



リモート・アップデート回路 (ALTREMOTE_UPDATE) メガファンクション・ ユーザーガイド



101 Innovation Drive
San Jose, CA 95134
www.altera.com

ソフトウェア・バージョン : 9.0
ドキュメント・バージョン : 2.4
ドキュメント・デート : 2009年4月

Copyright © 2009 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

第1章 このメガファンクションについて

サポートされるデバイス・ファミリ	1-1
はじめに	1-1
特長	1-1
概要	1-2
一般的なアプリケーション	1-2
機能説明	1-2
サポートされるデバイスの最高クロック周波数 (f_{MAX})	1-5
リモート・システム・コンフィギュレーション・モード	1-5
リモート・コンフィギュレーション・モード	1-5
ローカル・コンフィギュレーション・モード	1-7
リモート・システム・コンフィギュレーション・コンポーネント	1-8
ページ・モード機能	1-9
ファクトリ・コンフィギュレーション	1-9
アプリケーション・コンフィギュレーション	1-10
ウォッチドッグ・タイマ	1-10
リモート・アップデート・サブブロック	1-11
リモート・コンフィギュレーション・レジスタ	1-11
リソース使用率およびパフォーマンス	1-11

第2章 使用法

ソフトウェアおよびシステム要求	2-1
MegaWizard Plug-In Manager のカスタマイズ	2-1
MegaWizard Plug-In Manager ページのオプションと説明	2-1
HDL コードまたは回路図デザインにおけるメガファンクションのインスタンス化	2-5
EDA ツールの使用によるネットリストの生成	2-6
ポートおよびパラメータ定義の使用	2-6
コンパイル後のメガファンクションの識別	2-6
シミュレーション	2-7
Quartus II のシミュレーション	2-7
EDA シミュレーション	2-7
SignalTap II エンベデッド・ロジック・アナライザ	2-7
デザイン例 1: パラメータの書き込みとリコンフィギュレーション	2-8
デザイン・ファイル	2-8
ALTREMOTE_UPDATE メガファンクション・バリエーションの生成	2-8
デザインの実装	2-12
機能の結果 —Quartus II 開発ソフトウェアによるデザインのシミュレーション	2-14
機能の結果 —ModelSim-Altera ソフトウェアによるデザインのシミュレーション	2-17
デザイン例 2: パラメータの読み出し	2-18
デザイン・ファイル	2-18
ALTREMOTE_UPDATE メガファンクション・バリエーションの生成	2-18
デザインの実装	2-22
機能の結果 —Quartus II 開発ソフトウェアによるデザインのシミュレーション	2-24
機能の結果 —ModelSim-Altera ソフトウェアによるデザインのシミュレーション	2-27
結論	2-28

第3章 仕様

ポートおよびパラメータ	3-1
-------------	-----

追加情報

改訂履歴	Info-1
参考資料	Info-1

サポートされるデバイス・ファミリ

ALTREMOTE_UPDATE メガファンクションでは、以下のアルテラのデバイス・ファミリに対してサポートを提供しています。

- Arria GX
- Arria®II GX
- Cyclone® III
- Stratix
- Stratix II
- Stratix GX
- Stratix II GX
- Stratix III
- Stratix® IV

はじめに

デザインの複雑化に伴って、ベンダ固有の IP (Intellectual Property) ブロックの使用が一般的なデザイン手法となりました。アルテラは、アルテラのデバイス・アーキテクチャ用に最適化された、パラメータ化可能なメガファンクションを提供しています。独自のロジック作成の代わりにメガファンクションを使用することで、デザインの作業時間を有効活用することができます。Quartus II™ メガファンクションは、より効率的なロジックの合成およびデバイスの実装を提供します。

特長

ALTREMOTE_UPDATE メガファンクションは、リモート・アップデート・ファンクションを実装し、以下の追加機能を提供します。

- ページ・モード
- ファクトリ・コンフィギュレーション
- アプリケーション・コンフィギュレーション
- ウォッチドッグ・タイマ
- リモート・アップデート・サブブロック
- リモート・コンフィギュレーション・レジスタ

概要

ALTREMOTE_UPDATE メガファンクションにより、サポートされるデバイス用に提供されている専用リモート・システム・アップグレード回路の利点をデザインに活用できます。

リモート・システム・アップグレードは、システム設計者が製品回収を行わずに機能強化やバグ修正を行うことができ、製品の市場投入の短縮や製品寿命の延長に役立ちます。ALTREMOTE_UPDATE メガファンクションおよび専用回路を使用することにより、遠隔地から新しいコンフィギュレーション・イメージをダウンロードし、それをコンフィギュレーション・メモリに格納し、更に専用リモート・システム・アップグレード回路にリコンフィギュレーション・サイクルの開始を指示することができます。

専用回路は、コンフィギュレーション・プロセス中およびコンフィギュレーション・プロセス後に、エラー検出を実行します。エラーが検出された場合、回路は、安全なコンフィギュレーション・イメージまで戻り、エラー・ステータス情報を提供することにより、システム・リカバリを容易にします。

リモート・システム・アップグレードでは、以下のコンフィギュレーション方法がサポートされています。

- ファースト・パッシブ・パラレル (FPP)
- パッシブ・シリアル (PS)
- パッシブ・パラレル非同期 (PPA)
- アクティブ・シリアル (AS)
- アクティブ・パラレル (AP)

リモート・システム・アップグレードは、コンフィギュレーション・データのリアルタイム復元、および安全で効率的なフィールド・アップグレード用の AES (Advanced Encryption Standard) を使用したデザイン・セキュリティを組み込むことも可能です。

一般的なアプリケーション

今後の製品との互換性を保証するための追加のプロトコル・サポートが必要なとき、または問題を解決する必要があるときなど、自動コンフィギュレーション・アップデートを必要とするアプリケーションでこの機能を使用します。

機能説明

専用リモート・システム・アップグレード回路は、リモート・コンフィギュレーションを管理し、エラー検出、回復、およびステータス情報を提供します。ユーザー・ロジックまたは Nios® II プロセッサは、リモート・コンフィギュレーション・データ・ソースへのアクセスおよびシステムのコンフィギュレーション・メモリへのインタフェースを提供します。

リモート・システム・アップグレード・プロセスでは、以下のステップを実行します。

1. ユーザー・モードで動作する場合、デバイスは、ネットワーク経由でリモート・ソースから、または他のデータ転送ソースから、新規データまたはアップデートされたデータを受信します。Nios II™ (32 ビット ISA) エンベデッド・プロセッサを Arria GX、Stratix シリーズ、Cyclone III デバイス内に実装するか、あるいは外部プロセッサを実装して、リモート・ソースからメモリ・デバイスへのコンフィギュレーション・ファイルの読み出し／書き込み機能を制御できます。

2. デバイスは、新規データまたはアップデートされたデータをメモリ・デバイスに格納します。メモリ・デバイスには、エンハンスド・コンフィギュレーション・デバイス、シリアル・コンフィギュレーション・デバイス (Arria GX、Arria II GX、Stratix IV、Stratix III、Stratix II、Stratix II GX、および Cyclone III デバイス専用)、業界標準フラッシュ・メモリ・デバイス、または別のストレージ・デバイスがあります (図 1-1)。これは、FPGA デバイスがユーザー・モードの場合も同様です。
3. FPGA デバイスが、必要となるコンフィギュレーション・パラメータを `ALTREMOTE_UPDATE` メガファンクションに書き込んだ後、デザインは、新規またはアップデートされたコンフィギュレーション・データでリコンフィギュレーション・サイクルを開始します。
4. デバイスは、`FPGA_MSEL[2..0]` または `FPGA_MSEL[3..0]` ピンを使用して選択されたコンフィギュレーション手法により、メモリからの新しいデータでアップデートされます。3 ピンまたは 4 ピンの選び方は、選択するターゲット・デバイスによって決定されます。
5. 専用リモート・システム・アップグレード回路は、リコンフィギュレーション・サイクル中またはリコンフィギュレーション・サイクル後に発生する可能性のあるエラーの検出およびエラー状態からの回復を実行し、ユーザー・デザインにエラー・ステータス情報を提供します。


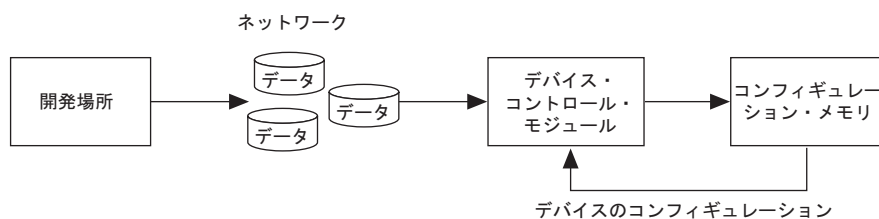
 リモート・アップデート回路は、エラーを検出し、デフォルト・ファクトリ・コンフィギュレーションを再ロードすることによってエラーから回復し、エラー・ステータス情報を提供します。

図 1-1 は、リモート・システム・コンフィギュレーションの例を示します。

図 1-1. リモート・システム・コンフィギュレーション



Arria GX、Stratix シリーズ、および Cyclone III デバイスは、表 1-1 に示されたコンフィギュレーション手法で、リモート・システム・アップグレードをサポートしています。

表 1-1. リモート・システム・アップグレード・コンフィギュレーション・サポート

デバイス・ファミリ	リモート・システム・コンフィギュレーション手法				
	FPP (1)、(2)	AS (3)	PS (1)、(2)	PPA (1)	AP
Arria II GX	✓	✓	✓	—	—
Arria GX	✓	✓	✓	✓	—
Stratix IV	✓	✓	✓	—	—
Stratix III	✓	✓	✓	—	—
Stratix II	✓	✓	✓	✓	—
Stratix II GX	✓	✓	✓	✓	—
Stratix	✓	—	✓	✓	—
Stratix GX	✓	—	✓	✓	—
Cyclone III	—	✓	—	—	✓

表 1-1 の注：

- (1) MAX[®] II デバイスとフラッシュ・メモリ（またはマイクロプロセッサとフラッシュ・メモリ）は、FPP、PS、または PPA 手法を使用して Stratix シリーズ・デバイスをコンフィギュレーションします。
- (2) エンハnst・コンフィギュレーション・デバイスは、FPP または PS コンフィギュレーション手法を使用して、Stratix シリーズ・デバイスをコンフィギュレーションします。
- (3) シリアル・コンフィギュレーション・デバイスは、AS 方式を使用して、Arria GX、Arria II GX、Stratix IV、Stratix III、Stratix II、Stratix II GX、および Cyclone III FPGA をコンフィギュレーションします。


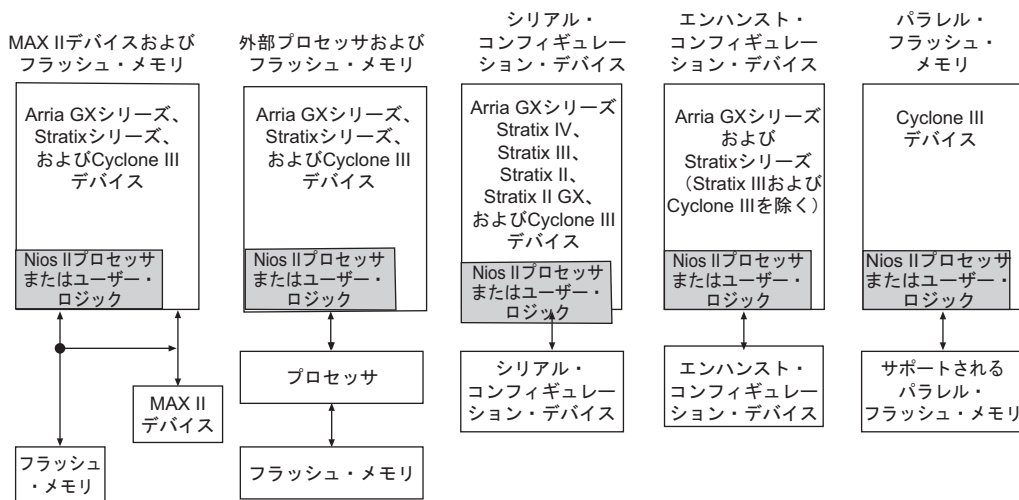

 デバイスのコンフィギュレーション・モードが JTAG に設定されていると、リモート・システム・アップグレードはサポートされません。

図 1-2 に、Arria GX、Stratix IV、Stratix シリーズ、および Cyclone III デバイスのさまざまなコンフィギュレーション手法を使用して行うリモート・システム・アップグレード実装のブロック図を示します。

図 1-2. リモート・システム・アップグレード・ブロック図



 コンフィギュレーション手法について詳しくは、関連デバイス・ハンドブックの「Configuration and Remote System Upgrades」セクションを参照してください。

サポートされるデバイスの最高クロック周波数 (f_{MAX})

f_{MAX} は、内部セットアップ・タイム (t_{SU}) およびホールド・タイム (t_H) 要求値に違反しないで達成できる最高クロック周波数です。制限なし f_{MAX} は、デザインが実行できる最高周波数です。制限付き f_{MAX} は、最大トグル・レートのようなデバイス制限を考慮したときにデザインが実行できる最高周波数です。

表 1-2 に、サポートされるデバイスのリモート・システム・コンフィギュレーションの制限付き f_{MAX} 値を示します。

表 1-2. サポートされるデバイスのリモート・アップデート用の制限付き f_{MAX}

デバイス	f_{MAX}
Arria GX	20 MHz
Arria II GX	10 MHz
Cyclone III	40 MHz
Stratix	20 MHz
Stratix II	20 MHz
Stratix III	10 MHz
Stratix IV	10 MHz

リモート・システム・コンフィギュレーション・モード


リモート・システム・アップグレードには、以下の 2 つの動作モードがあります。


- 「リモート・コンフィギュレーション・モード」
- 「ローカル・コンフィギュレーション・モード」

リモートおよびローカル・コンフィギュレーション・モードは、さまざまな機能を提供し、電源投入時にシステムの機能を決定することができます。

リモート・コンフィギュレーション・モード

エンハンスド・コンフィギュレーション・デバイスを使用する Arria GX および Stratix シリーズ・デバイス (Stratix III および Cyclone デバイスを除く) に対して最高 7 つの異なるアプリケーション・コンフィギュレーションと 1 つのファクトリ・コンフィギュレーションを管理するためには、リモート・コンフィギュレーション・モードを使用します。デバイスの PGM[] ピンで選択できるページ数により、コンフィギュレーション・ファイルは 7 つまでに制限されます。

 システムが、ネットワーク経由でリモート・ソースから、または他のデータ・ソースから 8 つ以上のコンフィギュレーション・ファイルを受信した場合、前のコンフィギュレーション・ファイルは上書きされます。

 リモート・アップグレードのサポートされるエンハンスド・コンフィギュレーション・デバイスについて詳しくは、関連デバイス・ハンドブックの「Configuration and Remote System Upgrades」セクションを参照してください。

リモート・アップデート・モードをシリアル・コンフィギュレーション・デバイスとともに使用すると、コンフィギュレーション・スペースは、任意のフラッシュ・セクタ境界から始まります。これにより、EPCS64 デバイスで最大 128 ページ、および EPCS16 デバイスで最大 32 ページが可能になります。各ページの最小サイズは 512 Kbit です。更に、アプリケーション・コンフィギュレーションには、機能エラーの検出が可能なユーザー・ウォッチドッグ・タイマの機能があります。

Cyclone III デバイスは、アルテラのデバイス用の AP コンフィギュレーション手法をサポートします。AP コンフィギュレーション手法では、Cyclone III デバイスは汎用 16 ビット・パラレル・フラッシュ・メモリを使用してコンフィギュレーションされます。フラッシュ・メモリにより、デバイスは 16 ビット幅のパラレル・データ・バス上でデータを検索できるため、コンフィギュレーション・データへのアクセスに高速インタフェースを実現できます。

