

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

Q1152017-8.0.0

### はじめに

プログラマブル・ロジックは、デザイン・サイクルの後半でのシステム仕様の変更に対応できます。一般的なエンジニアリング・プロジェクト開発サイクルでは、プログラマブル・ロジック部分の仕様は、エンジニアリング・プロジェクト開発の開始後、またはすべてのシステム・エレメントの統合時に変更される可能性があります。

このようなデザインの最終段階での変更は、一般に ECO (Engineering Change Orders) と呼ばれ、デザインがフルにコンパイルされた後に行われるデザイン機能の小さな変更です。デザインは合成および配置配線が完了するとフル・コンパイルされます。

Chip Planner は、デザイン・サイクル後半でロジックへの迅速かつ効率的な変更を可能にすることにより、ECO をサポートします。Chip Planner は、選択された FPGA のデバイス・アーキテクチャにマップされた配置配線後のデザインを、デバイスの LAB 配置から、マップされた各ロジック・エレメント (LE) またはアダプティブ・ロジック・モジュール (ALM) まで視覚的に表示します。この視覚表示でデザインを解析して、ECO をサポートするためのデバイス・リソースのマップ方法を変更することができます。

この章では、ECO がデザイン・サイクルに及ぼす影響を検討し、ECO を実行するためのデザイン・フローおよび ECO を実行するための Chip Planner の使用方法について説明します。



Chip Planner は ECO の実行のほかに、配線の輻輳、関連するリソース使用量、ロジック配置、LogicLock™、およびカスタム領域、ファンインおよびファンアウト、レジスタ間のパス、およびパスの遅延推定値についての詳細な解析を可能にします。



デザイン解析のための Chip Planner の使用について詳しくは、「Quartus II ハンドブック Volume 2」の「デザイン・フロアプランの解析および最適化」の章を参照してください。

Chip Planner は、以下のデバイス・ファミリをサポートしていません。

- MAX® 7000
- ACEX®
- APEX™ シリーズ
- FLEX® シリーズ

これらのデバイスを使用したフロアプラン解析には、タイミング・クロージャ・フロアプランを使用します。ECO はこれらのデバイス・ファミリーではサポートされていません。

## ECO

ECO は通常、デザイン・サイクルの検証ステージで実行されます。デザインで、PLL を別のクロック周波数に変更したり、解析のためにピンに信号を配線するなど小さな変更が必要なおと、特に大規模なデザインではデザイン全体をリコンパイルすると時間がかかることがあります。検証サイクル中に小さなデザイン変更の繰り返しは何回も発生する可能性があるため、細かなリコンパイル時間が積もって、すぐに長い時間になることがあります。さらに、小さなデザイン変更のためにフル・リコンパイルを行うと、以前のデザインの最適化が失われる場合があります。デザインに対してフル・リコンパイルを行わないで ECO を行うと、変更はロジックの影響を受ける部分にのみ限定されます。

この項では、ECO がシステム・デザインに影響を与える領域、およびユーザーがそれらの領域でデザインを最適化するのに Quartus® II ソフトウェアがどのように役立つかを説明します。この項では、以下のトピックについて説明します。

- 「性能」
- 13-3 ページの「コンパイル時間」
- 13-3 ページの「検証」
- 13-4 ページの「ドキュメント」

### 性能

デザイン機能に小さな変更を加えると、以前のデザイン最適化が失われる可能性があります。デザイン最適化の代表的な例は、フロアプラン最適化とフィジカル・シンセシスです。理想的には、以前のデザインの最適化を維持する必要があります。

Chip Planner により、デザインの配置配線後のデータベース上で直接 ECO を実行することができます。加えた変更はすべて影響があるデバイス・リソースに限定されるため、デザインの残り部分のタイミング性能は影響を受けません。Chip Planner は、すべての変更に対してデザイン・ルール・チェックを実行して、デザインの不正な変更を防止します。

また、Quartus II ソフトウェアは、リコンパイル中にデザインの最適化および配置を維持するインクリメンタル・コンパイル機能を提供します。この機能により、デザインのパーティションを作成できるため、デザインの完全な配置および最適化の後で変更が必要な場合、影響を受けるパーティションのみリコンパイルされて変更が実装されます。

インクリメンタル・コンパイル・フローは、Chip Planner による ECO の実行を完全にサポートしています。

Quartus II インクリメンタル・コンパイルを有効にしてプロジェクトをリコンパイルすると、コンパイラは変更されなかったパーティションで Chip Planner が実行したすべての ECO を維持します。



Quartus II ソフトウェアでのインクリメンタル・コンパイル機能の使用方法について詳しくは、「Quartus II ハンドブック Volume 1」の「階層およびチーム・ベース・デザインのための Quartus II インクリメンタル・コンパイル」の章を参照してください。

ECO フローでのインクリメンタル・コンパイルの使用について詳しくは、13-38 ページの「ECO フローにおけるインクリメンタル・コンパイルの使用」を参照してください。

## コンパイル時間

従来のプログラマブル・ロジック・デザイン・フローでは、デザインの小さな変更にはデザインの完全なリコンパイルが必要です。デザインの完全なリコンパイルは合成と配置配線から成ります。デザインに小さな変更を行ってからボード上で最終実装に至るまで、長いプロセスとなる場合があります。Chip Planner は配置配線後のデータベース上でのみ動作するため、フル・コンパイルを実行しないで数分でデザイン変更を実装することができます。

## 検証

デザイン変更後に、デザインへの影響を検証できます。タイミングに違反していないことを検証するには、Chip Planner 内でネットリストの変更点をチェックし保存した後、Quartus II クラシック・タイミング・アナライザまたは Quartus II TimeQuest タイミング・アナライザを使用してスタティック・タイミング解析を実行できます。



Quartus II TimeQuest タイミング・アナライザについて詳しくは、「Quartus II ハンドブック Volume 3」の「Quartus II TimeQuest Timing Analyzer」の章を参照してください。Quartus II クラシック・タイミング・アナライザについて詳しくは、「Quartus II ハンドブック Volume 3」の「Quartus II クラシック・タイミング・アナライザ」の章を参照してください。

さらに、Quartus II ソフトウェアにより生成された配置配線後のネットリストを使用して、ECO 変更されたデザインのゲート・レベル・シミュレーションまたはタイミング・シミュレーションを実行できます。

## ドキュメント

Chip Planner で行われたすべての ECO は、すべての変更の追跡記録を提供するために Change Manager にログされます。Change Manager を使用すると、簡単に元のフィッティング後のネットリストに戻ったり、適用する ECO を選択できます。

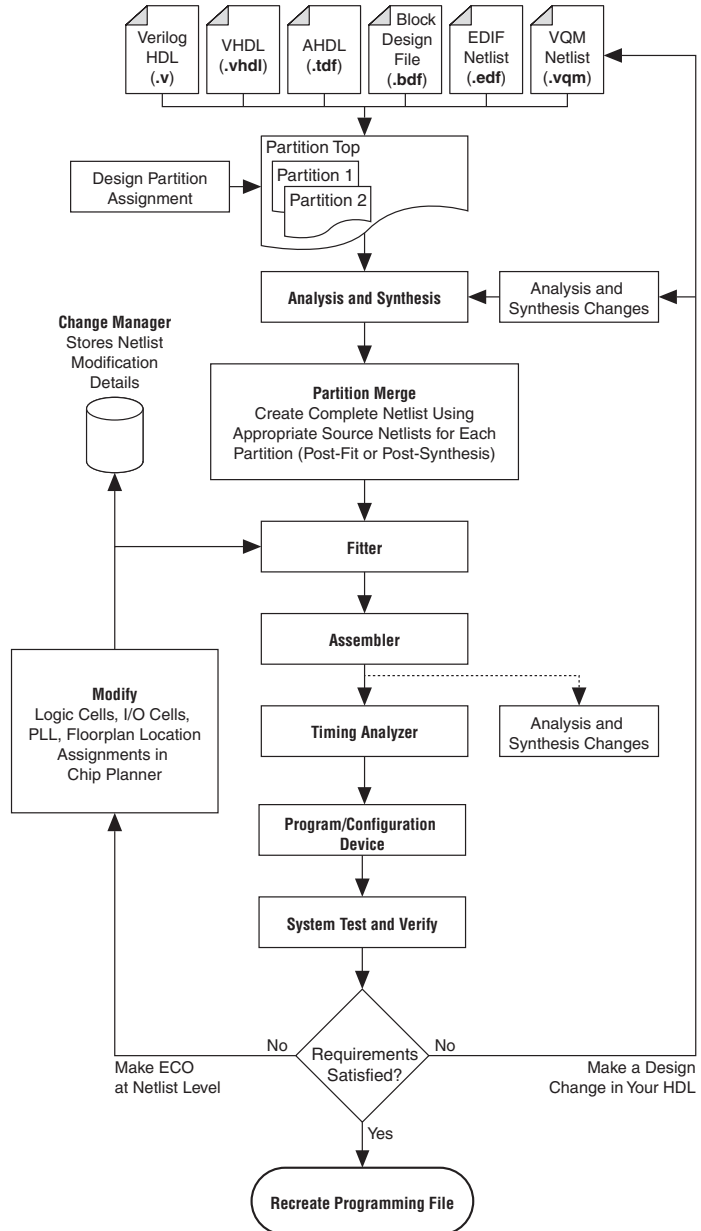
Change Manager について詳しくは、13–36 ページの「[Change Manager](#)」を参照してください。

さらに、Quartus II ソフトウェアは、同じプロジェクトの複数コンパイル・リビジョンをサポートします。Chip Planner で行った ECO とリビジョン・サポートを併用すれば、いくつかの異なる ECO 変更を比較して、以前のプロジェクト・リビジョンに戻る機能を提供することができます。

# ECO の デザイン・ フロー

図 13-1 に、ECO を実行するためのデザイン・フローを示します。

図 13-1. ECO 変更をサポートするデザイン・フロー



繰り返し検証サイクルでは、ネットリスト・レベルでの小さなデザイン変更の実装は、RTL コードの変更を行うより速い場合があります。このように、ECO 変更は、シリコン上でデザインをデバッグし、システム・デバッグ用のプログラミング・ファイルを生成するのに高速ターンアラウンドが必要なときに特に役立ちます。

ECO の代表的な用途には、ボードで問題を発見し、それをデバイス上の適切なノードまたは I/O セルに分離するときの利用が含まれます。機能を素早く訂正して、新しいプログラミング・ファイルを生成できなければなりません。Chip Planner を使用して小さな変更を行うと、配置配線後のネットリストを直接修正できます。これにより、合成およびロジック・マッピングを実行する必要がなくなり、検証サイクル中のプログラミング・ファイルの生成のためのターンアラウンド時間が短縮されます。変更により問題が訂正された場合は、HDL ソース・コードを修正する必要はありません。Chip Planner を使用して、デザインに対して以下の ECO 関連の変更を実行することができます。

- Change Manager で行われる変更を記録する
- デザイン変更を行うために必要なステップを簡単に再作成する
- デザイン検証のための EDA シミュレーション・ネットリストを生成する
- デザインでスタティック・タイミング解析を実行する



Quartus II ソフトウェアは、HDL ソース・コードの修正が必要となるより複雑な変更に対するインクリメンタル・リコンパイルでのリコンパイル時間の役立ちます。

## Chip Planner の概要

Chip Planner はデバイス・リソースを視覚的に表示します。Chip Planner は、ターゲットのデバイス・アーキテクチャにおけるリソース Atom の配置と使用量を表示します。リソース Atom は、ALM、LE、PLL、DSP ブロック、メモリ・ブロック、または IOE などのデバイスのためのビルディング・ブロックです。

Chip Planner は、デザイン解析および配置配線後のデザインに対する ECO のための統合プラットフォームも提供します。このツールセットは、Chip Planner (マップされたデザインのデバイス・フロアプラン・ビューを提供) と 2 つの統合サブツール —Resource Property Editor と Change Manager— から成ります。

