



# CIC MegaCore ファンクション ユーザーガイド

---



101 Innovation Drive  
San Jose, CA 95134  
[www.altera.com](http://www.altera.com)

ソフトウェア・バージョン : 11.0  
ドキュメント・デート : 2011年5月

Copyright © 2011 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.

|   |        |
|---|--------|
| <b>第 1 章 . この MegaCore ファンクションについて</b>            |        |
| 機能 .....  | 1-1    |
| リリース情報 .....                                      | 1-2    |
| デバイス・ファミリのサポート .....                              | 1-2    |
| MegaCore 検証 .....                                 | 1-3    |
| パフォーマンスおよびリソース使用率 .....                           | 1-4    |
| インストールおよびライセンス .....                              | 1-5    |
| OpenCore Plus 評価機能 .....                          | 1-6    |
| OpenCore Plus タイム・アウト動作 .....                     | 1-6    |
| <b>第 2 章 . 使用法</b>                                |        |
| デザイン・フロー .....                                    | 2-1    |
| DSP Builder フロー .....                             | 2-1    |
| MegaWizard Plug-In Manager フロー .....              | 2-2    |
| 生成されるファイル .....                                   | 2-6    |
| デザインのシミュレーション .....                               | 2-7    |
| NativeLink を使用したサードパーティ・シミュレーション・ツールのシミュレーション ... | 2-7    |
| デザインのコンパイルおよびデバイスのプログラム .....                     | 2-8    |
| <b>第 3 章 . パラメータの設定</b>                           |        |
| パラメータ設定の例 .....                                   | 3-1    |
| パラメータの説明 .....                                    | 3-4    |
| <b>第 4 章 . 機能の説明</b>                              |        |
| カスケード・インテグレータ・コーム (CIC) フィルタ .....                | 4-1    |
| 標準的な周波数応答 .....                                   | 4-2    |
| データ・ストレージ .....                                   | 4-3    |
| 出力のオプション .....                                    | 4-3    |
| 出力データ幅 .....                                      | 4-3    |
| 丸め処理の出力 .....                                     | 4-4    |
| Hogenauer プルーニング .....                            | 4-4    |
| マルチチャンネル・サポート .....                               | 4-4    |
| 多重入力・シングル出力 (MISO) .....                          | 4-5    |
| シングル入力・多重出力 (SIMO) .....                          | 4-6    |
| デシメーションおよび補間に対する可変レート変換率 .....                    | 4-7    |
| FIR フィルタの補償係数 .....                               | 4-8    |
| Avalon Streaming インタフェース .....                    | 4-9    |
| Avalon-ST インタフェースのデータ転送のタイミング .....               | 4-11   |
| パケット・データ転送 .....                                  | 4-11   |
| 信号 .....  | 4-12   |
| 参考資料 .....  | 4-13   |
| <b>追加情報</b>                                       |        |
| 改訂履歴 .....  | Info-1 |
| アルテラへのお問い合わせ .....                                | Info-1 |
| 表記規則 .....  | Info-2 |



本章では、Altera® CIC MegaCore® ファンクションについて説明します。アルテラの CIC MegaCore ファンクションは、Avalon® Streaming (Avalon-ST) インタフェースと互換性があるデータ・ポートを持っているカスケード積分櫛形 (CIC) のフィルタを実装します。Hogenauer フィルタとしても知られる、CIC フィルタは、デシメーションによる狭帯域ソースからのベースバンド信号の抽出において、また補間による処理済ベースバンド信号からの狭帯域信号の構築において、計算効率の良いフィルタです。

CIC フィルタは、加算器とレジスタのみを使用し、大きなレート変換を処理するための乗算器は必要としません。したがって、CIC はハードウェア実装に適した経済的なフィルタ・アーキテクチャで、デジタル・ダウン・コンバータ (DDC) およびデジタル・アップ・コンバータ (DUC) のようなサンプル・レート変換デザインで広く使用されます。

詳細については、[4-1 ページの「カスケード・インテグレータ・コーム \(CIC\) フィルタ」](#)を参照してください。

## 機能

アルテラ CIC MegaCore ファンクションは、以下のような機能もサポートしています。

- 可変レート変換率 (2 ~ 32,000)、ステージのコンフィギュレーション可能な数 (1 ~ 12)、および 2 つの差動遅延オプション (1 または 2) を持つ補間およびデシメーション・フィルタのサポート。
- 選択可能なインタフェースの数および 1,024 チャンネルの最大値を持つシングル・クロック・ドメイン。
- パイプライン化された積分器を使用するオプションを持つ選択可能なデータ・ストレージ・オプション。
- 設定可能な入力データの幅 (1 ~ 32 ビット) および出力データの幅 (1 ~ フル解像度のデータ幅)。
- 選択可能な出力の丸めモード (切り捨て、収束丸め、丸め、または飽和) および Hogenauer のブルーニングのサポート。
- 積分器ごとによって使用されるパイプラインのステージを指定することにより、速度のための最適化。
- 補償フィルタ係数の生成。
- パラメータ化およびハードウェアの生成のための使いやすい MegaWizard™ インタフェース。
- アルテラがサポートしている VHDL および Verilog HDL シミュレータ上で使用可能な IP ファンクション・シミュレーション・モデル。
- DSP Builder Ready。

## リリース情報

表 1-1 に、アルテラの CIC MegaCore ファンクションのこのリリースに関する情報を示します。

表 1-1. CIC MegaCore ファンクションのリリース情報

| 項目       | 説明         |
|----------|------------|
| バージョン    | 11.0       |
| リリース・デート | 2011 年 5 月 |
| 製品コード    | IP-CIC     |
| プロダクト ID | 00BB       |
| ベンダ ID   | 6AF7       |

 このリリースについて詳しくは、「[MegaCore IP Library Release Notes and Errata](#)」を参照してください。

アルテラは Quartus® II ソフトウェアの現行バージョンが、各 MegaCore ファンクションの前のバージョンをコンパイルできることを検証します。「[MegaCore IP Library Release Notes and Errata](#)」は、この検証に例外がある場合に報告します。アルテラは、1 リリースより前の MegaCore ファンクション・バージョンのコンパイルは検証しません。

## デバイス・ファミリのサポート

表 1-2 に、アルテラの IP コアのデバイス・サポートのレベルを定義します。

表 1-2. アルテラの IP コアのデバイス・サポート・レベル

| FPGA デバイス・ファミリ  | HardCopy デバイス・ファミリ   |
|---|--|
| <b>暫定サポート</b> —IP コアはこのデバイス・ファミリのための暫定タイミング・モデルと検証されています。IP コアがデバイス・ファミリの機能要件はすべて満たしているが、タイミング要件については評価中です。製造デザインでの使用は注意が必要なことを意味します。 | <b>HardCopy コンパニオン</b> —IP コアは HardCopy コンパニオン・デバイスのための暫定タイミング・モデルと検証されています。IP コアが HardCopy デバイス・ファミリの機能要件はすべて満たしているが、タイミング要件については評価中です。製造デザインでの使用は注意が必要なことを意味します。 |
| <b>最終的なサポート</b> —IP コアはこのデバイス・ファミリのための最終的なタイミング・モデルと検証されています。IP コアがデバイス・ファミリの機能要求およびタイミング要求をすべて満たしており、製造デザインとして使用可能であることを意味します。       | <b>HardCopy コンパイル</b> —IP コアは HardCopy デバイス・ファミリのための最終的なタイミング・モデルと検証されています。IP コアがデバイス・ファミリの機能要求およびタイミング要求をすべて満たしており、製造デザインとして使用可能であることを意味します。                      |

表 1-3 に、CIC MegaCore ファンクションによる各アルテラ・デバイス・ファミリへのサポートのレベルを示します。

表 1-3. サポートされるデバイス・ファミリ (その 1)

| デバイス・ファミリ   | サポート |
|-------------|------|
| Arria® GX   | 最終的  |
| Arria II GX | 最終的  |

表 1-3. サポートされるデバイス・ファミリ (その 2)

| デバイス・ファミリ       | サポート           |
|-----------------|----------------|
| Arria II GZ     | 最終的            |
| Cyclone®        | 最終的            |
| Cyclone II      | 最終的            |
| Cyclone III     | 最終的            |
| Cyclone III LS  | 最終的            |
| Cyclone IV GX   | 最終的            |
| HardCopy® II    | HardCopy コンパイル |
| HardCopy III    | HardCopy コンパイル |
| HardCopy IV E   | HardCopy コンパイル |
| HardCopy IV GX  | HardCopy コンパイル |
| Stratix®        | 最終的            |
| Stratix II      | 最終的            |
| Stratix II GX   | 最終的            |
| Stratix III     | 最終的            |
| Stratix IV GT   | 最終的            |
| Stratix IV GX/E | 最終的            |
| Stratix V       | 暫定サポート         |
| Stratix GX      | 最終的            |
| その他のデバイス・ファミリ   | サポートなし         |

## MegaCore 検証

CIC MegaCore ファンクションの更新バージョンをリリースする前に、アルテラは品質や正確性を検証するための包括的な回帰テストを実施します。

CIC MegaCore ファンクションのカスタム・バリエーションは、さまざまなパラメータのオプションを行使するために生成され、結果として得られるシミュレーション・モデルは、マスタのシミュレーション・モデルに対して検証結果を徹底的にシミュレートされます。

## パフォーマンスおよびリソース使用率

ここでは、Cyclone III および Stratix IV と共に Quartus II ソフトウェアを使用し、CIC MegaCore ファンクションの標準的な期待パフォーマンスを示します。

次の一般的な設定は、すべてのパラメータ化に適用されます。

- 段数 : 5
- レート変更係数 : 8
- 遅延差 : 1
- 積分器のデータ・ストレージ : Memory (可能な限り)
- 微分器のデータ・ストレージ : Memory (可能な限り)
- 入力データ幅 : 16
- 出力データ幅 : 16 (表で特に記述がない限り)
- 出力の丸め処理 : Truncation



Cyclone III デバイスはルックアップ・テーブル (LUT) とロジック・レジスタを使用し、Stratix IV デバイスはアダプティブ・ルックアップ・テーブル (ALUT) とロジック・レジスタを使用しています。

表 1-4 に、ターゲット  $f_{MAX}$  を 1GHz に設定した場合の CIC MegaCore ファンクションのパフォーマンスを示します。

表 1-4. CIC MegaCore ファンクション・パフォーマンス (その 1)

| デバイス・ファミリー   | 組み合わせ LUT/ALUT 数 | ロジック・レジスタ | メモリ (3) |     |        | $f_{MAX}$ (MHz) |
|--|------------------|-----------|---------|-----|--------|-----------------|
|  |                  |           | ビット数    | M9K | ALUT 数 |                 |
| デシメータ、フル出力解像度 (出力データ幅 = 31) インタフェースごとに 1 チャンネル、1 インタフェース           |                  |           |         |     |        |                 |
| Cyclone III (1)  | 480              | 806       | —       | —   | —      | 288             |
| Stratix IV (2)   | 391              | 725       | 128     | —   | 16     | 502             |
| デシメータ、Hogenauer プルーニング・オン、出力データ幅 = 16、インタフェースごとに 1 チャンネル、1 インタフェース |                  |           |         |     |        |                 |
| Cyclone III (1)  | 388              | 606       | —       | —   | —      | 312             |
| Stratix IV (2)   | 314              | 525       | 128     | —   | 16     | 526             |
| 補間器、フル出力解像度 (出力データ幅 = 29)、インタフェースごとに 1 チャンネル、1 インタフェース             |                  |           |         |     |        |                 |
| Cyclone III (1)  | 401              | 582       | —       | —   | —      | 304             |
| Stratix IV (2)   | 297              | 501       | 128     | —   | 16     | 509             |
| 補間器、収束丸め、インタフェースごとに 1 チャンネル、1 インタフェース                              |                  |           |         |     |        |                 |
| Cyclone III (1)  | 408              | 560       | —       | —   | —      | 292             |
| Stratix IV (2)   | 315              | 479       | 128     | —   | 16     | 470             |
| デシメータ、Hogenauer プルーニング・オン、インタフェースごとに 5 チャンネル、1 インタフェース             |                  |           |         |     |        |                 |
| Cyclone III (1)  | 763              | 893       | 875     | 11  | —      | 268             |
| Stratix IV (2)   | 537              | 952       | 1,003   | 8   | 74     | 492             |
| デシメータ、Hogenauer プルーニング・オン、インタフェースごとに 5 チャンネル、3 インタフェース             |                  |           |         |     |        |                 |
| Cyclone III (1)  | 1,357            | 1,411     | 3,211   | 21  | —      | 261             |



