



RAM ベースのシフト・レジスタ (ALTSHIFT_TAPS) 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

UG-01009-2.1

ドキュメント・バージョン: 10.1
ドキュメント・デート: 2010年11月



Subscribe

© 2010 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX are Reg. U.S. Pat. & Tm. Off. and/or trademarks of Altera Corporation in the U.S. and other countries. All other trademarks and service marks are the property of their respective holders as described at www.altera.com/common/legal.html. 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. 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 MegaCore 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
機能	1-2
概要	1-2
第2章 . 使用法	
システム要件	2-1
MegaWizard Plug-In Manager のカスタマイズ	2-1
MegaWizard Plug-In Manager ページの説明	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	2-7
EDA ツールの使用によるネットリストの生成	2-8
ポートおよびパラメータ定義の使用	2-8
コンパイル後のメガファンクションの識別	2-9
シミュレーション	2-9
Quartus II ソフトウェア・シミュレータ	2-9
EDA シミュレータ	2-9
デザイン例：タップを備えたシフト・レジスタ	2-10
デザイン・ファイル	2-10
コンフィギュレーションの設定	2-10
ModelSim-Altera シミュレータの機能シミュレーション	2-10
シミュレーション結果の理解	2-11
結論	2-14
第3章 . 仕様	
ALTSHIFT_TAPS メガファンクションの Verilog HDL プロトタイプ	3-1
ALTSHIFT_TAPS メガファンクションの VHDL コンポーネント宣言	3-1
VHDL Library-Use の宣言	3-2
ALTSHIFT_TAPS メガファンクションのポートとパラメータ	3-2
追加情報	
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

サポートされるデバイス・ファミリ

ALTSHIFT_TAPS メガファンクションは、次の Altera® デバイス・ファミリをサポートします。

- Arria® GX
- Cyclone® III
- Cyclone II
- Cyclone
- HardCopy® II
- HardCopy Stratix®
- Stratix IV
- Stratix III
- Stratix II
- Stratix II GX
- Stratix
- Stratix GX
- ACEX® 1K
- APEX™ II
- APEX 20KC
- APEX 20KE
- FLEX® 10K
- FLEX 10KA
- FLEX 10KE

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・コード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。さらに、アルテラが提供するファンクションは、より効果的なロジックの合成およびデバイスの実装を提供する場合があります。パラメータを設定して、メガファンクションのサイズを調整することができます。

アルテラは、ALTSHIFT_TAPS と呼ばれる RAM ベースのシフト・レジスタ・メガファンクションを提供して、従来のシフト・レジスタにはない機能が含まれています。標準的なフリップ・フロップで実装された従来のシフト・レジスタは、大きなシフト・レジスタのために多数のロジック・セルを使用します。ALTSHIFT_TAPS メガファンクションは、デバイス・メモリ・ブロックに実装されて、ロジック・セルと配線リソースが節約されます。ローカル・データ・ストレージを必要とする DSP（デジタル信号処理）アプリケーションなどの複雑なデザインには、シフト・レジスタとして ALTSHIFT_TAPS メガファンクションを実装する方が効率的です。

ALTSHIFT_TAPS メガファンクションはタップを備えたパラメータ化されたシフト・レジスタです。タップは、シフト・レジスタ・チェーンの特定のポイントで、シフト・レジスタからデータ出力を提供します。更に、アプリケーションのために、これらのタップから出力を使用する追加のロジックを追加することができます。メガファンクションの出力タップの機能は、リニア・フィードバック・シフト・レジスタ (LFSR) および有限インパルス応答 (FIR) フィルタなどのアプリケーションに便利です。

機能

ALTSHIFT_TAPS メガファンクションはタップを備えたシフト・レジスタを実装し、次の追加の機能が含まれています。


- 選択可能な RAM ブロック・タイプ
- shiftin ポートおよび shiftout ポートの幅の広い範囲
- シフト・レジスタ・チェーン内の特定のポイントで出力タップのサポート
- タップ間の選択可能な距離

概要

Quartus® II ソフトウェアのシフト・レジスタ (RAM ベース) MegaWizard® Plug In Manager を使用して、ALTSHIFT_TAPS メガファンクションを簡単にコンフィギュレーションし、作成することができます。

第 2 章の「[使用法](#)」は、各オプションの説明で MegaWizard Plug-In Manager の各ページを案内します。

ALTSHIFT_TAPS メガファンクションは、シンプルなデュアル・ポート RAM 付きのすべてのサポートされたデバイス・ファミリのエンベデッド・メモリ・ブロックに実装されます。必要な容量に応じて RAM ブロックのタイプを選択することができます。メモリ・ブロックの幅と深さで表示される容量は、ALTSHIFT_TAPS メガファンクションの TAP_DISTANCE、NUMBER_OF_TAPS、および WIDTH パラメータに依存します。

-  標準的なメモリ・ブロックの機能と容量について詳しくは、TriMatrix エンベデッド・メモリ・ブロックの情報を含むデバイス・ハンドブックの章を参照してください。

ALTSHIFT_TAPS メガファンクションは、shiftin ポートと shiftout ポートの幅に応じて1クロック・サイクルでデータ・シフトのシングル・ビットおよび複数のビットをサポートします。例えば、shiftin ポートと shiftout ポートがシングル・ビット・データである場合、クロック・サイクルあたりに1ビットのみをシフトされます。shiftin ポートと shiftout ポートは複数のビット・データである場合、1ワード・データ（8ビット）のように、クロック・サイクルあたりにすべてのワードをシフトされます。

また、メガファンクションは、シフト・レジスタ・チェーンの特定のポイントで出力タップをサポートしますが、タップ・ポイントは等間隔にする必要があります。タップ間のスペースは、MegaWizard Plug-in Manager の TAP_DISTANCE パラメータによって設定されます。


 MegaWizard Plug-in Manager のタップ間の距離のオプションを設定する方法については、第2章の「[使用法](#)」を参照してください。TAP_DISTANCE パラメータについては、第3章の「[仕様](#)」を参照してください。

図 1-1 にある図 (a) は、従来の 12 ワードの深さのシフト・レジスタを示します。図 (b) は、シフト・レジスタ・チェーンのデータが ALTSHIFT_TAPS メガファンクションの出力タップで、等間隔（1-4-7-10 番目）でタップされている方法を示します。