解析では Chip Planner は、ロジック配置、LogicLock およびカスタム領域、関連するリソース使用量、詳細な配線情報、配線の輻輳、ファンインおよびファンアウト、レジスタ間のパス、およびパスの遅延推定値を表示することができます。さらに、Chip Planner により、デバイス・フ

ロアプランを使用したロジック・セルまたは I/O 素子の移動や削除などの位置の制約またはリソース・アサインメントの変更を作成できます。ECO 変更では、Chip Planner で高速にプログラミング・ファイルを生成するために、配置配線後のネットリスト内のロジック・セルを作成、移動、または削除することができます。また、Chip Planner から Resource Property Editor を開いて、リソース Atom プロパティの編集や、それらの間の接続の編集を行うことができます。リソース Atom および接続への変更はすべて、Change Manager により自動的に記録されます。

## Chip Planner を開く

Chip Planner を開くには、Tools メニューの **Chip Planner** をクリックします。または、Quartus II ソフトウェアのツールバーの **Chip Planner** アイコンをクリックします。

Quartus II ソフトウェアはオプションで、Chip Planner を開くためのクロス・プロービングをサポートします。クロス・プロービングで Chip Planner を開くには、以下のツールのショートカット・メニューを使用します。

- Compilation Report
- Project Navigator ウィンドウ
- RTL ソース・コード
- Timing Closure Floorplan
- Node Finder
- Simulation Report
- RTL Viewer

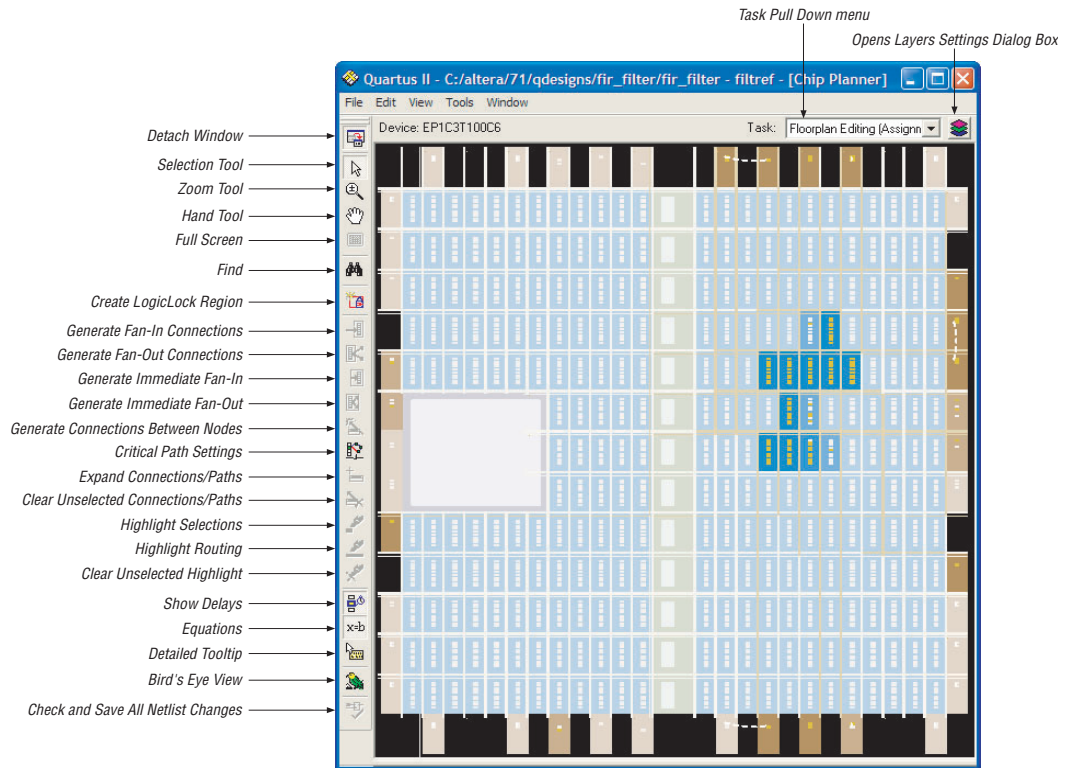


Timing Closure Floorplan について詳しくは、「Quartus II ハンドブック Volume 2」の「[デザイン・フロアプランの解析および最適化](#)」の章を参照してください。

## Chip Planner ツールバー

Chip Planner は、使いやすい GUI によりデザイン解析のための機能を提供します。Chip Planner の多くの機能は、メニュー項目から、あるいはツールバーのアイコンをクリックすることによって実行できます。[図 13-2](#) に、Chip Planner ツールバーの例を示し、一般的に使用されるアイコンについて説明します。

図 13-2. Chip Planner ツールバー



Chip Planner ツールバーのアイコンはカスタマイズすることもできます。Chip Planner ウィンドウがアタッチされている場合に、アイコン・ツールバーをカスタマイズするには、Tools メニューの **Customize Chip Planner** をクリックします。Chip Planner ウィンドウがデタッチされている場合は、Tools メニューの **Customize** をクリックします。



デザイン解析のための Chip Planner の使用について詳しくは、「Quartus II ハンドブック Volume 2」の「デザイン・フロアプランの解析および最適化」の章を参照してください。

## Chip Planner のタスクおよびレイヤ

Chip Planner では、タスクをセットアップして、ECO 変更を素早く実装したり、デバイスのフロアプランのアサインメントを操作することができます。各タスクは、編集モードと 1 組のカスタマイズされたレイヤ設定から成ります。

Chip Planner で使用できる編集モードは、Assignment モードと ECO モードです。Assignment モードでは、LogicLock 領域を作成または操作でき、デザインで使用されるリソース Atom にロケーション制約を作成することができます。アサインメントは、Quartus II 設定ファイル (.qsf) および Assignment Editor に反映されます。ECO モードでは、Atom を作成したり、削除したり、既存の Atom を異なるロケーションに移動することができます。ECO モードで行われる変更は配置配線後のデータベース内で行われます。両方のモードでデザインを解析できます。

レイヤ設定により、特定のタスクに対して表示されるグラフィック・エレメントを指定できます。特定のグラフィック・エレメントの表示をオフにして、ウィンドウのリフレッシュ速度を高め、複雑なデザインを表示するときに現れる乱れを減らすことができます。バックグラウンド・カラー・マップは、デバイスの異なるエリアに対するリソース使用率の相対レベルを示します。例えば、**Routing Utilization** は相対的な配線使用率、**Physical Timing Estimate** は相対的な物理的タイミングを示します。

Chip Planner には、ECO 変更の迅速な実装やデバイスのフロアプランのアサインメント操作を可能にする定義済みタスクがあります。Chip Planner は、以下の定義済みタスクを提供します。

- Post-Compilation Editing (ECO)
- Floorplan Editing (Assignment)
- Partition Display (Assignment)
- Global Clock Network (Assignment)
- Power Analysis (Assignment) — Stratix® III、Stratix II、Stratix II GX、Cyclone® III、Cyclone II、および HardCopy® II デバイスに対してのみ使用可能

Chip Planner フロアプラン・ビューの右上隅にある **Task** プルダウン・メニューを選択することにより、定義済みタスクを選択できます。

ユーザー独自のタスクをカスタマイズするには、Task プルダウン・メニューの隣の **Layers** アイコンをクリックして、**Layers Settings** ダイアログ・ボックスを開きます。



Chip Planner でのアサインメントおよび解析について詳しくは、「Quartus II ハンドブック Volume 2」の「[デザイン・フロアプランの解析および最適化](#)」の章を参照してください。

ECO モードを使用した ECO の実行について詳しくは、[13–15 ページの「Chip Planner での ECO の実行 \(フロアプラン・ビュー\)」](#)を参照してください。

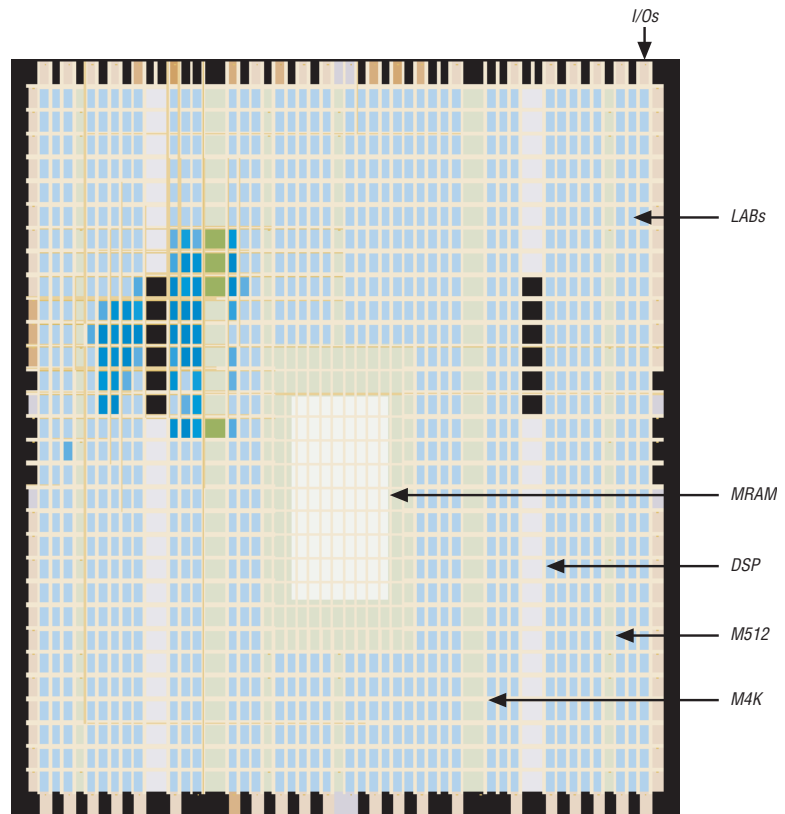
## Chip Planner フロアプラン・ビュー

Chip Planner は、ターゲットのアルテラ・デバイスの様々な抽象化レベルを示す階層ズーム・ビューワを使用しています。ズーム・レベルを上げると、抽象化レベルが下がり、デザインがより詳細に表示されます。

### ファースト・レベル・ビュー

ファースト・ズーム・レベルは、デバイス・フロアプラン全体のハイレベルな表示を提供します。このビューは、Quartus II Timing Closure Floorplan の Field View に類似する詳細レベルを提供します。デザインでの任意ノードの配置を検索および表示することができます。[図 13–3](#)に、Stratix デバイスの Chip Planner フロアプランのファースト・レベル・ビューを示します。

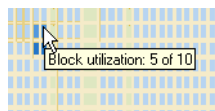
図 13-3. Chip Planner のファースト・レベル (最高) フロアプラン・ビュー (Stratix ファミリ・デバイス)



各リソースは互いに区別しやすいように異なる色で表示されます。Chip Planner フロアプランでは階調カラー方式を採用しており、リソースの使用率が高くなるほど色が濃くなります。例えば、LAB で LE の使用が増えると、LAB の色が濃くなります。

マウス・ポインタをこのレベルでのリソースの上に置くと、高いレベルでのリソースの使用率を示すツールチップが表示されます (図 13-4)。

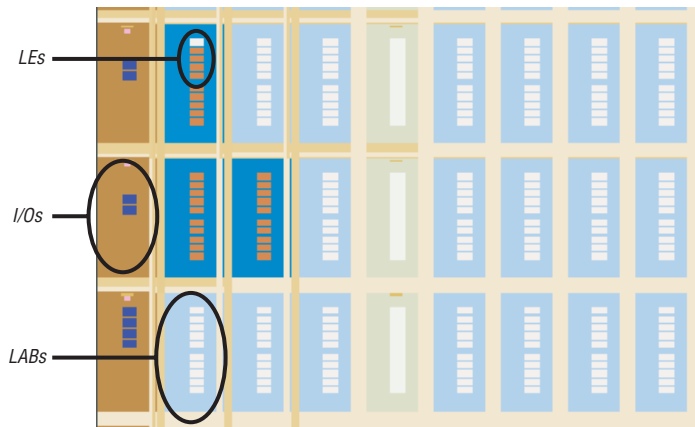
図 13-4. ツールチップ・メッセージ：ファースト・レベル・ビュー



## セカンド・レベル・ビュー

拡大表示すると、詳細レベルが高くなります。図 13-5 に、Stratix デバイスの Chip Planner フロアプランのセカンド・レベル・ビューを示します。