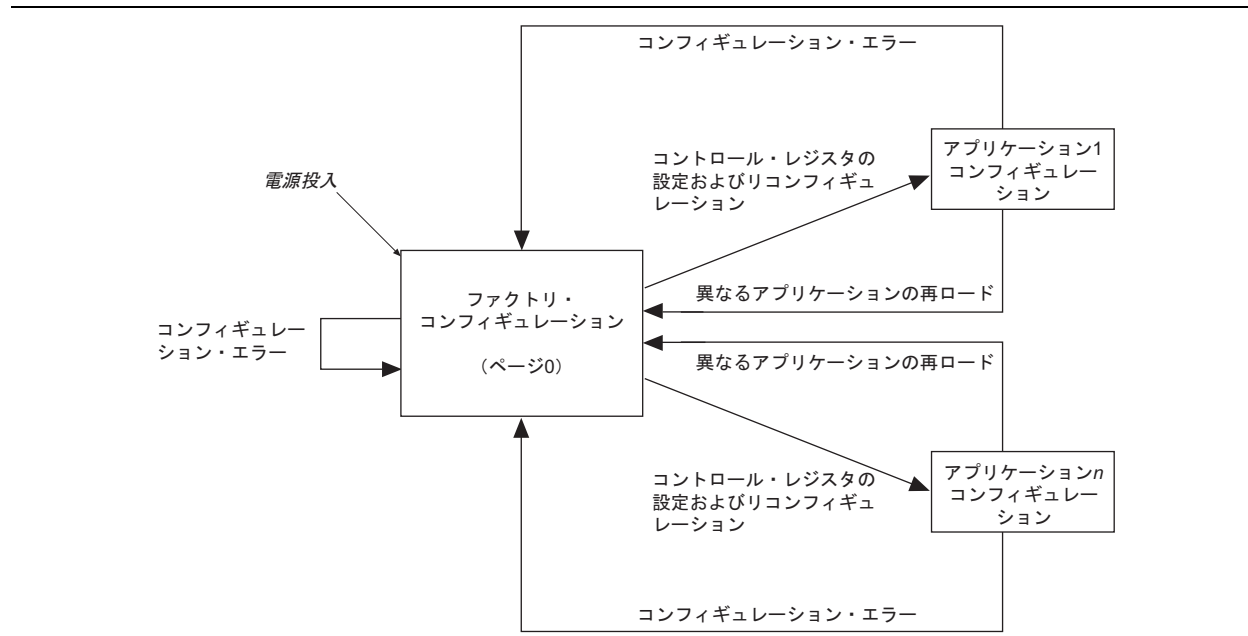


詳しくは、「Cyclone III デバイス・ハンドブック」の「Cyclone III デバイスのコンフィギュレーション」の章、および「Cyclone III デバイスのリモート・システム・アップグレード」の章を参照してください。

Arria GX、Stratix シリーズ、および Cyclone III デバイスは、PS、FPP、PPA コンフィギュレーション手法のリモート・コンフィギュレーション・モードをサポートしています。Cyclone III デバイスのみが、AP コンフィギュレーション手法をサポートしており、Arria GX、Arria II GX、Stratix IV、Stratix III、Stratix II、Stratix II GX、および Cyclone III デバイスのみが、AS コンフィギュレーション手法をサポートしています。リモート・コンフィギュレーション・モードでの電源投入時に、デバイスは、ユーザー指定のファクトリ・コンフィギュレーション・ファイルをロードします。このファイルは、エンハンスド・コンフィギュレーション・デバイスのデフォルトのページ・アドレス 000 に配置されています。デバイスがリモート・アップデート専用回路によってコンフィギュレーションされると、リモート・コンフィギュレーション・コントロール・レジスタは、デバイスにロードされる必要のある、アプリケーション・コンフィギュレーションのページ・アドレスを指し示します。アプリケーション・コンフィギュレーションのユーザー・モード時にエラーが発生した場合、デバイスはデフォルトのファクトリ・コンフィギュレーションを再ロードします。

1-7 ページの図 1-3 に、リモート・コンフィギュレーション・モードのブロック図を示します。

図 1-3. リモート・コンフィギュレーション・モード



ローカル・コンフィギュレーション・モード

ローカル・コンフィギュレーション・モードは、リモート・コンフィギュレーション・モードを簡略化したものです。このモードは、Arria GX、Stratix II、Stratix II GX、Stratix、および Stratix GX デバイスでのみサポートされており、Arria II GX、Stratix III、Stratix IV、および Cyclone III デバイスではサポートされていません。電源投入時、直ちにアプリケーションをロードするシステムには、ローカル・コンフィギュレーション・モードを使用します。ローカル・コンフィギュレーション・モードでは、1つのコンフィギュレーションしか使用できません。このコンフィギュレーションは、リモートまたはローカルでアップデートできます。

電源投入時、または nCONFIG のアサーション時に、デバイスは、アプリケーション・コンフィギュレーションをロードします。ファクトリ・コンフィギュレーション・モードは、アプリケーション・コンフィギュレーションのユーザー・モード時にエラーが発生した場合にのみロードされます。エンハンスド・コンフィギュレーション・デバイスを使用する場合、ページ・アドレス 001 は、アプリケーション・コンフィギュレーション・データ用の位置で、ページ・アドレス 000 は、ファクトリ・コンフィギュレーション・データ用の位置です。

ページ・アドレス 001 のコンフィギュレーション・データが巡回冗長検査 (CRC) の失敗により正常にロードできない場合、エンハンスド・コンフィギュレーション・デバイスがタイムアウトになった場合、あるいは外部プロセッサがタイムアウトになった場合は、デフォルト ページ (ページ・アドレス 000) にあるファクトリ・コンフィギュレーションがデバイスにロードされます。このモードでは、ユーザー・ウォッチドッグ・タイマ機能はサポートされていません。リモート・システム・アップグレード・コントロール・レジスタのライト・アクセスがディセーブルされます。ただし、このデバイスはエラー・ステータス情報を取得するためのリード・アクセスをサポートしています。


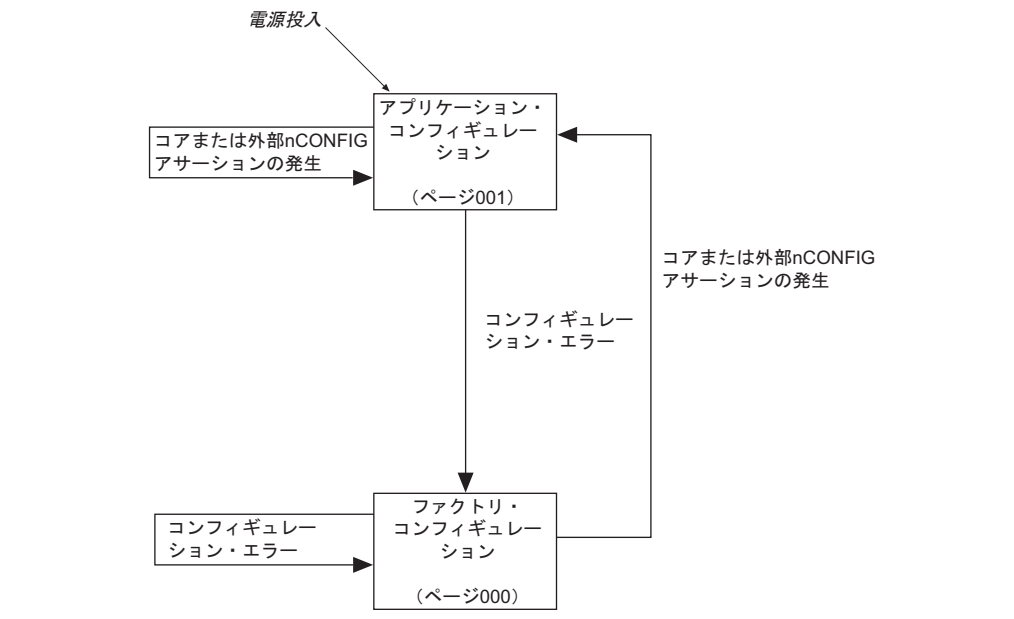
 AS コンフィギュレーション手法 (シリアル・コンフィギュレーション・デバイスを使用) では、ローカル・アップデート・モードはサポートされません。

図 1-4 に、ローカル・コンフィギュレーション・モードのコンフィギュレーション間の遷移を示します。

図 1-4. ローカル・コンフィギュレーション・モード




リモート・システム・コンフィギュレーション・コンポーネント

以下のコンポーネントは、リモートおよびローカル・コンフィギュレーション・モードをサポートするために、すべてのサポートされるデバイスにおいて使用可能です。

- ページ・モード機能
- ファクトリ・コンフィギュレーション
- アプリケーション・コンフィギュレーション
- ウォッチドッグ・タイマ
- リモート・アップデート・サブブロック
- リモート・コンフィギュレーション・レジスタ

次のセクションでは、各コンポーネントについて詳しく説明します。ただし、すべてのデバイス固有機能の全詳細が提供されるわけではありません。

 デバイス固有の機能について詳しくは、関連デバイス・ハンドブックの「Configuration and Remote System Upgrades」セクションを参照してください。

ページ・モード機能

ページ・モード機能では、デザインがコンフィギュレーション・データにアクセスする位置を選択できます。エンハンスド・コンフィギュレーション・デバイスは、8つまでの異なるコンフィギュレーション・ファイル（1つのファクトリ・ファイルと7つのアプリケーション・ファイル）を格納できます。ページ選択は、デバイス上の PGM[2..0] ピンで実行されます。FPGA 上において、これらのピンは、標準コンフィギュレーション・モード時にユーザー I/O ピンとして指定される場合がありますが、リモート・システム・コンフィギュレーション・モードの場合は、専用出力ピンとして使用されます。

リモート・コンフィギュレーション・モードで電源投入時、ファクトリ・コンフィギュレーションは、Stratix シリーズ・デバイス PGM[2..0] 出力ピンによって、ユーザー指定ページ・アドレスを選択します。これらのピンは、エンハンスド・コンフィギュレーション・デバイスの PGM[2..0] 入力ピンをドライブし、メモリ内の要求されたページを選択します。

エンハンスド・コンフィギュレーション・デバイスの代わりにインテリジェント・ホストが使用される場合、エンハンスド・コンフィギュレーション・デバイスの設定に類似したページ・モード設定をサポートするロジックをインテリジェント・ホストに作成する必要があります。



リモート・システム・アップグレード用の AS コンフィギュレーションは、Arria GX、Stratix II、および Stratix II GX デバイスでのみ使用できることに注意してください。EPCS デバイスが AS インタフェースに pgm ピンの入力を持たないため、remote_update ブロックの pgmout ピンを読み出すことができないため、このコンフィギュレーション手法に pgmout ピンはありません。代わりに、ASMI コントローラが、このコンフィギュレーション・プロセスを処理します。

AS コンフィギュレーションでリモート・システム・アップグレードを実装する際、Stratix II FPGA 内の専用 7 ビット・ページ開始アドレス・レジスタにより、シリアル・コンフィギュレーション・デバイス内のコンフィギュレーション・ページの開始アドレスが決定されます。PGM[6..0] レジスタは、24 ビット開始アドレスのうちビット [22..16] の部分を構成します。その他の 17 ビットは、 $StAdd[23..0] = \{1'b0, PGM[6..0], 16'b0\}$ のようにゼロに設定されます。

Stratix III デバイスでは、専用の 24 ビット開始アドレス・レジスタ PGM[23..0] が開始アドレスを保持しています。AS および AP の両コンフィギュレーションの場合、Cyclone III デバイスは、24 ビット・ブート開始アドレスを使用します。このアドレスには上位 22 ビットを設定します。Cyclone III デバイスは、pgmout ポートをサポートしていません。

AS コンフィギュレーション時に、Arria GX、Arria II GX、Stratix IV、Stratix III、Stratix II、Stratix II GX、または Cyclone III デバイスは、シリアル・コンフィギュレーション・デバイスからコンフィギュレーション・データを取得するために、この 24 ビット・ページ開始アドレスを使用します。

ファクトリ・コンフィギュレーション

ファクトリ・コンフィギュレーションはデフォルトのコンフィギュレーション・データ・セットアップです。エンハンスド・コンフィギュレーション・デバイスでは、デフォルトのページ・アドレスは 000 です。ファクトリ・コンフィギュレーション・データは、システム・メーカーによって一度だけメモリ・デバイスに書き込まれるもので、リモートのアップデートや変更を行ってはなりません。リモート・コンフィギュ

レーション・モードの場合、ファクトリ・コンフィギュレーションは電源投入時に Arria GX、Stratix シリーズ、および Cyclone III デバイスにロードされます。同様に、シリアル・コンフィギュレーション・デバイスを使用する AS リモート・コンフィギュレーション手法の場合、ファクトリ・コンフィギュレーションは電源投入時にシリアル・コンフィギュレーション・デバイスにロードされます。

システムがアプリケーション・コンフィギュレーション・データのロード中にエラーを検出した場合、または nCONFIG アサーションによりデバイスがリコンフィギュレーションした場合、デバイスはファクトリ・コンフィギュレーションをロードします。リモート・システム・コンフィギュレーション・レジスタは、ファクトリ・リコンフィギュレーションの理由を判断します。この情報に基づいて、ファクトリ・コンフィギュレーションは、どのアプリケーション・コンフィギュレーションをロードすべきか決定します。

Cyclone III デバイスの場合、AP コンフィギュレーション手法のリモート・アップデートでは、電源投入時に、Cyclone III デバイスが、以下のアドレスに配置されたデフォルトのファクトリ・コンフィギュレーションをロードします。

`boot_address[23:0] = 24'h010000 = 24'b1 0000 0000 0000 0000`。デフォルトのファクトリ・コンフィギュレーション・アドレスは、JTAG 命令 `APFC_BOOT_ADDR` を使用して任意のアドレスに変更することができます。ファクトリ・コンフィギュレーション・イメージは不揮発性メモリに保存され、リモート・アクセスを使用してアップデートまたは変更することはできません。これは、サポートされているパラレル・フラッシュ・メモリでのデフォルト開始アドレス位置 `0x010000` (デフォルト・アドレスが変更されている場合はアップデートされたアドレス) に対応します。`0x010000` は、AP フラッシュ・メモリ用の 16 ビット・ワードのアドレスであることに注意してください。ただし、Quartus II 開発ソフトウェアは、8 ビット・バイト・アドレッシングを実装しています。したがって、このアドレス用の正しい Quartus II 開発ソフトウェア設定は `0x020000` です。

 詳しくは、「Cyclone III デバイス・ハンドブック」の「Cyclone III デバイスのコンフィギュレーション」の章、および「Cyclone III デバイスのリモート・システム・アップグレード」の章を参照してください。

 Stratix III デバイスのファクトリ・コンフィギュレーションについて詳しくは、「Stratix III デバイス・ハンドブック」の「Configuring Stratix III Devices」の章、および「Stratix III デバイスのリモート・システム・アップグレード」の章を参照してください。


アプリケーション・コンフィギュレーション

アプリケーション・コンフィギュレーションは、リモート・ソースから受信されたコンフィギュレーション・データで、メモリ・ストレージ・デバイスのさまざまなロケーションまたはページ (ファクトリ・デフォルト・ページは除く) に格納されます。

ウォッチドッグ・タイマ

ウォッチドッグ・タイマは、別のメカニズム機能が適切に動作しているかどうかを判断する回路です。ウォッチドッグ・タイマは、アプリケーションが適切に実行している間は、リセット状態のままの時間遅延リレーのように動作します。Arria GX、Stratix シリーズ、および Cyclone III デバイスは、誤ったアプリケーション・コンフィギュレーションによってデバイスが停止したままになるのを防止するために、リモート・システム・コンフィギュレーション用にビルトイン・ウォッチドッグ・タイマを備えています。タイマは 29 ビット・カウンタですが、ウォッチドッグ・タイマの値を設

定するには上位 12 ビットのみを使用します。タイマは、デバイスがユーザー・モードに入るとカウントを開始します。アプリケーション・コンフィギュレーションがカウント・アップ前にユーザー・ウォッチドッグ・タイマをリセットしない場合、専用回路は、デバイスをファクトリ・コンフィギュレーションでリコンフィギュレーションし、ユーザー・ウォッチドッグ・タイマをリセットします。

 ユーザー・ウォッチドッグ・タイマはローカル・コンフィギュレーションではディセーブルされます。


リモート・アップデート・サブブロック


リモート・アップデート・サブブロックは、リモート・コンフィギュレーション機能を管理します。リモート・コンフィギュレーション・ステートマシンによって制御されるこのサブブロックは、さまざまなリモート・コンフィギュレーション・レジスタを制御するのに必要な制御信号を生成します。

リモート・コンフィギュレーション・レジスタ

リモート・コンフィギュレーション・レジスタは、ページ・アドレスおよびコンフィギュレーション・エラーの原因を常に把握しています。アップデート・レジスタとシフト・レジスタの両方を制御できます。ステータス・レジスタとコントロール・レジスタは内部ロジックによって制御されますが、読み出しはシフト・レジスタによって行われます。Stratix III デバイスでは、最新コントロール・レジスタは 38 ビット・レジスタですが、Stratix II デバイスでは、21 ビット・レジスタです。

Cyclone III デバイスの場合、リモート・システム・アップグレード・ステータス・レジスタには追加機能があります。3 セットのレジスタは、現在のアプリケーション・コンフィギュレーションとその前の 2 つのアプリケーション・コンフィギュレーションのステータスを格納します。

 Cyclone III デバイスの機能について詳しくは、「Cyclone III デバイス・ハンドブック」の「Cyclone III デバイスのリモート・システム・アップグレード」の章を参照してください。