図 1-1. 特定のポイントでシフト・レジスタ・チェーンのデータ・タッピング（注 1), (2), (3)

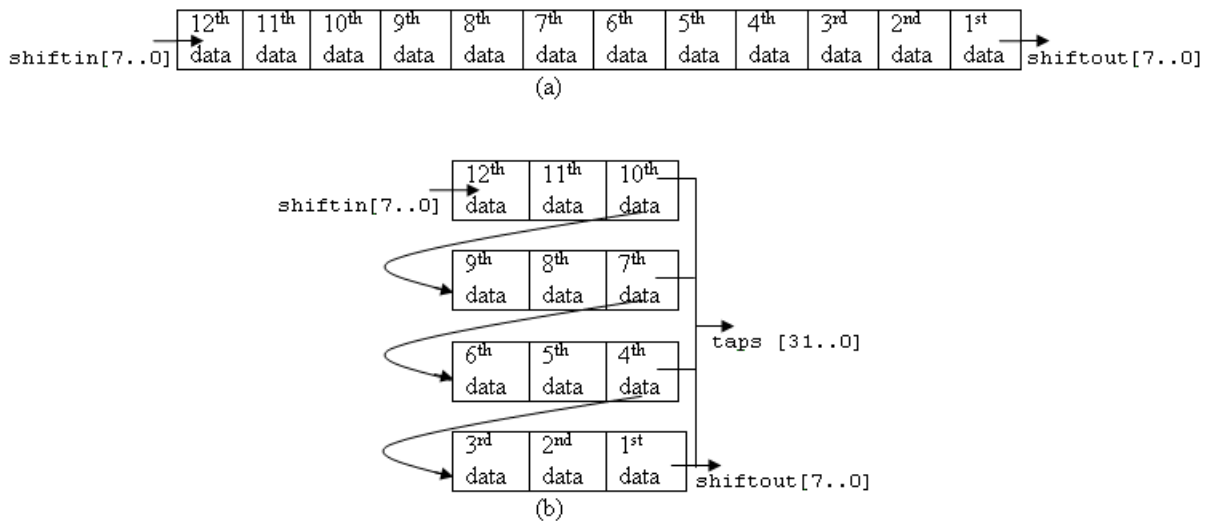


図 1-1 の注：

- (1) ここに記載されている ALTSHIFT_TAPS メガファンクションは、TAP_DISTANCE = 3 および NUMBER_OF_TAPS = 4 を持っています。
- (2) タップされたデータは taps [31..0] に出力します。一度に 4 ワードをタップするため、taps [31..0] が 32 ビット出力であることを注意してください。タップの MSB からの最初のワード（taps [31..24]）は、最初のデータを示し、4 番目、7 番目および 10 番目のデータが続きます。
- (3) shiftout [7..0] ワードは、taps [31..24] と同等です。

システム要件

この項の説明では、以下のソフトウェアが要求されます。

- Quartus® II ソフトウェア v8.0 以降
- オペレーティング・システム・サポート情報は、www.altera.co.jp/support/software/os_support/oss-index.html を参照してください。

MegaWizard Plug-In Manager のカスタマイズ

MegaWizard® Plug-In Manager は、デザイン・ファイルでインスタンス化することができるカスタム・メガファンクション・バリエーションを取めたデザイン・ファイルを作成または変更します。MegaWizard Plug-In Manager は、デザインで ALTSHIFT_TAPS メガファンクション機能のオプションを指定できるウィザードを提供します。

MegaWizard Plug-In Manager は、以下のいずれかの方法で開始します。

- Tools メニューの **MegaWizard Plug-In Manager** をクリックします。
- Block Editor で、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 ウィンドウの **MegaWizard Plug-In Manager** をクリックします。
- コマンド・プロンプトで次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。

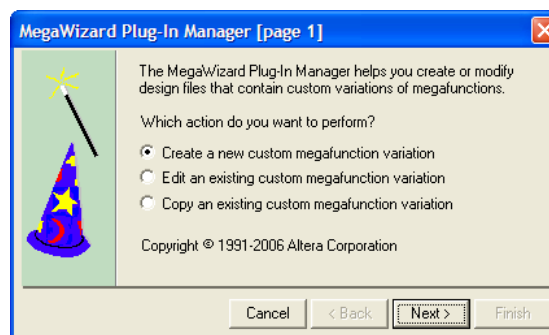
```
qmegawiz←
```

MegaWizard Plug-In Manager ページの説明

この項では、シフト・レジスタ (RAM ベース) MegaWizard Plug-In Manager の個々のページで提供されているオプションについて説明します。

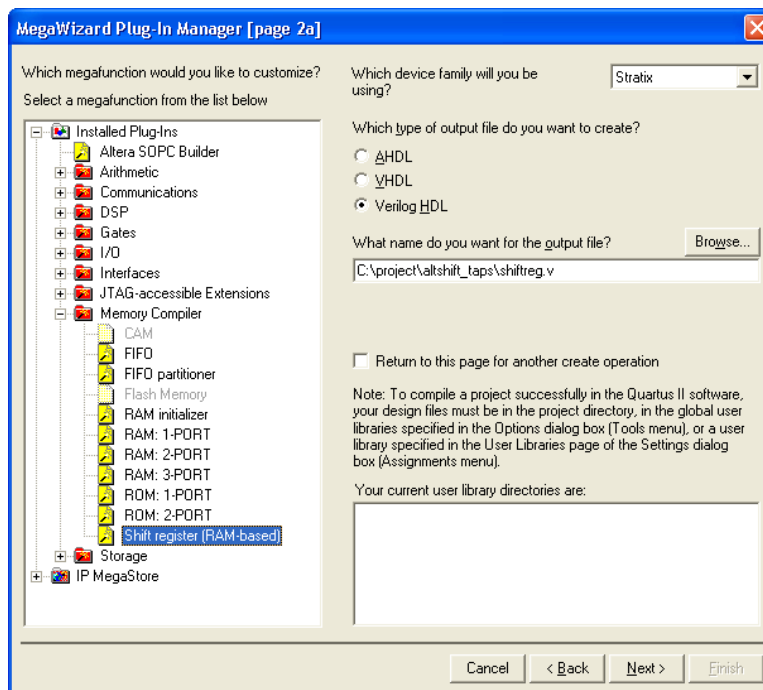
MegaWizard Plug-In Manager の 1 ページ目では、**Create a new custom megafunction variation**、**Edit an existing custom megafunction variation**、または **Copy an existing custom megafunction variation** を選択できます (図 2-1)。