図 13-5. Chip Planner のセカンド・レベル・フロアプラン・ビュー (Stratix ファミリー・デバイス)



このレベルでは、LAB および I/O バンクの内容を確認することができます。リソースの接続に使用される配線チャンネルも確認できます。このレベルでは、マウス・ポインタを LE または ALM の上に置くと、LE/ALM 名、LE/ALM の位置、およびその LAB で使用されるリソース数を示すツールチップが表示されます (図 13-6)。マウス・ポインタをインタコネクタ上に置くと、ツールチップはそのインタコネクタで使用される配線チャンネルを示します。

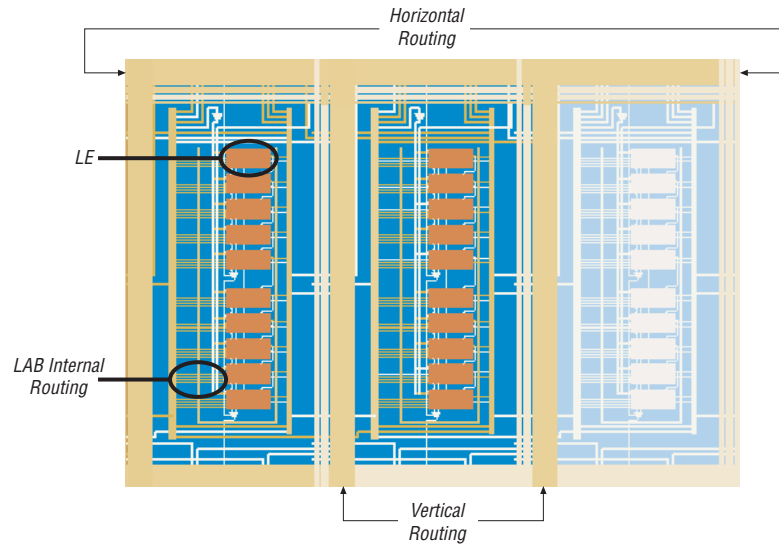
図 13-6. ツールチップ・メッセージ：セカンド・レベル・ビュー



## サード・レベル・ビュー

サード・レベルでは、最大の詳細レベルが提供され、FPGA の LAB 内で使用される各配線リソースを確認することができます。図 13-7 に、Stratix デバイスのサード・レベル・ビューにおける詳細レベルを示します。

図 13-7. Chip Planner のサード・レベル・ビュー



セカンド・レベルおよびサード・レベルの拡大表示からは、LE、ALM、およびI/Oをある物理的位置から別の位置に移動できます。リソースは、選択して希望の位置にドラッグ・アンド・ドロップして移動できます。これらのレベルでは、新しいLEおよびI/Oを作成することも可能です。

Atom の作成、Atom の削除、またはデバイス素子のリアロケートについて詳しくは、13-15 ページの「[Chip Planner](#)でのECOの実行（フロアプラン・ビュー）」の項を参照してください。

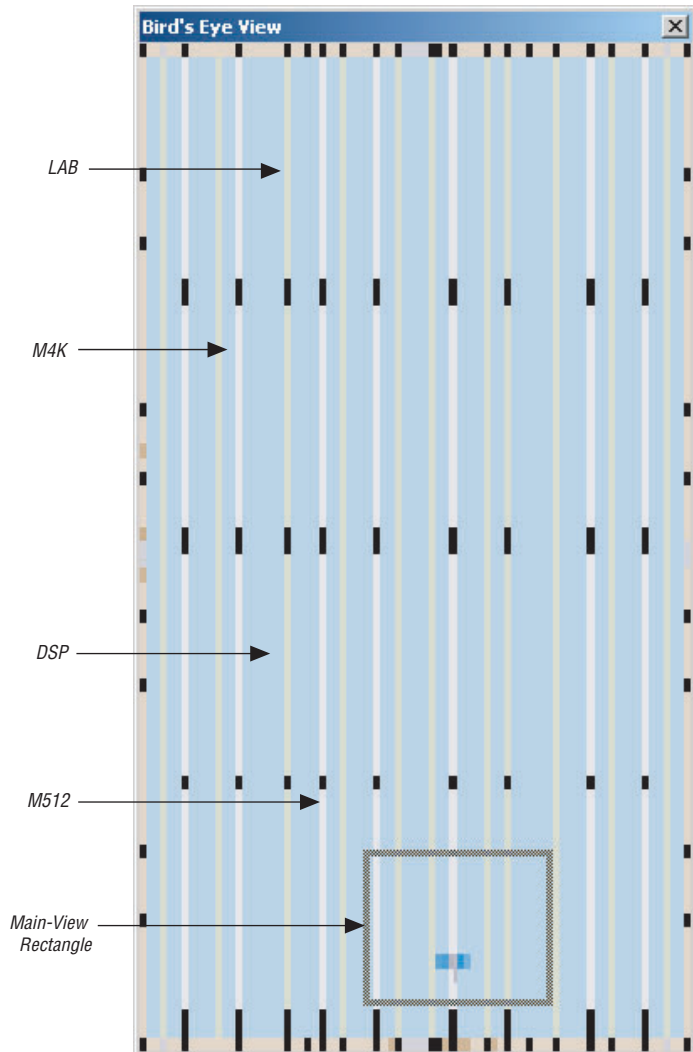


フロアプラン・アサインメントの作成について詳しくは、「[Quartus II](#) ハンドブック Volume 2」の「[デザイン・フロアプランの解析および最適化](#)」の章を参照してください。

## Bird's Eye View

Bird's Eye View (図 13-8) では、チップ全体のリソース使用量を高レベル画像で表示し、高速かつ効率的に Chip Planner 内をナビゲートします。

図 13-8. Bird's Eye View



Bird's Eye View は、Chip Planner フロアプランにリンクされている独立したウィンドウとして表示されます。Bird's Eye View 内のエリアを選択すると、Chip Planner フロアプランが自動的にリフレッシュされ、選択されたデバイスの領域を表示します。Bird's Eye View ウィンドウのメイン表示矩形のサイズを変更すると、Chip Planner フロアプラン・ウィンドウも拡大 (または縮小) します。Bird's Eye View のメイン表示矩形を縮小すると、Chip Planner フロアプラン・ウィンドウでデザインがより詳細に表示されます。

Bird's Eye View は、表示したいデザインの部分がチップの反対側にあり、基準座標系を失わずにリソース・エレメント間を素早く移動したい場合に特に便利です。

## Chip Planner での ECO の 実行 (フロアプラン・ ビュー)

ECO 編集モードを選択すると、Chip Planner でリソース Atom を操作できます。以下の ECO 変更を Chip Planner フロアプラン・ビューで行うことができます。

- [「Atom の作成」](#)
- [13-20 ページの「Atom の削除」](#)
- [13-20 ページの「Atom の移動」](#)



異なる LE/ALM 間の接続管理などの Atom プロパティをコンフィギュレーションするには、Resource Property Editor を使用します。

Atom のリソース・プロパティの編集について詳しくは、[13-21 ページの「Resource Property Editor」](#)を参照してください。

Chip Planner で ECO 編集モードを選択するには、Chip Planner が開いた状態で以下のステップを実行します。

1. View メニューの **Layers Settings** をクリックするか、Task メニューの隣の Layers アイコンをクリックします。Layers Settings ダイアログ・ボックスが表示されます。
2. **Editing Mode** で ECO を選択します。

### Atom の作成

ECO 編集モードのときに、Chip Planner で、目的のリソース Atom にマウス・ポインタを移動して右クリックし、**Create Atom** をクリックして、簡単に Atom を作成できます。Atom が作成された後、リソース Atom をダブル・クリックして Resource Property Editor を開き、プロパティを編集することができます。

作成できる Atom のタイプは以下のとおりです。

- ALM (該当するデバイス・ファミリの)
- LE (該当するデバイス・ファミリの)
- I/O エLEMENT

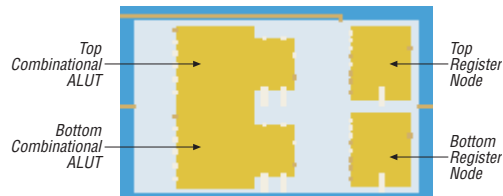
 リソース Atom の作成は **Assignment** 編集モードではサポートされていません。

デバイスでサポートされるアーキテクチャの詳細およびリソース Atom については、デバイス・データシートを参照してください。

### ALM Atom の作成

各 ALM には、2つの組み合わせ LUT 出力と 2つのレジスタ付き入力があります。Chip Planner で、出力パスのタイプに従って、各 ALM を 4つのリソース Atom に分割することができます。図 13-9 に、Chip Planner に表示される ALM を示します。

図 13-9. Chip Planner の ALM



組み合わせ ALM LUT Atom を作成するには、以下のステップを実行します。

1. 未使用 (網掛けなし) ALM の左側を右クリックし、**Create Atom** をクリックします。**Resource Selection** ダイアログ・ボックスが表示されます。
2. **Resource Selection** ダイアログ・ボックスで、作成する Atom を選択します。小さいインデックス番号は上の組み合わせノードを指し、大きいインデックス番号は下の組み合わせノードを指します。
3. **OK** をクリックします。**Create <Altera device> LUT Atom** ダイアログ・ボックスが表示されます。
4. **Atom Name** ボックスで、リソース Atom の名前を入力します。

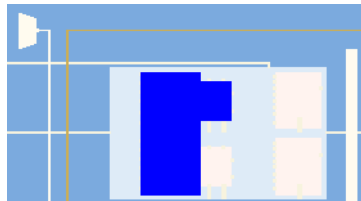
5. **LUT Mode** で、**Normal**、**Extended**、または **Arithmetic** から選択します。
6. 該当する場合は、**Partition** リストで、新しく作成された Atom が存在するはずのパーティションを選択します。新しく作成された Atom のデフォルト・パーティションは、トップレベル・パーティションです。
7. **OK** をクリックします。



LUT のモードについて詳しくは、該当するデバイスのデータシートを参照してください。

組み合わせ出力を正常に作成すると、Chip Planner で組み合わせエレメントに色が付きます。図 13-10 に、組み合わせ ALUT Atom を示します。

図 13-10. 組み合わせ ALUT Atom



レジスタ付き ALM Atom を作成するには、以下のステップを実行します。

1. 任意の ALM レジスタ・リソースを右クリックし、**Create Atom** をクリックします。**Create Register Atom** ダイアログ・ボックスが表示されます。
2. **Atom Name** ボックスで Atom 名を入力します。
3. **OK** をクリックします。

### ロジック・エレメント Atom の作成

Chip Planner は、Stratix、Cyclone、および MAX デバイス・ファミリのリソース Atom をロジック・エレメントとして表示します。他のすべての LE ベースのデバイス・ファミリでは、Chip Planner は、ロジック・エレメント LUT の組み合わせ出力およびロジック・エレメントのレジスタ付き出力としてリソース Atom を表示します。図 13-11 に、Stratix、

Cyclone、および MAX デバイスの Chip Planner における Atom リソース例を示します。図 13-12 に、Chip Planner における Cyclone II リソース Atom を示します。

