナミック・データ・レートのリコンフィギュレーション・プロセスをイネーブルします。また rate_reconfig[1:0] 信号が有効になることを示します。srio_softctrl ユーザー・ロジックからの出力、またリコンフィギュレーション・コントローラへの入力。
reconfig rate_reconfig(2)	現在のリコンフィギュレーション・データ・レートを示す内部信号。srio_softctrl ユーザー・ロジックが reconfig_ena 信号をアサートすると、rate_reconfig 値はリコンフィギュレーションする現在のデータ・レート、および値が有効であることを示します。次の値は、現在のデータ・レートを示しています。 <ul style="list-style-type: none"> ■ 0h: 5.000 GBaud ■ 1h: 3.125 GBaud ■ 2h: 2.500 GBaud ■ 3h: 1.250 GBaud
reconfig reconfig_done(2)	リコンフィギュレーション・プロセスが完了したことを示します。それらのデフォルト値にすべての状態と出力信号を返します。
controller(3)	コントローラ内の .mif からの。MIF からの信号データ、アドレス、および信号です。これらの信号は、ALTGX_RECONFIG メガファンクションへの入力です。

表 5 の注:

- (1) srio_softctrl ユーザー・ロジック (SOPC Builder システムの Avalon パス) への Nios II プロセッサからの信号。
- (2) コントローラに srio_softctrl ユーザー・ロジック (SOPC Builder システム) からのコンジジット信号。
- (3) ALTGX_RECONFIG メガファンクションへのコントローラからの信号。

図 10 ~ 22 ページの図 12 は、ダイナミック・データ・レートのリコンフィギュレーション・プロセスのための SignalTap II の波形を示します。

図 10 には、ダイナミック・データ・レートのリコンフィギュレーション・プロセスの開始時に SignalTap II ロジック・アナライザの波形の例を示します。SignalTap ファイル内のトリガ・ポイントとして reconfig_ena 信号を設定することにより、この波形を得ることができます。

図 10. ダイナミック・データ・レート・リコンフィギュレーション・プロセスの開始時 (注 1)

Name	-512	0	512	1024	1536	2048	2560
port_initialized							
⊕ reset_state_machine		st done				st wait for gxbpll lock	
⊕ softctrl reset_srio_n		1h				0h	
⊕ softctrl sys_mnt_s_address							
⊕ softctrl sys_mnt_s_chipselect		0h				0h	
⊕ softctrl sys_mnt_s_waitrequest		0h				1h	
⊕ softctrl sys_mnt_s_write		0h				0h	
⊕ softctrl sys_mnt_s_writedata							
⊕ reconfig reconfig_ena		0h				1h	
⊕ reconfig rate_reconfig		0h				3h	
⊕ reconfig reconfig_done						0h	
⊕ controller busy	0h	1h	1h	1h	1h	1h	1h
⊕ controller reconfig_address	00h	02h	03h	04h	05h	06h	07h
⊕ controller reconfig_data	0000h	0C30h	C000h	3000h	0000h	6054h	2200h
⊕ controller write_pulse	0h	0h	0h	0h	0h	0h	0h

図 10 の注:

- (1) レジスタ・リセットのために、18 ページの表 5 で述べられるように、5.0 GBaud を表わす代わりに、rate_reconfig 信号のための初期値は 0h です。reconfig_ena 信号の値は、処理用のコントローラに送信する rate_reconfig 信号をイネーブルする reconfig_ena 信号 (1'b1) がアサートされたときにのみ有効です。

リコンフィギュレーション・プロセスを開始する場合は、nios2-terminal でのソフトウェア・コマンド rate_reconfig<desired data rate> を入力します。ソフトウェア・コマンドを入力すると、reconfig_ena 信号をアサートするために、Nios II エンベデッド・プロセッサは srio_softctrl ユーザー・ロジックをプロンプトして表示されます。rate_reconfig 信号は 1.250 GBaud (3h) にリコンフィギュレーションされます。コントローラが rate_reconfig と reconfig_ena 信号を受信した後、それは ALTGX_RECONFIG メガファンクションへの書き込みパルスのシリーズを開始します。コントローラで生成されたすべての書き込みパルスで、reconfig_data と reconfig_address 信号は、ALTGX_RECONFIG メガファンクションでの選択された .mif ファイルの内容をロードして、トランシーバをリコンフィギュレーションします。

リコンフィギュレーション・プロセスが開始されると、srio_softctrl のユーザー・ロジックは、RapidIO MegaCore ファンクションをリセットするリセット信号を送信します。RapidIO リンクがダウンし、port_initialized 信号をデアサートします。リコンフィギュレーション・プロセスが完了するまでリセットが保持します。

図 11 は、ダイナミック・データ・レートのリコンフィギュレーション・プロセスが完了した後に SignalTap II ロジック・アナライザの波形の例を示します。SignalTap ファイル内のトリガ・ポイントとして reconfig_done 信号を設定することにより、この波形を得ることができます。

図 11. ダイナミック・データ・レートのリコンフィギュレーション・プロセスが完了した後

Name	-512	0	512	1024	1536	2048	2560
port_initialized							
reset_state_machine						st_gxbsppll_locked	
softctrl reset_srio_n	0h					1h	
softctrl sys_mnt_s_address							
softctrl sys_mnt_s_chipselect					0h		
softctrl sys_mnt_s_waitrequest	1h					0h	
softctrl sys_mnt_s_write					0h		
softctrl sys_mnt_s_writedata							
reconfig reconfig_ena	1h					0h	
reconfig rate_reconfig					1h		
reconfig reconfig_done	0h					1h	
controller busy	1h	1h				0h	
controller reconfig_address		36h				00h	
controller reconfig_data		0400h				F800h	
controller write_pulse	0h					0h	