表 1-4. CIC MegaCore ファンクション・パフォーマンス (その 2)

| デバイス・ファミリー   | 組み合わせ LUT/ALUT 数 | ロジック・レジスタ | メモリ (3) |     |        | f <sub>MAX</sub> (MHz) |
|--|------------------|-----------|---------|-----|--------|------------------------|
|  |                  |           | ビット数    | M9K | ALUT 数 |                        |
| Stratix IV (2)   | 890              | 1,497     | 3,211   | 24  | 12     | 459                    |
| デシメータ、Hogenauer プルーニング・オフ、切り捨て、インタフェースごとに 1 チャネル、1 インタフェース |                  |           |         |     |        |                        |
| Cyclone III (1)  | 465              | 746       | —       | —   | —      | 302                    |
| Stratix IV (2)   | 391              | 665       | 128     | —   | 16     | 509                    |
| 補間器、切り捨て、インタフェースごとに 5 チャネル、1 インタフェース                       |                  |           |         |     |        |                        |
| Cyclone III (1)  | 738              | 847       | 660     | 10  | —      | 283                    |
| Stratix IV (2)   | 512              | 938       | 788     | 6   | 90     | 486                    |
| 補間器、切り捨て、インタフェースごとに 5 チャネル、3 インタフェース                       |                  |           |         |     |        |                        |
| Cyclone III (1)  | 1,501            | 1,657     | 2,426   | 17  | —      | 188                    |
| Stratix IV (2)   | 913              | 1,753     | 2,426   | 20  | 16     | 345                    |

表 1-4 の注 :

- (1) EP3C10F256C6 デバイスを使用します。
- (2) EP4SGX70DF29C2X デバイスを使用します。
- (3) ターゲット f<sub>MAX</sub> をより低く設定することで、メモリ使用率を大幅に低減できる場合があります。

## インストールおよびライセンス

CIC MegaCore ファンクションは、MegaCore IP ライブラリの一部であり、Quartus® II ソフトウェアと共に配布されます。また、アルテラのウェブサイト ([www.altera.co.jp](http://www.altera.co.jp)) からダウンロードすることもできます。


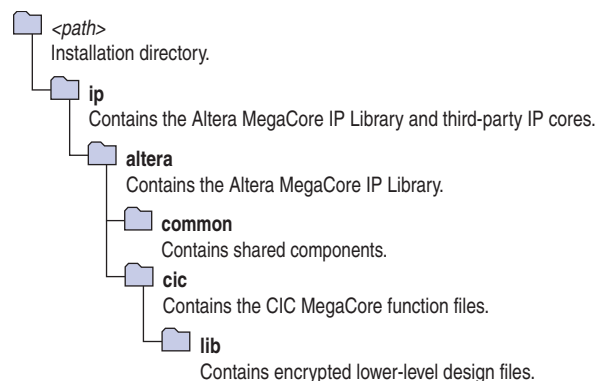
 システム要求およびインストールの手順については、「アルテラ・ソフトウェアのインストールおよびライセンス」のマニュアルを参照してください。

図 1-1 に、CIC MegaCore ファンクションをインストールした後のディレクトリ構造を示します。この場合、<path> が Quartus II ソフトウェアのインストール・ディレクトリです。

図 1-1. ディレクトリ構造



Windows でのデフォルトのインストール・ディレクトリは、**c:\altera\<version>** です。  
Linux では、**/opt/altera<version>** です。

## OpenCore Plus 評価機能

アルテラの無償 OpenCore Plus 評価機能により、以下の処理を実行できます。

- 作成したシステム内のメガファンクション（アルテラ MegaCore ファンクションまたは AMPP<sup>SM</sup> メガファンクション）の動作をシミュレーションする。
- デザインの機能を検証したり、サイズやスピードを迅速かつ簡単に評価したりする。
- メガファンクションを含むデザインに対し、実行時間に制限のあるデバイス・プログラミング・ファイルを生成する。
- デバイスをプログラムし、デザインを実機上で検証する。

MegaCore を製品に組み込む場合にのみ、ライセンスを購入していただく必要があります。

ライセンス購入後は、アルテラ・ウェブサイト（[www.altera.co.jp/licensing](http://www.altera.co.jp/licensing)）からライセンス・ファイルを要求して、コンピュータにインストールできます。ライセンス・ファイルを要求すると、アルテラから電子メールで **license.dat** ファイルが送信されます。インターネットをご利用いただけないお客様は、アルテラの販売代理店にお問い合わせください。



OpenCore Plus ハードウェア評価について詳しくは、「[AN320: OpenCore Plus 評価機能によるメガファンクションの評価](#)」を参照してください。

## OpenCore Plus タイム・アウト動作

OpenCore Plus ハードウェア評価機能は、以下の動作モードでメガファンクションの実機評価をサポートします。

- *Untethered*（アンテザード） — デザインは限定時間のみ実行されます。
- *Tethered*（テザード） — ボードとホスト・コンピュータ間に接続が必要です。デザイン内のすべてのメガファンクションが *Tethered* モードをサポートしている場合、デバイスはより長時間または無制限に動作できます。

OpenCore Plus 評価機能モードで動作中のメガファンクションのうちの1つでも評価時間に達すると、デバイス内のすべてのメガファンクションが同時にタイム・アウトします。デザイン内に複数のメガファンクションがある場合、特定のメガファンクションのタイム・アウト動作は、他のメガファンクションのタイム・アウト動作によってマスクされることがあります。

CIC MegaCore ファンクションの場合、アンテザード・タイム・アウトは1時間、テザード・タイム・アウト値は無制限です。

データ出力信号はハードウェア評価が期限切れになると、0に強制されます。

## デザイン・フロー

CIC MegaCore® ファンクションは、以下のデザイン・フローをサポートします。

- **DSP Builder**: CIC MegaCore ファンクションのバリエーションを含む DSP Builder モデルを作成する場合、このフローを使用してください。
- **MegaWizard™ Plug-In Manager**: ユーザーのデザインで、手動でインスタンス化できる CIC MegaCore ファンクションのバリエーションを作成する場合、このフローを使用してください。

この章では、これらのフローのいずれかの CIC MegaCore ファンクションを使用する方法について説明します。パラメータ化は、各フローに同じであり、[第 3 章のパラメータの設定](#)で説明されています。

これらのフローのいずれかのデザインをパラメータ化およびシミュレートした後、Quartus II ソフトウェアで完成したデザインをコンパイルすることができます。

## DSP Builder フロー

アルテラの DSP Builder 製品は、設計者がアルゴリズム開発の環境下で行った DSP デザインをハードウェアで実現するための支援を行い、デジタル信号処理 (DSP) のデザイン・サイクルを短縮します。


DSP Builder は、MathWorks 社の MATLAB® および Simulink® システム・レベルのデザイン・ツールのアルゴリズム開発、シミュレーション、および検証の機能と、アルテラの Quartus II 開発ソフトウェアおよびサード・パーティの合成およびシミュレーション・ツールを組み合わせます。既存の Simulink ブロックをアルテラの DSP Builder ブロックおよび MegaCore ファンクションのバリエーション・ブロックと組み合わせ、システム・レベル仕様を検証し、シミュレーションを実行できます。


DSP Builder の中で、MegaCore ファンクションの Simulink シンボルが、Altera DSP Builder Blockset から MegaCore Functions ライブラリの Simulink ライブラリ・ブラウザに表示されます。

次の手順を実行して、MATLAB/ Simulink の環境で CIC MegaCore ファンクションを使用することができます。


1. 新規 Simulink モデルを作成します。
2. Simulink ライブラリ・ブラウザで、**MegaCore Functions** ライブラリから `cic_<version>` ブロックを選択し、ユーザーのモデルに追加し、ブロックに一意的な名前を付けます。
3. モデル内の `cic_<version>` ブロックをダブル・クリックして、MegaWizard インタフェースを表し、MegaCore ファンクションのバリエーションをパラメータ化します。CIC MegaCore ファンクションのパラメータを設定する方法の例について、[第 3 章のパラメータの設定](#)を参照してください。
4. MegaWizard インタフェースで **Finish** をクリックして、パラメータ化を完了し、CIC MegaCore ファンクションのバリエーションを生成します。生成されたファイルについて、[2-6 ページの表 2-1](#)を参照してください。

5. CIC MegaCore ファンクションのバリエーションはモデル内の他のブロックに接続します。
6. DSP Builder モデルの CIC MegaCore ファンクションのバリエーションをシミュレートします。

 DSP Builder のフローについて詳しくは、『*DSP Builder User Guide*』の『*Using MegaCore Functions*』の章を参照してください。

 DSP Builder のフローを使用している場合、デバイスの選択、シミュレーション、Quartus II のコンパイルおよびデバイスのプログラミングは DSP Builder 環境内ですべて制御されます。

DSP Builder は、Avalon® Memory-Mapped (Avalon-MM) マスタ/スレーブ、および Avalon ストリーミング (Avalon-ST) ソース/シンク・インタフェースを使用して、SOPC Builder での統合をサポートします。

 これらのインタフェース・タイプについて詳しくは、『*Avalon Interface Specifications*』を参照してください。

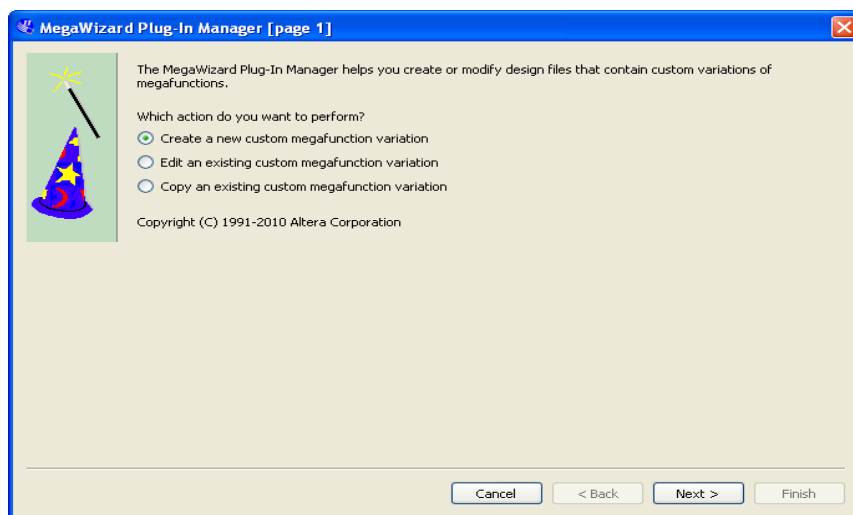
## MegaWizard Plug-In Manager フロー

MegaWizard™ Plug-In Manager フローでは、CIC MegaCore ファンクションをカスタマイズし、手動で MegaCore ファンクションのバリエーションを Quartus II デザインに組み込むことができます。