---

**図 13-11. Chip Planner での Stratix、Cyclone、および MAX デバイスのロジック・エレメント**

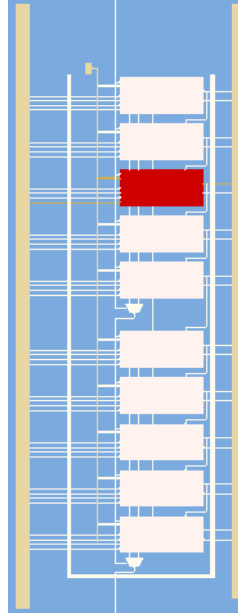
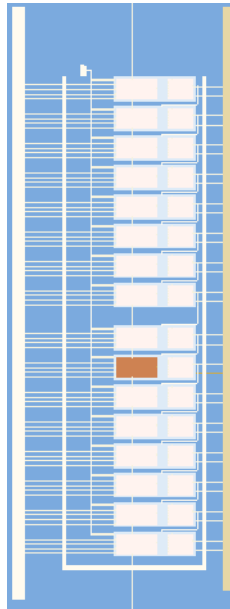


図 13-12. Chip Planner での Cyclone II デバイスのロジック・エレメント



Stratix、Cyclone、および MAX デバイス・ファミリのロジック・エレメント・リソースを作成するには、以下のステップを実行します。

1. 使用可能な (網掛けなし) LE リソースを右クリックし、**Create Atom** をクリックします。 **Create Logic Cell Atom** ダイアログ・ボックスが表示されます。
2. 該当する場合は、**Partition** リストで、新しく作成された Atom が存在するはずのパーティションを選択します。新しく作成された Atom のデフォルト・パーティションは、トップレベル・パーティションです。
3. **Atom Name** ボックスで Atom 名を入力します。
4. **OK** をクリックします。

他のすべての LE ベース・デバイス・ファミリの組み合わせリソース Atom を作成するには、以下のステップを実行します。

1. 使用可能な (網掛けなし) LE リソースの左側を右クリックし、**Create Atom** をクリックします。 **Create <device family> LUT Atom** ダイアログ・ボックスが表示されます。

2. 該当する場合は、**Partition** リストで、新しく作成された Atom が存在するはずのパーティションを選択します。新しく作成された Atom のデフォルト・パーティションは、トップレベル・パーティションです。
3. **Atom Name** ボックスで Atom 名を入力します。
4. **OK** をクリックします。

Cyclone II デバイスのレジスタ・リソース Atom を作成するには、以下のステップを実行します。

1. 使用可能な(網掛けなし)LE リソースの右側を右クリックし、**Create Atom** をクリックします。**Create <device family> Register Atom** ダイアログ・ボックスが表示されます。
2. 該当する場合は、**Partition** リストで、新しく作成された Atom が存在するはずのパーティションを選択します。新しく作成された Atom のデフォルト・パーティションは、トップレベル・パーティションです。
3. **Atom Name** ボックスで Atom 名を入力します。
4. **OK** をクリックします。

## Atom の削除

リソース Atom を削除するには、Chip Planner で削除したいリソース Atom を右クリックし、**Delete Atom** をクリックします。

リソースのすべてのファンアウト接続を削除した後でのみ、リソースを削除することができます。メガファンクションや IP コアのリソースなどの保護されたリソースは削除できません。

ファンアウト接続の削除について詳しくは、[13-21 ページ](#)の「**Resource Property Editor**」を参照してください。

## Atom の移動

希望のリソースをクリックし、選択された Atom を未使用のリソース Atom にドラッグして、リソース ALM、LE、および FPGA I/O の各 Atom を移動することができます。ECO としてノードを移動するのは、ECO 編集モードでのみ行えます。Assignment モード時に変更を行うと、デザイン上で位置制約が作成され、変更を取り込むにはリコンパイルが必要です。

メガファンクション IP コアのリソースなど、保護されたリソースからのリソース Atom は移動できません。

## ネットリスト変更のチェックおよび保存

すべての ECO を行った後、Chip Planner ツールバーの **Check and Save Netlist Changes** アイコンをクリックし、Fitter を実行して変更を取り込むことができます。Fitter は ECO 変更をコンパイルし、デザイン上でデザイン・ルール・チェックを実行し、プログラミング・ファイルを生成します。

## Resource Property Editor

Resource Property Editor で、以下のリソースを表示および編集できます。

- 13-21 ページの「ロジック・エレメント」
- 13-25 ページの「アダプティブ・ロジック・モジュール」
- 13-27 ページの「FPGA の I/O エレメント」
- 13-44 ページの「PLL プロパティ」
- 13-33 ページの「FPGA RAM ブロック」
- 13-34 ページの「FPGA DSP ブロック」

## ロジック・エレメント

アルテラの LE には、4 入力 LUT、つまり 4 変数からなる任意の機能を実装できるファンクション・ジェネレータが搭載されています。また、各 LE には、LUT の出力または別の LE で生成される独立したファンクションによって供給されるレジスタが含まれています。

Resource Property Editor を使用して、FPGA 内の任意の LE を表示および編集することができます。Project メニューの **Locate** をポイントし、以下のビューの 1 つで **Locate in Resource Property Editor** をクリックして、LE に対して Resource Property Editor を開きます。

- RTL Viewer
- Technology Map Viewer
- Node Finder
- Chip Planner



特定のデバイス・ファミリの LE アーキテクチャについては、デバイス・ファミリのハンドブックまたはデータシートを参照してください。

Resource Property Editor を使用して、以下の LE プロパティを変更できます。

- LUT へのデータ入力
- LUT マスクまたは LUT 等式

ロジック・エレメントの回路図ビュー

図 13-13 に、Resource Property Editor での LE の表示を示します。

図 13-13. Stratix LE アーキテクチャ 注 (1)、(2)

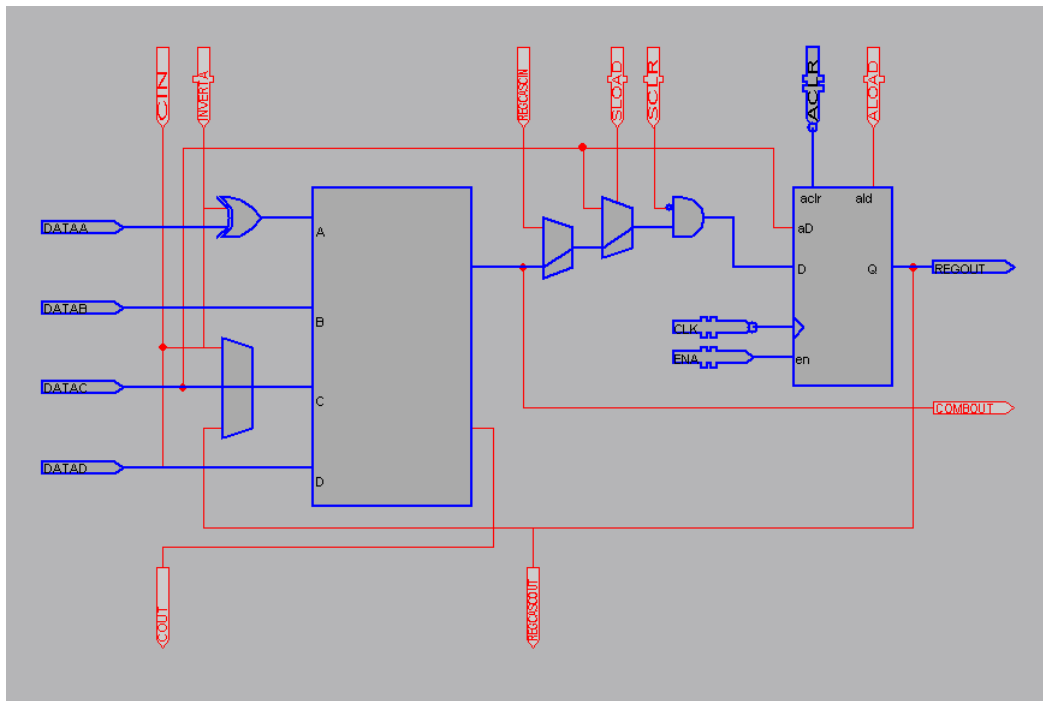


図 13-13 の注：

- (1) デフォルトでは、Quartus II ソフトウェアは、使用済みリソースを青、未使用のソースをグレーで表示します。図 13-13 では、使用済みリソースを青、未使用リソースを赤で表示しています。
- (2) Stratix デバイスの LE アーキテクチャについて詳しくは、「Stratix デバイス・ハンドブック」を参照してください。

## LE プロパティ

図 13-14 に、Resource Property Editor で選択された LE に対して表示可能なプロパティの例を示します。LE プロパティを表示するには、View メニューの **View Properties** をクリックして表示します。

図 13-14. LE プロパティ

Properties/Modes	Values	Sum Equation	D & (B # C) # ID & A & B	Node:	fitrefitaps.instn_1[7]	
LUT Mask	FC22	Carry Equation	N/A		COMBOUT(0)	REGOUT(0)
Sum LUT Mask	FC22			ACLRO(0)	N/A	880 ps
Carry LUT Mask	N/A			CLK(0)	N/A	560 ps
Operation Mode	normal			DATAA(0)	459 ps	N/A
Synchronous Mode	on			DATAB(0)	332 ps	N/A
Register Cascade Mode	off			DATAD(0)	87 ps	N/A
Latch Type	none					

## 動作モード

LE 内の LUT はノーマル・モードまたは演算モードで動作できます。



LE の動作モードについて詳しくは、該当するデバイス・ハンドブックの Volume 1 を参照してください。

LE がノーマル・モードで構成されていると、LE 内の LUT は 4 入力のファンクションを実装できます。

LE が演算モードで構成されているとき、LE での LUT は 2 つの 3 入力 LUT に分割されます。1 つの LUT は、LUT の出力をドライブする信号を生成し、もう 1 つの LUT はキャリー・アウト信号を生成します。キャリー・アウト信号は、別の LE のキャリー・イン信号しかドライブできません。

## 加算およびキャリーの式

加算およびキャリーの式を変更することにより、LUT で実装されるロジック・ファンクションを変更できます。LE がノーマル・モードで構成されている場合は、加算式の変更しかできません。LE が演算モードで構成されているときは、加算およびキャリー式の両方を変更できます。

LUT マスクは LUT 式出力の 16 進表現です。LUT 式を変更すると、Quartus II ソフトウェアは LUT マスクを自動的に変更します。逆に、LUT マスクを変更すると、Quartus II ソフトウェアは LUT 式を自動的に計算します。

## sload および sclear 信号

各 LE レジスタは、同期ロード (sload) 信号と同期クリア (scclr) 信号を備えています。LE に供給される sload 信号または scclr 信号のいずれかを反転することができます。デザインが LE 内で sload 信号を使用する場合、その信号およびその反転状態は、同じ LAB 内の他のすべての LE で同一でなければなりません。例えば、LAB の 2 個の LE で sload 信号が接続されている場合、両方の LE の sload 信号は同じ値に設定されている必要があります。これは scclr 信号についても当てはまります。

## レジスタ・カスケード・モード

レジスタ・カスケード・モードがイネーブルされていると、カスケード・イン・ポートがレジスタに入力を供給します。レジスタ・カスケード・モードは、デザインがシフト・レジスタを実装するときに最もよく使用されます。ポートでカスケードを接続 (または切断) することにより、レジスタ・カスケード・モードを変更できます。ただし、このポートを作成する場合、必ずソース・ポート LE がデスティネーション LE の真上にあるようにしなければなりません。

## セル遅延テーブル

セル遅延テーブルは、選択された LE に対するすべての入力からすべての出力までの伝播遅延を表します。

## LE の接続

View メニューの **View Port Connections** をクリックして、LE に供給および LE から供給される接続を表示します。図 13-15 に、Connectivity ウィンドウでの LE の接続を示します。

図 13-15. コネクションの表示

Input Port name	Signal name	Latch info	Output Port name	Signal name	Latch info
ACLRL	filterfreset	N/A	COMBOUT	filterfaps_inst1Selector0~14	N/A
ALOAD	<Disconnected>	N/A	COULT	<Disconnected>	N/A
CIN	<Disconnected>	N/A	REGCASCOUT	<Disconnected>	N/A
CLK	filterfclk	N/A	REGOUT	filterfaps_inst1bn_1[7]	N/A
DATAA	filterfaps_inst1bn[7]	N/A			
DATAB	filterfstate_m_inst1sel[1]	N/A			
DATAD	filterfstate_m_inst1sel[0]	N/A			
ENA	filterfcout	N/A			

