

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。

2007年7月 ver 1.0

Application Note 469

## はじめに

Stratix® III デバイスは、低スタティックおよびダイナミック消費電力、業界最高のシグナル・インテグリティを備えた、高速コア性能および高速 I/O を実現するよう設計されています。また、高いロジック集積度も達成しているため、製品のより多くの部分を集積でき、コストとボード・スペースを削減することが可能です。

高集積、高性能 Stratix III デザインのデザイン・プロセス全体を通して、アルテラの推奨事項に従うことが重要です。デザインを成功させるには、デザイン・プロセスの早期に FPGA とシステムをプランニングすることが不可欠です。本資料では、使いやすいガイドラインを提供し、Stratix III デザインで考慮すべき要素について説明します。詳細な仕様、デバイスの機能説明、およびその他のガイドラインが記載されたその他の文書への参照を示しています。Stratix III デバイスのアーキテクチャ、およびデザインで使用する Quartus® II ソフトウェアおよびサードパーティ・ツールの機能について説明します。

本資料に示すガイドラインに従うと、生産性が向上し、デザインでよく発生する誤りを回避することができます。本資料では、表 1 に示すとおり、各ステージが一般的に実行される順序でデザイン・フローのさまざまなステージを説明します。67 ページの「[デザイン・チェックリスト](#)」を使用して、各ガイドラインに従っていることを確認できます。

表 1. デザイン・フローのステージの要約およびガイドライン・トピック (1 / 2)

デザイン・フローのステージ	ガイドライン・トピック
2 ページの「 <a href="#">デバイスの選択</a> 」	デバイス情報、デバイス集積度、パッケージ、スピード・グレード、およびコア電圧の決定
5 ページの「 <a href="#">デバイス・コンフィギュレーションのプランニング</a> 」	コンフィギュレーション手法の概要、コンフィギュレーション機能、QuartusII の設定、オプション・ピン
12 ページの「 <a href="#">早期システム・プランニング</a> 」	プランニング：デザイン仕様、IP の選択、オン・チップ・デバッグ、早期消費電力見積もり
18 ページの「 <a href="#">ボード・デザインの検討事項</a> 」	パワーアップ、電源ピン、コンフィギュレーション・ピン、シグナル・インテグリティ、ボード・レベル検証
27 ページの「 <a href="#">I/O およびクロックのプランニング</a> 」	ピン・アサインメント、早期ピン・プランニング、I/O 機能および接続、クロックおよび PLL 選択、SSN
43 ページの「 <a href="#">デザインおよびコンパイル</a> 」	合成ツール、コーディング・スタイルおよび推奨事項、階層およびチーム・ベース・デザインのプランニング、SOPC Builder

表 1. デザイン・フローのステージの要約およびガイドライン・トピック (2 / 2)

デザイン・フローのステージ	ガイドライン・トピック
52 ページの「タイミング・クロージャおよび検証」	デバイス利用率、タイミング制約および解析、面積およびタイミングの最適化、検証
59 ページの「消費電力の解析および最適化」	解析ツール、最適化テクニック、熱管理オプション



Stratix III デバイス・アーキテクチャの詳細資料は、「[オンライン資料 : Stratix III デバイス](#)」ページを参照してください。Stratix III FPGA に関連する最新のテクニカル情報については、[Knowledge Database](#) を参照してください。

## デバイスの 選択

Stratix III デザイン・プロセスにおける最初のステップは、デザインニーズに最適なデバイス・ファミリー、デバイス集積度、スピード・グレード、パッケージ、およびコア電圧を選択することです。サードパーティ合成ツールまたは Quartus II ソフトウェアでデザインのコンパイルを開始する前に、正しいターゲット・デバイスを設定します。



各デバイスで提供される集積度、ロジック・メモリ・ブロック数、乗算器数、および PLL (Phase-Locked Loop) 数、またパッケージの種類および I/O ピン数については、「[Stratix III デバイス・ハンドブック Volume 1](#)」の「Stratix III デバイス・ファミリーの概要」の章を参照してください。