 コンフィギュレーション・レジスタについて詳しくは、関連デバイス・ハンドブックの「Configuration and Remote System Upgrades」セクションを参照してください。

リソース使用率およびパフォーマンス

特定のデバイスのリソース使用率およびパフォーマンスについて詳しくは、MegaWizard™ Plug-In Manager およびデバイスのコンパイル・レポートを参照してください。

ソフトウェアおよびシステム要求

このセクションの説明では、以下のソフトウェアが要求されます。

- オペレーティング・システム・サポート情報は、以下を参照してください。
www.altera.com/support/software/os_support/oss-index.html
- Quartus® II 開発ソフトウェア

MegaWizard Plug-In Manager のカスタマイズ

MegaWizard™ Plug-In Manager は、カスタム・メガファンクション・バリエーションを含むデザイン・ファイルを作成または変更します。このバリエーションは、デザイン・ファイル内でインスタンス化できます。MegaWizard Plug-In Manager は、デザイン内で、ALTREMOTE_UPDATE メガファンクション機能を指定することが可能な GUI を提供します。

MegaWizard Plug-In Manager は、以下のいずれかの方法で起動します。

- Tools メニューの **MegaWizard Plug-In Manager** をクリックします。
- Block Editor を実行しているときは、Edit メニューの **Insert Symbol as Block** をクリックするか、または Block Editor を右クリックして、**Insert** をポイントし、**Symbol as Block** をクリックします。**Symbol** ダイアログ・ボックスで **MegaWizard Plug-In Manager** をクリックします。
- コマンド・プロンプトで以下のコマンドを入力して、MegaWizard Plug-In Manager のスタンドアロン・バージョンを起動します。

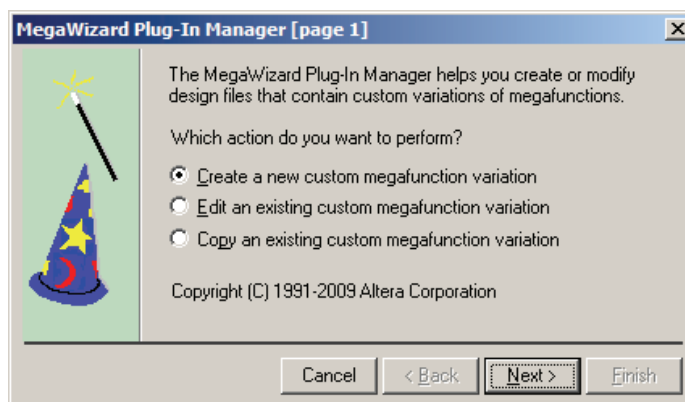
```
qmegawiz ←
```

MegaWizard Plug-In Manager ページのオプションと説明

このセクションでは、ALTREMOTE_UPDATE ウィザードの各ページで使用できるオプションについて説明します。

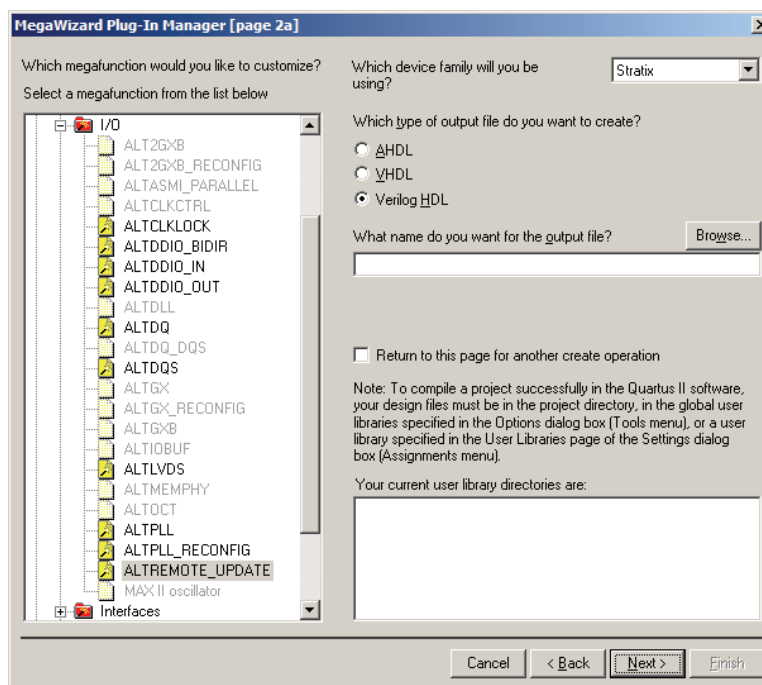
MegaWizard Plug-In Manager の 1 ページ目では、カスタム・メガファンクション・バリエーションの作成、編集、またはコピーを実行できます (図 2-1)。

図 2-1. メガファンクション・バリエーションの作成、編集またはコピー



ALTREMOTE_UPDATE MegaWizard Plug-In Manager の 2a ページ目では、使用するデバイス・ファミリ、作成する出力ファイルのタイプ、および出力ファイルの名前を指定します。出力ファイルのタイプには、AHDL (.tdf)、VHDL (.vhd)、または Verilog HDL (.v) を選択します (図 2-2)。

図 2-2. MegaWizard Plug-In Manager [ページ 2a]



ALTREMOTE_UPDATE MegaWizard Plug-In Manager の 3 ページ目では、動作モードのタイプ、リモート・アップデートの初期化設定、リコンフィギュレーション時にロードされるページ、およびウォッチドッグ・タイマを使用するかどうかを指定して、タイマ値を入力します。リコンフィギュレーションを開始する条件を指定します (図 2-3)。



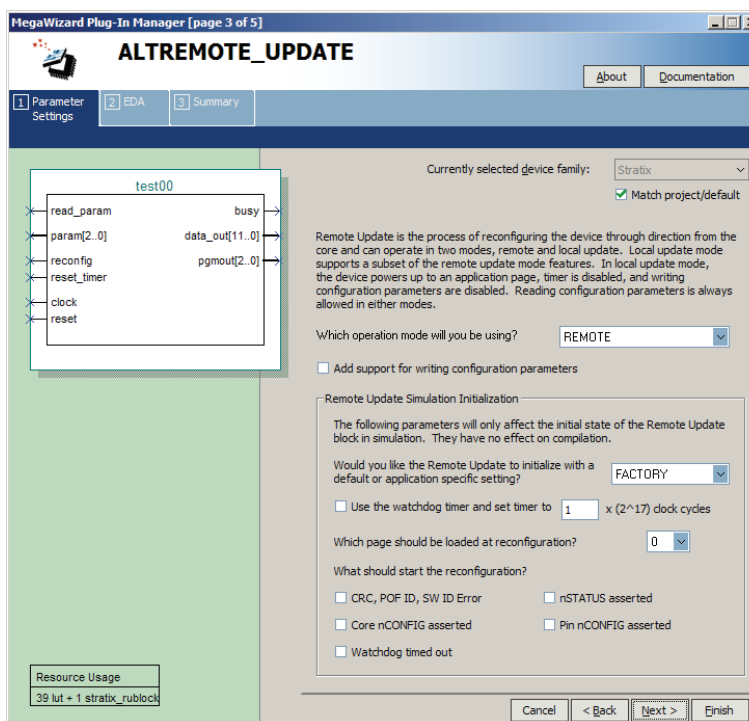


-  現在選択されているデバイス・ファミリが Arria® GX, Stratix® II, あるいは Stratix II GX デバイス・ファミリである場合、**Which operation mode will you be using?** プルダウン・リストで、**ACTIVE_SERIAL_REMOTE** を選択します。この選択を実行すると、Arria GX, Stratix II, または Stratix II GX デバイスのリモート・アップデート・ブロックにおいて、アクティブ・シリアル (AS) コンフィギュレーション手法を使用できます。これらのデバイス・ファミリのいずれかが選択されている場合は、生成されたブロックに pgmout 出力ピンがないので注意してください。
-  現在選択されているデバイス・ファミリが Stratix III または Cyclone® III デバイス・ファミリである場合、動作モードの選択がディセーブルされます。Stratix III デバイスは AS リモート・オプションのみをサポートし、Cyclone III デバイスは AS およびアクティブ・パラレル (AP) リモート・オプションのみをサポートします。ローカル・アップデート・モードは、Stratix III および Cyclone III デバイスでは使用できません。

図 2-3. MegaWizard Plug-In Manager—ALIREMOTE_UPDATE [ページ 3/5] : Parameter Settings

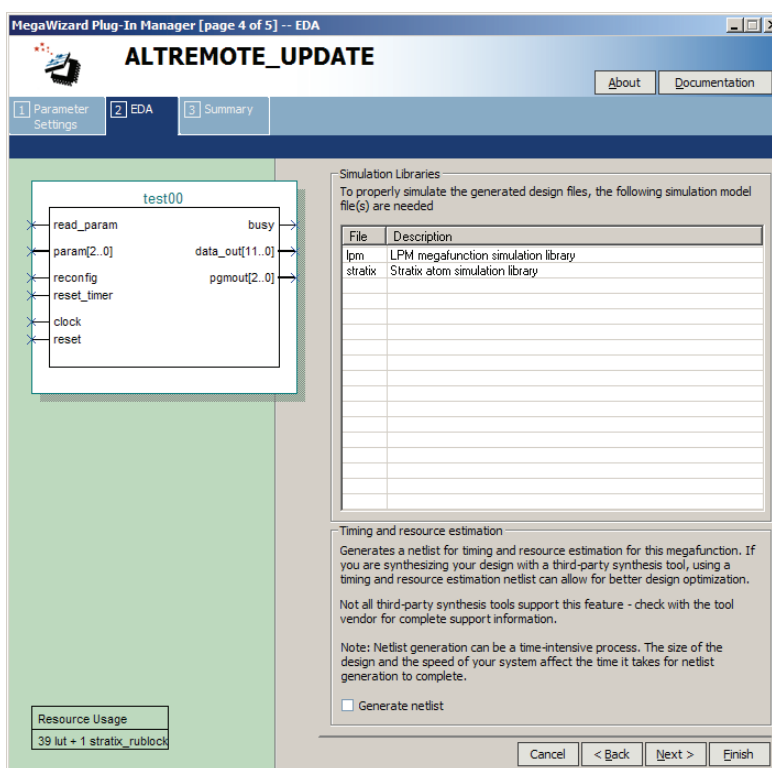


-  動作モードが **LOCAL** に設定されている場合、コンフィギュレーション・パラメータの書き込みはサポートされません。シミュレーション目的の場合、(初期化設定のような) オプションに対してウォッチドッグ・タイマ/タイムアウト設定を使用します。リコンフィギュレーション中にロードされるページを指定する設定がディセーブルされます。ただし、特定のリコンフィギュレーション・トリガ・オプションは使用できます。これらのシミュレーション・オプションは、Stratix III および Cyclone III デバイスでは使用できません。

 nSTATUS 信号、および nCONFIG ピンがアサートされた場合、あるいはウォッチドッグがタイムアウトになった場合、使用できるリコンフィギュレーション・トリガ・オプションは、巡回冗長検査 (CRC)、プログラマ・オブジェクト・ファイル ID (POF ID)、またはサンプリング・ウィンドウ ID (SW ID) によるエラー信号です。コンフィギュレーション・オプションは、イネーブルされると、機能シミュレーション中にトリガされます。しかし、これらのリコンフィギュレーション・トリガ・オプションは、Stratix III および Cyclone III デバイスでは使用できません。

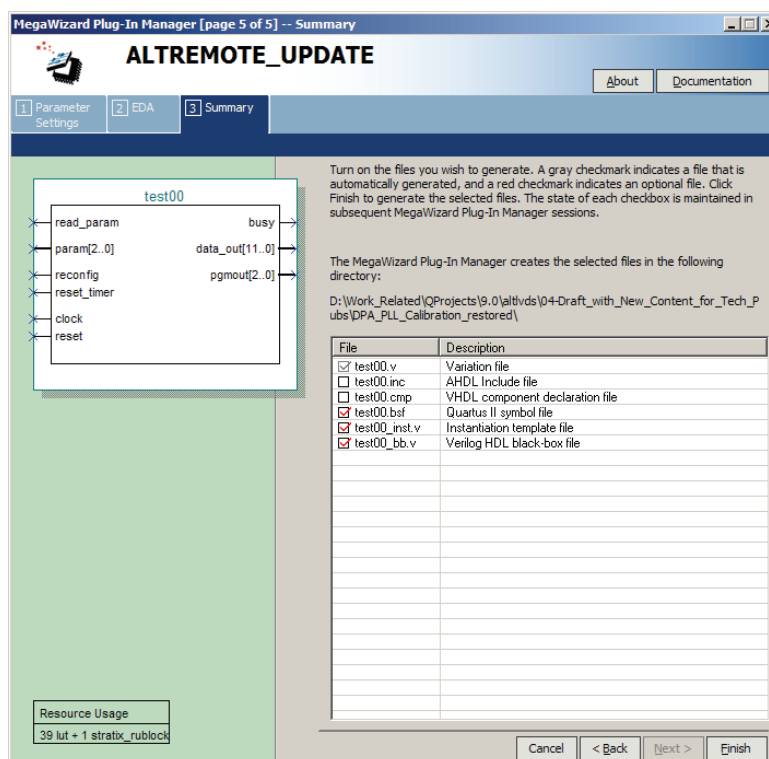
ALTREMOTE_UPDATE ウィザードの 4 ページ目には、生成されたデザイン・ファイルのシミュレーションに必要なシミュレーション・モデル・ファイルがリストされます (図 2-4)。サードパーティ・ツールで使用するための合成領域およびタイミング見積りネットリストを Quartus II 開発ソフトウェアで生成するようにできます。サードパーティ・ツールで使用されるこのメガファンクションのシミュレーション・モデルは、Stratix III または Cyclone III デバイスでは使用できないので注意してください。

図 2-4. MegaWizard Plug-In Manager—ALTREMOTE_UPDATE [ページ 4/5] – EDA



ALTREMOTE_UPDATE ウィザード (図 2-5) の 5 ページ目では、カスタム・メガファンクション用に生成するファイルを指定します。グレーのチェック・マークは、ファイルが常に生成されることを示します。その他のファイルは、オプションであり、選択された場合にのみ生成されます (赤のチェック・マークで表示)。

図 2-5. MegaWizard Plug-In Manager—AL TREMOTE_UPDATE [ページ 5/5] – Summary



AL TREMOTE_UPDATE メガファンクションのポートについては、第 3 章、仕様を参照してください。


HDL コードまたは回路図デザインにおけるメガファンクションのインスタンス化

MegaWizard Plug-In Manager を使用してメガファンクションをカスタマイズしパラメータ化すると、MegaWizard Plug-In Manager により出力ファイルのセットが作成されます。これにより、カスタマイズされた機能をデザイン内にインスタンス化できます。MegaWizard Plug-In Manager は、正しいパラメータ値でメガファンクションをインスタンス化し、MegaWizard Plug-In Manager で選択した言語に応じて、Verilog-HDL フォーマット (.v)、VHDL フォーマット (.vhd)、または AHDL フォーマット (.tdf) のいずれかを使用したメガファンクション・バリエーション・ファイル (ラッパー・ファイル) を、他のサポートするファイルとともに生成します。

MegaWizard Plug-In Manager は、以下のファイルを作成するオプションを提供します。

- バリエーション・ファイル (`_inst.v`、`_inst.vhd`、または `_inst.tdf`) の言語用のインスタンス化例のテンプレート
- .vhd ファイルで使用できるコンポーネント宣言ファイル (.cmp)
- .tdf で使用できる AHDL インクルード・ファイル (.inc)
- 回路図デザインで使用できる Quartus II のブロック・シンボル・ファイル (.bsf)


- サードパーティ合成ツール内にメガファンクションをブラック・ボックスとしてインスタンス化する際に使用できる Verilog HDL モジュール宣言ファイル (`_bb.v`)

 ウィザードで生成されたファイルについて詳しくは、Quartus II Help または「Quartus II ハンドブック Volume 1」の「推奨される HDL コーディング構文」の章を参照してください。

EDA ツールの使用によるネットリストの生成


サードパーティ EDA 合成ツールを使用すると、メガファンクション・バリエーション・ファイルを合成用のブラック・ボックスとしてインスタンス化できます。VHDL コンポーネント宣言または Verilog モジュール宣言のブラック・ボックス・ファイルを使用して、合成ツール機能を定義し、次にメガファンクション・バリエーション・ファイルを Quartus II プロジェクトに含めます。