## LE の削除

LE を削除するには、以下のステップを実行します。

1. Chip Planner において目的の LE を右クリックし、**Locate** をポイントし、**Locate in Resource Property Editor** をクリックします。
2. 削除前に LE からすべてのファンアウト接続を削除しなければなりません。ファンアウト接続を削除するには、接続された各出力信号を右クリックし、**Remove** を選択して、**Fanouts** をクリックします。表示されるダイアログ・ボックスですべてのファンアウト信号を選択して、**OK** をクリックします。
3. すべてのファンアウト接続を削除した後に **Atom** を削除するには、**Chip Planner** で **Atom** を右クリックし、**Delete Atom** をクリックします。

### アダプティブ・ロジック・モジュール

各 ALM には、2つのアダプティブ・ルック・アップ・テーブル (ALUT) 間で分割できる LUT ベースのリソースが含まれています。2 個の ALUT への最大 8 本の入力により、それぞれの ALM で 2 つのファンクションの様々な組み合わせを実装できます。この適応性により、ALM は 4 入力 LUT アーキテクチャとの完全な下位互換性を提供します。1 個の ALM で、最大 6 本の入力を持つ任意のファンクションおよび特定の 7 入力ファンクションを実装することができます。アダプティブ LUT ベースのリソースに加えて、各 ALM には 2 個のプログラマブル・レジスタ、2 個の専用の全加算器、1 本のキャリー・チェーン、1 本の共有演算チェーン、および 1 本のレジスタ・チェーンも含まれています。これらの専用リソースを使用して、ALM は様々な演算ファンクションやシフト・レジスタを効率的に実装することができます。

1 個の ALM に以下のタイプのファンクションを実装できます。

- 2 個の独立した 4 入力ファンクション
- それぞれ独立した 1 個の 5 入力ファンクションと 3 入力ファンクション
- 1 つの入力を共有する場合、1 個の 5 入力ファンクションと 1 個の 4 入力ファンクション
- 2 つの入力を共有する場合、2 個の 5 入力ファンクション
- 独立した 1 個の 6 入力ファンクション
- 4 つの入力とファンクションを共有する場合、2 個の 6 入力ファンクション
- 特定の 7 入力ファンクション

Resource Property Editor を使用して、以下の ALM プロパティを変更できます。

- LUT へのデータ入力
- LUT マスクまたは LUT 等式

### ALM 回路図

RTL Viewer、Node Finder、または Chip Planner 上で ALM を右クリックし、**Locate in Resource Property Editor** をクリックして任意の ALM Atom を Resource Property Editor で表示および編集できます (図 13-16)。



ALM についての詳しい説明は、ALM アーキテクチャに基づくデバイスのデバイス・ハンドブックを参照してください。

図 13-16. ALM 回路図 注(1)

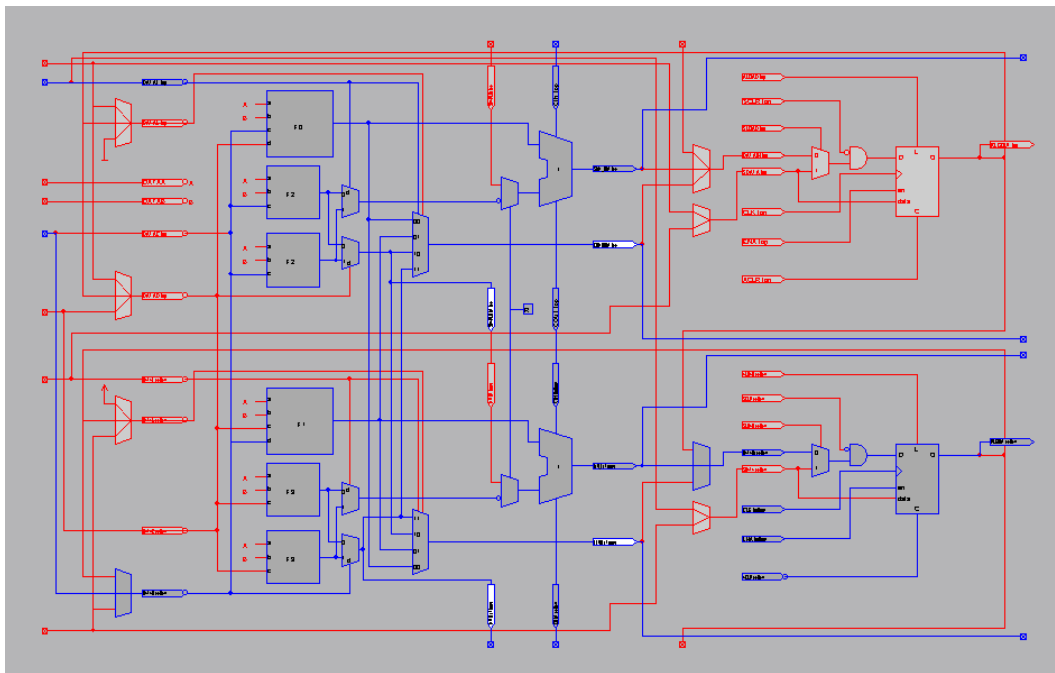


図 13-16 の注 :

- (1) デフォルトでは、Quartus II ソフトウェアは、使用済みリソースを青、未使用のソースをグレーで表示します。図 13-16 では、使用済みリソースを青、未使用リソースを赤で表示しています。

## ALM プロパティ

ALM に対して表示できるプロパティには、ALM 内の 2 つの組み合わせノードおよび 2 つのレジスタ・ノードのそれぞれの名前および位置を示す式テーブル、各組み合わせノードに対する個別の LUT 式、および各組み合わせノードに対する combout、sumout、carryout、および shareout 式などがあります。

## ALM の接続

View メニューの **View Port Connections** をクリックして、ALM に供給および ALM に供給される接続を表示します。

## FPGA の I/O エLEMENT

アルテラの FPGA は、最大 6 個のレジスタを含む高性能 I/O エLEMENT を備えており、最大速度でのデザインの実行が可能な多数の I/O 規格をサポートします。Resource Property Editor を使用して、I/O エLEMENT のプロパティの表示、接続の変更および編集を行うことができます。Chip Planner (Floorplan ビュー) を使用して配置の変更、削除、および新しい I/O エLEMENT の作成を行うことができます。



デバイスの I/O エLEMENT の詳しい説明は、該当するデバイス・ハンドブックを参照してください。

以下の I/O プロパティを変更することができます。

- 遅延チェーン
- バス・ホールド
- ウィーク・プルアップ
- 低速スルー・レート
- I/O 規格
- 電流容量
- OE ディセーブルの延長
- PCI I/O
- レジスタ・リセット・モード
- レジスタ同期リセット・モード
- レジスタのパワーアップ
- レジスタ・モード

### Arria GX、Stratix II、Stratix、および Stratix GX I/O エLEMENT

Stratix シリーズ・デバイス・ファミリおよび Arria GX デバイスの I/O エLEMENTは、1 個の双方向 I/O バッファ、6 個のレジスタ、および完全に双方向のシングル・データ・レートまたは DDR 転送に対応するラッチで構成されています。図 13-17 に、Stratix および Stratix GX I/O エLEMENT 構造を示します。I/O エLEMENT 構造は、2 個の入力レジスタ ( および 1 個のラッチ )、2 個の出力レジスタ、および 2 個の出力カインープル・レジスタを備えています。

図 13-17. Stratix および Stratix GX デバイスの I/O エLEMENT および構造 注 (1)、(2)

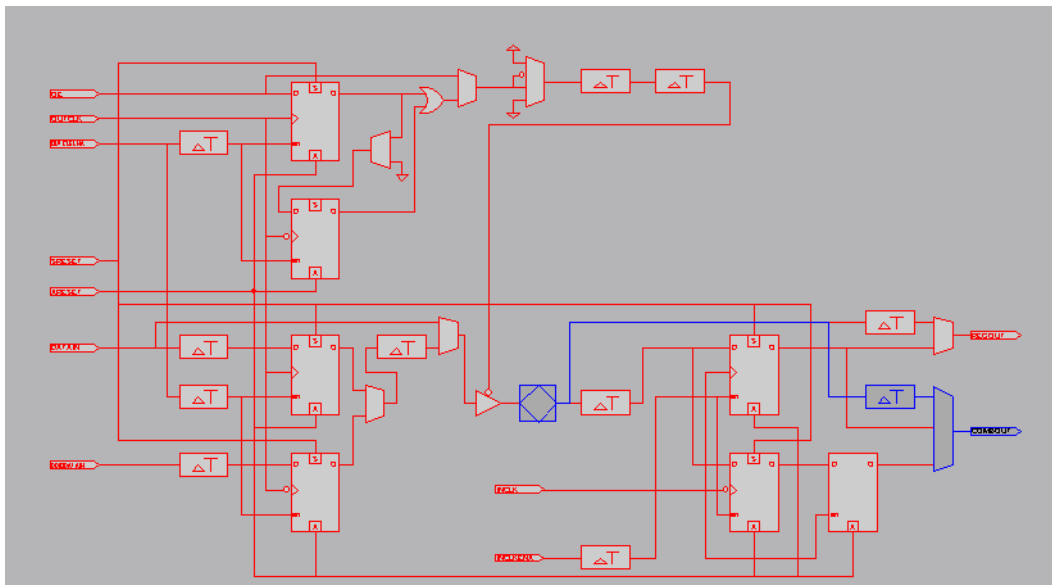


図 13-17 の注:

- (1) デフォルトでは、Quartus II ソフトウェアは、使用済みリソースを青、未使用のリソースをグレーで表示します。図 13-17 では、使用済みリソースを青、未使用リソースを赤で表示しています。
- (2) Stratix および Stratix GX デバイスの I/O エLEMENT について詳しくは、「Stratix デバイス・ハンドブック」および「Stratix GX デバイス・ハンドブック」を参照してください。





Stratix III デバイスのプログラマブル I/O エLEMENT について詳しくは、「AN 474: Quartus II ソフトウェアでの Stratix III プログラマブル I/O 遅延設定の実装」を参照してください。

図 13-19. Stratix III デバイスの I/O エLEMENT および構造 注 (1)、(2)

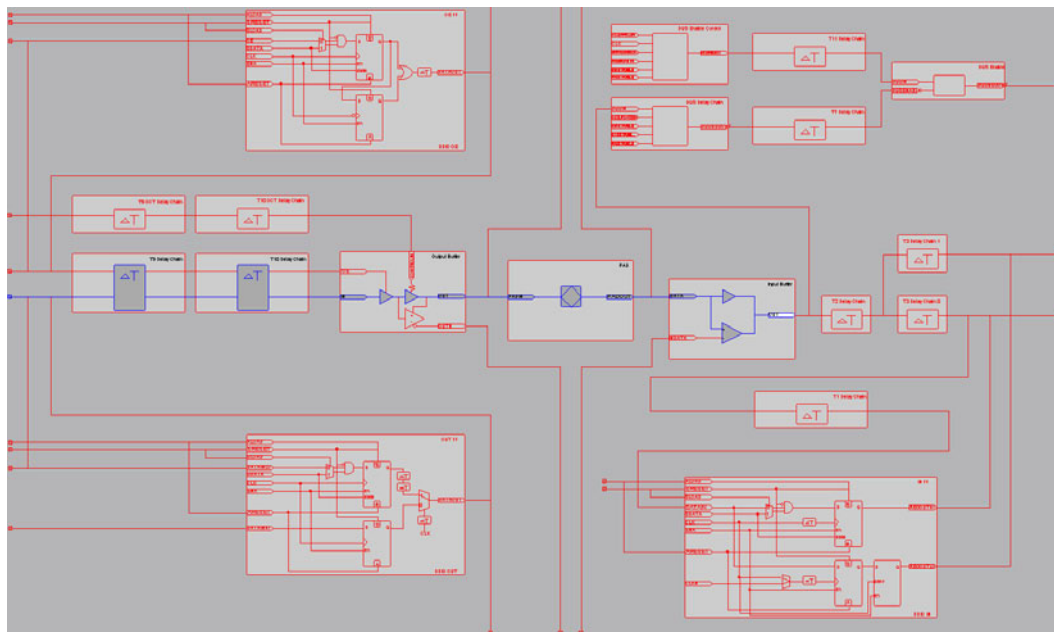


図 13-19 の注:

- (1) デフォルトでは、Quartus II ソフトウェアは、使用済みリソースを青、未使用のリソースをグレーで表示します。図 13-19 では、使用済みリソースを青、未使用リソースを赤で表示しています。
- (2) Stratix III デバイスの I/O エLEMENT について詳しくは、「Stratix III デバイス・ハンドブック」を参照してください。

### Cyclone II および Cyclone の I/O エLEMENT

Cyclone II、および Cyclone デバイスの I/O エLEMENT は、1 個の双方向の I/O バッファ、および完全に双方向のシングル・データ・レート転送に対応する 3 個のレジスタで構成されています。図 13-20 に Cyclone II および Cyclone の I/O エLEMENT 構造を示します。I/O エLEMENT は、1 個の入力レジスタ、1 個の出力レジスタ、および 1 個の出力カインエーブ・レジスタで構成されています。



図 13-21. Cyclone III デバイスの I/O エLEMENT および構造 注 (1)、(2)

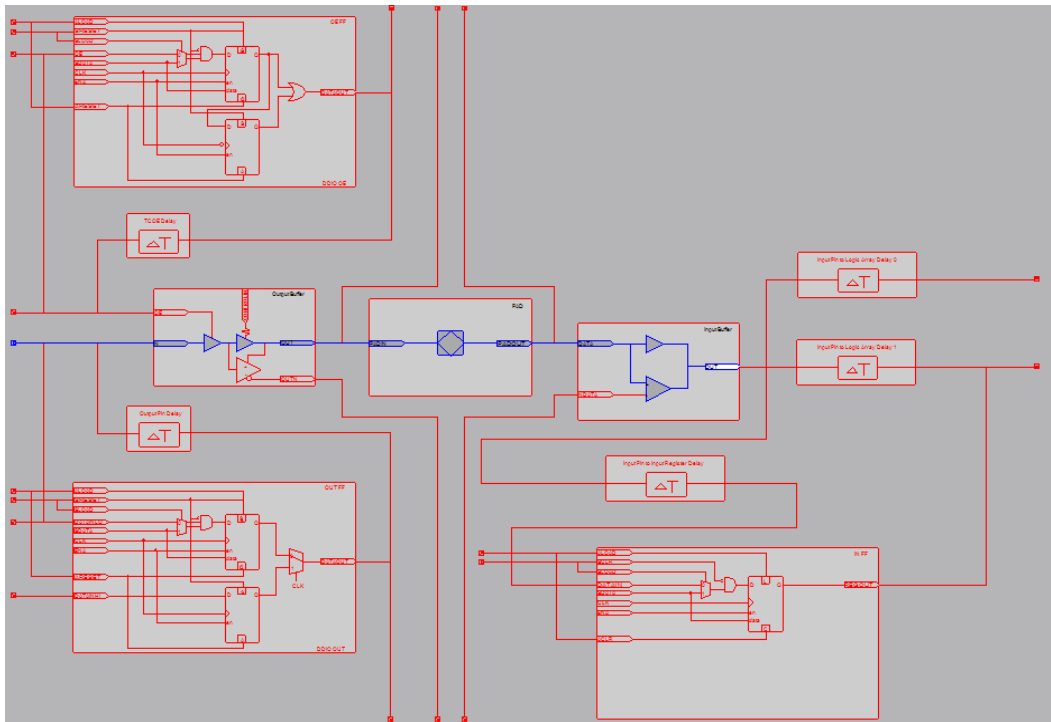


図 13-21 の注：

- (1) デフォルトでは、Quartus II ソフトウェアは、使用済みリソースを青、未使用のリソースをグレーで表示します。  
図 13-21 では、使用済みリソースを青、未使用リソースを赤で表示しています。
- (2) Cyclone III デバイスの I/O エLEMENT について詳しくは、「Cyclone III デバイス・ハンドブック」を参照してください。

## MAX II の I/O エLEMENT

MAX II デバイスの I/O エLEMENT は双方向 I/O バッファを備えています。図 13-22 に MAX II の I/O エLEMENT 構造を示します。隣接する LAB からのレジスタは、I/O エLEMENT の双方向 I/O バッファにドライブでき、またこのバッファからドライブすることもできます。

図 13-22. MAX II デバイスの I/O エlement および構造 注 (1)、(2)

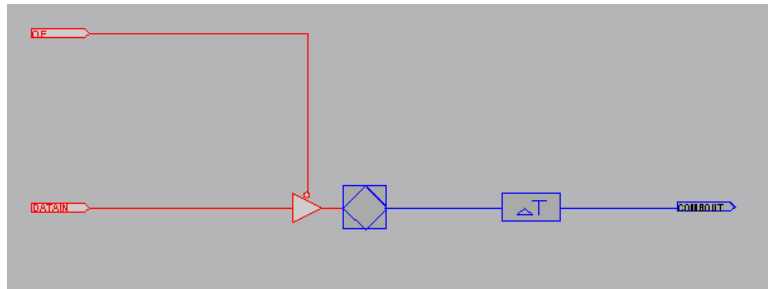


図 13-22 の注：

- (1) デフォルトでは、Quartus II ソフトウェアは、使用済みリソースを青、未使用のリソースをグレーで表示します。図 13-22 では、使用済みリソースを青、未使用リソースを赤で表示しています。
- (2) MAX II デバイスの I/O エlement について詳しくは、「MAX II デバイス・ハンドブック」を参照してください。

## FPGA RAM ブロック

Resource Property Editor で、デバイスの異なる RAM ブロックのアーキテクチャを表示し、RAM ブロックからの入力レジスタおよび出力レジスタを変更し、入力および出力ポートの接続を変更することができます。図 13-23 に、Stratix III デバイスの M9K RAM 表示を示します。



図 13-24. Stratix III デバイスでの DSP ブロックの表示 注 (1)

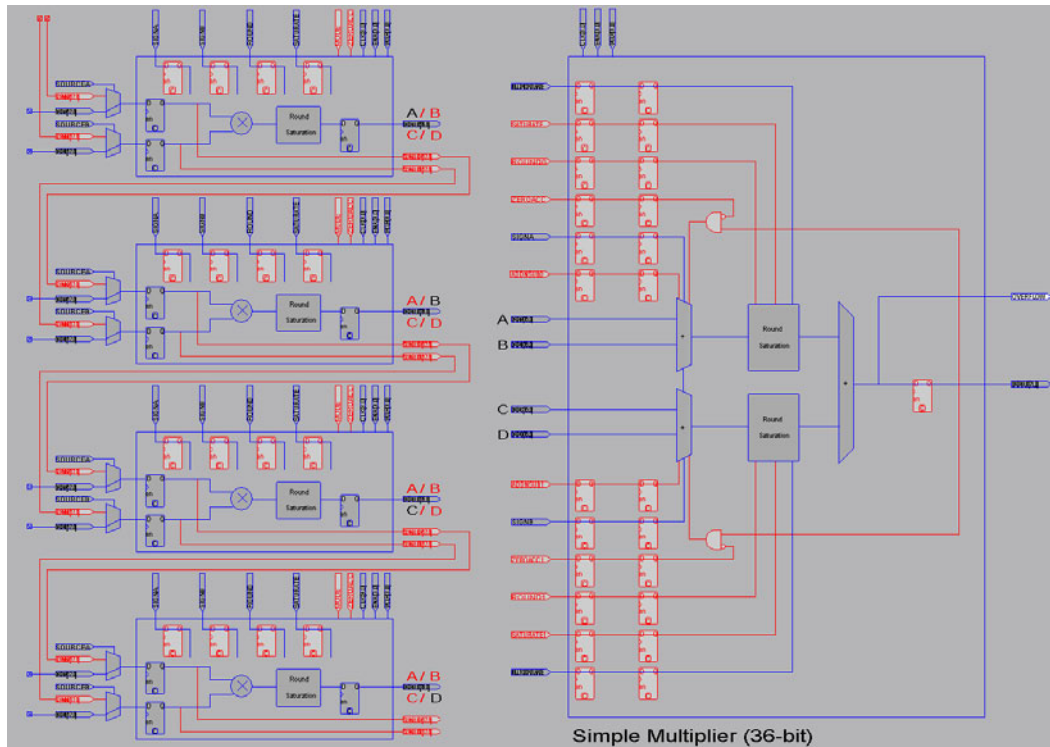


図 13-24 の注:

- (1) デフォルトでは、Quartus II ソフトウェアは、使用済みリソースを青、未使用のリソースをグレーで表示します。  
 図 13-24 では、使用済みリソースを青、未使用リソースを赤で表示しています。

## Change Manager

Change Manager は、Resource Property Editor で実行するすべての変更の記録を維持します。Change Manager の各ロウは、実行された 1 つの ECO を表します。変更には順番に番号が付けられ、大きい番号がほど新しい変更です。

より複雑な変更には、Change Manager で + アイコンが付けられます。実施されたすべての変更を確認するには、+ アイコンをクリックして、Change Manager で複雑なエントリを拡大表示することができます。複雑な変更の一例は Atom の作成または削除です。

表 13-1 に、Change Manager により表示される情報をまとめます。

カラム名	説明
Index	Chip Planner または Resource Property Editor で行われる変更に対応する変更記録を連続番号により識別します。  変更記録が複雑な場合、インデックス・カラムは主な変更だけでなくコンポーネントの変更もすべて識別します。
Node Name	変更が行われたリソースを個々に識別します。
Change Type	リソースに行われた変更のタイプを識別します。
Old Value	変更が行われる直前のリソースの値をリストします。
Target Value	Resource Property Editor、Chip Planner、または SignalProbe を使用して、確立した希望のターゲット値（新しい値）をリストします。
Current Value	メモリで現在アクティブなネットリストのリソースの値をリストします（ディスクに保存されたネットリストの値ではありません。変更を行ったが、まだ <b>Check &amp; Save All Netlist Changes</b> コマンドを使用していない場合、この値は異なることがあります）。
Disk Value	ディスク上のリソースの現在の値をリストします。この値は、現在のフィッティング後のネットリストに取り込まれた変更を反映しています。
Comment	これで Change Manager で変更記録にコメントを追加できます。 変更記録にコメントを追加するには、注釈を付けたいレコードの <b>Comment</b> フィールドをダブル・クリックして、希望のコメントを入力します。

デザイン変更がすべて終了した後に、Change Manager で右クリックし、**Check & Save All Netlist Changes** をクリックしてネットリストの整合性をチェックします。加えられた変更がネットリストのチェックに合格すると、ディスクに書き込まれます。変更がネットリストのチェックに合格しない場合は、正常に実行したネットリストのチェック以降の変更はすべて元に戻されます。図 13-25 に Change Manager を示します。

**Current Value** および **Disk Value** カラムの色付きインジケータは、これらのカラムにおけるデータの現在のステータスを示します。**Current Value** カラムが緑色の場合、変更が記録済みであることを示します。**Disk Value** カラムが青色の場合、変更が **Check & Save Netlist Changes** 操作に合格したことを示します。