MegaWizard Plug-in Manager フローを使用するには、以下のステップを実行します。

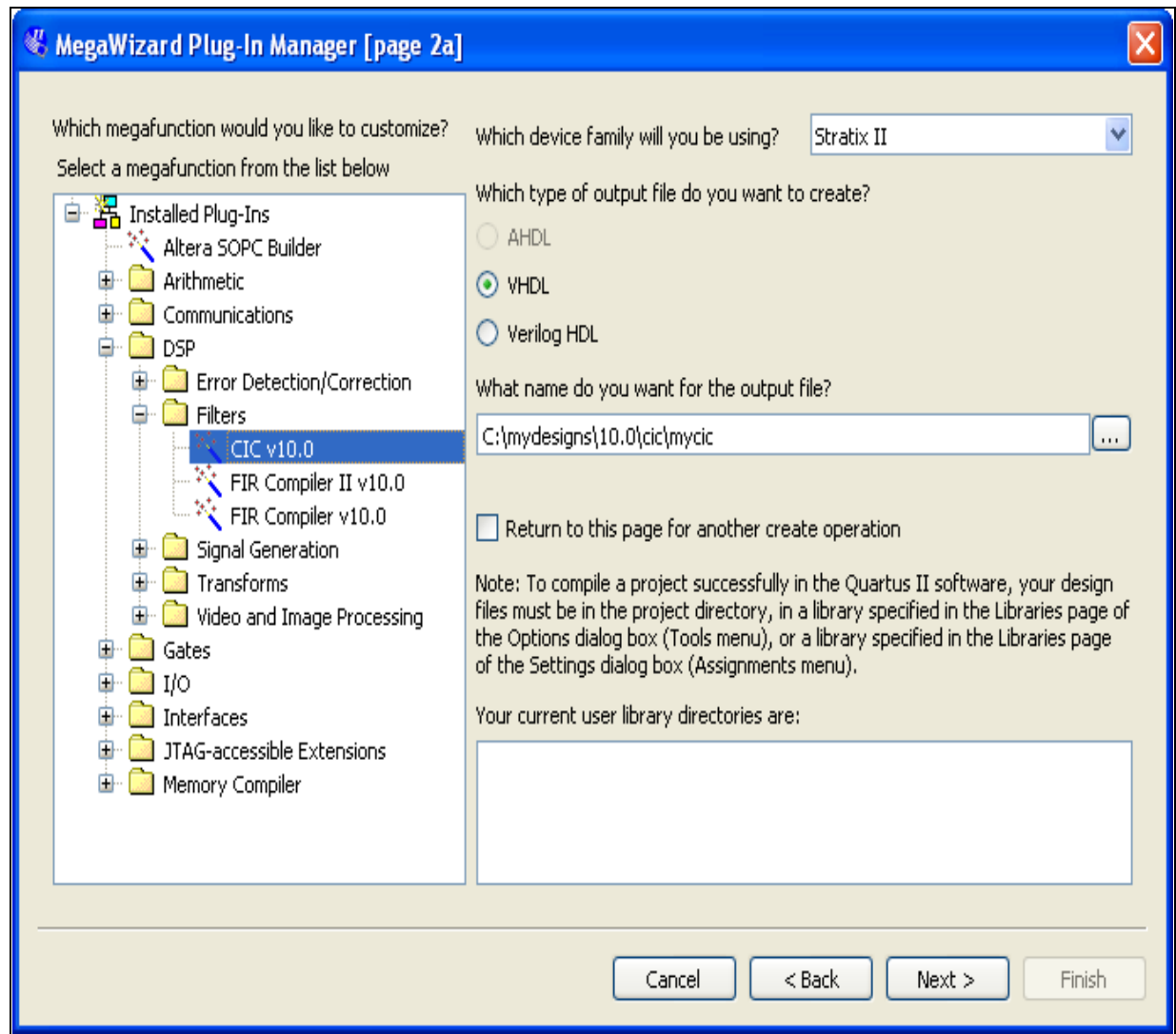
1. Quartus II ソフトウェアの File メニューから **New Project Wizard** を選択して、新規のプロジェクトを作成します。
2. Tools メニューから **MegaWizard Plug-in Manager** を起動し、新規のカスタム・メガファンクション・バリエーションを作成するには、オプションを選択します (図 2-1)。

図 2-1. MegaWizard Plug-In Manager



3. **Next** をクリックして、**Installed Plug-Ins** タブでの **DSP>Filters** セクションから **CIC** <version> を選択します (2-3 ページの 図 2-2)。

図 2-2. Megafunction の選択




4. デバイス・ファミリが **New Project Wizard** で指定したものと同一であることを確認します。
5. デザインのトップ・レベルの出力ファイル・タイプを選択します。ウィザードでは VHDL と Verilog HDL をサポートしています。
6. MegaCore ファンクション・バリエーションのトップ・レベルの出力ファイル名を指定し、**Next** をクリックして、MegaWizard インタフェースの **Parameter Settings** ページを表示します。MegaWizard インタフェースを使用して MegaCore ファンクション・バリエーションの必要なパラメータを指定します。CIC MegaCore ファンクションのパラメータを設定する方法について、第 3 章のパラメータの設定を参照してください。
7. **Next** をクリックして、パラメータ化を完了し、**EDA** ページを表示します (2-4 ページの 図 2-3)。

図 2-3. EDA ページ



8. EDA ページで、**Generate Simulation Model** をオンにします。

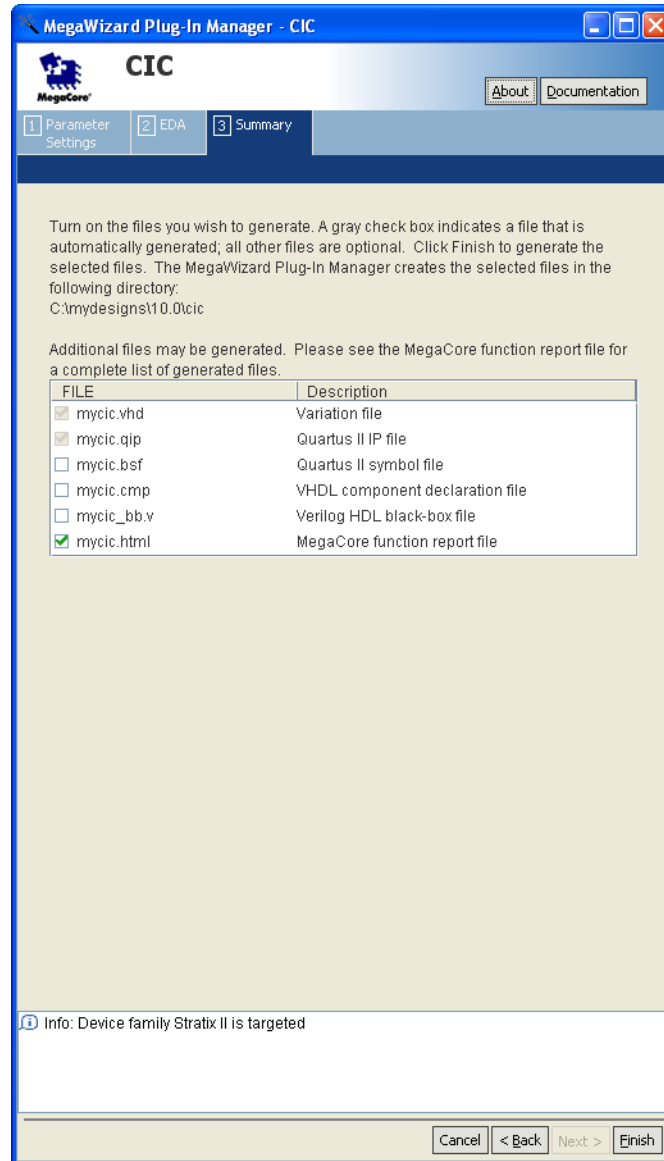
 IP 機能シミュレーション・モデルは、Quartus II ソフトウェアで生成するサイクル精度の正確な VHDL または Verilog HDL モデルです。



これらのシミュレーション・モデルは、シミュレーションの目的にのみ使用し、合成やその他の目的には使用しないでください。これらのモデルを合成に使用すると、動作しないデザインが作成されます。

9. 一部サード・パーティ合成ツールでは、詳細なロジックは含まず MegaCore ファンクションの構造のみを含むネットリストを使用して、MegaCore ファンクションを含むデザインの性能を最適化することができます。合成ツールがこの機能をサポートしている場合は、**Generate netlist** をオンにします。
10. Next をクリックして、**Summary** ページを表示します (図 2-4)。

図 2-4. Summary ページ



11. ファイルを生成する場合、**Summary** タブで、チェック・ボックスをオンにします。グレイのチェック・マークは、自動的に生成されるファイルを示します。その他のファイルはすべてオプションです。
12. **Finish** をクリックして、MegaCore ファンクションおよびサポートするファイルを生成します。ファイル生成フェーズを完了するには、数分かかる場合があります。生成の進捗状況およびステータスはレポート・ウィンドウに表示されます。

13. **Exit** をクリックして、進捗状況のレポート・ウィンドウを閉じます。**Quartus II IP Files** プロンプトで **Yes** をクリックして、現在の Quartus II プロジェクトにカスタム MegaCore ファンクションのバリエーションを説明する **.qip** ファイルを追加します。

 MegaWizard Plug-In Manager について詳しくは、「Quartus II Help」を参照してください。

## 生成されるファイル

表 2-1 に、プロジェクト・ディレクトリに生成されるファイルを示します。

ファイルのタイプおよび名前は、バリエーション名およびパラメータ化で指定する HDL タイプによって異なります。例えば、異なるファイル・セットはデザインを VHDL または Verilog HDL のいずれで作成したかによって作成されます。

表 2-1. 生成されるファイル (注 1) & (注 2)

| ファイル名                               | 説明   |
|-------------------------------------|--|
| <entity name>.ocp                   | 暗号化された OpenCore Plus ファイル。   |
| <entity name>.vhd または .v            | デザイン・エンティティを定義する VHDL または Verilog HDL ファイル。  |
| <variation name>.bsf                | MegaCore ファンクション・バリエーションの Quartus II ブロック・シンボル・ファイル。Quartus II ブロック図エディタでこのファイルを使用できます。  |
| <variation name>.cmp                | MegaCore ファンクション・バリエーション用の VHDL コンポーネント宣言ファイルです。このファイルの内容は、MegaCore ファンクションをインスタンスする VHDL アーキテクチャの 1 つに追加します。(3)   |
| <variation name>.html               | MegaCore ファンクションは、ハイパーテキスト・マークアップ言語 (HTML) (Hypertext markup language) でレポートし、生成されるファイルのリストおよび MegaCore ファンクション・バリエーションのポートを含みます。(3)   |
| <variation name>.qip                | Quartus II コンパイラの MegaCore ファンクション・バリエーションの処理に必要なすべてのアサインメントおよび他の情報が含まれる Quartus II IP ファイルを生成されます。MegaWizard から脱出すると、このファイルを現行の Quartus II プロジェクトに追加するように促されます。                 |
| <variation name>.log                | ログ・ファイル。   |
| <variation name>.vhd または .v         | カスタム MegaCore ファンクションの VHDL または Verilog HDL トップ・レベルの記述を定義する VHDL または Verilog HDL ファイルです。デザイン内部のこのファイルによって定義されたエンティティをインスタンスします。Quartus II ソフトウェアでのデザインのコンパイル時にこのファイルがインクルードされます。 |
| <variation name>.vho または .vo        | IP 機能シミュレーション・モデルの記述を定義する VHDL または Verilog HDL の出力ファイルです。  |
| <variation name>_bb.v               | MegaCore ファンクション・バリエーションの Verilog HDL ブラック・ボックス・ファイルです。サード・パーティ製 EDA ツールを使用してデザインを合成するときにこのファイルを使用します。(3)  |
| <variation name>_fir_comp_coeff.m   | 補償 FIR フィルタの係数を生成する MATLAB スクリプト。  |
| <variation name>_syn.vhd または _syn.v | 一部のサード・パーティ合成ツールで利用できるタイミングおよびリソース見積りネットリストです。(4)  |
| <variation name>_nativelink.tcl     | NativeLink シミュレーション・テストベンチ設定を Quartus II プロジェクトに割り当てるために使用する Tcl スクリプトです。  |



表 2-1. 生成されるファイル (注 1) & (注 2)

| ファイル名                          | 説明   |
|--------------------------------|--|
| <variation name>_quartus.tcl   | Quartus II ソフトウェアにコンパイルを実行するために使用する Tcl スクリプトです。   |
| <variation name>_tb_input.txt  | テストベンチ用の入力データを含むテキスト・ファイルです。   |
| <variation name>_tb.vhd または .v | CIC MegaCore ファンクション・バリエーション用の VHDL または Verilog HDL テストベンチ・ファイル。VHDL トップ・レベルが選択されているときは VHDL ファイルが生成され、Verilog HDL トップ・レベルが選択されているときは Verilog HDL ファイルが生成されます。 |

表 2-1 の注 :

- (1) <variation name> プリフィックスが MegaWizard Plug-In Manager で指定された出力・ファイル名のベースを使用して自動的に追加されます。p
- (2) <entity name> プリフィックスが自動的に追加されます。各 MegaCore インスタンスの VHDL コードは、Finish をクリックしたとき動的に生成され、<entity name> はインスタンスごとに異なります。それは、<variation name> に \_cic を追加して生成されます。
- (3) .cm、\_bb.v および .html ファイルは MegaWizard インタフェースの Summary ページでイネーブルになっている場合のみ生成されます。
- (4) \_syn.vhd または \_syn.v ファイルは MegaWizard インタフェースの EDA ページでイネーブルになっている場合にのみ生成されません。



MegaWizard Summary ページの **MegaCore function report file** のチェック・ボックスを有効にする場合、MegaCore ファンクションのバリエーション用に定義されたデザイン・ファイルおよびポートのリストを含む MegaCore ファンクション・レポート・ファイルは HTML ファイルとして保存されます。

MegaCore ファンクション・バリエーションの外部ポートでサポートされる信号については [4-12 ページ](#) の [表 4-3](#) を参照してください。

## デザインのシミュレーション

MegaWizard 生成された VHDL または Verilog HDL IP 機能シミュレーション・モデルを使用して、デザインをシミュレーションできます。

IP 機能シミュレーション・モデルは、指定した出力言語に応じて、.vo または .vho ファイルのいずれかです。シミュレーション環境で .vo または .vho ファイルをコンパイルして、MegaCore ファンクションのカスタム・バリエーションの機能シミュレーションを実行します。