図 2-1. MegaWizard Plug-In Manager [1 ページ]



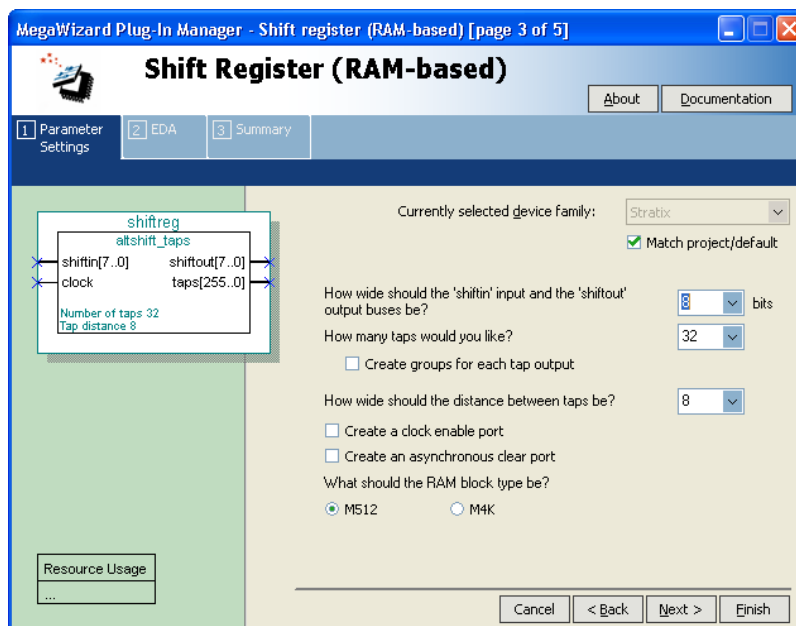
MegaWizard Plug-In Manager の 2a ページ目では、メガファンクション、使用するデバイス・ファミリ、作成する出力ファイルのタイプ、および出力ファイルの名前を指定します (図 2-2)。出力ファイルのタイプには、AHDL (.tdf)、VHDL (.vhd)、または Verilog HDL (.v) を選択します。

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



MegaWizard Plug-In Manager の 3 ページ目では、shiftin 入力バスと shiftout 出力バスの幅、タップ数、およびタップ間の距離を指定し、各タップの出力のグループを作成します。また、デザインに適用可能であれば、クロック・イネーブル・ポートと非同期クリア・ポートを作成することができ、使用する RAM ブロック・タイプを選択できます (図 2-3)。

図 2-3. MegaWizard Plug-In Manager – シフト・レジスタ (RAM ベース) [3/5 ページ]



 シフト・レジスタ (RAM ベース) MegaWizard Plug-In Manager の 3 ページ目から始めると、*Shift Register (RAM-based) (ALTSHIFT_TAPS) Megafunction User Guide*、ALTSHIFT_TAPS メガファンクションのオンライン・ヘルプを起動することができます。あるいは、**Documentation** ボタンをクリックしてサンプル波形を生成できます。

表 2-1 に、シフト・レジスタ (RAM ベース) MegaWizard Plug-In Manager の 3 ページ目で使用可能なオプションを示します。これらの表をハードウェア記述と共に使用して、最適な設定を決定します。

表 2-1. シフト・レジスタ (RAM ベース) MegaWizard Plug-in Manager [ページ 3] のオプション (その 1)

コンフィギュレーションの設定	説明
How wide should the 'shiftin' input and the 'shiftout' output buses be?	データ入力バスとデータ出力バスの幅を指定します。図 2-4 に示すように、この値はシフト・レジスタ・メモリのコンフィギュレーションの用語 w によって表示されます。(1)
How many taps would you like?	タップ数を指定します。図 2-4 に示すように、この値はシフト・レジスタ・メモリのコンフィギュレーションの用語 n によって表示されます。(2)
Create groups for each tap output	このオプションをオンにして、レジスタ・チェーンのタップされた出力データのために別々のグループを作成します。(3)

表 2-1. シフト・レジスタ (RAM ベース) MegaWizard Plug-in Manager [ページ 3] のオプション (その 2)

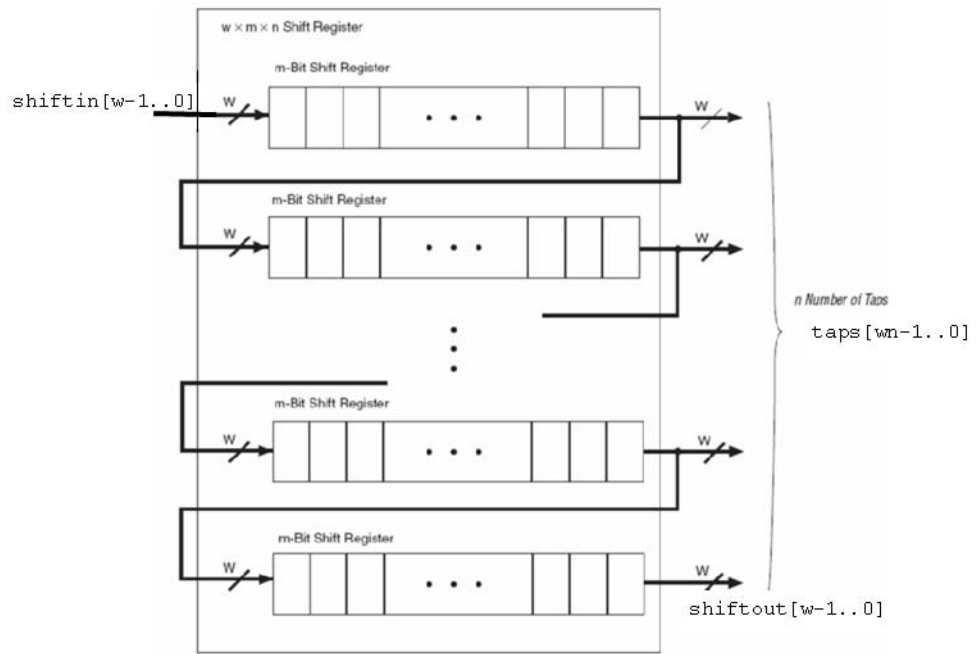
コンフィギュレーションの設定	説明
How wide should the distance between taps be?	タップ間の距離を指定します。図 2-4 に示すように、この値はシフト・レジスタ・メモリのコンフィギュレーションの用語 m によって表示されます。(4)
Create a clock enable port	このオプションをオンにしてレジスタ・ポートのイネーブル信号を作成します。このオプションがオンになっていない場合、レジスタ・ポートは常にイネーブルされます。(5)
Create an asynchronous clear port	このオプションをオンにして、非同期クリア信号を作成します。このオプションをアサートされると、シフト・レジスタの出力はすぐにクリアされます。
What should the RAM block type be?	機能をサポートされるメモリ・ブロックのタイプ、メモリ・コンフィギュレーション、およびアプリケーションの容量を選択します。(6)