MegaWizard Plug-In Manager の合成領域およびタイミング見積りネットリストを生成するためのオプションをイネーブルすると、ウィザードは追加ネットリスト・ファイル (`_syn.v`) を生成します。ネットリスト・ファイルは、Quartus II 開発ソフトウェアで使用される、カスタマイズされたロジックを表すものです。このファイルは、メガファンクションにおけるアーキテクチャ・エレメントの接続情報を提供しますが、真の機能を表していない場合があります。この情報により、特定のサードパーティ合成ツールは、レポート領域およびタイミング見積りを改善できます。更に、合成ツールは、タイミング情報を使用して、タイミング・ドリブンを最適化に専念し、結果の品質を改善できます。

 サードパーティ合成ツールにおけるメガファンクションの使用について詳しくは、「Quartus II ハンドブック Volume 1」の「Synthesis」セクションの該当する章を参照してください。

ポートおよびパラメータ定義の使用

MegaWizard Plug-In Manager の代わりに、メガファンクションを呼び出して、そのパラメータを他のモジュール、コンポーネントまたはサブデザインの設定の場合と同じように設定することにより、メガファンクションを Verilog HDL、VHDL、または AHDL コードでインスタンス化できます。

 アルテラは、複雑なメガファンクションに MegaWizard Plug-In Manager を使用することを推奨しています。MegaWizard Plug-In Manager により、すべてのメガファンクション・パラメータを適切に設定することができます。

メガファンクション・ポートおよびパラメータのリストについては、第3章、仕様を参照してください。

コンパイル後のメガファンクションの識別

Quartus II 開発ソフトウェアでのコンパイル中に、解析とエラーレクションが実行され、デザインの構造が構築されます。Project Navigator ウィンドウで、コンパイル階層を展開し、メガファンクションを名前前で検索すると、メガファンクションを見つけることができます。

同様に、メガファンクション内のノード名を（Node Finder を使用して）検索するには、**Look in** ダイアログ・ボックスで **Browse (…)** をクリックし、**Hierarchy** ダイアログ・ボックスでメガファンクションを選択します。


シミュレーション

Quartus II シミュレーション・ツールは、シミュレーションを実行するための使いやすしい統合ソリューションを提供します。シミュレーション・オプションについては、以下のセクションで説明しています。

Quartus II のシミュレーション

Quartus II シミュレータでは、機能とタイミングのシミュレーションを実行できます。Quartus II プログラムの機能シミュレーションでは、FPGA のタイミング遅延を考慮することなく、デザインの論理動作を検証することができます。このシミュレーションは、RTL コードのみを使用して実行されます。機能シミュレーションを実行するときは、合成前の段階で存在する信号を表示できます。これらの信号は、レジスタ（Node Finder 内の pre-synthesis フィルタ、Design Entry フィルタ、または Pin フィルタ）で検索できます。メガファンクションのトップ・レベル・ポートは、これら 3 つのフィルタを使用して検索されます。

これに対して、Quartus II 開発ソフトウェアのタイミング・シミュレーションでは、アノテートされたタイミング情報を使用してデザインの動作を検証します。このシミュレーションは、配置配線後のネットリストを使用して実行されます。タイミング・シミュレーションを実行するときは、配置配線後に存在する信号を表示できます。これらの信号は、Node Finder の Post-Compilation フィルタを使用して検索されます。合成および配置配線中、RTL 信号の名前が変更されます。したがって、Post-Compilation フィルタを使用して、メガファンクションのインスタンスから信号を検出するのは困難です。ただし、合成および配置配線段階で信号名を保持する場合は、keep または preserve の合成属性を使用する必要があります。これらの Verilog および VHDL 合成属性は、解析および合成エンジンに、特定のワイヤ、レジスタ、またはノードの状態を維持するよう指示します。これらの合成属性を使用して、組み合わせロジック・ノードを維持すると、シミュレーション中にノードを観察することができます。

 詳しくは、「Quartus II ハンドブック Volume 1」の「Quartus II インテグレートッド・シンセシス」の章を参照してください。


EDA シミュレーション

使用しているシミュレーション・ツールに応じて、「Quartus II ハンドブック Volume 3」の「Simulation」セクションの該当する章を参照してください。これらのツール固有の章では、メガファンクションを含む機能およびゲート・レベルのタイミング・シミュレーションの実行方法や、必要なファイルとそれらのファイルが配置されているディレクトリが示されています。

SignalTap II エンベデッド・ロジック・アナライザ

SignalTap®II エンベデッド・ロジック・アナライザは、デザイン内のすべてのアルテラ・メガファンクションをデバッグする方法を提供します。SignalTap II エンベデッド・ロジック・アナライザを使用すると、システムをフル・スピードで実行しながら、デザイン内のアルテラ・メガファンクションのトップ・レベル・ポートのサンプルデータをキャプチャおよび解析することができます。

アルテラ・メガファンクションからの信号をモニタするには、最初に Quartus II 開発ソフトウェアで SignalTap II エンベデッド・ロジック・アナライザをコンフィギュレーションし、次にアナライザを Quartus II プロジェクトの一部として含める必要があります。次に、Quartus II 開発ソフトウェアは、アナライザをデザインとともに、選択したデバイスにシームレスに埋め込みます。

 SignalTap II エンベデッド・ロジック・アナライザの使用方法について詳しくは、「Quartus II ハンドブック Volume 3」の「SignalTap II エンベデッド・ロジック・アナライザを使用したデザインのデバッグ」の章を参照してください。

デザイン例 1：パラメータの書き込みとリコンフィギュレーション

このデザイン例は、ウォッチドッグ・タイマのタイムアウト値を目的のカウントに設定して有効にするための制御信号のシーケンスを示します。また、アプリケーション・ページを選択し、その選択されたページを使用してデバイスのリコンフィギュレーションを開始するための reconfig 信号をアサートするシーケンスも示します。

デザイン・ファイル

デザイン・ファイルは、アルテラのウェブサイトの「[オンライン資料](#)」のページにあります。ファイルは、以下のセクションの下にあります。

- 「[オンライン資料：Quartus II 開発ソフトウェア](#)」ページで、「[メガファンクション](#)」のセクションを展開して、次に「[I/O](#)」セクションを展開します。
- アルテラのウェブサイトの「[オンライン資料：ユーザーガイド](#)」セクション

この例では、次の作業を実行する必要があります。

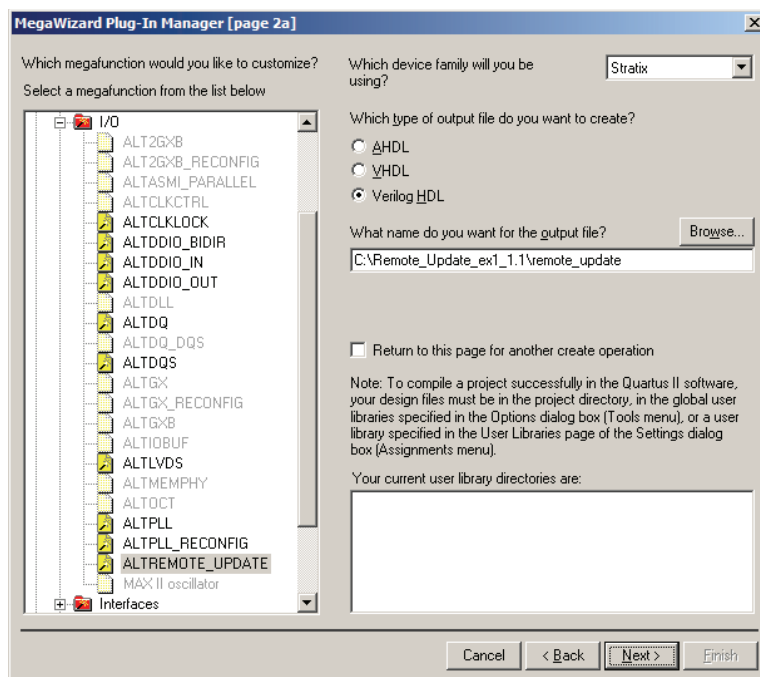
- ALTREMOTE_UPDATE メガファンクションおよび MegaWizard Plug-in Manager を使用して、リモート・アップデート・ブロックをインスタンス化します。
- デザインを実装して、EP1S10B672C6 デバイスをプロジェクトに割り当てます。
- デザインをコンパイルしてシミュレーションします。

ALTREMOTE_UPDATE メガファンクション・バリエーションの生成

この例で作成された新規の ALTREMOTE_UPDATE メガファンクションは、Quartus II プロジェクトのトップ・レベル・ファイルに追加されます。

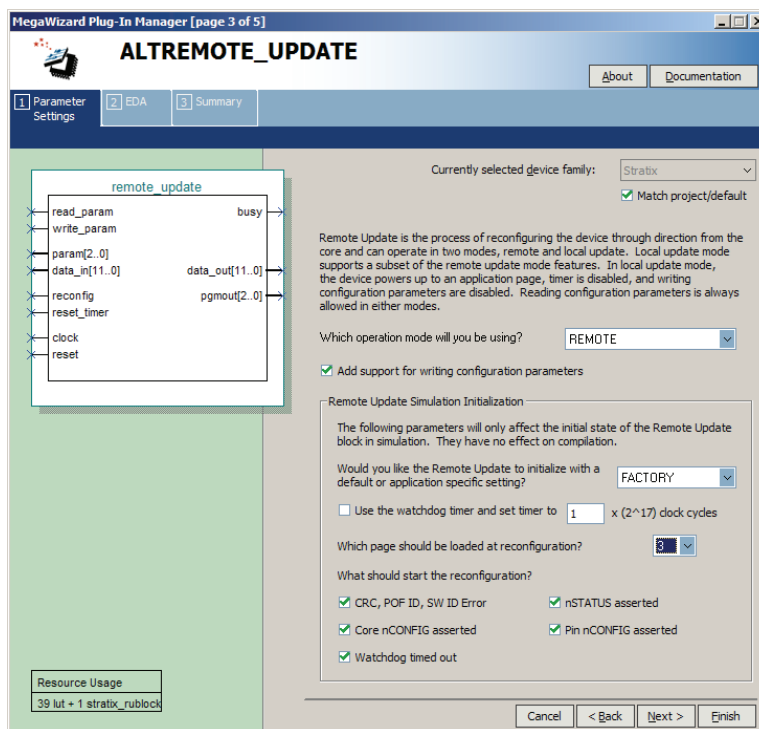
1. ALTREMOTE_UPDATE_DesignExample_ex1.zip の内容をコンピュータ上で使用している任意のハード・ディスクに解凍します。
2. Quartus II 開発ソフトウェアで、Remote_Update_ex1_1.1.qar プロジェクトを開きます。
3. Tools メニューの MegaWizard Plug-In Manager を選択します。
4. Create a new custom megafunction variation を選択し、Next をクリックします。MegaWizard Plug-In Manager ページが表示されます (図 2-6)。

図 2-6. MegaWizard Plug-In Manager [ページ 2a]



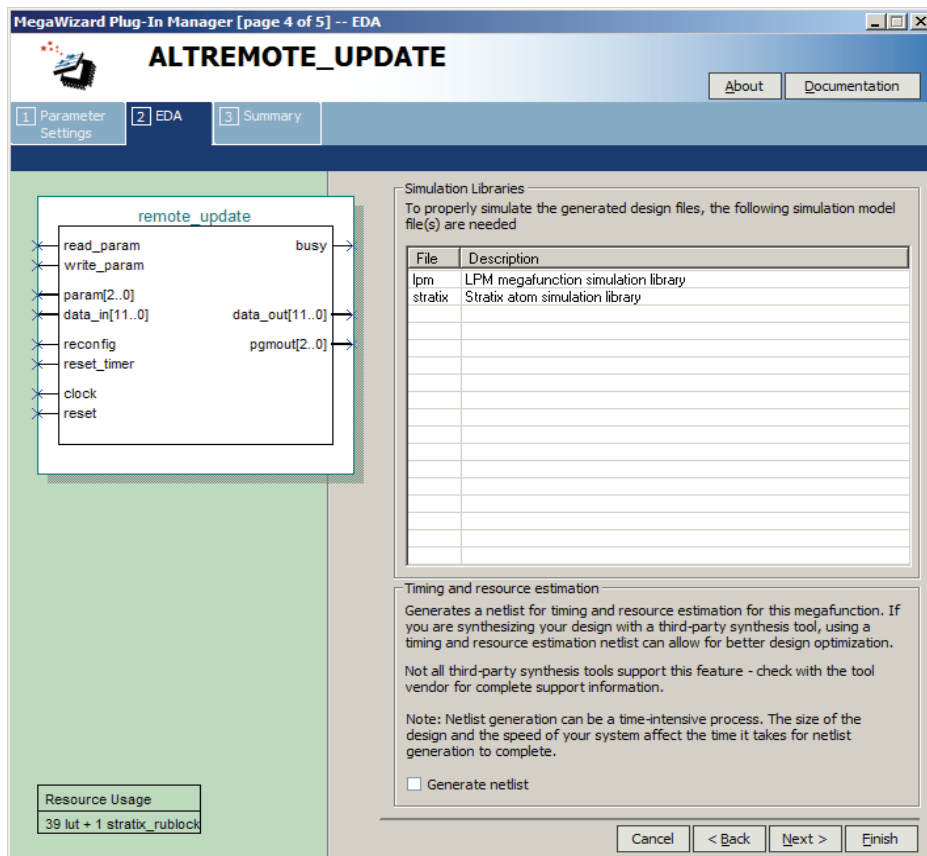
5. **Which device family will you be using?** プルダウン・リストから **Stratix** を選択します。
6. **Which type of output file do you want to create?** オプションで、**Verilog HDL** を選択します。
7. **Installed Plug-Ins** の下の **I/O** フォルダから **ALTREMOTE_UPDATE** を選択します。
8. 出力ファイルに名前 **remote_update** を付けます。図 2-6 に、上記の選択を行った後のウィザードを示します。
9. **Next** をクリックします。ページ 3 が表示されます (図 2-7)。

図 2-7. MegaWizard Plug-In Manager—AL TREMOTE_UPDATE [ページ 3/5]



10. **Which operation mode will you be using?** プルダウン・リストから、**REMOTE** を選択します。
11. **Add support for writing configuration parameters** のチェック・ボックスにチェックを付けます。
12. **Would you like Remote Update to initiate with a default or application specific settings?** プルダウン・リストから、**FACTORY** を選択します。
13. **Use the Watchdog timer and set timer to** チェック・ボックスのチェックを外します。
14. **Which page should be loaded at reconfiguration?** プルダウン・リストから、**3** を選択します。
15. **What should start the reconfiguration?** リストで、**CRC, POF ID, SW ID Error, Core nConfig asserted, nStatus asserted、および Pin nConfig asserted** チェック・ボックスにチェックを付けます。**Watchdog timed out** チェック・ボックスのチェックを外します。図 2-7 に、上記選択を行った後のダイアログ・ボックスを示します。
16. **Next** をクリックします。ページ 4 (図 2-8) が表示されます。

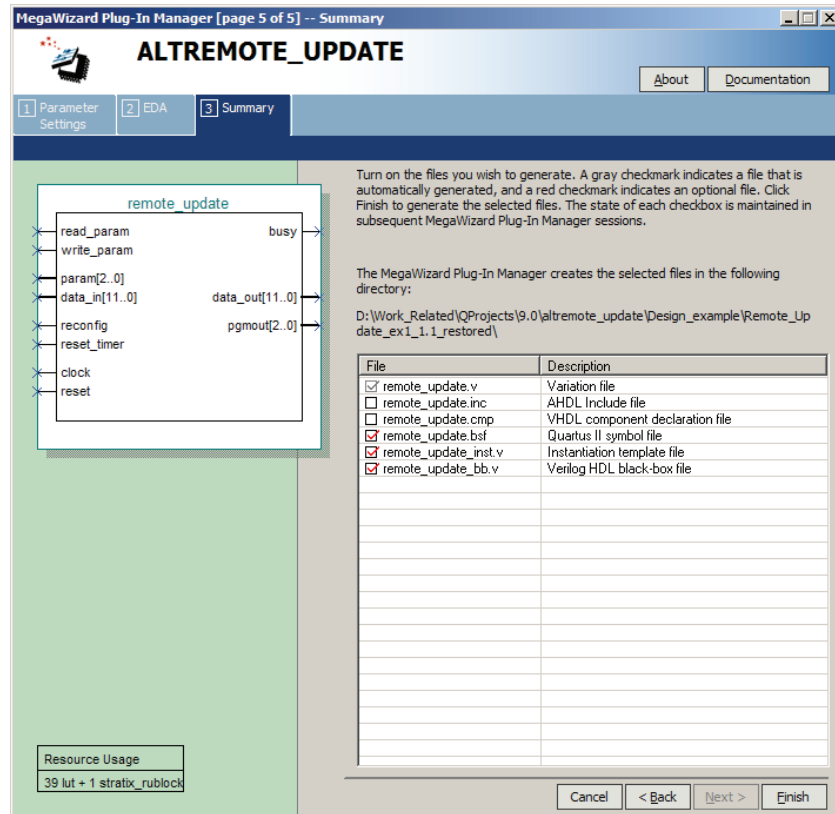
図 2-8. MegaWizard Plug-In Manager—AL TREMOTE_UPDATE [ページ 4/5] – EDA