IP 機能シミュレーション・モデルについて詳しくは、「Quartus II ハンドブック Volume 3」の [「Simulating Altera Designs」](#) の章を参照してください。

### NativeLink を使用したサードパーティ・シミュレーション・ツールのシミュレーション

シミュレーションは、NativeLink を使用して Quartus II ソフトウェアからサード・パーティ製シミュレーション・ツールを使用して実行できます。

Tcl スクリプト・ファイル <variation name>\_nativelink.tcl を使用して、デフォルトの NativeLink テストベンチ設定を Quartus II プロジェクトに割り当てることができます。

Quartus II ソフトウェアで NativeLink を使用してシミュレーションを実行するには、以下のステップを実行します。

1. この章で前述した手順に従って、カスタム MegaCore ファンクションのバリエーションを作成します。バリエーション名が Quartus II プロジェクト名と一致しているかを確認します。
2. Quartus II ソフトウェアの Tools メニューの Options ページに、サード・パーティ製 EDA ツールへの絶対パスが設定されているかを確認します。
3. Processing メニューから **Start** を選択し、**Start Analysis & Elaboration** をクリックします。
4. Tools メニューの **Tcl scripts** をクリックします。 **Tcl Scripts** ダイアログ・ボックスで `<variation name>_nativelink.tcl` の Tcl スクリプトを選択して、**Run** をクリックします。Tcl スクリプトが正常にロードされたことを確認するメッセージをチェックします。
5. Assignments メニューの **Settings** をクリックして、**EDA Tool Settings** を展開し、**Simulation** を選択します。 **Tool Name** でシミュレータを選択し、**NativeLink Settings** で **Compile Test Bench** を選択して、**Test Benches** をクリックします。
6. Tools メニューで **EDA Simulation Tool** をポイントして、**Run EDA RTL Simulation** をクリックします。

Quartus II ソフトウェアは、シミュレータを選択し、アルテラのライブラリ、デザイン・ファイル、およびテストベンチをコンパイルします。テストベンチが実行して、波形ウィンドウには、分析のためのデザイン信号を示します。



詳細は、「Quartus II ハンドブック Volume 3」の「*Simulating Altera Designs*」の章を参照してください。

## デザインのコンパイルおよびデバイスのプログラム

Quartus II ソフトウェアを使用して、デザインをコンパイルすることができます。デザインをコンパイルした後、ターゲットのアルテラ・デバイスをプログラムし、ハードウェア内でデザインを検証できます。




MegaWizard Plug-In Manager および Quartus II ソフトウェアについて詳しくは、「Quartus II Help」を参照してください。

この章では CIC MegaCore ファンクションをパラメータ化する方法の例を提供します。

**Parameter Settings** ページでは、DSP Builder または MegaWizard Plug-In Manager のフローから MegaWizard インタフェースが開いているかどうかの同じオプションを提供します。しかし、DSP Builder フローを使用しているときに、EDA および Summary タブは表示されません。

MegaWizard インタフェースを開く方法について詳しくは、[2-1 ページの「デザイン・フロー」](#)を参照してください。

 MegaWizard インタフェースは、パラメータの適正な組み合わせのみの選択を可能にし、無効なパラメータの組み合わせに対しては警告を出します。


## パラメータ設定の例

CIC MegaCore ファンクションをパラメータ化するには、以下のステップを従います。

1. **Parameter Settings: Architecture** ページで、必要な値を指定します。例えば、[3-2 ページの「Architecture ページ」](#)に、[表 3-1](#)に示されたパラメータ設定を示します。

表 3-1. CIC MegaCore Function Architecture タブのサンプル・パラメータ

| パラメータ                                    | 値     |
|--|-------|
| Target                                   | デシメータ |
| Number of stages 段数                      | 3     |
| Differential delay                       | 1     |
| Rate change factor                       | 4     |
| Enable variable rate change factor       | オフ    |
| Number of interfaces                     | 1     |
| Number of channels per interface         | 7     |
| Integrator data storage                  | メモリ   |
| RAM type of integrator data storage      | AUTO  |
| Differentiator data storage              | メモリ   |
| RAM type of differentiator data storage  | AUTO  |
| Use pipelined integrators                | オン    |
| Number of pipeline stages per integrator | 2     |

 可変レート変換率のオプションをイネーブルする場合、実際のレート変換率を指定する代わりに、最小値と最大値を指定することができます。データ・ストレージ・オプションを使用できるのは、インタフェースあたりのチャンネル数が 4 より大きい場合のみです。

これらのパラメータについて詳しくは、[3-4 ページの「CIC MegaCore Function Architecture ページ」](#)を参照してください。

図 3-1. Architecture ページ

MegaWizard Plug-In Manager - CIC

**CIC**

About Documentation

1 Parameter Settings 2 EDA 3 Summary

Architecture Input/Output Options

Device Family

Target: Stratix II

Filter Specifications

Filter type: Decimator

Number of stages: 3

Differential delay: 1

Rate change factor: 4

Variable Rate Change Factor Options

Enable variable rate change factor

Minimum: 2 Maximum: 3

Multi-channel Options

Number of interfaces: 1

Number of channels per interface: 7

Data Storage Options

Integrator data storage: Memory

RAM type of integrator data storage: AUTO

Differentiator data storage: Memory

RAM type of differentiator data storage: AUTO

Optimize for Speed

Use pipelined integrators

Number of pipeline stages per integrator: 1

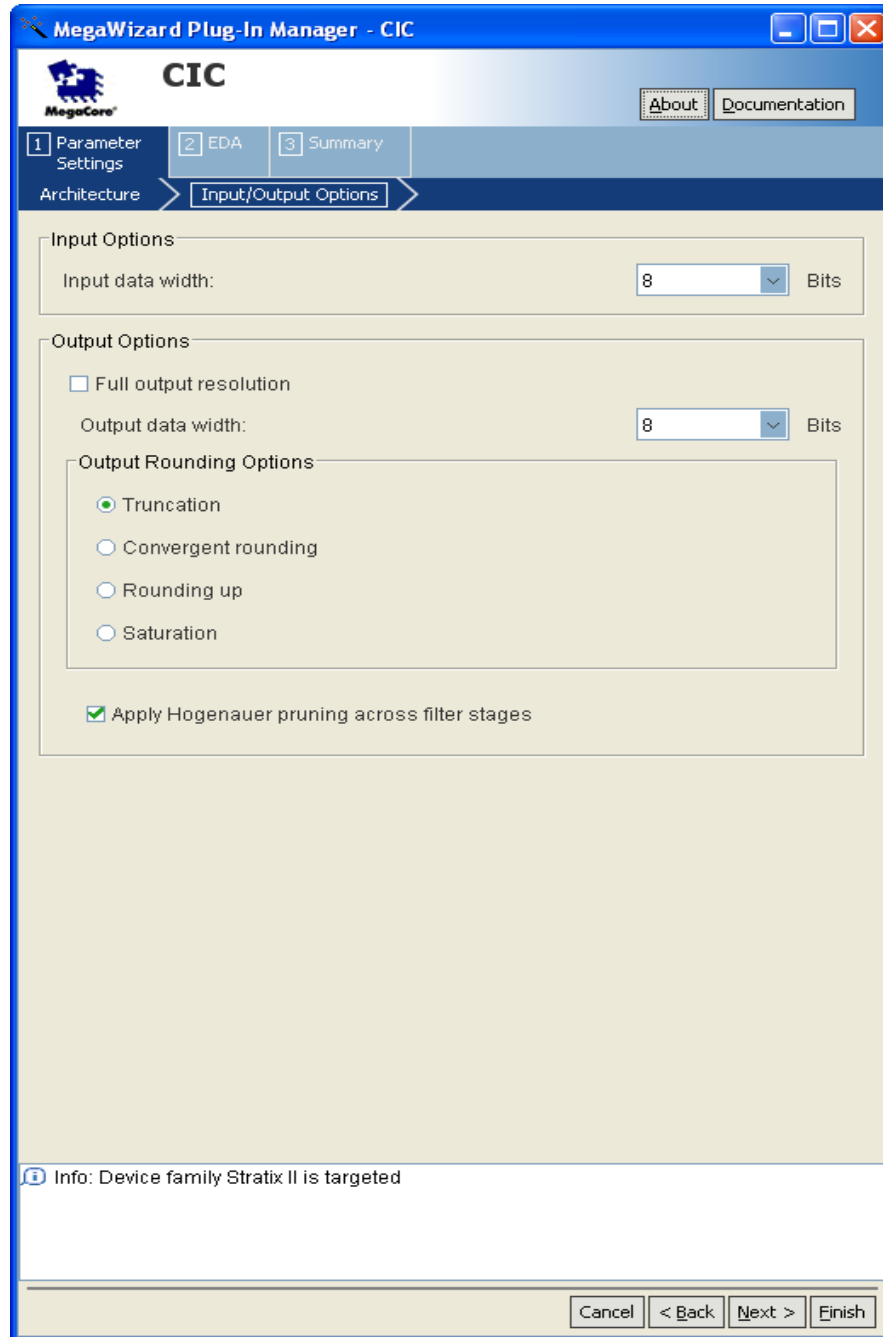
Info: Device family Stratix II is targeted

Cancel < Back Next > Finish

 **Parameter Settings - Architecture** ページは、現在選択されていることを示す読み出し専用フィールドとしてターゲット・デバイス・ファミリーを表示します。ターゲット・デバイスは、DSP Builder フローを使用している場合、Signal Compiler、または MegaWizard Plug-In Manager フローを使用している場合、New Project Wizard で指定することができます。

2. **Next** をクリックして **Parameter Settings: Input/Output Options** ページを表示します (図 3-2)。

図 3-2. Input/Output Options ページ



3. **Parameter Settings: Input/Output Options** ページを使用して [表 3-2](#) にリストされた値を指定します。

**表 3-2. CIC MegaCore Function Input/Output Options タブのサンプル・パラメータ**

| パラメータ  | 値    |
|--|------|
| Input data width                             | 8    |
| Full output resolution                       | オフ   |
| Output data width                            | 8    |
| Output Rounding Options                      | 切り捨て |
| Apply Hogenauer pruning across filter stages | オン   |

これらのパラメータについて詳しくは、[3-5 ページ](#)の「[CIC MegaCore Function Input/Output Options ページのパラメータ](#)」を参照してください。

## パラメータの説明

この項では、MegaWizard インタフェースで設定可能な CIC MegaCore ファンクション・パラメータを説明します ([3-1 ページ](#)の「[パラメータ設定の例](#)」を参照)。

[表 3-3](#) に、**Architecture** ページで設定可能なパラメータを示します。

**表 3-3. CIC MegaCore Function Architecture ページ (その 1)**

| パラメータ                               | 値   | 説明   |
|-------------------------------------|---|--|
| Target                              | サポートされたデバイスのリストについては、 <a href="#">1-2 ページ</a> の「 <a href="#">サポートされるデバイス・ファミリ</a> 」を参照してください。 | Quartus II プロジェクトを作成したときに指定したターゲット・デバイス・ファミリが表示されます。   |
| Filter type                         | デシメータ、補間器   | デシメータまたは補間器のどちらを実装するかを選択できます。  |
| Number of stages                    | 1-12  | 必要な段数を指定します。   |
| Differential delay                  | 1, 2  | サイクルの遅延差を指定します。  |
| Rate change factor                  | 2-32000   | レート変換率を指定します。  |
| Enable variable rate change factor  | On または Off  | ラン・タイムに変更できる可変レート変換率をイネーブルするためにオンにします。このオプションがオンの場合、 <b>Rate change factor</b> パラメータは使用できませんが、最小値および最大値を指定することができます。 |
| Number of interfaces                | 1-128   | MISO の入力数および SIMO の出力数を指定します。(1)   |
| Number of channels per interface    | 1-1024  | インタフェースあたりのチャンネル数を指定します。(1)  |
| Integrator data storage             | ロジック・エレメント、メモリ  | ロジック・エレメントまたはメモリとして積分器データ・ストレージを実装するかどうかを選択できます。(2)  |
| RAM type of integrator data storage | AUTO、M512、M4K、M9K、M144K、MLAB  | <b>Memory</b> を選択した場合、積分器データ・ストレージに使用する RAM タイプを選択できます。(4)   |
| Differentiator data storage         | ロジック・エレメント、メモリ  | ロジック・エレメントまたはメモリとして微分器データ・ストレージを実装するかどうかを選択できます。(3)  |