ALTGX_RECONFIG メガファンクションは、コントローラから選択された .mif ファイルの内容を受信すると、reconfig_togxb 信号を介しての RapidIO MegaCore ファンクションのトランシーバをリコンフィギュレーションします。ALTGX_RECONFIG メガファンクションは、リコンフィギュレーション・プロセスを完了するには .mif ファイルにすべての内容を書き込みます。ALTGX_RECONFIG メガファンクションが完全にトランシーバをリコンフィギュレーションした後、リコンフィギュレーション・プロセスが完了したことを示すために reconfig_done 信号をアサートします。

srio_softctrl ユーザー・ロジックのステート・マシンは、reconfig_ena 信号をデアサートします。そして、srio_softctrl ユーザー・ロジックとリコンフィギュレーション・コントローラの両方はアイドル状態に戻ります。

リコンフィギュレーション・プロセスが完了すると、reset_srio_n 信号は、RapidIO MegaCore ファンクションのリセットをリリースします。RapidIO MegaCore ファンクションは、リンクを再確立するために、トランシーバ PLL と周波数にロックしようとします。

図 12 に、リンクが再確立された SignalTap II ロジック・アナライザの波形の例を示します。SignalTap ファイル内のトリガ・ポイントとして port_initialized 信号を設定することにより、この波形を得ることができます。


図 12. リンクが再確立される場合

Name	-512	0	512	1024	1536	20
port_initialized						
reset_state_machine						
softctrl reset_srio_n						
softctrl sys_mnt_s_address						
softctrl sys_mnt_s_chipselect						
softctrl sys_mnt_s_waitrequest						
softctrl sys_mnt_s_write						
softctrl sys_mnt_s_writedata						
reconfig reconfig_ena						
reconfig rate_reconfig						
reconfig reconfig_done						
controller busy						
controller reconfig_address						
controller reconfig_data						
controller write_pulse						

データ・レートは、TX と RX ポートで同一である場合、RapidIO MegaCore ファンクションは、リンクを再確立するために、トランシーバ PLL と周波数にロックされます。RapidIO MegaCore ファンクションは、リンクが確立して準備されていることを示すため port_initialized 信号をアサートします。このとき、RapidIO MegaCore ファンクションは、希望のデータ・レートでデータを送受信します。

リコンフィギュレーションのイネーブル

この項では、ご使用のデザインにリコンフィギュレーション・プロセスを実装する際には、必要に応じて追加の情報を提供します。

 ファレンス・デザインには、以下の手順に従って最終的な結果を含む SRAM オブジェクト・ファイル (.sof) を提供します。


RapidIO MegaCore ファンクションのトランシーバをリコンフィギュレーションするには、以下の手順を実行します。


1. SOPC Builder システムでの RapidIO MegaCore ファンクションをインスタンス化します。
2. SOPC Builder システムを生成します
3. RapidIO の階層の下に、ALTGX パラメータ・エディタを起動するには **rapidio_riophy_gxb** をクリックします。
4. ALTGX_RECONFIG メガファンクションをイネーブルして、トランシーバでチャネルおよび CMU PLL をリコンフィギュレーションするために、ALTGX_RECONFIG パラメータ・インタフェースの **Reconfiguration settings** タブの **Enable Channel and Transmitter PLL reconfiguration** をオンにします。
5. **Finish** をクリックします。トランシーバのファイルが再生成されます。

ALTGX_RECONFIG メガファンクションをインスタンス化してチャンネルおよび CMU PLL のリコンフィギュレーション・モードを選択するには、以下の手順を実行します。

1. Tools メニューの **MegaWizard Plug-In Manager** をクリックします。
2. **Next** をクリックします。
3. メガファンクション・リストでは、**I/O category** の横の「+」アイコンをクリックして、**ALTGX_RECONFIG** を選択します。 **Next** をクリックします。
4. ALTGX_RECONFIG パラメータ・エディタの **Reconfiguration settings** タブでは、**Channel and TX PLL select/reconfig** をオンにします。

reconfig_togxb および reconfig_fromgxb 信号を介しての RapidIO MegaCore ファンクションのトランシーバと ALTGX_RECONFIG メガファンクションのデータ交換を行います。他の信号は、選択された **.mif** ファイルからリコンフィギュレーション・データを取得するためにコントローラに接続します。

 ALTGX_RECONFIG について詳しくは、「*Stratix IV デバイス・ハンドブック Volume 2*」の「*Stratix IV ダイナミック・リコンフィギュレーション*」の章を参照してください。

 デザインが異なるトランシーバ設定が必要な場合（たとえば、デザインはリファレンス・デザインで指定された周波数の異なるクロック周波数を使用している場合）、アルテラは、4つのデータ・レートのために希望のトランシーバ設定で4つの **.mif** ファイルを再生成することを推奨します。 **reconfig_mif** フォルダ内の **.mif** ファイルにこの RapidIO MegaCore ファンクション用のトランシーバ設定にのみ含まれているため、**.mif** ファイルを再生成する必要があります。

 複数の **.mif** ファイルをインスタンス化する方法の詳細については、「*Stratix IV デバイス・ハンドブック Volume 2*」の「*Stratix IV ダイナミック・リコンフィギュレーション*」の章を参照してください。

改訂履歴

表 6 に、このアプリケーション・ノートの改訂履歴を示します。

表 6. 改訂履歴

日付	バージョン	変更内容
2010 年 12 月	1.0	初版。