表 2-1 の注：

- (1) `shiftin` 入力バスと `shiftout` 出力バスの幅は同一であり、そしてそれらがレジスタされていません。しかし、メモリ・ブロックの内部リード・アドレスはクロックに同期しているので、出力データはクロックに同期すると見なすことができます。
- (2) 出力タップの幅は w (入力データの幅) と n (タップ数) の逡倍です。また、出力タップの MSB からのワードは `shiftout` 出力バスと等しいです。
- (3) これらのグループの組み合わせは、`taps[wn-1:0]` バスを示します。
- (4) タップ間の距離 m は、少なくとも 3 でなければなりません。
- (5) レジスタされたポートは、メモリ・アドレス・ポートで内部レジスタと呼ばれます。`shiftin` ポートおよび `shiftout` ポートはレジスタされません。
- (6) 選択されたメモリ・ブロックのタイプについては、TriMatrix エンベデッド・メモリ・ブロックの情報含むデバイス・ハンドブックの章を参照してください。使用する RAM ブロックのタイプについて、特定されていない場合、`AUTO` を選択することができます。`AUTO` オプションを使用すると、メモリ・ブロックのタイプはコンパイル時に Quartus II ソフトウェアのシンセサイザーや Fitter によって決定されます。使用されるメモリ・ブロックのタイプを決定するには、Quartus II Fitter Report をチェックします。

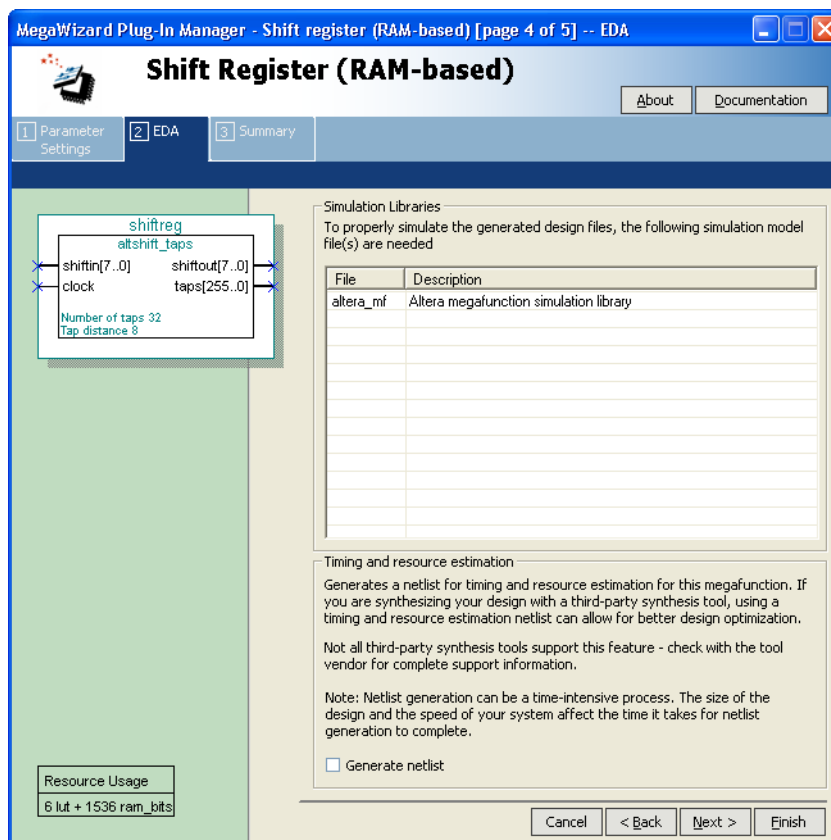
図 2-4 に、シフト・レジスタ・チェーンの例を示します。

図 2-4. シフト・レジスタ・チェーンの例



シフト・レジスタ (RAM ベース) MegaWizard Plug-In Manager の 4 ページ目でデザイン・ファイルを正確にシミュレートするために必要なファイルを示します (図 2-5)。

図 2-5. MegaWizard Plug-In Manager – シフト・レジスタ (RAM ベース) [4/5 ページ]