表 3-3. CIC MegaCore Function Architecture ページ ( その 2 )

| パラメータ                                    | 値                            | 説明   |
|--|------------------------------|--|
| RAM type of differentiator data storage  | AUTO、M512、M4K、M9K、M144K、MLAB | <b>Memory</b> を選択した場合、微分器データ・ストレージに使用する RAM タイプを選択できます。(4)   |
| Use pipelined integrators                | On または Off                   | パイプライン化された積分器を使用するには、オンにします。 <b>Number of channels per interface</b> が 2 以上である (または積分器データ・ストレージ用に <b>Memory</b> オプションを選択した場合、6 以上である) ときは、このオプションが利用可能です。<br><br>大規模な入力ビット幅を持ち、高い $f_{MAX}$ を必要とするマルチチャンネル・デザイン用に、このオプションを使用します。このオプションは、 <b>Stratix</b> ファミリー・デバイスのターゲット・デザインに推奨しますが、 <b>Cyclone</b> ファミリー・デバイスに推奨しません。             |
| Number of pipeline stages per integrator | 1-4                          | 積分器ごとに使用されたパイプライン・ステージ数を指定します。追加インテグレータを追加すると、 $f_{MAX}$ を向上させるが、リソースの使用率を増加させます。<br><br>使用できるパイプライン・ステージの最大数は、チャンネル数、および積分器データ・ストレージに <b>Memory</b> または <b>Logic Cells</b> のどちらを選択するかによって依存します。 <b>Memory</b> が選択されている場合、パイプライン・ステージの最大数は、チャンネル数に 5 を引き算した値に等しくなります。 <b>Logic Cells</b> が選択されている場合、パイプライン・ステージの最大数は、チャンネル数に等しくなります。 |

表 3-3 の注 :

- (1) **Number of interfaces** および **Number of channels per interface** のプロダクトは 1024 以下にする必要があります。
- (2) **Number of channels per interface** は 4 より大きい時、**Memory** オプションが積分器データ・ストレージに使用できます。
- (3) **Differential delay**、**Number of channels per interface** および **Number of interfaces** は 4 より大きい時、**Memory** オプションが微分器データ・ストレージに使用できます。
- (4) 使用可能なオプションはターゲット・デバイス・ファミリーに依存します。**AUTO** を選択された場合、Quartus II ソフトウェアは自動的に現在選択されているデバイス・ファミリーに最適な RAM タイプを選択します。

表 3-4 に、**Input/Output Options** ページで設定可能なパラメータを示します。

表 3-4. CIC MegaCore Function Input/Output Options ページのパラメータ ( その 1 )

| パラメータ                   | 値                | 説明  |
|-------------------------|------------------|---|
| Input data width        | 1-32             | ビットの入力データ幅を指定します。   |
| Full output resolution  | On、Off           | フル出力の分解能を使用するには、オンにします。選択すると、出力データ幅は最大に設定され、出力の丸め処理のオプションが無効になっています。    |
| Output data width       | 1 ~ 計算された最大のデータ幅 | ビットの出力データ幅を指定します。   |
| Output Rounding Options | 切り捨て、収束丸め、丸め、飽和  | 必要な丸め処理の出力モードを選択します。飽和限界がオーバーフローの最大値、または負のオーバーフローの最小値であることを注意してください。(1) |

表 3-4. CIC MegaCore Function Input/Output Options ページのパラメータ (その 2)

| パラメータ  | 値      | 説明   |
|--|--------|--|
| Apply Hogenauer pruning across filter stages | On、Off | このオプションは <b>Architecture</b> ページで <b>Decimator</b> フィルタ・タイプを選択する場合のみ、使用できます。フィルタ・ステージを介して Hogenauer プルーニングを適用するときに、オンにします。 |

## 表 3-4 の注:

- (1) これらのオプションについて詳しくは、4-4 ページの「丸め処理の出力」を参照してください。



## カスケード・インテグレータ・コーム (CIC) フィルタ

カスケード・インテグレータ・コーム (CIC) フィルタは現代の通信システムで広く使用されています。CIC フィルタを使用すると、サンプル・レート変換を実行するためのシリコン効率的なアーキテクチャを提供します。これは、デシメーションによる狭帯域ソースからのベースバンド信号を抽出し、補間による処理済ベースバンド信号からの狭帯域信号を構築することによって達成されます。CIC フィルタの有利な点は、加算器とレジスタのみを使用することで、大きなレートの変更を処理するためのハードウェアで実装するために乗算器を必要としません。

Hogenauer フィルタとしても知られる、カスケード・インテグレータ・コーム (CIC) フィルタは、デシメーションまたは補間を実行するために使用することができます。デシメーション CIC フィルタは、積分器のカスケード (積分器のセクションと呼ばれる)、次にダウン・サンプリング・ブロック (デシメータ) および、微分器のカスケード (微分器またはコーム・セクションと呼ばれる) で構成されています。同様に、補間 CIC フィルタは、微分器のカスケード、次にアップ・サンプリング・ブロック (補間器) および積分器のカスケードで構成されています。

CIC フィルタでは、積分器やコーム・セクションの両方では、積分器と微分器の数が同じです。積分器と微分器の各ペアは、ステージと呼ばれます。段数 ( $N$ ) は、CIC フィルタの周波数応答に直接影響を及ぼします。フィルタの応答は、段数  $N$ 、レート変換率  $R$ 、および微分器の遅延数 (遅延差と呼ばれる)  $M$  のコンフィギュレーションによって決定されます。実際には、遅延差が 1 または 2 に設定されます。

図 4-1 に、積分器を示します。

図 4-1. 積分器

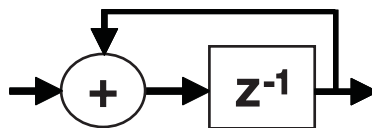


図 4-2 に、微分器を示します。

図 4-2. 微分器

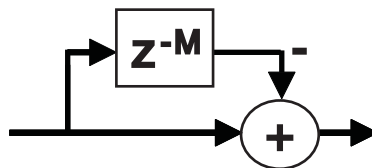


図 4-3 に、3つのステージ・デシメーション CIC フィルタを作成するために、デシメータを使用して結合される3つの積分器と3つの微分器を示します。

図 4-3. 3つのステージ・デシメーション CIC フィルタ

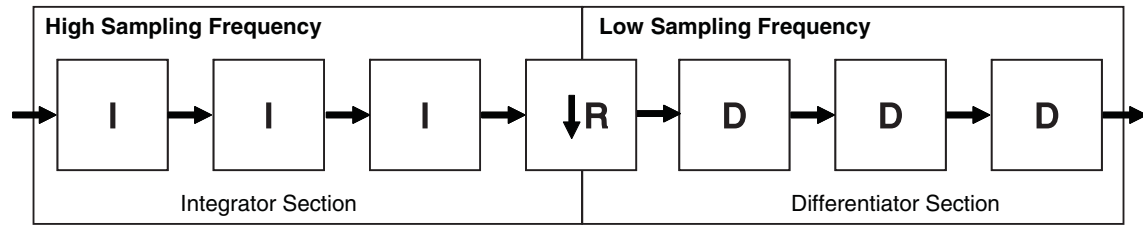
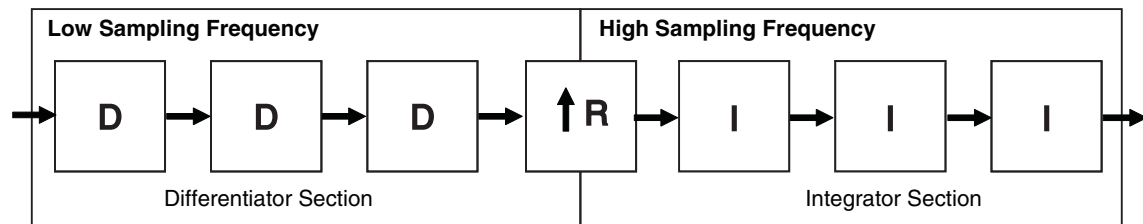


図 4-4 に、3つのステージ補間 CIC フィルタを作成するために、補間器を使用して結合される3つの微分器と3つの積分器を示します。

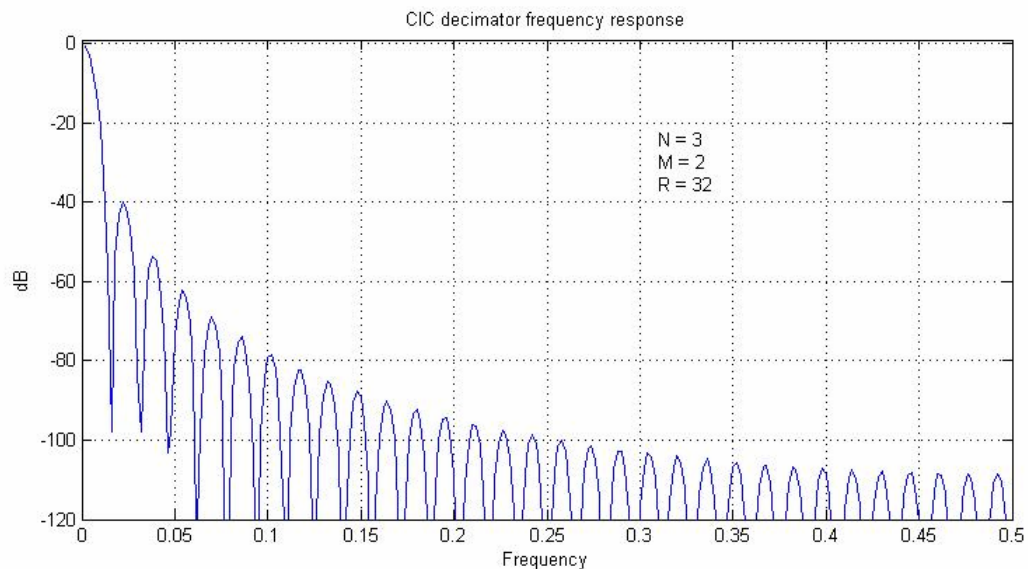
図 4-4. 3つのステージ補間フィルタ



## 標準的な周波数応答

図 4-5 は、 $N=3$ 、 $M=2$ 、 $R=32$  を持つ CIC デシメーション・フィルタの周波数応答を示します。

図 4-5. 3つのステージ CIC デシメーション・フィルタの周波数応答



## データ・ストレージ

積分器と微分器のデータ・ストレージにはロジック・セルまたはメモリ・ブロックのいずれかを利用することができます。データ・ストレージのオプションがイネーブルされている場合、MegaWizard インタフェースでデータ・ストレージのタイプが選択できます。次に、メモリ・タイプが選択されたデバイスに基づいて、Quartus II ソフトウェアにより自動的に選択されます。

メモリのオプションがイネーブルになっている場合について詳しくは、[3-4 ページの「CIC MegaCore Function Architecture ページ」](#)を参照してください。

## 出力のオプション

出力データ・ビット幅と丸め処理の出力オプションを選択することができます。

### 出力データ幅


デシメーション・フィルタの場合、フィルタの出力でのゲインは、次の通りです。

$$G = (RM)^N$$

したがって、フル解像度の場合、出力ステージでのデータ幅は、次の通りです。

$$B_{out} = \lceil B_{in} + N \log_2(RM) \rceil$$

ここで、 $B_{in}$  は入力データ幅です。

 各積分器と微分器には、データ損失なしのために  $B_{out}$  のデータ幅が必要です。各積分器と微分器には、データ損失なしのために  $B_{out}$  のデータ幅が必要です。

フル解像度の出力データ幅より小さい出力データ幅を選択した場合、Hogenauer プルーニング技術はフィルタ・ステージを介してデータ幅と全体のリソース使用率を減らすために提供することができます。

補間フィルタの場合、フィルタ・ステージごとのゲインは、次の通りです。

$$G_i = \begin{cases} 2^i & i = 1, 2, \dots, N \\ \frac{2^{2N-1} (RM)^{i-N}}{R} & i = N+1, \dots, 2N \end{cases}$$

したがって、 $i$  番目のステージに必要なデータ幅は、次の通りです。


$$W_i = \lceil B_{in} + \log_2(G_i) \rceil$$

また、出力ステージでのデータ幅は、次の通りです。

$$B_{out} = \lceil B_{in} + N \log_2(RM) - \log_2(R) \rceil$$

ここで、 $B_{in}$  は入力データ幅です。


遅延差が 1 であるとき、積分器ステージごとのビット幅は安定性を確保するために、1 ずつ増加します。

 これらの計算について詳しくは、Hogenauer, Eugene. *An Economical Class of Digital Filters For Decimation and Interpolation*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-29, pp. 155-162, April 1981 を参照してください。