17. **Generate a netlist for synthesis area and timing estimation** チェック・ボックスのチェックを外します。

18. **Next** をクリックします。ページ 5 (図 2-9) が表示されます。

図 2-9. MegaWizard Plug-In Manager—AL TREMOTE_UPDATE [ページ 5/5] – Summary



19. **Quartus II symbol file**、**Instantiation template file**、および **Verilog HDL black box declaration file** チェック・ボックスにチェックを付け、**AHDL Include file** および **VHDL Component declaration file** オプションのチェックを外します。図 2-9 に、ステップ 19 のチェック・ボックスを選択した後のダイアログ・ボックスを示します。

20. **Finish** をクリックします。

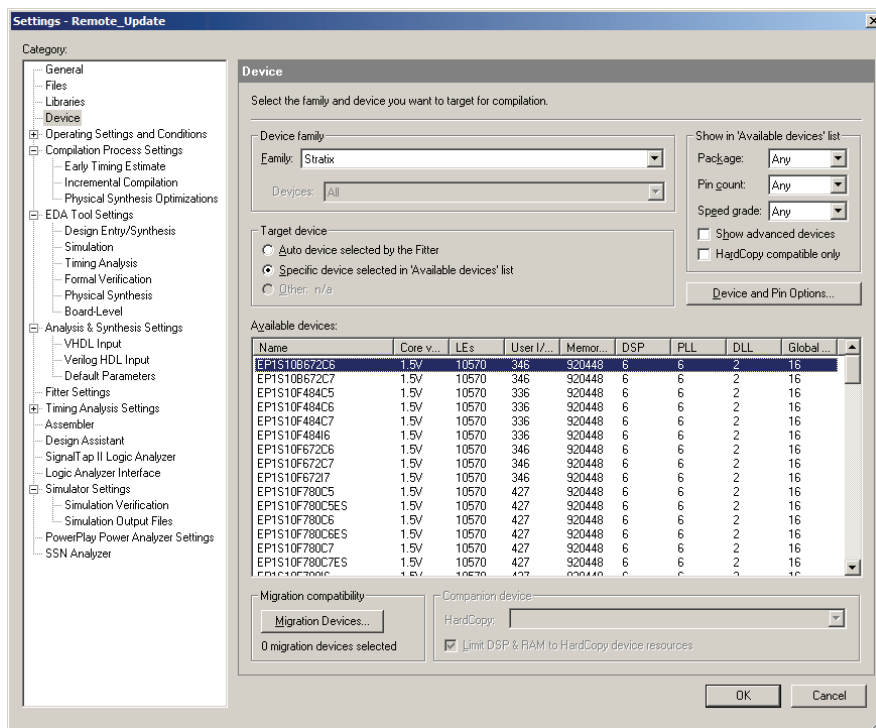
AL TREMOTE_UPDATE モジュールが構築されます。

デザインの実装

このセクションでは、EP1S10B672C6 デバイスをプロジェクトに割り当てて、プロジェクトをコンパイルする方法を説明します。

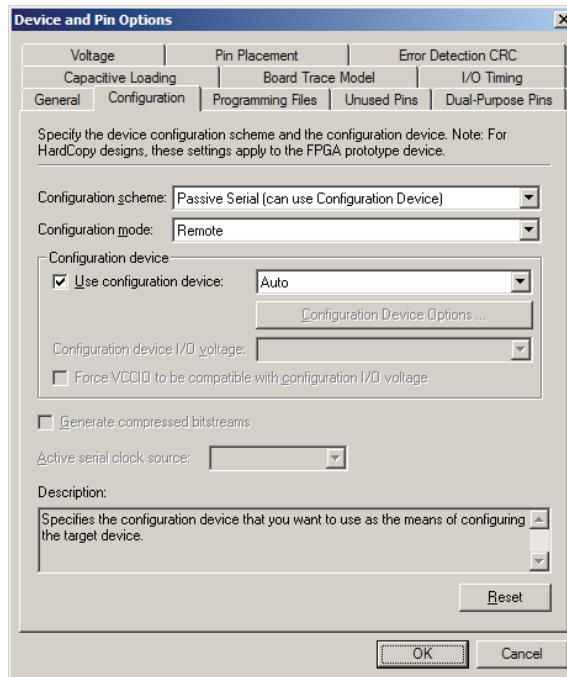
1. Quartus II 開発ソフトウェアで、**Remote_Update_ex1_1.1.qar** を開いた状態で、Assignments メニューの **Settings** を選択します。**Settings** ダイアログ・ボックスが表示されます。
2. **Category** リストから、**Device** を選択します。デバイス選択用のウィンドウが表示されます (図 2-10)。

図 2-10. デバイスの選択



3. **Which device family will you be using?** プルダウン・リストから **Stratix** を選択します。
4. **Target device** リストで、**Specific device selected in 'Available devices'** オプションにチェックを付けます。
5. **Available devices** リストから、**EP1S10B672C6** を選択します。
6. **Device & Pin Options...** ボタンをクリックします。**Device & Pin Options** ダイアログ・ボックスが表示されます。

図 2-11. Device and Pin Options ダイアログ・ボックス



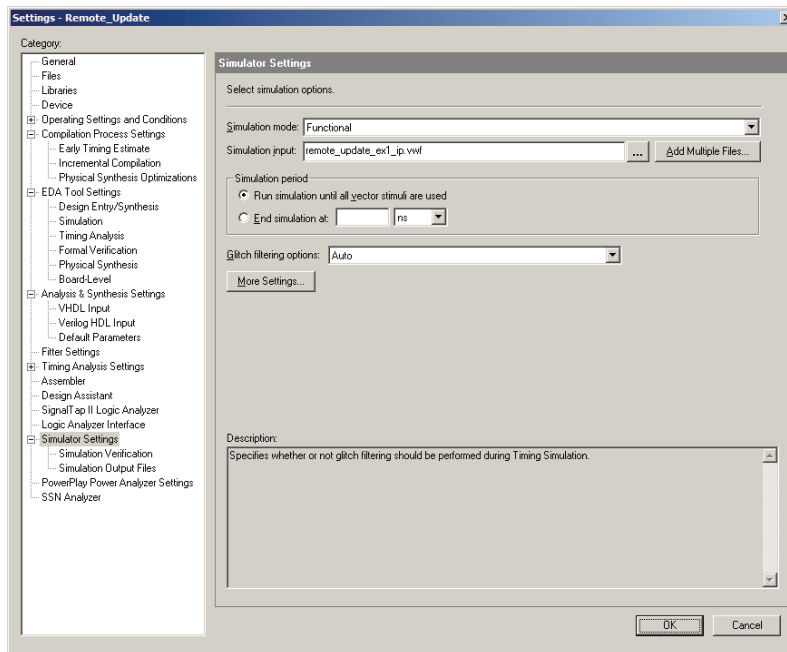
7. **Configuration** タブをクリックします。
8. **Configuration mode** プルダウン・リストから **Remote** を選択します。
9. 他のチェック・ボックスは、デフォルトの状態のままにして **OK** をクリックします。図 2-10 に、上記のすべてのオプションの選択および設定を行った後のダイアログ・ボックスを示します。
10. **Settings** ダイアログ・ボックスで、他のオプションをデフォルトの状態のままにして、**OK** をクリックします。
11. Processing メニューの **Start Compilation** を選択します。
12. **Full Compilation was successful** ダイアログ・ボックスが表示されたら、**OK** をクリックします。

機能の結果—Quartus II 開発ソフトウェアによるデザインのシミュレーション

このセクションは、Quartus II シミュレータでデザインをシミュレーションすることにより作成したデザイン例の検証方法について説明します。Quartus II シミュレータを設定するには、以下のステップを実行します。

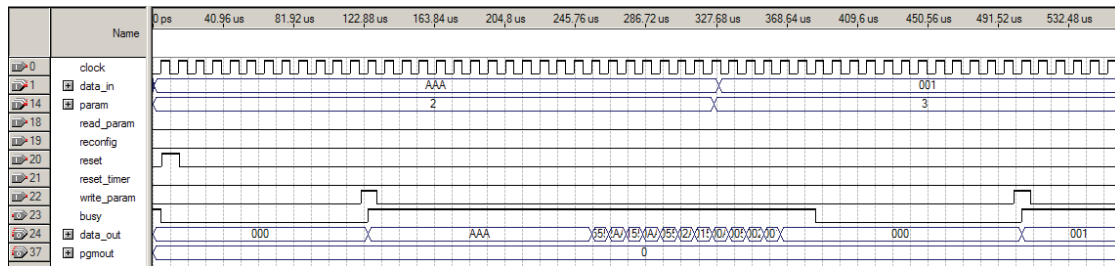
1. Quartus II 開発ソフトウェアで、**Remote_Update_ex1_1.1.qar** を開いた状態で、Processing メニューの **Generate Functional Simulation Netlist** を選択します。
2. **Functional Simulation Netlist Generation was successful** ダイアログ・ボックスが表示されたら、**OK** をクリックします。
3. Assignments メニューの **Settings** を選択します。**Settings** ダイアログ・ボックスが表示されます (図 2-12)。

図 2-12. シミュレータの設定



4. **Category** プルダウン・リストから **Simulator Settings** を選択します。
5. **Simulation mode** リストから **Functional** を選択します。
6. **Simulation input** ダイアログ・ボックスに `remote_update_ex1_ip.vwf` を入力するか、**Browse (...)** をクリックして、プロジェクト・フォルダ内からファイルを選択します。
7. **Run simulation until all vector stimuli are used** オプションにチェックを付けます。
8. **OK** をクリックします。図 2-12 に、上記設定を行った後のダイアログ・ボックスを示します。
9. シミュレーションを実行するには、**Processing** メニューの **Start Simulation** をクリックします。
10. **Simulation was successful** ダイアログ・ボックスが表示されたら、**OK** をクリックします。
11. シミュレーションの波形結果を検証します (図 2-13)。

図 2-13. シミュレーション波形



以下のセクションでは、このデザイン例に対して Quartus II 開発ソフトウェアで行われた機能シミュレーションについて説明します。使用されるステイミュラスは、ウォッチドッグ・タイマ・タイムアウト値の **AAA** (357,826,560 クロック・サイクル) への設定、ウォッチドッグ・タイマのイネーブル、リコンフィギュレーション用のページ 6 の選択、リコンフィギュレーションの開始を行うためのものです。

シミュレーションの最初は、ウォッチドッグのタイムアウト値の設定です。

- `data_in[]` のデフォルトは 1010_1010_1010 (357,826,560 クロック・サイクル) に設定され、ウォッチドッグ・タイマのタイムアウト値が指定されます。
- `param[]` のデフォルトは 010 に設定されます。この設定により、ウォッチドッグ・タイマ値パラメータが選択されます。
- $t = 121.4 \mu\text{s}$ の時点で、`write_param` が 1 クロック・サイクルの間アサートされます。これによって、タイマ値のパラメータをリモート・アップデート・ブロックに書き込む動作が始まります。
- `busy` 信号は、データ・ロード開始時の $t = 125 \mu\text{s}$ からパラメータが正常に書き込まれる時点 $t = 385.8 \mu\text{s}$ までアクティブになります。

シミュレーションの 2 番目は、ウォッチドッグ・タイマのイネーブルです。

- `data_in[]` のデフォルトは 0001 に設定され、ウォッチドッグ・タイマがイネーブルされます。このパラメータには、1 (イネーブル) か 0 (ディセーブル) の 2 つの値しかありません。
- `param[]` のデフォルトは 011 に設定されます。この設定により、ウォッチドッグ・イネーブル・パラメータが選択されます。
- $t = 501 \mu\text{s}$ の時点で、`write_param` が 1 クロック・サイクルの間アサートされます。これによって、ウォッチドッグをイネーブルするためのパラメータをリモート・アップデート・ブロックに書き込む動作が始まります。
- `busy` 信号は、データ・ロード開始時の $t = 505 \mu\text{s}$ からパラメータが正常に書き込まれる時点 $t = 765 \mu\text{s}$ までアクティブになります。

シミュレーションの 3 番目は、ページ選択パラメータを 6 に設定することです。

- `data_in[]` を 0110 (ページ 6) に設定して、リコンフィギュレーション中にロードする目的のページを指定します。このパラメータで使用できる値は、0 (ファクトリ・コンフィギュレーション)、または 1 ~ 7 (アプリケーション・コンフィギュレーション) の 8 つのみです。

- param[] は 100 に設定されます。これによって、ページ選択パラメータが選択されます。
- $t = 870 \mu\text{s}$ の時点で、write_param が 1 クロック・サイクルの間アサートされます。これによって、ページ選択パラメータを設定するためのパラメータをリモート・アップデート・ブロックに書き込む動作が始まります。
- busy 信号は、データ・ロード開始時の $t = 875 \mu\text{s}$ からパラメータが正常に書き込まれる時点 $t = 1135 \mu\text{s}$ までアクティブになります。

シミュレーションの最後は、チップのリコンフィギュレーションを現在の設定で開始することです。これは、 $t = 1270 \mu\text{s}$ の時点で、reconfig がアサートされると実行されます。この後、チップはリコンフィギュレーションを開始します。pgmout[] ピンはリコンフィギュレーション中にロードされたページ (6 = 0110) の値でドライブされます。

機能の結果—ModelSim-Altera ソフトウェアによるデザインのシミュレーション

両方のシミュレータの結果を比較するには、ModelSim-Altera ソフトウェアでデザインをシミュレーションします。このユーザーガイドは、デザイン例を使用する前に ModelSim-Altera ソフトウェアの使用方法を十分に理解しているユーザーを対象としています。ModelSim-Altera ソフトウェアをよく知らない場合は、アルテラ・ウェブサイトの「[ModelSim-Altera Software Support](#)」セクションを参照してください。サポート・ページには、インストール、使用方法、およびトラブルシューティングのようなトピックへのさまざまなリンクがあります。

ModelSim-Altera ソフトウェアを設定するには、以下のステップを実行します。

1. **ALTREMOTE_UPDATE_ex1_msim.zip** ファイルを PC 上の任意の作業ディレクトリに解凍します。
2. ファイルを解凍したフォルダに移動します。
3. **remote_update_ex1.do** ファイルをテキスト・エディタで開きます。
4. **remote_update_ex1.do** ファイルの 1 行目の `<directory_path>` を適切なライブラリ・ファイルのディレクトリ・パスに置き換えます。例：
`C:/Modeltech_ae/altera/verilog/stratix`
5. File メニューの **Save** をクリックします。
6. **ModelSim-Altera** ソフトウェアを起動します。
7. File メニューの **Change Directory** をクリックします。
8. ファイルを解凍したフォルダを選択します。 **OK** をクリックします。
9. Tools メニューの **Execute Macro** をクリックします。
10. **remote_update_ex1.do** ファイルを選択して **Open** をクリックします。このファイルは、シミュレーションに必要なすべての設定を自動的に行うための ModelSim-Altera ソフトウェア用スクリプト・ファイルです。
11. Waveform Viewer ウィンドウに表示された結果を検証します。

Quartus II シミュレータで、信号の配列を変更したり、冗長信号を取り除いたり、結果に合わせて基数を変更したりしなければならない場合もあります。

デザイン例2：パラメータの読み出し

このデザイン例は、ウォッチドッグ・イネーブル・ステータス、ウォッチドッグ・タイムアウト値、リコンフィギュレーション用のページ・セット、および現在のコンフィギュレーション状態を読み出すための制御信号のシーケンスを示します。

デザイン・ファイル

デザイン・ファイルは、アルテラのウェブサイトの「[オンライン資料](#)」のページにあります。ファイルは、以下のセクションの下にあります。

- 「[オンライン資料：Quartus II 開発ソフトウェア](#)」ページで、「[メガファンクション](#)」のセクションを展開して、次に「[I/O](#)」セクションを展開します。
- アルテラのウェブサイトの「[オンライン資料：ユーザーガイド](#)」セクション