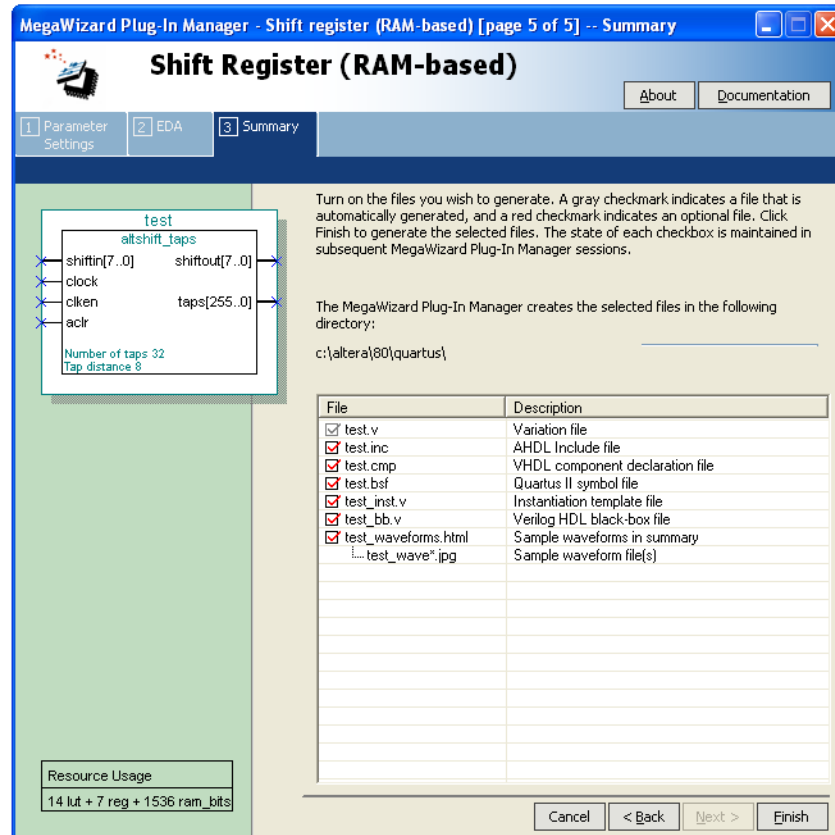


シフト・レジスタ (RAM ベース) MegaWizard Plug-In Manager の 5 ページ目で生成されるファイルのタイプを表示します。Variation ファイルは、自動的に生成され、2a ページで指定された言語のラッパー・コードが含まれています。MegaWizard Plug-In Manager の 5 ページで、生成されるファイルのタイプを指定します。以下のファイル・タイプから選択できます。

- AHDL インクルード・ファイル (<function name>.inc)
- VHDL コンポーネント宣言ファイル (<function name>.cmp)
- Quartus II シンボル・ファイル (<function name>.bsf)
- インスタンス化のテンプレート・ファイル (<function name>_inst.v)
- Verilog HDL ブラック・ボックス・ファイル (<function name>_bb.v)

MegaWizard Plug-In Manager の4ページで、**Generate netlist** を選択された場合、ネットリスト用のファイルも利用可能です。グレー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、赤色のチェックマークはオプションのファイルを示します (図 2-6)。

図 2-6. MegaWizard Plug-In Manager - シフト・レジスタ (RAM ベース) [5/5 ページ]




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、出力ファイルの設定が作成されます。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、ウィザードで選択した言語に応じて、Verilog-HDL (.v)、VHDL (.vhd)、または AHDL (.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共に生成します。

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
- VHDL デザイン・ファイルで使用できるコンポーネント宣言ファイル (`.cmp`)
- テキスト・デザイン・ファイル (`.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は、Quartus II Help または「Quartus II ハンドブック Volume1」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は、「Quartus II ハンドブック v1」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一覧については、[第3章の「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、解析とエラーボレーションが実行され、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。

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** 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 シミュレータは、シミュレーションを実行するための使いやす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 ソフトウェア・シミュレータ

Quartus II シミュレータでは、機能とタイミングの 2 種類のシミュレーションを実行することができます。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のみを追加します。Node Finder の Filter オプションで、次のいずれかを使用して検索することが出来ます：Registers : pre-synthesis、Design Entry、または Pins。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のみ追加します。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が困難な場合があります。

合成および配置配線段階で信号名を保持するには、合成属性 keep または preserve を使用します。これらは Verilog および VHDL の合成属性であり、解析および合成時に特定のワイヤ、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。

 これらの属性の使用について詳しくは、「Quartus II ハンドブック v1」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 シミュレータ

「Quartus II ハンドブック」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、および必要なファイルとこれらのファイルが配置されているディレクトリの詳細について説明しています。

 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 v3」の「Simulation」セクションの該当する章を参照してください。

デザイン例：タップを備えたシフト・レジスタ

このデザイン例の目的は、シフト・レジスタ（RAM ベース）の MegaWizard Plug-In Manager を使用して作成された ALTSHIFT_TAPS メガファンクションを実装し、インスタンス化することです。この例では、8 ビットのデータ幅 (w) を持つシフト・レジスタ、3 のタップ距離 (m) および 4 に等しいタップ数 (n) を使用します。また、シフト・レジスタ・チェーンの特定なポイントでデータをタップする方法を示します。

デザイン・ファイル

デザイン・ファイル例は、Altera[®] ウェブサイト (www.altera.co.jp) の「ユーザーガイド」のセクションで入手できます。

コンフィギュレーションの設定

シフト・レジスタ（RAM ベース）MegaWizard Plug-In Manager の 3 ページ目で、表 2-2 に示されているコンフィギュレーション設定を選択、または確認します。Next をクリックして、次のページを表示します。

表 2-2. シフト・レジスタ（RAM ベース）MegaWizard Plug-In Manager のコンフィギュレーション設定

コンフィギュレーション設定	値
Currently selected device family	Stratix III
How wide should the 'shiftin' input and the 'shiftout' output buses be?	8 ビット
How many taps would you like?	4
Create groups for each tap output	選択した
How wide should the distance between taps be?	3
Create a clock enable port	選択した
Create an asynchronous clear port	選択した
What should the RAM block type be?	自動

ModelSim-Altera シミュレータの機能シミュレーション

デバイス動作の波形表示を生成するために ModelSim[®]-Altera ソフトウェアのデザインをシミュレートします。

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する必要があります。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 (www.altera.co.jp) のソフトウェア製品のサポート・ページ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。

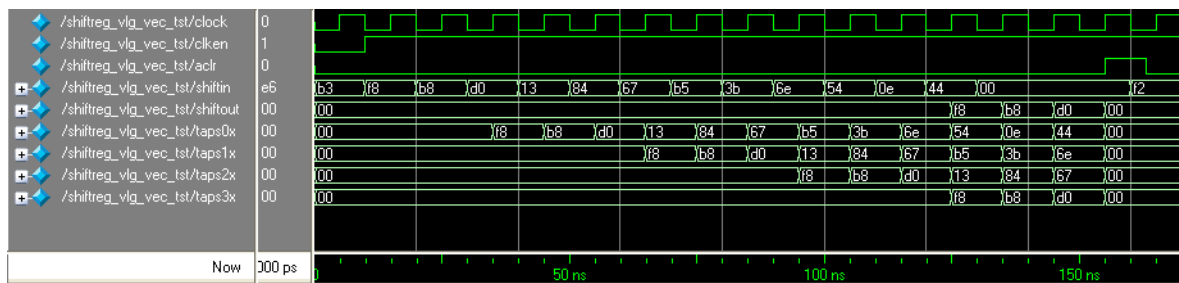
以下のステップを実行して、ModelSim-Altera ソフトウェアを設定し、シミュレートします。