### 丸め処理の出力

高レート変換率の場合、データ損失なしのため、必要な最大データ幅は、多くの場合に大きいです。入力レベルに出力データの幅を減らすために、量子化は、通常に出力ステージの最後に適用されます。そうする場合は、次の丸め処理または飽和のオプションが提供されます。


- **Truncation** : LSB は廃棄されます。(これは、負の無限大への丸めと同等である)。
- **Convergent rounding** : バイアスのない丸め処理としても知られます。最も近い偶数に丸められます。最も重要な削除されたビットが1の場合、おおよびいずれかの残りのビットまたは他の削除されたビットは少なくとも1つの最下位が1の場合には、残りのビットに1つを追加します。
- **Round up** : 正の無限大への丸め処理と知られます。キャリー・インを介して正と負の数のため、破棄されたビットの **MSB** を追加します。
- **Saturation** : 入力許容範囲を超えると、出力で限界値 (オーバーフローの場合は上限値、または負のオーバーフローの場合は下限値) を入れます。上限は  $+2^{n-1}$  であり、下限は  $-2^n$  です。

 これらの丸めのオプションは、フィルタの出力ステージにのみ適用することができます。中間ステージでのデータ幅は変更されません。次のセクションでは、中間ステージでデータ幅を変更できるケースを説明します。


## Hogenauer プルーニング

Hogenauer プルーニング [参照] は、導入された合計エラーが丸めフル精度の出力によってまだ導入された量子化誤差を超えていない間に、ステージからステージに単調減少ビットの保持数を持つ中間ステージでの切り捨てや丸め処理を利用する技術です。この技術は、フィルタによって使用されるロジック・セル数を減らすことができ、優れた性能を得ることができます。

$N$  および  $R$  の値に対して Hogenauer のビット幅の成長を計算するための存在のアルゴリズムは計算コストが高いです。

 これらのアルゴリズムについて詳しくは、U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, 2nd Edition, Spinger, 2004 を参照してください。


CIC MegaCore ファンクションは、MegaCore ファンクション内に格納されている計算済み Hogenauer プルーニングのビット幅を持っています。Hogenauer プルーニングがデシメーション・フィルタにイネーブルされる場合、Hogenauer プルーニングのビット幅を計算されるまでに待つ必要はありません。

 Hogenauer プルーニングは選択された出力データ幅がフル出力解像度のデータ幅より小さいときのみ、デシメーション・フィルタに使用できます。

## マルチチャネル・サポート

同じコンフィギュレーションがある CIC フィルタによるフィルタリングが必要なデジタル信号処理 (DSP) システムで通常は多くのチャネルがあります。これらはステージごとに存在する加算器を共有し、全体的なリソース使用率を減少される1つのフィルに統合することができます。

この組み合わせフィルタは、多くの個々の CIC フィルタを使用するよりも少ないリソースを使用します。例えば、2チャネルの平行・フィルタは、2つの出力を計算するために2クロック・サイクルが必要です。結果としてハードウェアは、個々のフィルタの2倍データ・レートで実行する必要があります。これは、加算器が特に大規模な成長でより高いレートの変換に対して特に便利です。

 ロジック・エレメントの数を最小限に抑えるために、多重入力・シングル出力 (MISO) アーキテクチャはデシメーション・フィルタに使用することができ、シングル入力・多重出力 (SIMO) アーキテクチャは補間フィルタに使用することができます (次の項で説明)。

### 多重入力・シングル出力 (MISO)

多くの実際のデザインにおいてはチャネル信号が異なる入力インタフェースから来ます。入力インタフェースごとに、レート変換率を含む同じパラメータは、CIC フィルタが処理しようとしているチャンネル・データに適用されます。CIC MegaCore ファンクションは、柔軟性がロー・レート微分器セクションのタイム・シェアリングを利用することができます。

これは高レート部分に複数入力インタフェースと処理チェーンを提供して、次に、シングル処理チェーンに低いレート部分に関連するすべての処理を組み合わせることによって達成されます。この方法は、リソースのフル使用率が発生し、最も効率的なハードウェアの実装を表します。これらのアーキテクチャは、多重入力・シングル出力 (MISO) のデシメーション・フィルタとして知られています。


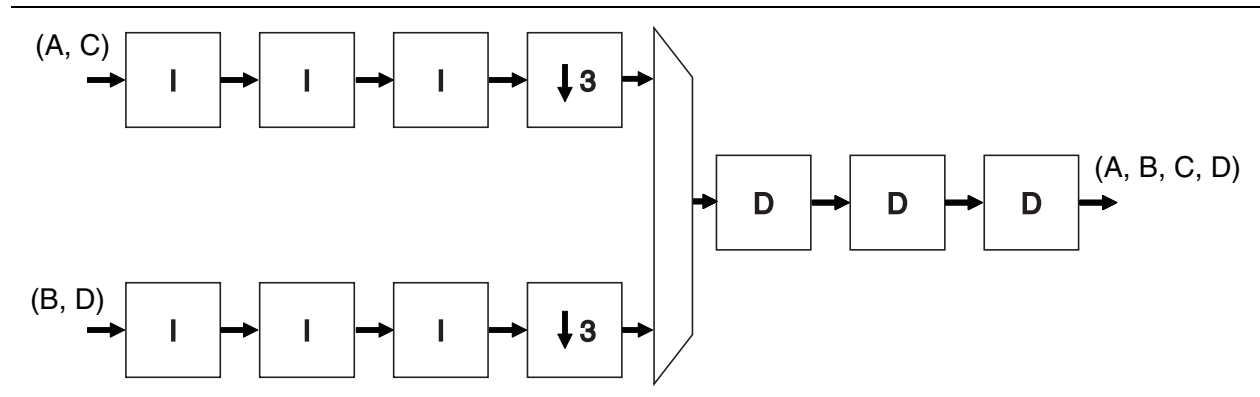

 図 4-6 に、4チャネルの合計を処理する CIC フィルタの MISO アーキテクチャの例を示します。この例では、シンボル A、B、C、D が1つの出力 A、B、C、D に多重化されます。

図 4-6. 多重入力・シングル出力のアーキテクチャ



入力データのサンプリング周波数は、バスごとに時間多重の2つのチャンネルにのみ使用できます。したがって、CIC フィルタは、2つの入力インタフェースでコンフィギュレーションされる必要があります。2つのインタフェースが必要であるため、このアーキテクチャを利用するには少なくともレート変換率2でなければなりません。これにより、複数の入力インタフェースを使用して最大1,024チャンネルをサポートすることができます。

 MISO アーキテクチャは、デシメーション・フィルタ・タイプが選択されたとき、および MegaWizard インタフェースで選択されたインタフェース数が1より大きいときに適用されます。

### シングル入力・多重出力 (SIMO)

シングル入力・多重出力 (SIMO) は補間 CIC フィルタに関連付ける機能です。このアーキテクチャでは、フィルタリングに提示されたすべてのチャンネル信号はシングル入力インタフェースから来ます。

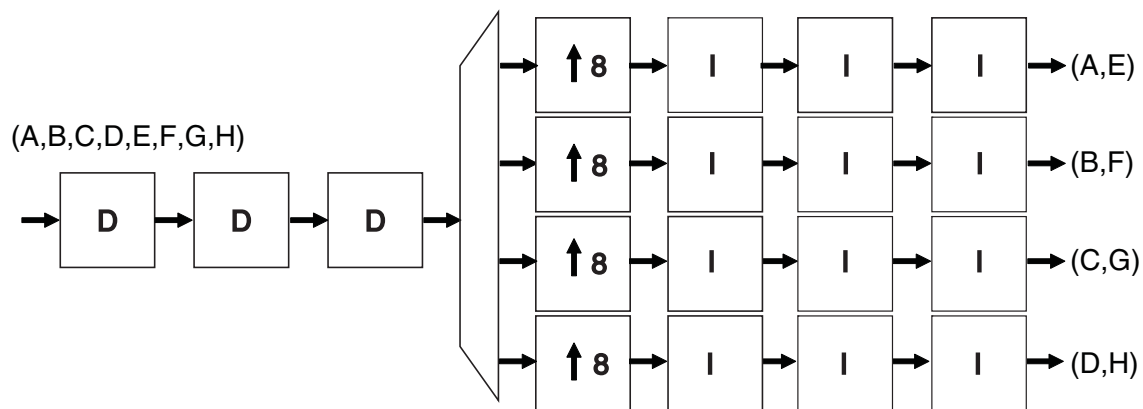
MISO のケースと同様に、それはより高いサンプリング周波数の積分器のセクションよりも、多くのチャンネル間で低いサンプリング・レート微分器のセクションを共有することが可能です。したがって、このアーキテクチャでは、微分器のセクションのシングル・インスタンスおよび積分器のセクションの複数のパラレル・インスタンスを提供します。

微分器のセクションで処理した後、チャンネル信号は積分器のセクションで、高いサンプリング周波数で処理するために複数のパラレル・セクションに分割されます。


 図4-7に、8チャンネルの合計を処理するCIC フィルタのSIMO アーキテクチャの例を示します。

この例では、シンボル A、B、C、D、E、F、G、H が4つの出力 A、E；B、F；C、G；および D、H に多重化されます。

図4-7. シングル入力・多重出力のアーキテクチャ




出力データの必要なサンプリング周波数は、バスごとに時間多重の2つのチャンネルにのみ使用できます。したがって、CIC フィルタは、4つの出力インタフェースでコンフィギュレーションされる必要があります。4つのインタフェースが必要であるため、このアーキテクチャを利用するには少なくともレート変換率4でなければなりません。しかし、この例では、8つのレート変換が示されています。

 SIMO アーキテクチャは、補間フィルタ・タイプが選択されたとき、および MegaWizard インタフェースで選択されたインタフェース数が 1 より大きいときに適用されます。


入力チャンネルの総数は、インタフェース数の倍数でなければなりません。この要求を満たすために、ダミー・チャンネルを挿入または複数の CIC MegaCore ファンクションを使用することのいずれかが必要です。

データは Avalon Streaming (Avalon-ST) インタフェースを使用してパケットとして転送されます。これらのインタフェースの一般的な説明については、[4-9 ページの「Avalon Streaming インタフェース」](#)を参照してください。

 マルチ・チャンネル MISO および SIMO アーキテクチャのデザイン例については、[「AN442: Tool Flow Design of Digital IF for Wireless Systems」](#)を参照してください。

## デシメーションおよび補間に対する可変レート変換率

必要に応じて、デシメータまたは補間器のレート変換率の最小値と最大値を設定し、レート変換率は、実行時に設定することができるようにすることができます。これらをセットすると、追加の rate ポートが提供されて、レート変換率を指定することができます。可変レート変換率のオプションをイネーブルする方法については詳しくは、[3-1 ページの「パラメータ設定の例」](#)を参照してください。

 可変レート変換率がイネーブルされている場合、MegaCore ファンクションは、レート変換率を変更されたときにリセットする必要があります。それ以外の場合は、以前のメモリ/レジスタの値が使用されます。フィルタ・モード（補間またはデシメーション）は、実行時には変更できません。


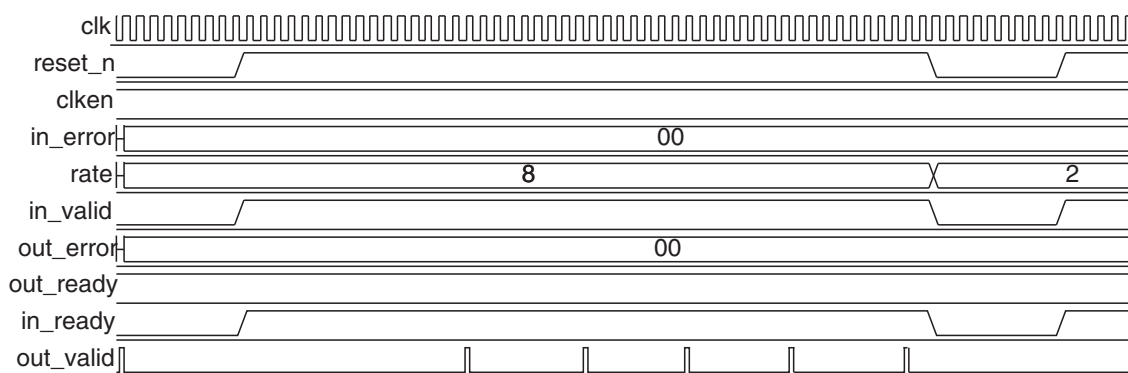
 [図 4-8](#) に、可変レート変換率のデシメーション・フィルタの入力および出力のタイミング関係を示します。out\_valids 信号は、可変レート変換に応じてその期間をどのように変化させるかに注意してください。