この例では、次の作業を行う必要があります。

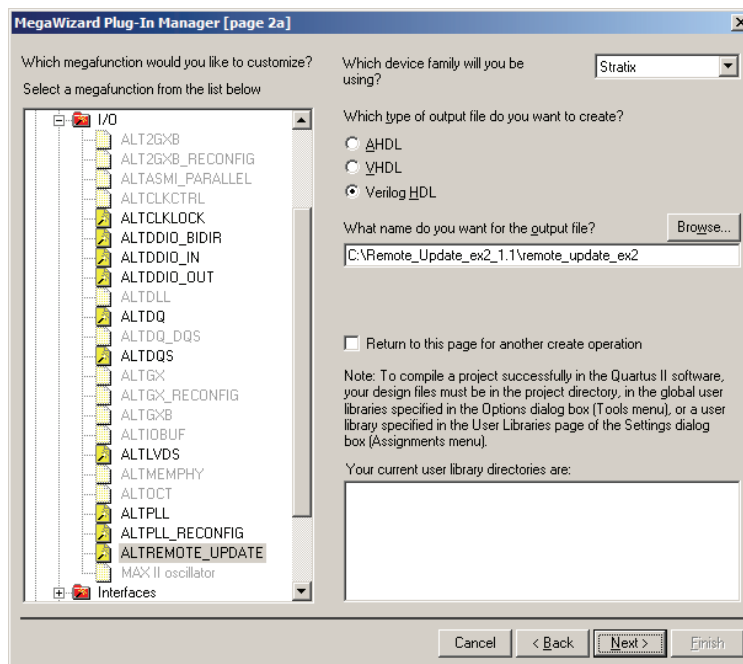
- ALTREMOTE_UPDATE メガファンクションおよび MegaWizard Plug-in Manager を使用して、リモート・アップデート・ブロックをインスタンス化します。
- デザインを実装して、EP1S25F1020C5 デバイスをプロジェクトに割り当てます。
- デザインをコンパイルしてシミュレーションします。

ALTREMOTE_UPDATE メガファンクション・バリエーションの生成

この例で作成された新規の ALTREMOTE_UPDATE メガファンクションは、Quartus II プロジェクトのトップ・レベル・ファイルに追加されます。

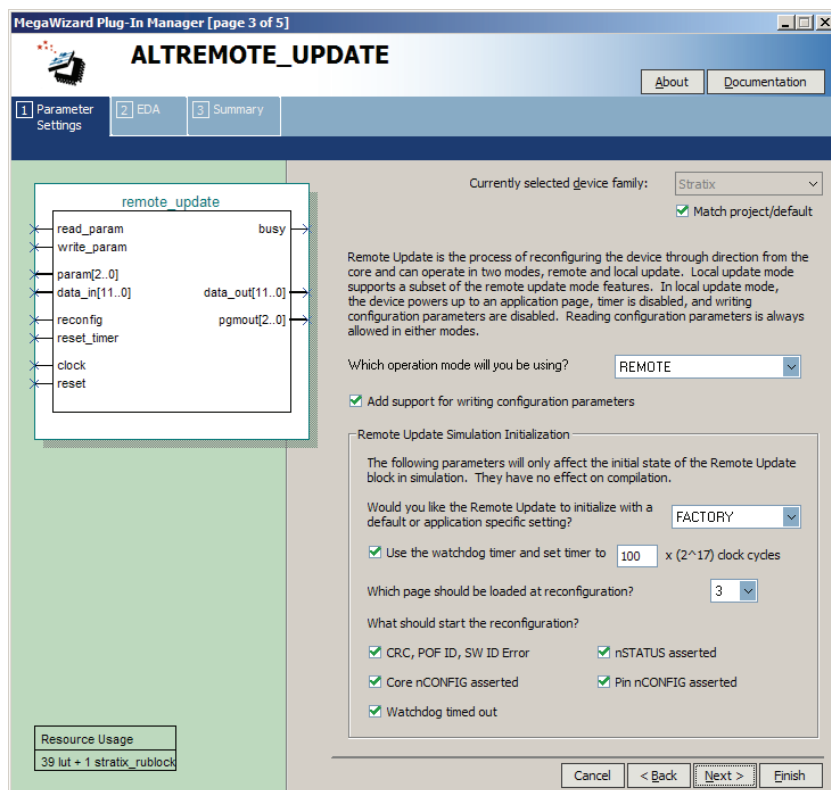
1. ALTREMOTE_UPDATE_DesignExample_ex2_msim.zip ファイルの内容を PC 上の任意の作業ディレクトリに解凍します。
2. Quartus II 開発ソフトウェアで、Remote_Update_ex2_1.1.qar プロジェクトを開きます。
3. Tools メニューの **MegaWizard Plug-In Manager** を選択します。
4. **Create a new custom megafunction variation** を選択し、**Next** をクリックします。MegaWizard Plug-In Manager ページが表示されます (図 2-14)。

図 2-14. MegaWizard Plug-In Manager [ページ 2a]



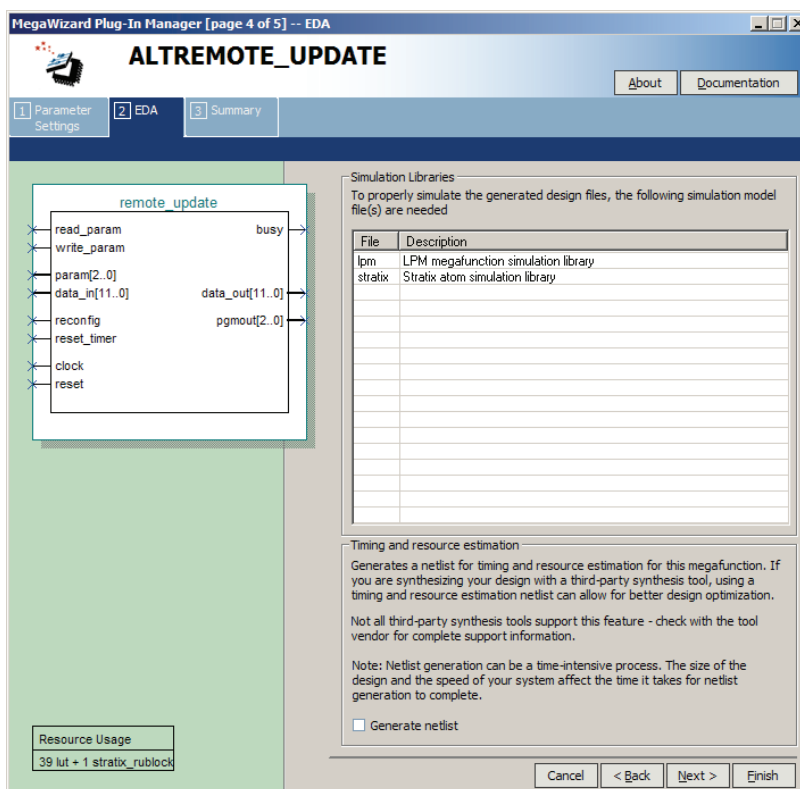
5. **Which device family will you be using?** プルダウン・リストから **Stratix** を選択します。
6. **Which type of output file do you want to create?** リストで、**Verilog HDL** オプションにチェックを付けます。
7. **Installed Plug-Ins** の下の **I/O** フォルダから **ALTREMOTE_UPDATE** を選択します。
8. 出力ファイル **remote_update_ex2** を指定します。図 2-14 に、上記選択を行った後のダイアログ・ボックスを示します。
9. **Next** をクリックします。ページ 3 (図 2-15) が表示されます。

図 2-15. MegaWizard Plug-In Manager—ALIREMOTE_UPDATE [ページ 3/5]



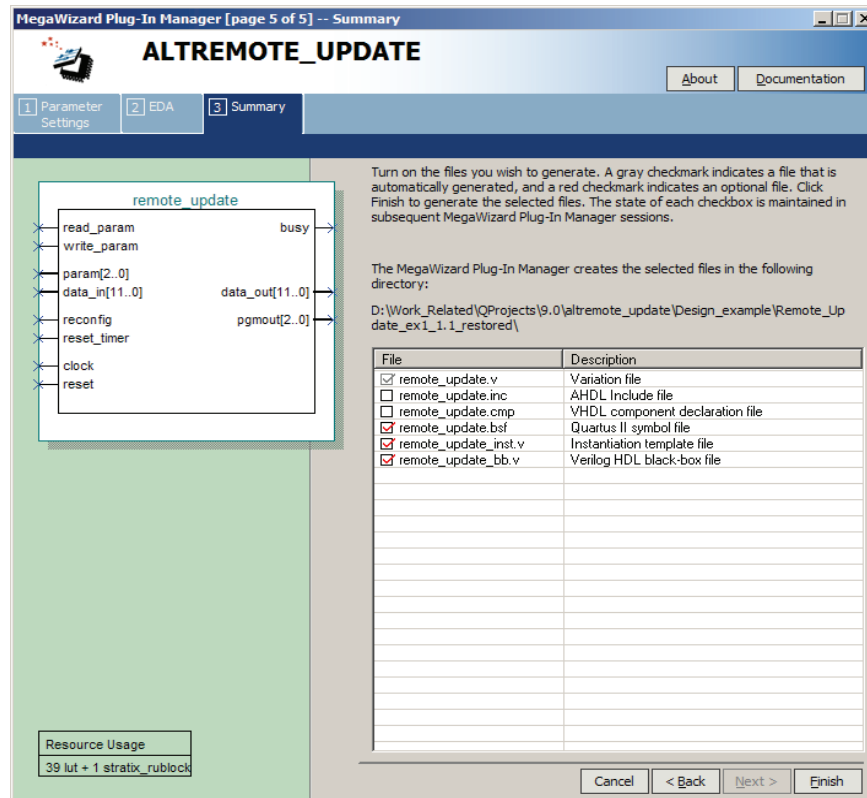
10. **Which operation mode will you be using?** プルダウン・リストから、**REMOTE** を選択します。
11. **Add support for writing configuration parameters** のチェック・ボックスにチェックを付けます。
12. **Would you like Remote Update to initiate with a default or application specific settings?** プルダウン・リストから、**FACTORY** を選択します。
13. **Use the Watchdog timer and set timer to** のチェック・ボックスにチェックを付けて、**100** を指定します。
14. **Which page should be loaded at reconfiguration?** プルダウン・リストから、**3** を選択します。
15. **What should start the reconfiguration?** リストのすべてのチェック・ボックスにチェックを付けます。図 2-15 に、これらの選択を行った後のダイアログ・ボックスを示します。
16. **Next** をクリックします。ページ 4 (図 2-16) が表示されます。

図 2-16. MegaWizard Plug-In Manager—AL TREMOT E_UPDATE [ページ 4/5] – EDA



17. **Generate a netlist for synthesis area and timing estimation** チェック・ボックスのチェックを外します。
18. **Next** をクリックします。ページ 5 (図 2-17) が表示されます。

図 2-17. MegaWizard Plug-In Manager—ALTRMOTEM_UPDATE [ページ 5/5] – Summary



19. **Quartus II symbol file**、**Instantiation template file**、および **Verilog HDL black-box file declaration file** のチェック・ボックスにチェックを付けます。**AHDL Include file** および **VHDL component declaration file** チェック・ボックスのチェックを外します。図 2-17 に、上記オプションすべてを選択した後のダイアログ・ボックスを示します。

20. **Finish** をクリックします。

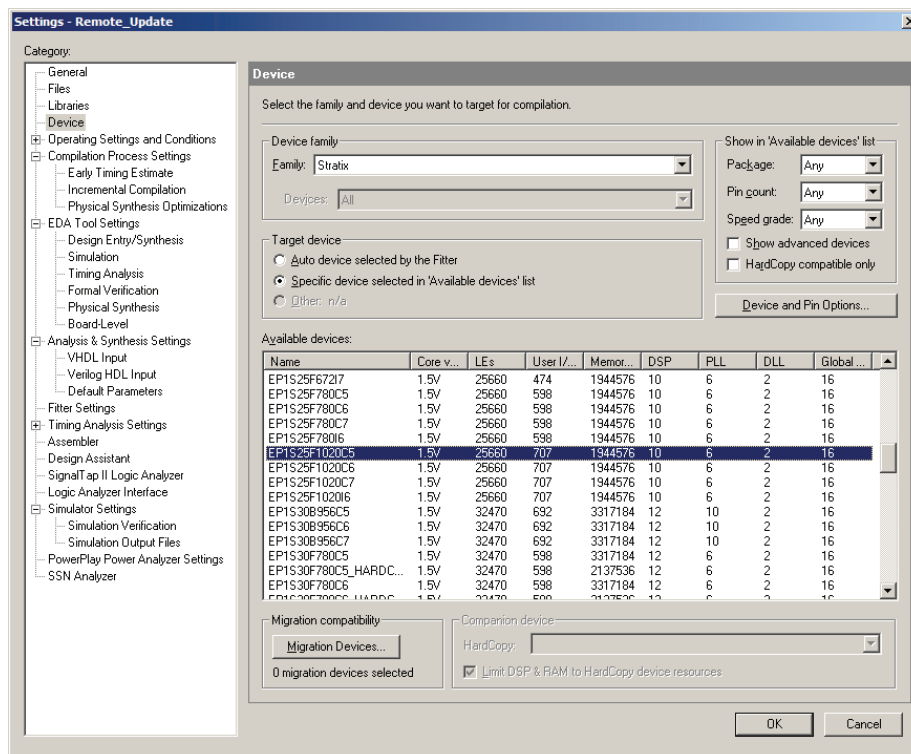
ALTRMOTEM_UPDATE_ex2 モジュールが構築されます。

デザインの実装

このセクションでは、EP1S25F1020C5 デバイスをプロジェクトに割り当てて、プロジェクトをコンパイルする方法を説明します。

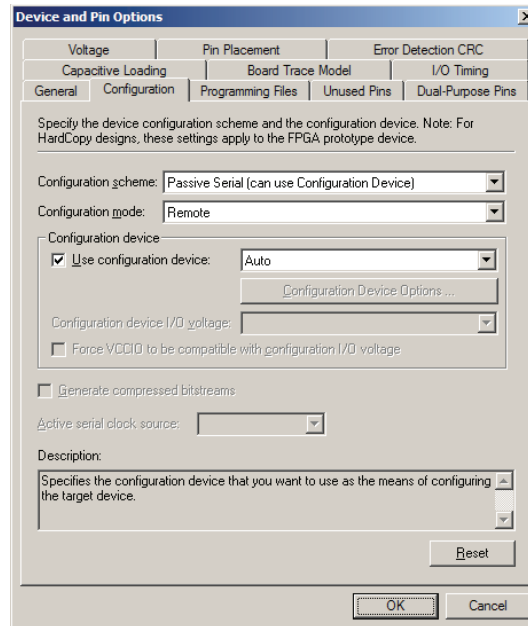
1. Quartus II 開発ソフトウェアで、**Remote_Update_ex2_1.1.qar** を開いた状態で、**Assignments** メニューの **Settings** を選択します。**Settings** ダイアログ・ボックスが表示されます (図 2-18)。

図 2-18. デバイスの設定



2. **Category** リストから **Device** を選択します。 **Device** ウィンドウが表示されます。
3. **Target device** リストで、 **Specific device selected in 'Available devices' list** オプションにチェックを付けます。
4. **Available devices** リストから、 **EP1S25F1020C5** を選択します。
5. **Device & Pin Options** ボタンをクリックします。 **Device & Pin Options** ダイアログ・ボックスが表示されます。

図 2-19. Device and Pin Options ダイアログ・ボックス



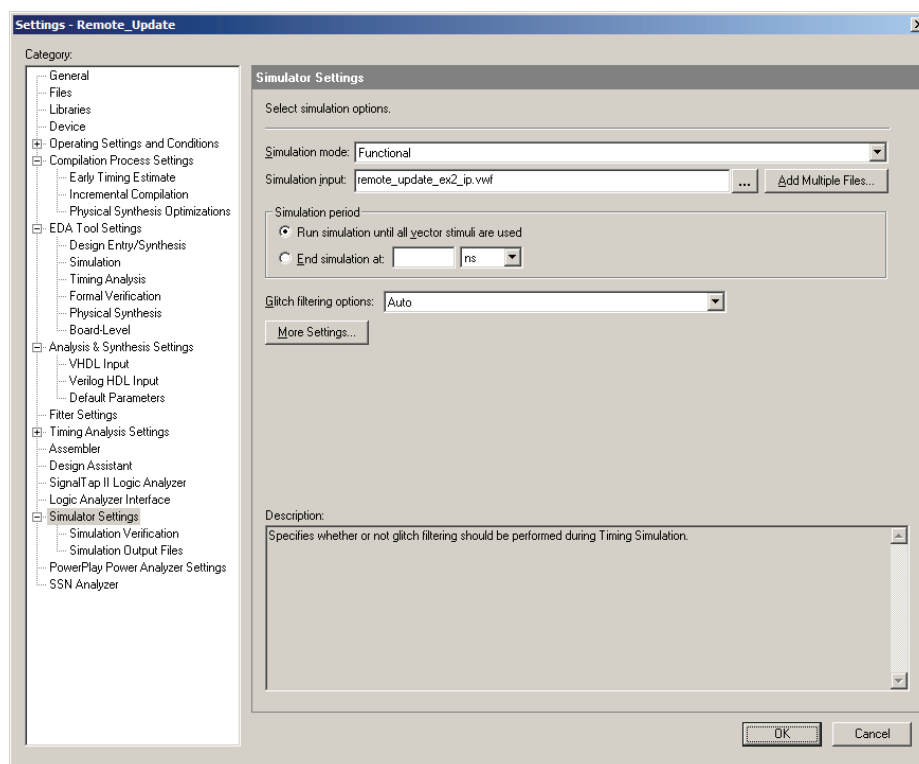
6. **Configuration** タブをクリックします。**Configuration mode** プルダウン・リストから **Remote** を選択します。
7. 他のオプションはデフォルト状態のままにして、**OK** をクリックします。
8. **Settings** ダイアログ・ボックスで、他のオプションはデフォルトの状態のままにして **OK** をクリックします。2-23 ページの図 2-18 に、上記のオプション選択および設定がすべて行われた後のページが表示されます。
9. Processing メニューの **Start Compilation** をクリックします。
10. **Full Compilation was successful** ダイアログ・ボックスが表示されたら、**OK** をクリックします。