### 図 13-25. Change Manager の結果

Index	Node Name	Change Type	Old Value	Target Value	Current Value	Disk Value
1	test1	SignalProbe	Disconnected	fitrefinst4	Disconnected	Disconnected
2	mult_inst6lpm_mult_lpm_mult_componentimu...	Modify Source	fitrefitaps.in...	Disconnected	fitrefitaps.instxn[4]	fitrefitaps.instxn[4]
3	mult_inst6lpm_mult_lpm_mult_componentimu...	Modify Source	fitrefitaps.in...	Disconnected	fitrefitaps.instxn[4]	fitrefitaps.instxn[4]
4	taps_instxn_T[4]~feeder.DATAF.0	Modify Source	fitrefitaps.in...	Disconnected	fitrefitaps.instxn[4]	fitrefitaps.instxn[4]
5	mult_inst6lpm_mult_lpm_mult_componentimu...	Modify Source	fitrefitaps.in...	Disconnected	Disconnected	Disconnected
6	mult_inst6lpm_mult_lpm_mult_componentimu...	Modify Source	fitrefitaps.in...	Disconnected	Disconnected	Disconnected
7	taps_instxn_T[0]~feeder.DATAF.0	Modify Source	fitrefitaps.in...	Disconnected	Disconnected	Disconnected

Netlist Check Required -- 3 Pending Changes

## Change Manager における複雑な変更

Resource Property Editor または Chip Planner で行う特定タイプの変更 (Atom の作成または削除、および接続の変更など) は、自己完結型に見えますが、これらの変更は実際には複数の動作で構成されています。複雑な変更は **Index** カラムに + アイコン付きで表示されます。

**Change Manager** での変更記録は、発生した実際の変更動作を 1 行で表現したものです。+ アイコンをクリックして変更記録を拡大し、変更を構成するコンポーネント動作を表示します。

**Change Manager** で変更エントリを拡大すると、Atom の作成が以下の 3 つの動作で構成されることが分ります。

- 新しいロジック・セルの作成
- 新しく作成されたロジック・セルの出力ポートの作成
- 新しく作成されたロジック・セルへの位置インデックスのアサインメント

複雑な変更記録の個別コンポーネントを選択することはできません。複雑な変更記録の一部を選択すると、複雑な変更記録全体が選択されます。



**Change Manager** を使用した変更の管理例については、「Quartus II Help」の「Change Manager による変更の管理例」を参照してください。

## SignalProbe 信号の管理

**SignalProbe Pins** ダイアログ・ボックスから作成する SignalProbe ピンは、Change Manager に記録されます。SignalProbe アサインメント作成後、Change Manager を使用し、Change Manager の右クリック・メニューから **Revert to Last Saved Netlist** を選択して、SignalProbe アサインメントを素早くディセーブルすることができます。



SignalProbe ピンについて詳しくは、「Quartus II ハンドブック Volume 3」の「**Quick Design Debugging Using SignalProbe**」の章を参照してください。

## 変更のエクスポート

ツール・コマンド言語 (Tcl) スクリプト、カンマ区切り値 (.csv) ファイル、または Text (.txt) ファイルにすべての変更をエクスポートできます。Tcl ファイルでは、コンパイルにより削除された変更を再適用するスクリプトを記述することができます。作成するその他の Quartus II ソフトウェア・プロジェクトに適用されるスクリプトを記述することもできます。カンマ区切り値または Text ファイルは、表形式で変更リストを提供します。変更をエクスポートするには、以下のステップを実行します。

1. 右クリック・メニューで **Export Changes** をクリックします。
2. Tcl ファイル名を指定します。
3. **OK** をクリックします。

結果の Tcl スクリプトで、別の Quartus II デザインに同様の変更を実装することもできます。

## ECO フローにおけるインクリメンタル・コンパイルの使用

Quartus II ソフトウェア v6.1 からは、デフォルトでインクリメンタル・コンパイル機能がオンになります。インクリメンタル・コンパイル機能がオンのとき、トップレベル・デザインは自動的にデザイン・パーティションに設定されます。インクリメンタル・コンパイル中のデザイン・パーティションには、異なるネットリスト・タイプが存在する場合があります (ネットリスト・タイプは、ソース HDL、合成後、またはフィッティング後に設定できます)。ネットリスト・タイプは、そのパーティションがリコンパイル中に再合成または再フィッティングされる必要があることを示します。デザイン・サイクルの後半で一部のパーティションに対して小さな変更を行う必要がある場合、インクリメンタル・コンパイルによって時間が節約され、デザインで変更されないパーティションの配置が維持されます。



パーティション、それらのネットリスト・タイプおよび Quartus II インクリメンタル・コンパイルについて詳しくは、「Quartus II ハンドブック Volume 1」の「階層ベースおよびチーム・ベースのデザインのための Quartus II インクリメンタル・コンパイル」の章を参照してください。

インクリメンタル・コンパイル中の ECO の動作は、デザイン・パーティションのネットリスト・タイプによって異なります。Quartus II ソフトウェアは、ECO を含むパーティションが以下の 2 つの条件を満たす場合は、ECO を維持します。

- 影響を受けるパーティションのネットリスト・タイプが Post-Fit に設定され、Fitter preservation level が Placement and Routing に設定されている場合。
- リコンパイル中にパーティションの再合成を引き起こす可能性があるソース・コードの変更が影響を受けたパーティションにない場合。

デザイン内の複数のパーティションに影響する ECO がある場合、Quartus II ソフトウェアは、影響されるパーティションのどれかがフィッティング後に設定されていれば、リコンパイル中に ECO を維持します。ECO が複数のパーティションに影響するときは常に、影響を受けるパーティションのすべてがリンクされることに注意してください。最も近い共通の親まで上位レベルのすべての“親”パーティションもリンクされます。この場合、パーティション間の接続は、直ちに影響を受ける 2 つのパーティションの外部で実際に定義されるため、すべてのパーティションをまとめてコンパイルする必要があります。リンクされているパーティションは、最高レベルの維持が設定されているパーティションのネットリスト・タイプを継承します。例えば、ECO が Post-Fit タイプに設定された低いレベルのパーティションで行われ、トップレベルのパーティションは Post-Synthesis タイプに設定されている場合、これら 2 つのパーティションはリンクされて Post-Fit ネットリスト・タイプになります。

パーティションがソース・コードまたは合成後のネットリストを使用するよう設定されている場合、ソフトウェアは警告を発行し、フィッティング後の ECO 変更は新しいコンパイルにインクルードされません。

例えば、トップレベル・パーティションのネットリスト・タイプが合成後に設定されている場合、およびその他の下位レベルのパーティションが存在しないか、リコンパイル中に下位レベルのパーティションのネットリスト・タイプも合成後に設定される場合、ECO は維持されず、ECO 変更は破棄されるが、ECO 情報はすべて Change Manager に保持されることを示す警告メッセージがメッセージ・ウィンドウに表示されます。この場合、Change Manager から ECO を適用し、13-40 ページの「Quartus II

インクリメンタル・コンパイルを使用しない ECO フロー」で説明する  
とおり、**Check & Save All Netlist Changes** ステップを実行することが  
できます。

## Quartus II インクリメンタル・コンパイルを使用しない ECO フロー

Quartus II インクリメンタル・コンパイル機能を使用しないで ECO を実  
装した場合、それらの ECO はデザインのリコンパイル時には維持され  
ませんが、ECO はすべて **Change Manager** 内に残ります。ECO を適用  
するには、**Change Manager** を右クリックし、**Apply Selected Change**  
をクリックします。(画面下部に **Change Manager** ウィンドウが表示さ  
れない場合は、**View** メニューから **Utility Windows** をポイントして  
**Change Manager** をクリックします。)

選択された ECO を適用した後、以下のステップを実行します。

- ✓ **Change Manager** のメニューから、**Check & Save All Netlist Changes**  
をクリックします。

または

- ✓ **Processing** メニューから、**Start** をポイントして、**Start Check & Save  
All Netlist Changes** をクリックします。

## スクリプ ティング・ サポート

この章で説明する手順の実行と設定は Tcl スクリプトで行うことができ  
ます。また、一部の手順はコマンド・プロンプトでも実行できます。Chip  
Planner をコントロールする Tcl コマンドは、quartus\_cdb 実行ファイ  
ルの chip\_planner パッケージにあります。Chip Planner の Tcl コマ  
ンドの包括的なリストは、「[Quartus Scripting Reference Manual](#)」にあ  
ります。



Tcl スクリプトについて詳しくは、「[Quartus II ハンドブック Volume 2](#)」  
の「[Tcl スクリプト](#)」の章を参照してください。Quartus II ソフトウェア  
におけるすべての設定および制約については、「[Quartus II  
Settings File Reference Manual](#)」を参照してください。コマンドライン・  
スクリプトについて詳しくは、「[Quartus II ハンドブック Volume 2](#)」の  
「[Command-Line Scripting](#)」の章を参照してください。

## 一般的な ECO アプリケーション

この項では、ECO を使用してコンパイル後にデザインの変更を行う状況の例を示します。システムを素早く構築できるように、Chip Planner 機能を使用して以下の操作を実行することができます。

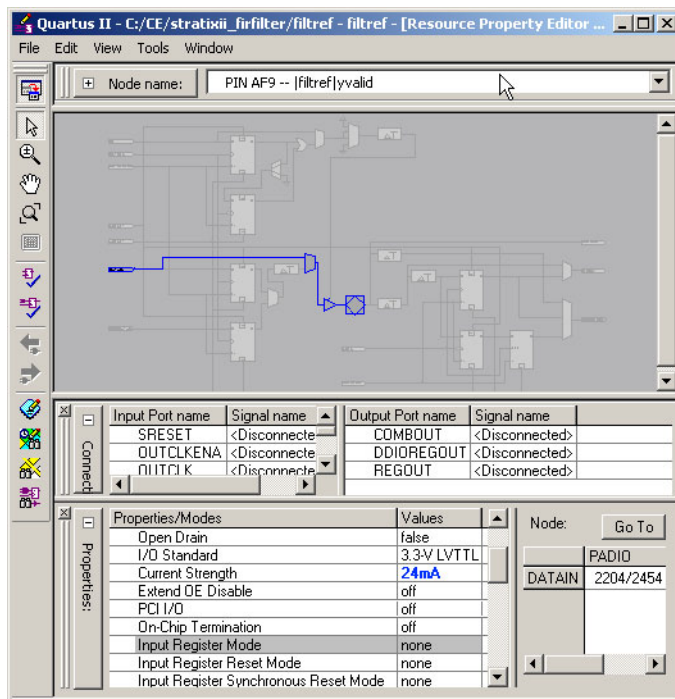
- Chip Planner を使用して I/O のドライブ強度を調整する
- Resource Property Editor を使用して PLL プロパティを変更する (13-43 ページの「Chip Planner を使用した PLL プロパティの変更」を参照)。
- 新しいリソース Atom 間の接続を変更する

### Chip Planner を使用した I/O のドライブ強度の調整

I/O のドライブ強度を調整するには、この項にあるステップに従って、Fitter および Assembler を実行して ECO 変更をデザインのネットリストに取り込みます。

1. Chip Planner で、**Post Compilation Editing (ECO)** タスクを選択します。
2. 図 13-26 に示すとおり、**Resource Property Editor** で I/O を探索します。

図 13-26. Resource Property Editor での I/O



3. 選択された I/O の **Current Strength** ボックスをクリックし、次に **Edit** をクリックします。
4. 希望の電流容量の値を変更します。
5. Change Manager ツールの ECO 変更を右クリックし、**Check & Save All Netlist Changes** をクリックして ECO 変更を適用します。



ECO フローを使用しても、入力 / 出力ポートのピン位置を変更できます。Chip Planner で Post Compilation Editing (ECO) タスクの間に、既存のピン位置から新しい位置に信号をドラッグして移動できます。その後、**Check & Save All Netlist Changes** を設定して、ECO をコンパイルできます。

## Chip Planner を使用した PLL プロパティの変更

PLL を使用して、デザイン要件を満たすようにクロック信号を変更し、生成します。さらに、デザインで異なるデバイスにクロック信号を分配するのに PLL を使用して、デバイス間のクロック・スキューを低減し、I/O タイミングを改善し、内部クロック信号を生成します。

Resource Property Editor で、PLL プロパティを表示および変更してデザイン要件を満たすようにすることができます。Stratix PLL を一例として使用して、この項の残りの部分で調整可能な PLL プロパティと、PLL 出力パラメータを制御する調整可能な PLL プロパティの関数としての式について説明します。図 13-27 に、Resource Property Editor に表示される Stratix PLL を示します。

図 13-27. Stratix デバイスでの PLL の表示

The screenshot displays the Resource Property Editor for a Stratix PLL. The main area shows a block diagram of the PLL with various input and output ports. Below the diagram are several tables for properties and modes.