図 4-8. 可変レート変換のデシメーション CIC フィルタのタイミング図





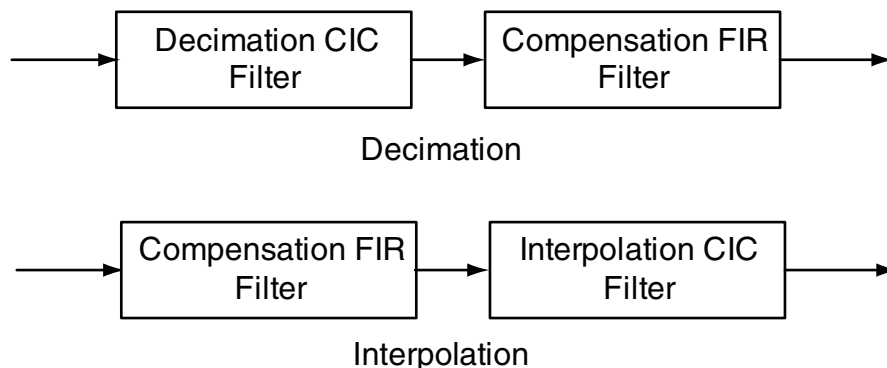
## FIR フィルタの補償係数

CIC フィルタは、ロー・パス・フィルタ特性を備えています。通過帯域の特性およびエイリアシング/画像の除去を変更するには、3つのパラメータ（レート変換率  $R$ 、ステージ数  $N$ 、遅延差  $M$ ）のみを変更することができます。

しかし、通過帯域ゲインの垂下および広い遷移領域のため、単独で CIC フィルタは、通常に、デシメーションまたは補間フィルタリング・アプリケーションで必要とされる平坦な通過帯域と狭い遷移領域フィルタ性能を提供することはできません。

この問題は、出力帯域幅を狭くし、通過帯域ゲインを平坦化するための補償 FIR フィルタにデシメーションまたは補間 CIC フィルタを接続することで緩和できます（図 4-9）。

図 4-9. CIC 補償 FIR フィルタの使用



CIC の望ましくない通過帯域垂下を等化する FIR フィルタの係数を決定し、理想周波数応答を構築するために周波数サンプリング法を使用することができます。

理想周波数応答は、応答を反転前に CIC の正規化された振幅応答をサンプリングすることによって決定されます。

一般に、それは通過帯域での応答を均等化することだけが必要ですが、フィルタ・チェーンのカスケード応答を微調整するために通過帯域よりもさらにサンプリングすることができます。

MATLAB スクリプト `<variation_name>_fir_comp_coeff.m` は、CIC MegaCore ファンクションによってプロジェクト・ディレクトリに生成されます。このスクリプトは、適切な通過帯域イコライゼーションを提供する FIR 係数を生成するには、MATLAB で実行できます。生成された係数は、アルテラの FIR コンパイラ MegaCore ファンクションで使用するための準備ができていてテキスト・ファイルに保存されます。

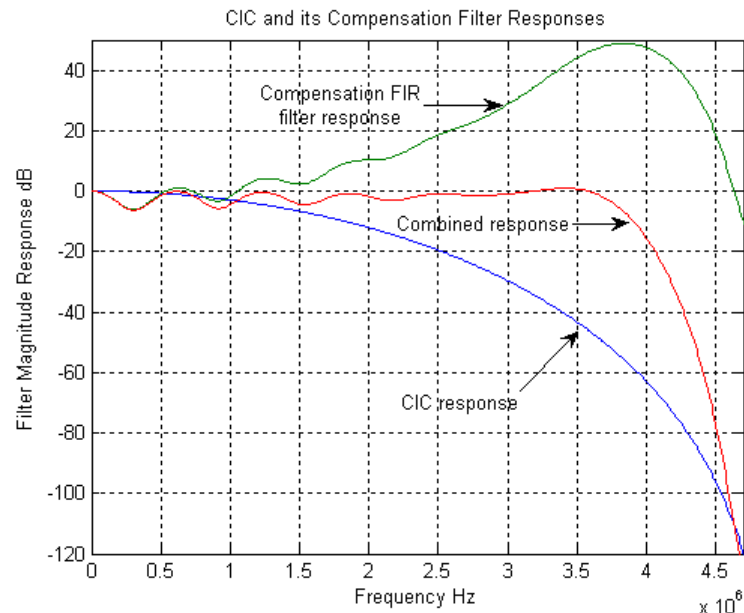
MATLAB スクリプトは補償 FIR フィルタに対して、次のパラメータが必要です。

- $L$  : タップの数や係数の数と同じである FIR フィルタの長さ
- $F_s$  : デシメーション/補間前の FIR フィルタのサンプル・レート (Hz)
- $F_c$  : FIR フィルタの遮断周波数 (Hz)
- $B$  : 係数が固定小数点数で書かれている場合での係数ビット幅

4-9 ページの「CIC および補償フィルタの応答」に、補償フィルタ応答の例を示します。



図4-10. CIC および補償フィルタの応答



詳細については、「AN455: CIC 補償フィルタの解説」を参照してください。

## Avalon Streaming インタフェース

Avalon<sup>®</sup> Streaming (Avalon-ST) インタフェースは、Atlantic インタフェースが進化したものです。Avalon<sup>®</sup> Streaming (Avalon-ST) インタフェースは、ソース・インタフェースからシンク・インタフェースへのデータ転送に対して標準的な柔軟性の高いモジュラ式プロトコルを定義しており、データパスにおけるデータ・フローのコントロール・プロセスを簡略化します。

Avalon-ST インタフェース信号は、チャンネルやパケット境界の概念のない従来の単一データ・ストリームをサポートします。このようなインタフェースは通常、data、ready、および valid 信号から構成されます。

Avalon-ST インタフェースは、複数のチャンネルに渡ってインタリーブされたパケットでバーストとパケット転送のために、より複雑なプロトコルをサポートすることができます。

Avalon-ST インタフェースは、本質的に複雑な制御ロジックを実装せずに効率的な達成を可能にするマルチ・チャンネル・デザイン時間多重実装を同期させます。

Avalon-ST インタフェースでは、シンクがデータの送信を停止するようにソースに通知できるフロー制御メカニズムであるバックプレッシャをサポートしています。FIFO バッファがいっぱいである場合、または、その出力で輻輳が発生している場合、シンクは常にデータの流れを止めるためにバックプレッシャを使用しています。

CIC MegaCore ファンクションを含むデータパスをデザインするときに、ダウンストリーム・コンポーネントが常にデータを受け取ることができることを知っていれば、バックプレッシャを必要としないことがあります。

*Avalon Interface Specifications* は、Avalon-ST インタフェースのいずれかのタイプを指定するために使用できるパラメータを定義します。表 4-1 に、CIC MegaCore ファンクションで使用される Avalon-ST インタフェースを定義されるパラメータ値を示します。表に、明示的に表示されないすべてのパラメータは未定義値を持っています。

表 4-1. Avalon-ST インタフェースのパラメータ

| パラメータ名            | 値   |
|-------------------|---|
| READY_LATENCY     | 0   |
| BITS_PER_SYMBOL   | データ幅  |
| SYMBOLS_PER_BEAT  | (注 1), (注 2), (注 3)   |
| SYMBOL_TYPE       | 符号あり  |
| ERROR_DESCRIPTION | 00 : エラーなし<br>01 : startofpacket (SOP) の欠落<br>10 : endofpacket (EOP) の欠落<br>11 : 予期しない EOP または他のエラー |

表 4-1 の注：

- (1) シングル入力およびシングル出力のアーキテクチャの場合、ソースとシンクではビートごとに1つのシンボルがあります。
- (2) MISO アーキテクチャの場合、シンクではビートごとに <number of interfaces> のシンボルがあり、ソースではビートごとにシングル・シンボルがあります。
- (3) SIMO アーキテクチャの場合、ソースでのビートごとに <number of interfaces> のシンボルがあり、シンクでのビートごとにシングル・シンボルがあります。

*Avalon Interface Specifications* はオプションで、多くの信号の種類を定義します。

表 4-2 に、CIC MegaCore ファンクションの Avalon-ST インタフェースで使用される信号のタイプを示します。

表 4-2. Avalon-ST インタフェース信号のタイプ

| 信号のタイプ        | 幅                 |
|---------------|-------------------|
| ready         | 1                 |
| valid         | 1                 |
| data          | データ幅              |
| channel       | $\log_2$ (チャンネル数) |
| error         | 2                 |
| startofpacket | 1                 |
| endofpacket   | 1                 |

表に明示的に記載されていない信号タイプは CIC MegaCore ファンクションによって使用されません。

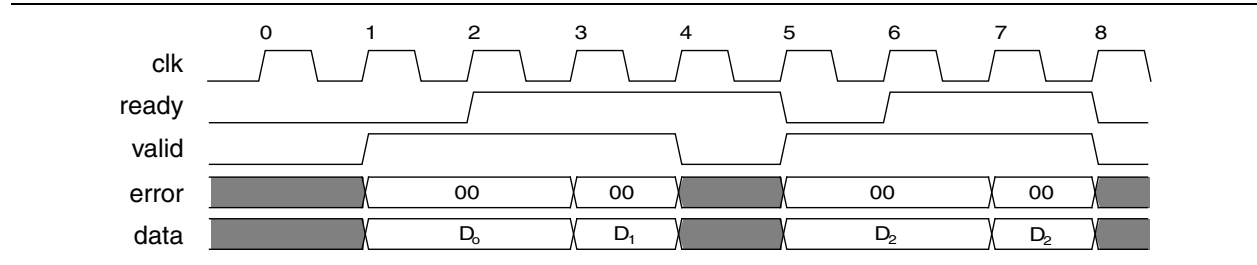


Avalon-ST インタフェース・プロトコルについて詳しくは、「[Avalon Interface Specifications](#)」を参照してください。

## Avalon-ST インタフェースのデータ転送のタイミング

図 4-11 に、データ転送のタイミングを示します。

図 4-11. READY\_LATENCY=0 の Avalon-ST インタフェースのタイミング



ソースは、シンクの準備ができていないにもかかわらず、データを提供し、1 サイクルで valid をアサートします。ソースは、シンクが次のデータ・サイクルに移る前に、ready がアサートされる時にサイクル 2 まで待機します。サイクル 3 では、ソースは同じサイクルでデータを駆動し、シンクはそれを受信する準備ができたので転送はすぐに発生します。サイクル 4 では、シンクは ready がアサートが、ソースが有効なデータを駆動しません。

### パケット・データ転送

ビートは、ソースとシンクのインタフェース間のデータの 1 単位の移転として定義されています。このデータ・ユニットは、1 つまたは複数の記号で構成され、各有効なサイクルに関する複数の情報を伝えるモジュールをサポートすることが可能です。

パケット・データ転送は、マルチ・チャネル転送に使用されます。2 つの追加信号 (startofpacket および endofpacket) は、パケット転送のために実装されます。

図 4-12 に、4 つのシンボルがビートごとに転送されている例を示しています。このビート・シナリオごとに複数のシンボルは、MISO CIC フィルタでシンク・インタフェースと SIMO CIC フィルタでソース・インタフェースの両方に適用されます。他のすべてのインタフェースは、ビートごとにシングル・シンボルに動作しますが、インタフェースはパケットを使用して複数のチャネルをサポートします。

図 4-12. パケット・データ転送

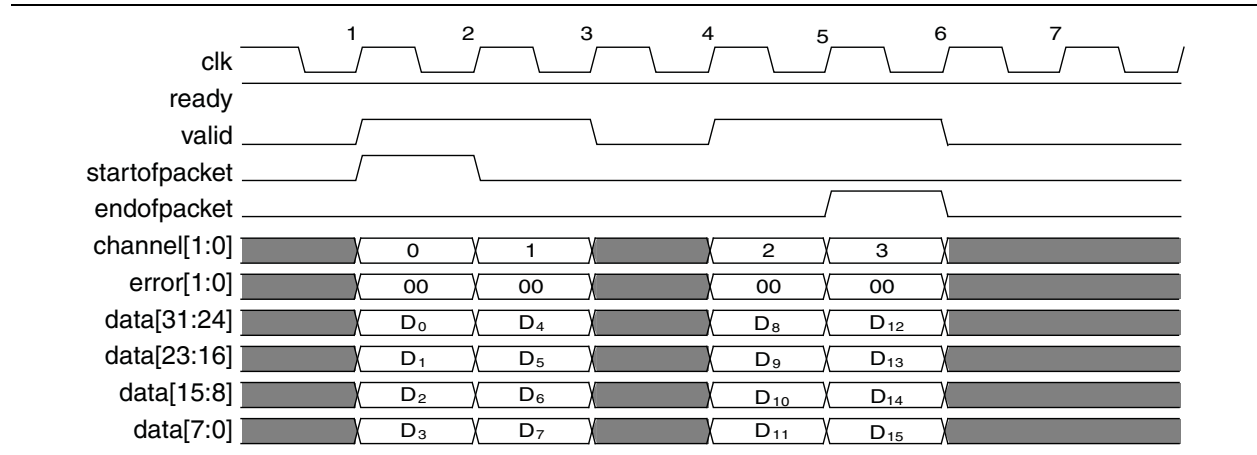


図 4-12 のデータ転送には、ready および valid がアサートされているとき、サイクル 1、2、4、および 5 で発生します。

サイクル1の間に、startofpacket がアサートされて、パケットの最初の4バイトが転送されます。サイクル5の間に、endofpacket がアサートされて、これはパケットの終わりであることを示しています。channel 信号は、データに関連付けられたチャンネルのインデックスを示します。例えば、サイクル1で、チャンネル0に関連付けられている D<sub>0</sub>、D<sub>1</sub>、D<sub>2</sub>、および D<sub>3</sub> のデータが入手可能です。