機能の結果 — Quartus II 開発ソフトウェアによるデザインのシミュレーション

このセクションは、Quartus II シミュレータでデザインをシミュレーションすることにより作成したデザイン例の検証方法について説明します。Quartus II シミュレータを設定するには、以下のステップを実行します。

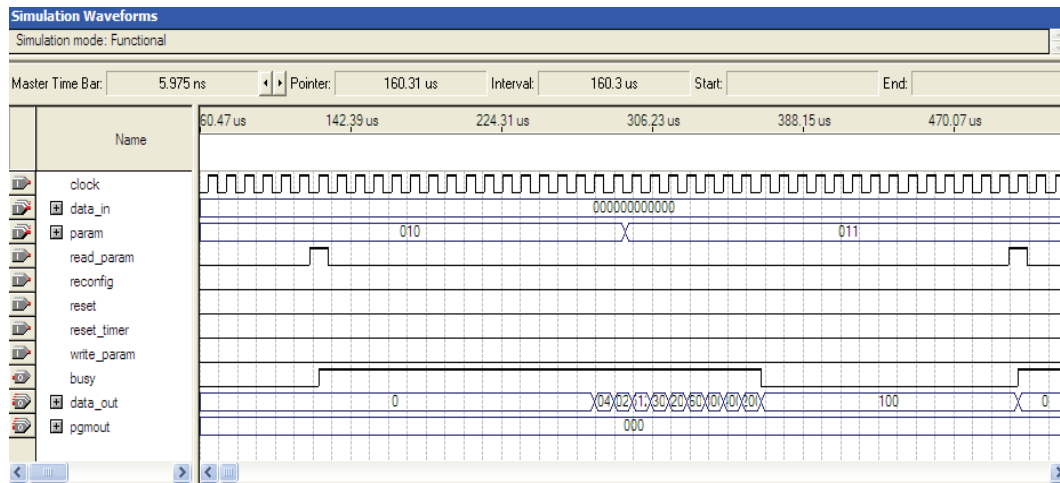
1. Quartus II 開発ソフトウェアで、**Remote_Update_ex2_1.1.qar** を開いた状態で、Processing メニューの **Generate Functional Simulation Netlist** を選択します。
2. **Functional Simulation Netlist Generation was successful** ダイアログ・ボックスが表示されたら、**OK** をクリックします。
3. Assignments メニューの **Settings** を選択します。**Settings** ダイアログ・ボックスが表示されます (図 2-20)。

図 2-20. シミュレータの設定



4. **Category** リストから **Simulator Settings** を選択します。
5. **Simulation mode** プルダウン・リストから、**Functional** を選択します。
6. **Simulation input** ダイアログ・ボックスに remote_update_ex2_ip.vwf を入力するか、**Browse (...)** をクリックして、プロジェクト・フォルダ内からファイルを選択します。
7. **Run simulation until all vector stimuli are used** オプションにチェックを付けます。
8. **OK** をクリックします。図 2-20 に、上記の設定および選択をすべて行った後のページを示します。
9. Processing メニューの **Start Simulation** をクリックします。
10. **Simulation was successful** ダイアログ・ボックスが表示されたら、**OK** をクリックします。
11. **Simulation Report** ウィンドウが表示されます。シミュレーションの波形結果を検証します (図 2-21)。

図 2-21. シミュレーション波形



以下のセクションでは、このデザイン例に対して Quartus II 開発ソフトウェアで行われた機能シミュレーションについて説明します。使用されるステイミュラスは、ウォッチドッグ・タイマ・タイムアウト値、ウォッチドッグ・タイムアウト・イネーブル・パラメータ、ページ・セットの値、およびコンフィギュレーション・パラメータの値を読み出すためのものです。

シミュレーションの最初は、ウォッチドッグ・タイムアウト・パラメータの値の読み出しです。以下にこの読み出しの方法を示します。

- param[] のデフォルトは 010 に設定され、ウォッチドッグ・タイマ値パラメータが選択されます。
- read_param のデフォルト設定が、1 クロック・サイクルの間アサートされます。t = 121 μs の時点で、リモート・アップデート・ブロックからのデータ読み出しが始まります。
- busy 信号は、データ読み出し開始時の t = 125 μs からパラメータが正常に読み出される時点の t = 365 μs までアクティブになります。

data_out[] は有効になり、読み出されたパラメータの現在の値が含まれます（この場合、値は 0000_0110_0100 です。これは、タイムアウト・カウンタ 29 ビットのうちの上位 12 ビットで、ウォッチドッグ・タイマが、カウンタ値 100 でタイムアウトすることを意味しています）。

シミュレーションの 2 番目は、ウォッチドッグ・タイムアウト・イネーブル・パラメータの値の読み出しです。以下にこの読み出しの方法を示します。

- param[] は 011 に設定され、ウォッチドッグ・タイムアウト・イネーブル・パラメータが選択されます。
- read_param が、1 クロック・サイクルの間アサートされる t = 500 μs の時点で、リモート・アップデート・ブロックからのデータ読み出しが始まります。
- busy 信号は、データ読み出し開始時の t = 505 μs からパラメータが正常に読み出される時点の t = 735 μs までアクティブになります。

data_out[] が有効になり、0000_0000_0001 が含まれます。設定されている LSB ビットは、ウォッチドッグ・タイマがイネーブルされていることを示しています。

シミュレーションの3番目は、ページ・セットの値の読み出しです。以下にこの読み出しの方法を示します。

- param[] は 100 に設定され、Page Select パラメータが選択されます。
- read_param が、1 クロック・サイクルの間アサートされる $t = 870 \mu\text{s}$ の時点で、リモート・アップデート・ブロックからのデータ読み出しが始まります。
- busy 信号は、データ読み出し開始時の $t = 875 \mu\text{s}$ からパラメータが正常に読み出される時点の $t = 1075 \mu\text{s}$ までアクティブになります。

data_out[] が有効になり、0000_0000_0011 が含まれます。3つのLSBビットは、ページ・セット値が3であることを示します。

シミュレーションの4番目は、コンフィギュレーションの値の読み出しです。以下にこの読み出しの方法を示します。

- param[] は 101 に設定され、コンフィギュレーション・パラメータが選択されません。
- read_param が、1 クロック・サイクルの間アサートされる $t = 1260 \mu\text{s}$ の時点で、リモート・アップデート・ブロックからのデータ読み出しが始まります。
- busy 信号は、データ読み出し開始時の $t = 1265 \mu\text{s}$ からパラメータが正常に読み出される時点の $t = 1455 \mu\text{s}$ までアクティブになります。

data_out[] が有効になり、0000_0000_0000 が含まれます。LSBビットは、現在のコンフィギュレーションがファクトリ・コンフィギュレーションであることを示します。

機能の結果 — ModelSim-Altera ソフトウェアによるデザインのシミュレーション

両方のシミュレータの結果を比較するには、ModelSim-Altera ソフトウェアでデザインをシミュレーションします。このユーザーガイドは、デザイン例を使用する前に ModelSim-Altera ソフトウェアの使用方法を十分に理解しているユーザーを対象としています。ModelSim-Altera ソフトウェアをよく知らない場合は、アルテラ・ウェブサイトの「[ModelSim-Altera Software Support](#)」セクションを参照してください。サポート・ページには、インストール、使用方法、およびトラブルシューティングのようなトピックへのさまざまなリンクがあります。

以下のステップを実行して、ModelSim-Altera ソフトウェアを設定します。

1. **ALTREMOTE_UPDATE_ex2_msim.zip** ファイルを PC 上の任意の作業ディレクトリに解凍します。
2. ファイルを解凍したフォルダに移動します。
3. **remote_update_ex2.do** ファイルをテキスト・エディタで開きます。
4. **remote_update_ex2.do** ファイルの1行目の `<directory_path>` を適切なライブラリ・ファイルのディレクトリ・パスに置き換えます。例：
C:/Modeltech_ae/altera/verilog/stratix
5. File メニューの **Save** をクリックします。
6. ModelSim-Altera ソフトウェアを起動します。
7. File メニューの **Change Directory** をクリックします。
8. ファイルを解凍したフォルダを選択します。 **OK** をクリックします。
9. Tools メニューの **Execute Macro** をクリックします。

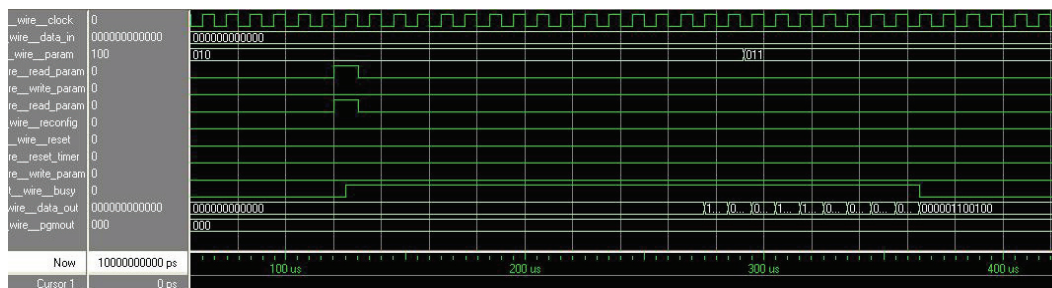
10. **remote_update_ex2.do** ファイルを選択して **Open** をクリックします。これはシミュレーションに必要なすべての設定を自動的に行うための ModelSim-Altera ソフトウェア用スクリプト・ファイルです。

11. Waveform Viewer ウィンドウに表示された結果を検証します。

Quartus II シミュレータで、信号の配列を変更したり、冗長信号を取り除いたり、結果に合わせて基数を変更したりしなければならない場合があります。

図 2-22 に、予測される ModelSim-Altera ソフトウェアでのシミュレーション結果を示します。

図 2-22. ModelSim-Altera ソフトウェアによるシミュレーション波形



結論


Quartus II 開発ソフトウェアは、加算器やカウンタなどの単純な演算ユニットから、最新 PLL (Phase-Locked Loop) ブロック、乗算器、およびメモリ構造に及ぶ、パラメータ化可能なメガファンクションを提供します。これらのメガファンクションはアルテラのデバイス用に性能が最適化されており、コーディング・プロセスを自動化して貴重な設計時間を節約するため、より効率的なロジック合成およびデバイス実装を可能にします。アルテラは、デザインの実装時にこれらの機能を使用することを推奨します。これによって、一貫してデザイン目標を達成することができます。

ポートおよびパラメータ

このセクションに示すオプションでは、アプリケーションに応じて ALTREMOTE_UPDATE メガファンクションをカスタマイズするための、各デバイスで使用できる、すべてのポートおよびパラメータについて説明します。

表 3-1 に、ALTREMOTE_UPDATE メガファンクションの入力ポート、表 3-2 に、ALTREMOTE_UPDATE メガファンクションの出力ポート、および表 3-3 に、ALTREMOTE_UPDATE メガファンクションのパラメータを示します。

表 3-3 に、Cyclone® III デバイスでの ALTREMOTE_UPDATE 動作と、その動作を指定するための設定を示します。この表では、read_source ポートの使用方法を説明します。

 ALTREMOTE_UPDATE メガファンクションのポートおよびパラメータの最新情報は、最新の Quartus® II Help を参照してください。

パラメータの詳細は、MegaWizard™ Plug-In Manager インタフェースを使用しないで、メガファンクションをデザイン内で直接パラメータ化されたインスタンスとして使用する場合にのみ関係しています。これらのパラメータの詳細は、MegaWizard Plug-In Manager インタフェースには表示されません。

表 3-1. ALTREMOTE_UPDATE メガファンクション入力ポート (1/5)

ポート名	必須	説明	備考
clock	○	リモート・アップデート・ブロックへのクロック入力	マシンを制御するため、およびパラメータのアップデート中にリモート・アップデート・ブロックをドライブするためのクロック入力です。このポートは有効なクロックに接続されていなければなりません。
reset	○	メガファンクションへの非同期リセット入力	マシンを有効な状態に初期化するための非同期リセット入力です。初回使用前にマシンをリセットする必要があります。リセットしない場合は、有効な状態が保証されません。
data_in[]	×	リモート・アップデート・ブロックにパラメータ・データを書き込むためのデータ入力	<p>パラメータ・データ用の入力バスです。パラメータによっては、一部のビットしか使用されないものがあります。その場合は、下位ビットが使用されます（例：ステータス値はビット 0-4 を使用します）。このバスは、未接続のままの場合、デフォルトは 0 です。現在のコンフィギュレーションがアプリケーション・コンフィギュレーションの場合、ポートは無視されます。このバスの幅はデバイスに依存します。</p> <ul style="list-style-type: none"> ■ Arria® GX、Stratix® II、Stratix II GX、Stratix、および Stratix GX デバイスの場合、12 ビット・バス ■ Stratix III デバイスの場合、24 ビット・バス ■ Cyclone III デバイスの場合、22 ビット・バス

表 3-1. ALTREMOTE_UPDATE メガファンクション入力ポート (2/5)

ポート名	必須	説明	備考
param[]	×	読み出したりはアップデートを行う必要のあるパラメータを指定するバス	<p>読み出したりはアップデートするパラメータを選択する3ビット・バスです。</p> <p>このポートが未接続のままの場合、このポートのデフォルト値は000です。</p> <p>Arria GX、Stratix III、Stratix II、Stratix II GX、Stratix、およびStratix GX デバイスの場合、各パラメータ・タイプおよび対応するパラメータ・ビット幅へのマッピングは以下のように定義されています。</p> <ul style="list-style-type: none"> ■ 000 - リコンフィギュレーション・トリガ条件（読み出しのみ）- 幅 5。トリガ・ビットは以下のように定義されています。 <ul style="list-style-type: none"> ビット 4： wdtimer_source：ユーザー・ウォッチドッグ・タイマのタイムアウト ビット 3： nconfig_source：外部コンフィギュレーション・リセット（nCONFIG）のアサーション ビット 2： runconfig_source：ロジック・アレイからトリガされたコンフィギュレーション・リセット ビット 1： nstatus_source：エラー発生時の外部デバイスによる nSTATUS アサーション ビット 0： crcerror_source：アプリケーション・コンフィギュレーション時の CRC エラー ■ 001 - （無効な値） ■ 010 - ウォッチドッグ・タイムアウト値 - 幅 12 ■ 011 - ウォッチドッグ・イネーブル - 幅 1

表 3-1. ALTREMOTE_UPDATE メガファンクション入力ポート (3/5)