1. **DE_ALTSHIFT_TAPS.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. ModelSim-Altera ソフトウェアを起動します。
3. File メニューの **Change Directory** をクリックします。
4. ファイルを解凍したフォルダを選択します。
5. **OK** をクリックします。
6. Tools メニューの **Execute Macro** をクリックします。
7. **DE_ALTSHIFT_TAPS.do** ファイルを選択し、**Open** をクリックします。
DE_ALTSHIFT_TAPS.do 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。

Wave ウィンドウにシミュレーション結果を示します。図 2-7 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 2-7. Simulation Waveform for Shift Register with Taps Design Example



シミュレーション結果の理解

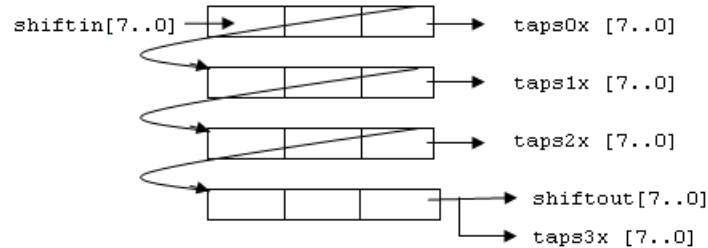
この例では、次のプロパティを持つようにシフト・レジスタをコンフィギュレーションされます。

- 8 ビットのデータ幅
- 3 に等しいタップ間の距離（タップ長）
- 4 に等しいタップ数
- 各タップの出力にグループを作成
- クロック・イネーブル信号および非同期クリア信号を作成

この例では、すべての 12 ワード・データがシフト・レジスタにシフトされた時、1-4-7-10 番目のデータ・ワード（2-5-8-11 番目および 3-6-9-12 番目が続く）を同時にタップする方法を示します。

図 2-8 に、この例の ALTSHIFT_TAPS メガファンクションで設定したコンフィギュレーションに類似するシフト・レジスタ・チェーンを示します。

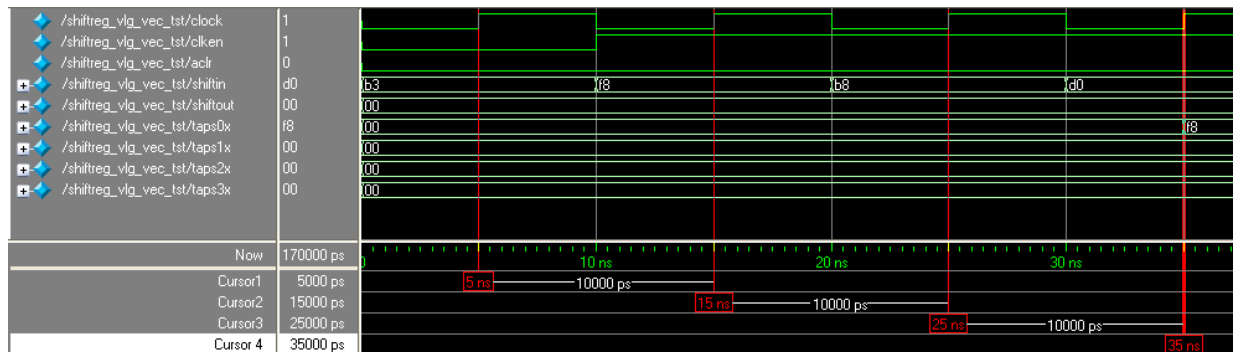
図 2-8. コンフィギュレーションされた ALTSHIFT_TAPS メガファンクションにシフト・レジスタ・チェーンのアナロジー



次のセクションでは、このシフト・レジスタ・チェーンを使用して ALTSHIFT_TAPS メガファンクションのシフト動作と出力動作を説明します。

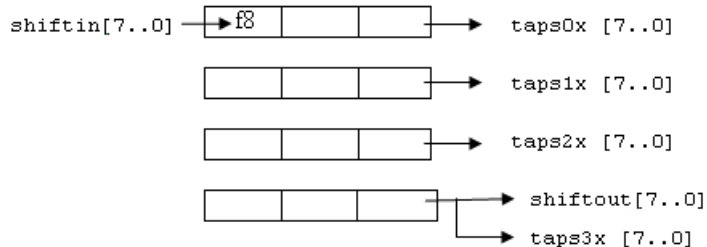
図 2-9 に、シフト・レジスタ・チェーンに書き込まれ、シフトされた最初の 3 つのデータ・ワード、および taps0x 出力で表示された最初のデータを示します。

図 2-9. シフト・レジスタに書き込まれ、シフトされた最初の 3 つのデータ



5 ns で、clken 信号は Low であり、したがって、何も動作が実行されません。この動作が開始するとして、最初の立ち上がりクロック・エッジは 15 ns 見なすことができます。図 2-10 に示すように、最初のデータ F8 は、シフト・レジスタにシフトされます。データが出力のいずれかにシフトされていないため、すべての出力は 00 を示します。

図 2-10. 15 ns でシフト・レジスタ・チェーンの内容



25 ns および 35 ns で、それぞれ 2 番目のデータ B8 および 3 番目のデータ D0 をシフト・レジスタにシフトされます。


 シフト・レジスタ・チェーンの既存のデータは、新しいデータのシフト・インの直前にシフトされます。

図 2-11 に、35 ns でシフト・レジスタ・チェーンの内容を示します。すべての出力は、00 を示しますが、taps0x の出力を除いて、最初のデータ F8 を示します。


 入力と出力のデータ・ポートはいずれもレジスタされていません。シフト・レジスタ内のメモリ・ブロックのアドレス・ポートのみをレジスタされます。したがって、出力ポートのいずれかでデータをシフトされた時、データはそれぞれの出力ポートですぐに表示されます。