## 信号

表 4-3 に、CIC MegaCore ファンクションの入力信号および出力信号を示します。

表 4-3. CIC MegaCore ファンクション信号 (その 1)

| 信号               | 入力/<br>出力 | 説明  |
|------------------|-----------|---|
| clk              | 入力        | クロック信号はすべての内部レジスタをクロックするために使用されます。  |
| reset_n          | 入力        | アクティブ Low リセット信号です。CIC MegaCore ファンクションは常にデータを受信する前にリセットする必要があります。MegaCore ファンクションがリセットされていない場合、CIC フィルタはフィードバック信号による予期しない結果が生じることがあります。  |
| clken            | 入力        | オプションのトップ・レベルのクロック・イネーブル。   |
| in_data          | 入力        | サンプル入力。複数の入力ケースの場合、入力データ・ポートは in0_data、in1_data などとして名前が付けられます。   |
| in_endofpacket   | 入力        | 着信サンプル・グループの終了をマークします。N チャンネルがある場合、サンプルが最後のチャンネルに属するとき、パケット終了 EOP (end of packet) 信号は High である必要があります。それは、チャンネル N-1 は、in_data で発表されています。  |
| in_error         | 入力        | 入力側の Avalon-ST プロトコル違反を示すエラー信号： <ul style="list-style-type: none"> <li>■ 00 : エラーなし</li> <li>■ 01 : SOP の欠落</li> <li>■ 10 : EOP の欠落</li> <li>■ 11 : 予期しない EOP</li> </ul> 他のエラーは 11 にマークします。 |
| in_ready         | 出力        | MegaCore ファンクションは、データを受け入れることができることを示します。   |
| in_startofpacket | 入力        | 着信サンプル・グループの開始をマークします。パケット開始 SOP (start of packet) は、チャンネル 0 からのサンプルとして解釈されます。  |
| in_valid         | 入力        | in_data におけるデータが有効であると、この信号がアサートされます。in_valid がアサートされていない場合、有効が再アサートされるまで処理が停止されます。clken が 0 の場合、in_valid がアサートされません。   |
| out_channel      | 出力        | out_data で表示されている結果のチャンネルを指定します。  |
| out_data         | 出力        | フィルタ出力。データ幅はパラメータ設定に依存します。複数の出力ケースの場合、出力データ・ポートは out0_data、out1_data などとして名前が付けられます。  |
| out_endofpacket  | 出力        | 送信結果グループの終了をマークします。「1」の場合、チャンネル N-1 に対する結果が出力されます。ここで、N はチャンネル数です。  |

表 4-3. CIC MegaCore ファンクション信号 (その 2)

| 信号                | 入力/<br>出力 | 説明  |
|-------------------|-----------|---|
| out_error         | 出力        | ソース側の Avalon-ST プロトコル違反を示すエラー信号：<br><ul style="list-style-type: none"> <li>■ 00：エラーなし</li> <li>■ 01：SOP の欠落</li> <li>■ 10：EOP の欠落</li> <li>■ 11：予期しない EOP</li> </ul> 他のエラーは 11 にマークします。 |
| out_ready         | 入力        | データを受け入れることができる場合、ダウンストリーム・モジュールによりアサートされます。  |
| out_startofpacket | 出力        | 送信結果グループの開始をマークします。「1」の場合、チャンネル 0 に対する結果が出力されます。  |
| out_valid         | 出力        | 「1」の場合、チャンネル 0 に対する結果が出力されます。   |
| rate              | 入力        | この信号は可変レート変換率のオプションがイネーブルになっているときに利用でき、ラン・タイム中のデシメーション/補間のレートを変換することができます。それは、サイズ Ceil ( $\log_2$ (最大レート)) があります。   |

## 参考資料

アルテラのアプリケーション・ノート、ホワイト・ペーパー、およびユーザーガイドで、MegaCore ファンクションおよび Quartus II ソフトウェアを使用した効率的なデザイン方法が提供されています。アルテラ・ウェブサイト ([www.altera.co.jp](http://www.altera.co.jp)) を参照してください。また、以下の参考資料もあります。

- Hogenauer, Eugene. *An Economical Class of Digital Filters For Decimation and Interpolation*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-29, pp. 155-162, April 1981.
- U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, 2nd Edition, Springer, 2004.
- [MegaCore IP Library Release Notes and Errata](#).
- [アルテラ・ソフトウェアのインストールおよびライセンスマニュアル](#)。
- [AN320: OpenCore Plus 評価機能によるメガファンクションの評価](#)。
- [DSP Builder User Guide](#)。
- [Avalon Interface Specifications](#)。
- Quartus II ハンドブックの volume 3 の「[Simulating Altera Designs](#)」の章。
- [AN442: Tool Flow Design of Digital IF for Wireless Systems](#)。
- [AN455: CIC 補償フィルタの解説](#)。



## 改訂履歴

以下の表に、このユーザーガイドの改訂履歴を示します。

| 日付       | バージョン | 変更内容  |
|----------|-------|---|
| 2011年5月  | 11.0  | <ul style="list-style-type: none"> <li>■ Arria® II GX、Arria II GZ、Cyclone® III LS、および Cyclone IV GX デバイスの最終的なサポートにサポート・レベルを更新。</li> <li>■ HardCopy III、HardCopy IV E、および HardCopy IV GX デバイスの HardCopy Compilation にサポート・レベルを更新。</li> </ul> |
| 2010年12月 | 10.1  | <ul style="list-style-type: none"> <li>■ Arria II GZ デバイスの暫定サポートを追加。</li> <li>■ Stratix® IV GT デバイスの最終的なサポートにサポート・レベルを更新。</li> </ul>  |
| 2010年7月  | 10.0  | <ul style="list-style-type: none"> <li>■ Stratix V デバイスの暫定サポートを追加。</li> <li>■ パラメータ名の <b>Rate change factor</b> に <b>Rate factor</b> を更新。</li> </ul>  |
| 2009年11月 | 9.1   | <ul style="list-style-type: none"> <li>■ メンテナンスを更新。</li> <li>■ Cyclone III LS、Cyclone IV、および HardCopy IV GX デバイスの暫定サポート。</li> </ul>   |
| 2009年3月  | 9.0   | <ul style="list-style-type: none"> <li>■ スピードを最適化するオプションを追加。</li> <li>■ Arria II GX デバイスの暫定サポート。</li> </ul>   |
| 2008年11月 | 8.1   | <ul style="list-style-type: none"> <li>■ Stratix III のフル・サポート。</li> <li>■ 新技術刊行物スタイルの応用。</li> <li>■ UNIX のサポートを撤回。</li> </ul>   |
| 2008年5月  | 8.0   | <ul style="list-style-type: none"> <li>■ Cyclone III のフル・サポート。</li> <li>■ Stratix IV の暫定サポート。</li> </ul>  |
| 2007年10月 | 7.2   | <ul style="list-style-type: none"> <li>■ Arria GX のフル・サポート。</li> </ul>  |
| 2007年5月  | 7.1   | <ul style="list-style-type: none"> <li>■ 補間/デシメーションの変数および補償フィルタ係数の生成に新しい機能の説明を追加。</li> <li>■ Arria™ GX の暫定サポート。</li> <li>■ Stratix II GX および HardCopy® II デバイスのフル・サポート。</li> </ul>  |
| 2006年12月 | 7.0   | <ul style="list-style-type: none"> <li>■ Cyclone III の暫定サポート。</li> </ul>  |
| 2006年12月 | 6.1   | <ul style="list-style-type: none"> <li>■ このユーザーガイドの初版。</li> </ul>   |

## アルテラへのお問い合わせ

Altera® 製品に関する最新情報については、次の表を参照してください。

| お問い合わせ先 (1) | お問い合わせ方法 | アドレス   |
|-------------|----------|--|
| 技術的なご質問     | ウェブサイト   | <a href="http://www.altera.co.jp/support">www.altera.co.jp/support</a> |
| 技術トレーニング    | ウェブサイト   | <a href="http://www.altera.com/training">www.altera.com/training</a>   |
|             | 電子メール    | <a href="mailto:custrain@altera.com">custrain@altera.com</a>           |




| お問い合わせ先 (1)                              | お問い合わせ方法 | アドレス   |
|--|----------|--|
| 製品資料                                     | ウェブサイト   | <a href="http://www.altera.co.jp/literature">www.altera.co.jp/literature</a> |
| 一般的なお問い合わせ<br>ソフトウェア・ライセンス<br>に関するお問い合わせ | 電子メール    | <a href="mailto:nacomp@altera.com">nacomp@altera.com</a>                     |
|  | 電子メール    | <a href="mailto:authorization@altera.com">authorization@altera.com</a>       |

## 表の注：



(1) 詳しくは、日本アルテラまたは販売代理店にお問い合わせください。

## 表記規則

本資料では、以下の表に示す表記規則を使用しています。

| 書体  | 意味  |
|---|---|
| 太字かつ文頭が大文字  | コマンド名、ダイアログ・ボックス・タイトル、ダイアログ・ボックス・オプション、およびその他の GUI ラベルを表します。例えば、 <b>Save As</b> ダイアログ・ボックス。   |
| 太字  | ディレクトリ名、プロジェクト名、ディスク・ドライブ名、ファイル名、ファイルの拡張子、およびソフトウェア・ユーティリティ名を表します。<br>例： <b>qdesigns</b> ディレクトリ、 <b>d:</b> ドライブ、および <b>chiptrip.gdf</b> ファイル。   |
| 斜体かつ文頭が大文字  | 資料のタイトルを表します。例： <i>AN 519: Stratix IV</i> デザイン・ガイドライン。  |
| 斜体  | 変数を表します。例： $n+1$ 。<br>変数名は、山括弧 ( ) で囲んでいます。例：(ファイル名) および (プロジェクト名) <b>.pdf</b> ファイル。  |
| 文頭が大文字  | キーボード・キーおよびメニュー名を表します。例：Delete キー、Options メニュー。   |
| 「小見出しタイトル」  | かぎ括弧は、資料内の小見出しおよび Quartus II Help トピックのタイトルを表します。例：「表記規則」   |
| Courier フォント  | 信号、ポート、レジスタ、ビット、ブロック、およびプリミティブ名を表します。例： <code>data1</code> 、 <code>tdi</code> 、および <code>input</code> 。アクティブ Low 信号は、サフィックス <code>n</code> で表されています。例： <code>resetn</code> 。<br>コマンドライン・コマンド、および表示されているとおりに入力する必要があるものを表します。例： <code>c:\qdesigns\tutorial\chiptrip.gdf</code> 。<br>また、 <b>Report</b> ファイルなどの実際のファイルのセクション、ファイルの構成要素への参照（例：AHDL キーワードの SUBDESIGN）、ロジック・ファンクション名（例：TRI）も Courier フォントで表記されています。 |
| 1、2、3、および a、b、c など  | 手順など項目の順序が重要なものは、番号が付けられリスト形式で表記されています。   |
| ■ ■   | 箇条書きの黒点などは、項目の順序が重要ではないものに付いています。   |
|  | 指差しマークは、要注意箇所を表しています。   |
|  | 注意は、製品または作業中のデータに損傷を与えたり、破壊したりするおそれのある条件や状況に対して注意を促します。   |
|  | 警告は、ユーザーに危害を与えるおそれのある条件や状況に対して注意を促します。  |



| 書体  | 意味                        |
|---|---------------------------|
|  | 矢印は、Enter キーを押すことを示しています。 |
|  | 足跡マークは、詳細情報の参照先を示しています。   |