Input Port name	Signal name	Output Port name	Signal name
SCANDATA	Interactive_Reconfig_pll_reconfig_pll_inst[pll_reconfig_pll_inst].pll_reconfig_pll_clk0_75i_pll_reconfig_pll_inst	ACTIVELOCK	<Disconnected>
SCANCLK	Interactive_Reconfig_pll_reconfig_pll_inst[pll_reconfig_pll_inst].pll_reconfig_pll_clk1_75i_pll_reconfig_pll_inst	CLK 0	Interactive_Reconfig_pll_inst[pll_inst].pll_component_clk0
PRFNPLN	<Disconnected>	CLK 1	Interactive_Reconfig_pll_inst[pll_inst].pll_component_clk1
		CLK 2	<Disconnected>

Properties/Modes	Values	Properties	Values
General PLL Properties		CLK 0	
PLL type	Enhanced	Multi	1
Scan chain	Long	Div	1
Feedback source	N/A	Output Frequency (MHz)	50.00
Compensate clock	clock0	Phase Shift (ps)	0
Switchover on loss of clock	N/A	Delay (ps)	0
Enable Switch Over Counter	N/A	Duty Cycle (%)	50
Switchover counter	N/A	Counter	60
Primary clock	N/A	CLK 1	
Input frequency 0 (MHz)	50.00	Multi	1
Input frequency 1 (MHz)	N/A	Div	1
Normal VCO frequency (MHz)	529.08	Output Frequency (MHz)	50.00
Free min lock (MHz)	24.98	Phase Shift (ns)	0

Node:	Interactive_Reconfig_pll	Go To
CLK	EXTCLK	LOCKED
ARESET	N/A	N/A
CLK	N/A	N/A
SCANACLK	N/A	N/A
SCANCLK	N/A	N/A
SCANDATA	N/A	N/A

## PLL プロパティ

Resource Property Editor で、位相シフト、出力クロック周波数、およびデューティ・サイクルなどの PLL オプションを変更できます。Resource Property Editor を使用して以下の PLL プロパティを変更することもできます。

- Input frequency
- M VCO Tap
- M initial
- M value
- N value
- M counter delay
- N counter delay
- M2 value
- N2 value
- SS counter
- Charge pump current
- Loop filter resistance
- Loop filter capacitance
- Counter delay
- Counter high
- Counter low
- Counter mode
- Counter initial
- VCO tap

コンパイル後の PLL プロパティは、コンパイル・レポートで表示することもできます。そのためには、**Compilation Report** で、**Fitter** を選択し、次に **Resource Section** を選択します。

### デューティ・サイクルの調整

以下のように計算式 1 を使用して、個々の出力クロックのデューティ・サイクルを調整します。

- (1)  $\text{High \%} = \text{Counter High} / (\text{Counter High} + \text{Counter Low})$   
 $\text{Low \%} = \text{Counter Low} / (\text{Counter High} + \text{Counter Low})$

### 位相シフトの調整

計算式 2 を使用して、PLL の出力クロックの位相シフトを調整します。

- (2)  $\text{位相シフト} = (\text{Period } V_{CO} \times 0.125 \times \text{Tap } V_{CO}) + (\text{Initial } V_{CO} \times \text{Period } V_{CO})$

ノーマル・モードでは Period  $V_{CO}$ 、Tap  $V_{CO}$ 、および Initial  $V_{CO}$  は以下の設定により制御されます。

$$\begin{aligned} \text{Tap } V_{CO} &= \text{Counter Delay} - M \text{ Tap } V_{CO} \\ \text{Initial } V_{CO} &= \text{Counter Initial} - M \text{ Initial} \\ \text{Period } V_{CO} &= \text{In Clock Period} \times N / M \end{aligned}$$

外部フィードバック・モードでは、Tap  $V_{CO}$ 、Initial  $V_{CO}$ 、および Period  $V_{CO}$  は以下の設定で制御されます。

$$\begin{aligned} \text{Tap } V_{CO} &= \text{Counter Delay} - M \text{ Tap } V_{CO} \\ \text{Initial } V_{CO} &= \text{Counter Initial} - M \text{ Initial} \\ \text{Period } V_{CO} &= \text{In Clock Period} \times N / (M + \text{Counter High} + \text{Counter Low}) \end{aligned}$$



設定の詳細な説明は、Quartus II Help を参照してください。Stratix デバイスの PLL について詳しくは、「Stratix デバイス・ハンドブック Volume 1」の「Stratix アーキテクチャ」の章を参照してください。Arria GX、Stratix II、Cyclone II、および Cyclone デバイスの PLL について詳しくは、該当するデバイス・ハンドブックを参照してください。

### 出力クロック周波数の調整

以下のように計算式 3 を使用して、ノーマル・モードでの PLL 出力クロックを調整します。

$$(3) \quad \text{Output Clock Frequency} = \text{Input Frequency} \cdot \frac{M \text{ value}}{N \text{ value} + \text{Counter High} + \text{Counter Low}}$$

以下のように計算式 4 を使用して、外部フィードバック・モードでの PLL 出力クロックを調整します。

$$(4) \quad \text{OUTCLK} = \text{INCLK} \cdot \frac{M \text{ value} + \text{External Feedback Counter High} + \text{External Feedback Counter Low}}{N \text{ value} + \text{Counter High} + \text{Counter Low}}$$

### スペクトラム拡散の調整

以下のように、式 5 を使用して PLL のスペクトラム拡散を調整します。

$$(5) \quad \% \text{spread} = 1 - \frac{M_2 N_1}{M_1 N_2}$$

## リソース Atom 間の接続の変更

Chip Planner と Resource Property Editor で、新しいリソース Atom を作成したり、フィッティング後のネットリストでのリソース Atom 間の既存の接続を操作できます。この機能は、タイミング・エラーを起こす組み合わせパスにパイプライン・レジスタを手動で挿入したり、解析用の予備 I/O ピンに信号を配線するといった、デバッグ段階での小さな変更役に立ちます。以下の手順を使用して、Cyclone III デバイスで新しいレジスタを作成し、レジスタ出力を予備の I/O ピンに配線します。この例は、新しいリソース Atom を作成し、リソース Atom 間の接続を変更する仕組みを示しています。

新しいリソース Atom を作成し、フィッティング後のネットリストにおけるリソース Atom 間の既存の接続を操作するには、以下のステップを実行します。

1. Chip Planner フロアプランで新しいレジスタを作成します
2. Resource Property Editor で Atom を探索します。
3. クロック信号をレジスタに割り当てます。レジスタのクロック入力ポートを右クリックし、**Edit connection** をポイントし、**Other** をクリックします。Node Finder を使用してデザインからのクロック信号を割り当てます。
4. SLOAD 入力ポートを VCC に接続します: レジスタのクロック入力ポートを右クリックし、**Edit connection** をポイントして、**VCC** をクリックします。
5. デザインからのデータ信号を SDATA ポートに割り当てます。
6. 接続ウィンドウにおいて、出力ポート名の下でレジスタのポート名をコピーします。
7. Chip Planner フロアプランで、未使用 I/O リソースを見つけて出力バッファを作成します。
8. Resource Property Editor で新しい I/O Atom を探索します。
9. 出力バッファへの入力ポートを右クリックし、**Edit connection** をポイントして、**Other** をクリックします。
10. 表示されるダイアログ・ボックスで、作成したレジスタの出力ポート名をペーストします。

11. ECO Fitter を実行し、**Check and Save Netlist Changes** ボタンをクリックして変更を適用します。



ECO 接続が成功するかどうかは、利用可能な配線リソースに依存します。**Layers Settings** ダイアログ・ボックスの **Background Color Map** として **Routing Utilization** を選択すると、相対的な配線利用率を表示できます。また、該当するリソース上にマウス・ポインタを置くと表示されるツールチップを使用して、ローカル、ロウ、およびカラム・インタコネクタからの個々の配線チャンネル利用率を表示することもできます。デバイスの配線インタコネクタのアーキテクチャについて詳しくは、デバイスのデータシートを参照してください。

## ECO 後の ステップ

この項では、Chip Planner で ECO を変更した後に実行できる操作について説明します。

### スタティック・タイミング解析の実行

Chip Planner で ECO の変更を行った後、Quartus II Classic Timing Analyzer または Quartus II TimeQuest Timing Analyzer のいずれかで、デザインのスタティック・タイミング解析を実行し、変更がデザインのタイミング性能に悪影響を及ぼしていないことを確認する必要があります。

例えば、特定ピンの遅延チェーン設定の1つをオンにすると、I/O タイミングが変化します。したがって、I/O タイミングが変化してもすべてのタイミング要件が満たされるように、スタティック・タイミング解析を実行する必要があります。

アルテラでは、Chip Planner を使用してデザインを変更するたびに、Quartus II Simulator またはサードパーティEDA シミュレーション・ツールのいずれかで、ゲート・レベルのタイミング・シミュレーションも実行することを推奨しています。



デザインのスタティック・タイミング解析の実行について詳しくは、「Quartus II ハンドブック Volume 3」の「**Quartus II クラシック・タイミング・アナライザ**」または「**Quartus II TimeQuest タイミング・アナライザ**」の章を参照してください。

## まとめ

「Time-to-Market」の圧力が高まるほど、最短時間で完全動作のデザインを作成することが重要になってきます。アルテラはこの課題に対処するために、Quartus II ソフトウェア・スイートの Chip Planner を開発しました。Chip Planner は、デザインのプロアプランの解析および変更を可能にします。また、Chip Planner で行われる ECO 変更にはフル・リコンパイルは不要なので、RTL の変更、再合成、および別の配置配線サイクルという長いプロセスが省略されます。要約すると、Chip Planner は検証サイクルを短縮し、短期間でデザインのタイミング・クロージャが達成されます。

## 参考資料

この章では以下のドキュメントを参照しています。

- 「Quartus II ハンドブック Volume 2」の「デザイン・フロアプランの解析および最適化」の章
- 「AN 474: Quartus II ソフトウェアにおける Stratix III プログラマブル I/O 遅延設定の実装」
- 「Quartus II ハンドブック Volume 2」の「コマンドライン・スクリプト」の章
- 「Cyclone デバイス・ハンドブック」
- 「MAX II デバイス・ハンドブック」
- 「Quartus II ハンドブック Volume 3」の「Quartus II クラシック・タイミング・アナライザ」の章
- 「Quartus II ハンドブック Volume 1」の「階層およびチーム・ベース・デザインのための Quartus II インクリメンタル・コンパイル」の章
- 「Quartus II ハンドブック Volume 3」の「Quartus II プログラマ」の章
- 「Quartus II Settings File Reference Manual」
- 「Quartus II ハンドブック Volume 3」の「Quartus II TimeQuest タイミング・アナライザ」の章
- 「Quartus II ハンドブック Volume 3」の「Quick Design Debugging Using SignalProbe」の章
- 「Stratix デバイス・ハンドブック Volume 1」の「Stratix アーキテクチャ」の章
- 「Stratix デバイス・ハンドブック」
- 「Quartus II ハンドブック Volume 2」の「Tcl Scripting」の章

## 改訂履歴

表 13-2 に、本資料の改訂履歴を示します。

表 13-2. 改訂履歴		
日付およびバージョン	変更内容	概要
2008年5月 v8.0.0	<ul style="list-style-type: none"><li>● デバイス・サポート・リストを更新</li><li>● ブロックRAMおよびDSPブロックに対するECOサポートの説明を変更</li><li>● Stratix PLL ECO の例を訂正</li><li>● リソース Atom 間の接続の変更を示すアプリケーション例を追加</li></ul>	Quartus II ソフトウェア・リリース 8.0 のための更新。
2007年10月 v7.2	14-47 頁の「参考資料」を再編集	—
2007年5月 v7.1.0	初版	—