ポート名	必須	説明	備考						
param[] (続き)			<ul style="list-style-type: none"> ■ 100 - ページ選択 Stratix および Stratix GX デバイス : <ul style="list-style-type: none"> ■ ページの読み出しおよび書き込み時は幅 3 です。 エンハンスド・コンフィギュレーション・デバイスの使用時にのみ適用されます。 Arria GX、Stratix II、および Stratix II GX デバイス : <ul style="list-style-type: none"> ■ ページの読み出しおよび書き込み時は幅 3 です。 エンハンスド・コンフィギュレーション・デバイスの使用時にのみ適用されます。 ■ 開始アドレスの読み出しおよび書き込み時は幅 7 です。 PGM[6..0] レジスタは、24 ビットの開始アドレスのうちビット [22..16] を構成します。残りの 17 ビットは 0 に設定されます。 シリアル・コンフィギュレーション・デバイスの使用時にのみ適用されます (AS モード)。 Stratix III デバイス : <ul style="list-style-type: none"> ■ 開始アドレスの読み出しおよび書き込み時は幅 24 です。PGM[23..0] レジスタは、24 ビット開始アドレスを形成します。残りの 17 ビットは 0 に設定されます。 シリアル・コンフィギュレーション・デバイスの使用時にのみ適用されます (AS モード)。 Stratix III デバイスは、このモードをリモート・コンフィギュレーションのみに使用します。 ■ 101 - コンフィギュレーション・モード (AnF) - 幅は 1 です。ローカル・アップデート・モードでは、このパラメータの読み出しのみが可能です。アプリケーション・ページでは 1 に、ファクトリ・ページでは 0 に設定されます。リモート・アップデート・モードでは、このパラメータの読み出しおよび書き込みができます。リモート・アップデート・モードでアプリケーション・ページをロードする前に、アルテラは、これを 1 に設定することを推奨します。 ■ 110 - (無効な値) ■ 111 - (無効な値) <p>Cyclone III デバイスの場合、各パラメータ・タイプへのマッピングおよび対応するパラメータ・ビット幅は以下のように定義されます。</p> <ul style="list-style-type: none"> ■ 000 - マスタ・ステート・マシンの現在の状態モード (読み出しのみ) - 幅は 2 です。値は以下のように定義されます。 <table style="margin-left: 20px; border: none;"> <tr> <td style="padding-right: 20px;">00</td> <td>ファクトリ・モード</td> </tr> <tr> <td>01</td> <td>アプリケーション・モード</td> </tr> <tr> <td>11</td> <td>マスタ・ステート・マシン・ユーザー・ウォッチドッグ・タイマがイネーブルされたアプリケーション・モード</td> </tr> </table> 	00	ファクトリ・モード	01	アプリケーション・モード	11	マスタ・ステート・マシン・ユーザー・ウォッチドッグ・タイマがイネーブルされたアプリケーション・モード
00	ファクトリ・モード								
01	アプリケーション・モード								
11	マスタ・ステート・マシン・ユーザー・ウォッチドッグ・タイマがイネーブルされたアプリケーション・モード								

表 3-1. ALTREMOTE_UPDATE メガファンクション入力ポート (4/5)

ポート名	必須	説明	備考
param[] (続き)			<ul style="list-style-type: none"> ■ 001 - 早期CONF_DONE (Cd_early)チェックの強制 - 幅1です。 ■ 010 - ウォッチドッグ・タイムアウト値 - 読み出し時は幅29、書き込み時は幅12です。書き込み用の12ビットは、29ビット・ウォッチドッグ・タイムアウト値の上位12ビットであることに注意してください。 ■ 011 - ウォッチドッグ・イネーブル - 幅1 ■ 100 - ブート・アドレス - 読み出し時は幅24、書き込み時は幅22です。ブート・アドレスは、24ビット・ブート・アドレスの上位22ビットに書き込まれる必要があります。 ■ 101 - (無効な値) ■ 110 - 開始ステート・マシン・クロック (Osc_int) オプション・ビットとして内部オシレータを強制します - 幅1です。 ■ 111 - リコンフィギュレーション・トリガ条件 (読み出しのみ) - 幅5です。トリガ・ビットは以下のように定義されています。 <ul style="list-style-type: none"> ビット4: nconfig_source : 外部コンフィギュレーション・リセット (nCONFIG) のアサーション ビット3: crcerror_source : アプリケーション・コンフィギュレーション時のCRCエラー ビット2: nstatus_source : エラー発生時の外部デバイスによる nSTATUS アサーション ビット1: wdtimer_source : ユーザー・ウォッチドッグ・タイマのタイムアウト ビット0: runconfig_source : ロジック・アレイからトリガされたコンフィギュレーション・リセット <p>すべてのパラメータは、ファクトリ・コンフィギュレーション・モードの場合にのみ書き込むことができます。</p> <p>デバイスによってサポートされるパラメータが異なります。関連するデバイス・ハンドブックの「リモート・システム・アップグレード」の章を参照してください。</p>
read_param	×	param[] 入力ポートで指定され、data_out[] 出力ポートに供給されるパラメータのリード信号	param[] ポートで指定されたパラメータを読み出す必要があることを示す信号です。data_out[] で設定されたビット数は、パラメータのタイプによって異なります。信号は立ち上がりクロック・エッジでサンプリングされます。パラメータが後続のクロック・サイクルで再び読み出されるのを防ぐために、信号は1クロック・サイクルの間だけアサートされる必要があります。busy信号は、read_paramがアクティブとして読み出されると直ちにアクティブになります。パラメータが読み出されている間、busy信号はアサートされたままで、data_out[] は無効です。busy信号が非アクティブになると、data_out[] は有効になり、次のパラメータを読み出すことができます。

表 3-1. ALTREMOTE_UPDATE メガファンクション入力ポート (5/5)

ポート名	必須	説明	備考
write_param	×	param[] で指定され、data_in[] で指定された値が書き込まれるパラメータのライト信号	param[] で指定されたパラメータを data_in[] で指定された値で RU ブロックに書き込む必要があることを示す信号です。data_in[] から読み出されたビット数は、パラメータのタイプによって異なります。信号は立ち上がりクロック・エッジでサンプリングされます。パラメータが後続のクロック・サイクルで再び書き込まれるのを防ぐために、信号は1クロック・サイクルの間だけアサートされる必要があります。busy 信号は、write_param がアクティブとして読み出されると直ちにアクティブになります。パラメータが書き込まれている間、busy 信号はアサートされたままで、data_in[] への入力は無視されます。busy 信号が非アクティブになると、別のパラメータを書き込むことができます。パラメータはアプリケーション・コンフィギュレーション・モードでは書き込むことができないため、この信号はファクトリ・コンフィギュレーション・モードの場合にのみ有効です。この信号は、ローカル・アップデート・モードでは使用できません。
reconfig	○	現在のパラメータを使用してリコンフィギュレーションの開始時を示す	デバイスのリコンフィギュレーションは、現在のパラメータ設定を使用して開始すべきであることを示す信号です。値が1の場合は、リコンフィギュレーションが開始することを示します。リコンフィギュレーションが開始される前にすべてのパラメータが完全に書き込まれるように、busy がアサートされている間、この信号は無視されます。
reset_timer	×	ウォッチドッグ・タイマのリセット信号	内部ウォッチドッグ・タイマがリセットされるべきであることを示す信号です。この信号は他の入力と異なり、busy 信号の影響を受けないため、busy がアサートされてもタイマをリセットできます。この信号は、ローカル・アップデート・モードでは使用できません。このパラメータのタイミング規格については、特定のデバイスのハンドブックを参照してください。
read_source	○	パラメータ値を現在の状態から読み出すか、以前の状態から読み出すかを指定。Cyclone III デバイスでのみ使用可能	Cyclone III デバイスでのみ、この2ビット・ポートは、パラメータ値の読み出し元の状態を指定します。この信号は、read_param 信号が有効な場合にのみ有効です。 選択されたソースへの read_source[1..0] のマッピングは以下のように定義されています。 <ul style="list-style-type: none"> ■ 00 - ステータス・レジスタ内の現在の状態の内容 ■ 01 - ステータス・レジスタ内の以前の状態のレジスタ 1 の内容 ■ 10 - ステータス・レジスタ内の以前の状態のレジスタ 2 の内容 ■ 11 - 入力レジスタの値 詳しくは、「Cyclone III デバイス・ハンドブック」の「Cyclone III デバイスのリモート・システム・アップグレード」の章を参照してください。

表 3-2 に、AL TREMOTE_UPDATE メガファンクションの出力ポートを示します。

表 3-2. AL TREMOTE_UPDATE メガファンクション出力ポート

ポート名	必須	説明	備考
data_out[]	×	パラメータ読み出し時のデータ出力	<p>このバスは、RU ブロックからのリード・パラメータ・データを保持します。param[] 値は読み出すパラメータを指定します。read_param 信号がアサートされると、パラメータ値はロードされ、このバス上でドライブされます。データは、busy 信号がデアサートされると有効になります。このポートが非接続のままの場合、このポートのデフォルト値は 000 です。このバスの幅はデバイスに依存します。</p> <ul style="list-style-type: none"> ■ Arria GX、Stratix II、Stratix II GX、Stratix、および Stratix GX デバイスの場合、12 ビット・バス ■ Stratix III デバイスの場合、24 ビット・バス ■ Cyclone III デバイスの場合、29 ビット・バス
busy	×	リモート・アップデート・ブロックがデータの読み出し中または書き込み中であることを示すビジー信号	<p>パラメータを読み出しているか、または書き込んでいるため、マシンがビジーであることを示す信号です。この信号がアサートされている間は、マシンは入力のほとんどを無視するため、この信号がデアサートされるまで変更はできません。したがって、マシンがビジーでないときのみ、変更が可能です。この信号は、read_param または write_param がアサートされると High になり、読み出しまたは書き込み動作が完了するまで High のままです。</p>
pgmout[]	○	リモート・アップデート・ブロックがリコンフィギュレーションされると、コンフィギュレーション・データのどのページ・アドレスがロードされるかを指定	<p>Arria GX、Stratix II、Stratix II GX、および Stratix デバイスでは、3 ビット・バスで、出力ピンに直接接続される必要があります。このバスは、デバイスがリコンフィギュレーションされる時にロードされるコンフィギュレーション・データのページ・アドレス (000 ~ 111) を保持します。異なるコンフィギュレーションを保持している外部デバイスがこのバスを読み出します。</p> <p>Arria GX、Stratix II、および Stratix II GX デバイスでは、ACTIVE_SERIAL_REMOTE が選択されている場合、7 PGM[6:0] ビットが代わりに使用されます。このビットは、通常のリモートおよびローカル・アップデート・モードの場合と同様に、ページ・モード・ピン PGM[2:0] 経由でフラッシュ・メモリに送信されません (ブロックには pgmout 出力ピンがありません)。代わりに、ASMI コントローラが読み出し命令を発行して、フラッシュ・メモリにシフトすると、ビットは ASMI コントローラの命令レジスタにロードされます。これらのビットは、フラッシュ・メモリのアドレスの add[22:16]bits にマッピングされます。</p> <p>このポートは、Stratix III および Cyclone III デバイスにはありません。</p>

表 3-3 に、Cyclone III デバイスでの ALTREMOTE_UPDATE 動作とそれらを指定するための設定を示します。この表では、read_source ポートの使用方法を説明します。

表 3-3. Cyclone III ALTREMOTE_UPDATE の動作 (1/2)

ポート				data_in Width (ビット)	Cyclone III リモート・ アップデート動作	data_out Width (ビット)	データ型	MSM モード
read_param	write_param	read_source	param					
1	0	[00]	[000]	—	マスタ・ステート・マシンの現在の状態モードを読み出す。	2	モード	ファクトリ／アプリケーション
1	0	[00]	[100]	—	現在のファクトリ・モード boot_address を読み出す。	24	アドレス	ファクトリ
1	0	[01]	[000]	—	以前のステータス 1 MSM モードを読み出す。	2	モード	ファクトリ
1	0	[01]	[010]	—	現在のアプリケーション・モードのウォッチドッグ・タイムアウト値を読み出す。	29	タイマ値	アプリケーション
1	0	[01]	[011]	—	現在のアプリケーション・モードのウォッチドッグ・イネーブルを読み出す。	1	フラグ	アプリケーション
1	0	[01]	[100]	—	以前のステータス 1 boot_address を読み出す。	24	アドレス	ファクトリ
1	0	[01]	[111]	—	以前のステータス 1 リコンフィギュレーション・トリガ条件のソースを読み出す。	5	リコンフィギュレーション・トリガの原因	ファクトリ
1	0	[10]	[000]	—	以前のステータス 2 MSM モードを読み出す。	2	モード	ファクトリ
1	0	[10]	[100]	—	以前のステータス 2 boot_address を読み出す。	24	アドレス	ファクトリ
1	0	[10]	[111]	—	以前のステータス 2 リコンフィギュレーション・トリガ条件のソースを読み出す。	5	リコンフィギュレーション・トリガの原因	ファクトリ
0	1	[00]	[001]	1	早期 CONF_DONE チェック (Cd_early) オプション・ビットを書き込む。	—	フラグ	ファクトリ
0	1	[00]	[010]	12 (29 ビットのウォッチドッグ・タイムアウト値の上位 12 ビット)	ウォッチドッグ・タイムアウト値を書き込む。	—	タイマ設定値	ファクトリ

表 3-3. Cyclone III ALTREMOTE_UPDATE の動作 (2/2)

ポート				data_in Width (ビット)	Cyclone III リモート・ アップデート動作	data_out Width (ビット)	データ型	MSM モード
read_param	write_param	read_source	param					
0	1	[00]	[011]	1	ウォッチドッグ・イネーブル を書き込む。	—	フラグ	ファクトリ
0	1	[00]	[100]	22 (24 ビットのア ドレス値の 上位 22 ビット)	boot_address を書き込む。	—	アドレス	ファクトリ
0	1	[00]	[110]	1	内部オシレータが開始ステ ート・マシン・クロックとし ての機能を許可する (Osc_int) オプション・ ビットを書き込む。	—	フラグ	ファクトリ
1	0	[11] (1)	[001]	—	入力レジスタから、早期 CONF_DONE チェック (Cd_early) オプション・ ビットを読み出す。	1	フラグ	ファクトリ
1	0	[11] (1)	[010]	—	入力レジスタからウォッチ ドッグ・タイムアウト値を 読み出す。	12	タイマ 設定値	ファクトリ
1	0	[11] (1)	[011]	—	入力レジスタからウォッチ ドッグ・イネーブル値を 読み出す。	1	フラグ	ファクトリ
1	0	[11] (1)	[100]	—	入力レジスタから boot_address 値を 読み出す。	22	アドレス	ファクトリ
1	0	[11] (1)	[110]	—	入力レジスタから Osc_int オプション・ビットを 読み出す。	1	フラグ	ファクトリ

表 3-3 の注：

(1) Quartus II 開発ソフトウェア v7.2 以降のバージョンでのみ有効。

改訂履歴

以下の表にこのユーザーガイドの章の改訂履歴を示します。

日付	ドキュメント・バージョン	変更内容
2009年4月	2.4	Quartus® II 開発ソフトウェア v9.0 用に更新。 <ul style="list-style-type: none"> ■ サポートされるデバイス・ファミリセクションを更新。 ■ サポートされるデバイスの最高クロック周波数 (f_{MAX}) を追加。 ■ 表 1-1 および表 3-3 を更新。
2007年5月	2.3	Quartus II 開発ソフトウェア v7.1 用に更新。 <ul style="list-style-type: none"> ■ Arria® GX デバイスのサポートを追加して更新。 ■ Cyclone® III デバイス情報を追加して更新。 ■ 「参考資料」のセクションを追加。
2007年3月	2.2	第1章に Cyclone III サポートを追加して更新。
2006年12月	2.1	第1章に Stratix® III サポートを追加して更新。
2006年9月	2.0	Quartus II 開発ソフトウェア v6.0 用の一般的な更新 (スクリーンショット、Modelsim®-Altera シミュレーション・ツールのセクションを第3章に追加など)。
2005年3月	1.0	初版

参考資料

本ユーザーガイドは以下のドキュメントを参照しています。

- 「Arria GX デバイス・ハンドブック」の「[Configuration and Remote System Upgrades](#)」セクション
- 「Arria II GX デバイス・ハンドブック」の「[Configuration, Design Security, and Remote System Upgrades](#)」セクション
- 「Stratix IV デバイス・ハンドブック」の「[Configuration, Design Security, Remote System Upgrades](#)」セクション
- 「Stratix III デバイス・ハンドブック」の「[Hot Socketing, Configuration, Remote Upgrades, and Testing](#)」セクション
- 「Stratix II デバイス・ハンドブック」の「[Configuration and Remote System Upgrades](#)」セクション
- 「Stratix II GX デバイス・ハンドブック」の「[Configuration and Remote System Upgrades](#)」セクション

- 「Stratix デバイス・ハンドブック」の「[Configuration and Remote System Upgrades](#)」セクション
- 「Stratix GX デバイス・ハンドブック」の「[Configuration and Remote System Upgrades](#)」セクション
- 「Cyclone III デバイス・ハンドブック」の「[コンフィギュレーション、ホットソケット、リモート・アップグレード、および SEU の緩和](#)」セクション
- 「Quartus II ハンドブック Volume 1」の「[推奨される HDL コーディング構文](#)」の章
- 「Quartus II ハンドブック Volume 1」の「[Synthesis](#)」セクション
- 「Quartus II ハンドブック Volume 1」の「[Quartus II インテグレートッド・シンセシス](#)」の章
- 「Quartus II ハンドブック Volume 3」の「[Simulation](#)」セクション
- 「Quartus II ハンドブック Volume 3」の「[SignalTap II エンベデッド・ロジック・アナライザを使用したデザインのデバッグ](#)」の章