図 2-11. 35 ns でシフト・レジスタ・チェーンの内容

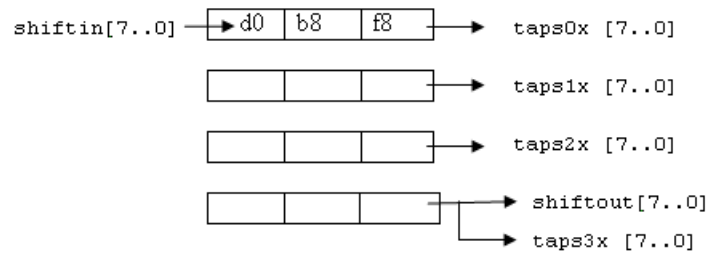
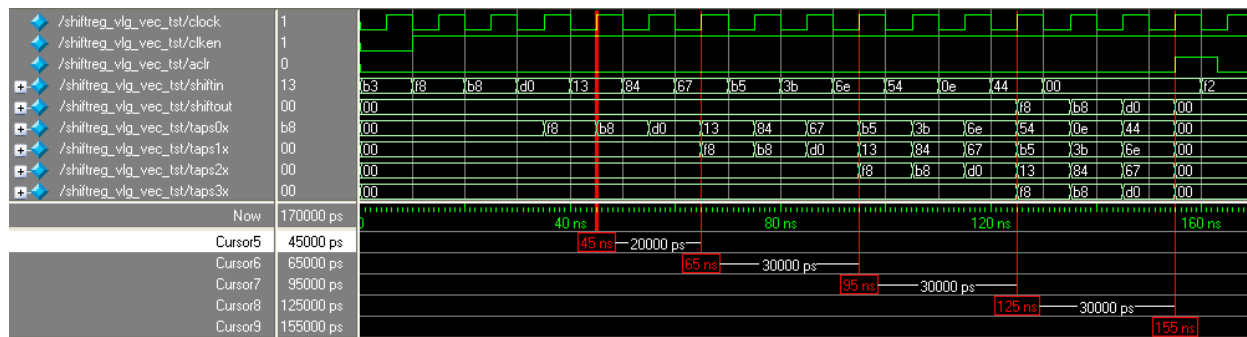


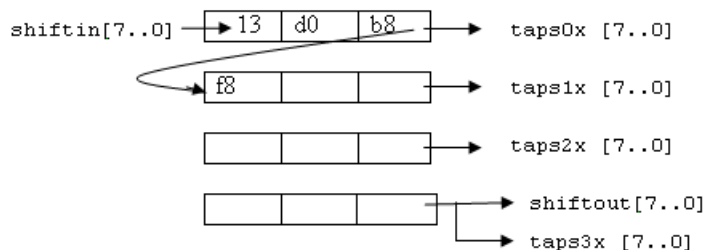
図 2-12 に、等間隔の間隔で、シフト・レジスタ・チェーンからのデータ・シフトと出力タッピングを示します。

図 2-12. データのシフトおよび出力のタッピング



45 ns で、[図 2-13](#) に示すように、最初のデータ F8 はタップの次の列にシフトされ、2 番目のデータ B8 は taps0x にシフトされます。他の出力ポートは、00 を示し続けます。また、同じ立ち上がりクロック・エッジで、新しいデータ 13 は、シフト・レジスタにシフトされます。

図 2-13. 45 ns でシフト・レジスタ・チェーンの内容

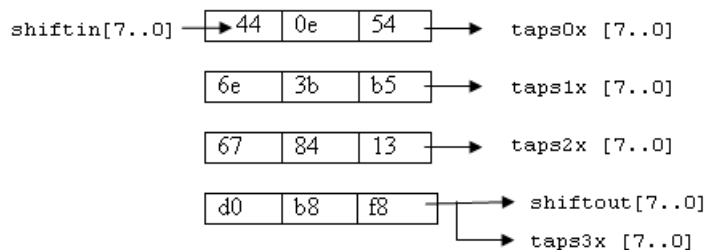


65 ns で、最初のデータ F8、および 4 番目のデータ 13 は、それぞれ taps1x と taps0x にシフトされます。95 ns で、最初のデータ F8、4 番目のデータ 13、および 7 番目のデータ B5 は、それぞれ taps2x、taps1x、および taps0x にシフトされます。最後に、125 ns で、すべての 12 のデータ・ワードはシフト・レジスタにシフトされます。そして、それぞれ taps3x、taps2x、taps1x、および taps0x に 1-4-7-10 番目のデータ・ワードを同時にタップをして、起動することができます。

shiftout 出力は taps3x に相当し、両方のポートは、同じ出力データを生成します。

次の立ち上がりクロック・エッジで、2-5-8-11 番目のデータ・ワードをタップすることができます。そして、次の立ち上がりエッジで、3-6-9-12 番目のデータ・ワードをタップ続きます。[図 2-14](#) に、すべての 12 のデータ・ワードはシフト・レジスタにシフトされるときに、シフト・レジスタ・チェーンの内容を示します。

図 2-14. 125 ns でシフト・レジスタ・チェーンの内容



155 ns で、すべてのデータをタップ・アウトされた後、出力ポートのデータとシフト・レジスタの内容をすぐにクリアするために aclr 信号をアサートすることができます。そして、次の 12 ワードのデータをシフトすることを開始できます。

このデザイン例では、動作のシフトとタッピングする方法を示します。それは、特定のアプリケーションの使用状況を表示するものではありません。ニーズに合わせて、追加のロジックでタッピング機能を使用することができます。

結論

シフト・レジスタはデジタル信号処理（DSP）アプリケーションに広く使用されています。標準的なフリップ・フロップで実装されている従来のシフト・レジスタに比べて、ALTSHIFT_TAPS メガファンクションは、DSP アプリケーションに適しています。それは、メガファンクションは、ロジック・セルと配線リソースを節約し、そしてより大きなメモリ容量を提供するエンベデッド・メモリ・ブロックを使用して、実装されるためです。

また、ALTSHIFT_TAPS メガファンクションは、特定の固定点でのデータをタップすることができるタップ機能が装備されています。選択可能な入力データ幅、タップの長さ、およびタップ数は、必要なシフト・レジスタの柔軟性のコンフィギュレーションを提供します。

この章では、ALTSHIFT_TAPS メガファンクションのプロトタイプ、宣言、ポート、およびパラメータについて説明します。アプリケーションに応じて ALTSHIFT_TAPS メガファンクションをカスタマイズするために、ポートとパラメータを使用することができます。

ALTSHIFT_TAPS メガファンクションの Verilog HDL プロトタイプ

次の Verilog HDL プロトタイプは、<Quartus II installation directory>\eda\synthesis ディレクトリの Verilog デザイン・ファイル (.v) **altera_mf.v** に配置することができます。

```

module    altshift_taps
#(
    parameter    intended_device_family = "unused",
    parameter    number_of_taps = 1,
    parameter    power_up_state = "CLEARED",
    parameter    taps_distance = 1,
    parameter    width = 1,
    parameter    lpm_type = "altshift_taps",
    parameter    lpm_hint = "unused")
(
    input wire    aclr,
    input wire    clken,
    input wire    clock,
    input wire    [width-1:0]    shiftin,
    output wire   [width-1:0]    shiftout,
    output wire   [width*number_of_taps-1:0]    taps)/*synthesis syn_black_box=1 */;
endmodule \\altshift_taps

```

ALTSHIFT_TAPS メガファンクションの VHDL コンポーネント宣言

次の VHDL デザイン・ファイル (.vhd) **altera_mf.vhd** は、<Quartus II installation directory>\libraries\bhd\altera_mf ディレクトリに配置することができます。

```

component altshift_taps
    generic (
        intended_device_family : string := "unused";
        number_of_taps : natural;
        power_up_state : string := "CLEARED";
        tap_distance : natural;
        width : natural;
        lpm_hint : string := "UNUSED";
        lpm_type : string := "altshift_taps"
    );
    port (
        aclr : in std_logic := '0';
        clken : in std_logic := '1';
        clock : in std_logic;
        shiftin : in std_logic_vector(width-1 downto 0);
        shiftout : out std_logic_vector(width-1 downto 0);
        taps : out std_logic_vector(width*number_of_taps-1 downto 0)
    );
end component;

```

VHDL Library-Use の宣言

VHDL LIBRARY-USE の宣言は、VHDL コンポーネント宣言をしようするとき必要とされません。

```
LIBRARY alterea_mf;
USE altera_mf.altera_mf_components.all;
```

ALTSHIFT_TAPS メガファンクションのポートとパラメータ

図 3-1 に、ALTSHIFT_TAPS メガファンクションのポートとパラメータを示します。

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ユーザー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 インタフェースのユーザーには見えません。

図 3-1. シフト・レジスタ (RAM ベース) ポートとパラメータ

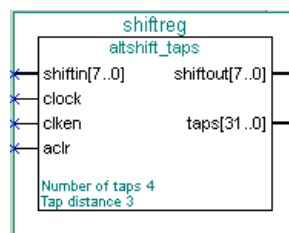


表 3-1 に、ALTSHIFT_TAPS メガファンクションの入力ポートを示します。

表 3-1. シフト・レジスタ (RAM ベース) MegaWizard Plug-In Manager の入力ポート

名称	必須	説明
shiftin[]	あり	シフターへのデータ入力。入力ポート WIDTH ビット幅です。
clock	あり	ポジティブ・エッジでトリガされるクロックです。
clken	なし	クロック・ポートのクロック・イネーブル。clken は V _{CC} にデフォルトします。
aclr	なし	シフト・レジスタ・チェーンの内容を非同期にクリアします。shiftout 出力は、aclr0 のアサーションで、すぐにクリアされます。

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

表 3-2. シフト・レジスタ (RAM ベース) MegaWizard Plug-In Manager の出力ポート

名称	必須	説明
shiftout []	あり	データ・レジスタの終了からの出力。出力ポート WIDTH ビット幅です。
taps []	あり	シフト・レジスタに沿って等間隔のタップから出力。出力ポート WIDTH * NUMBER_OF_TAPS 幅です。このポートは、シフト・レジスタに沿ってすべての等間隔のタップ (各 WIDTH ビット) の合計です。

表 3-3 に、ALTSHIFT_TAPS メガファンクションのパラメータを示します。

表 3-3. シフト・レジスタ (RAM ベース) MegaWizard Plug_In のパラメータ

名称	タイプ	必須	説明	
NUMBER_OF_TAPS	整数	あり	シフト・レジスタに沿って等間隔のタップの数を指定します。	
TAP_DISTANCE	整数	あり	クロック・サイクルの等間隔のタップ間の距離を指定します。この数は、使用される RAM のワード数に変換します。TAP_DISTANCE は、少なくとも 3 でなければなりません。	
WIDTH	整数	あり	入力パターンの幅を指定します。	
POWER_UP_STATE	文字列	なし	パワーアップ時にシフト・レジスタの内容を指定します。値は、CLEARED および DONT_CARE です。省略され場合、デフォルトは CLEARED になります。	
			値	説明
			CLEARED	ゼロ内容です。Stratix および Stratix II デバイス・ファミリの場合、M512 または M4K RAM ブロックを使用する必要があります。
			DONT_CARE	不定の内容です。この設定では、M-RAM ブロックを使用できません。

この章では、このドキュメントとアルテラの追加の情報について説明します。

改訂履歴

以下の表に、本資料の改訂履歴を示します。

日付	バージョン	変更内容
2010年11月	2.1	<ul style="list-style-type: none"> ■ ポートとパラメータを更新。 ■ プロトタイプとコンポーネント宣言を追加。
2008年7月	2.0	<ul style="list-style-type: none"> ■ このメガファンクションにサポートされるデバイス・ファミリのリストを更新。 ■ メガファンクションの機能および動作を示す説明で新しいデザイン例を作成。 ■ 新しい入力ピン ac1r の説明を追加。 ■ ドキュメント全体を再編成。
2007年3月	1.2	Cyclone [®] III サポートを追加。
2006年12月	1.1	Stratix [®] III のサポートを追加。
2006年9月	1.0	初版。

アルテラへのお問い合わせ

アルテラ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。









お問合せ先 (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	www.altera.co.jp/support
技術トレーニング	ウェブサイト	www.altera.co.jp/training
	電子メール	custrain@altera.com
製品資料	ウェブサイト	www.altera.co.jp/literature
一般的なお問い合わせ ソフトウェア・ライセンス に関するお問い合わせ	電子メール	nacomp@altera.com
	電子メール	authorization@altera.com

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

本書では、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、 Save As ダイアログ・ボックス。GUI エLEMENT の場合、大文字は GUI と一致します。
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、ソフトウェア・ユーティリティ名および GUI ラベルを表します。例： <code>\qdesigns</code> ディレクトリ、 D: ドライブ、および chiptrip.gdf 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 デザイン・ガイドライン</i> 。
斜体	変数を表します。例： $n+1$ 。 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) .pof 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> 。 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> 。 また、 Report 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。
	矢印は、 Enter キーを押すことを示しています。
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。
	疑問符は、関連情報を持つソフトウェア・ヘルプ・システムを案内しています。
	足跡マークは、詳細情報の参照先を示しています。
	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。
	エンベロープは、アルテラ・ウェブサイトの「 メール配信サービス・センター 」ページへのリンクです。ここでは、アルテラの文書の更新通知を受け取るためにサインアップすることができます。