### ロジック、メモリ、および乗算器のリソース

Stratix III ロジック・ファミリー (L) は、バランスのとれたロジック、メモリ、および乗算器を備え、幅広いアプリケーションに対応します。エンハンスド・ファミリー (E) は、ロジックに対しより豊富なメモリと乗算器を備え、無線、医療用画像処理、および軍用アプリケーションに理想的です。デザイン要件に対して最適なリソース・バランスを持つデバイス・ファミリーを選択します。

Stratix III デバイスは、メモリ、乗算器、およびアダプティブ・ロジック・モジュール (ALM) ロジック・セルなど、異なるデバイス・ロジック・リソース量を持つ幅広い集積度を提供します。また、PLL などの機能の数もデバイスの集積度によって異なる場合があります。必要なロジック集積度の決定は、デザイン・プランニング・プロセスの中でも困難な作業になる可能性があります。ロジック・リソースの多いデバイスほど、大きく複雑なデザインを実装することが可能ですが、通常コストも高くなります。デバイスが小さいほど、スタティック消費電力も少なくなります。デザイン・サイクルの後半でロジックを追加したり、デザインをアップグレードまたは拡張できるように、ある程度の安全マージンを持ってデザインのニーズに最適なデバイスを選択します。49 ページの「階層およびチーム・ベース・デザインのプランニング」で説明するとおり、デバイスの空きスペースを増やして、インクリメンタルまたはチーム・ベース・デザインのデザイン・フロアプランを容易に作成できるようにしたい場合もあります。13 ページの「オン・チップ・デバッグのプランニング」の説明に従って、デバッグ用リソースの確保について検討します。Stratix III デバイスは、4 ページの「デバイスのパーティカル・マイグレーション」で説明するような柔軟性を提供する、特定のデバイス集積度間のパーティカル・マイグレーションをサポートします。

多くの次世代デザインは、現在のデザインを出発点として使用します。アルテラ・デバイスをターゲットとする他のデザインがある場合、それらのリソース利用率を新しいデザインの見積もりとして使用することができます。Settings ダイアログ・ボックスの **Auto device selected by the Fitter** オプションを使用して、Quartus II ソフトウェアの既存のデザインをコンパイルします。リソース利用率を確認して、デザインに適合するデバイスの集積度を決定します。Quartus II ソフトウェアで使用されるコーディング・スタイル、デバイス・アーキテクチャ、および最適化オプションが、デザインのリソース利用率およびタイミング性能に大きく影響する可能性があることに留意してください。リソース利用率の決定について詳しくは、52 ページの「デバイスのリソース利用率レポート」を参照してください。

アルテラの IP (Intellectual Property) デザインの特定のコンフィギュレーションに対するリソース利用率を見積もるには、([www.altera.co.jp](http://www.altera.co.jp)) の資料セクションにある **IP メガファンクション** ページのアルテラ・メガファンクションおよび IP MegaCores のユーザガイドを参照してください。ユーザガイドに記載された値は、デザインのリソース利用率を見積もるのに役立ちます。

## I/O ピン数およびパッケージの種類

Stratix III デバイスは、さまざまな I/O ピン数を持つ省スペースの FineLine BGA パッケージで提供されます。「デバイスのパーティカル・マイグレーション」で説明するように、特定のパッケージ・サイズは異なるデバイス集積度内でのパーティカル・マイグレーションをサポートします。デザインの他のシステム・ブロックとのインタフェース要件を考慮して、必要な I/O ピン数を決定します。既存のデザインを、Quartus II ソフトウェアの Settings ダイアログ・ボックスの **Auto device selected by the Fitter** オプションを使用してコンパイルし、必要な入力ピンおよび出力ピン数を決定します。また、13 ページの「オン・チップ・デバッグのプランニング」の説明に従って、デバッグ用のピンを予約することについても検討します。

## デバイスのバーティカル・マイグレーション

Stratix III デバイスは、同一のパッケージにおけるバーティカル・マイグレーションをサポートしており、専用ピン、コンフィギュレーション・ピン、および電源ピンが同じ配置で、異なる集積度のデバイスにマイグレーションすることができます。これにより、ボード上の Stratix III デバイスを集積度が異なる別の Stratix III デバイスに置き換えることができるため、将来ボード・レイアウトを変更することなく、デザインをアップグレードまたは変更することが可能になります。使用可能なロジック量を増やすことなく、L ロジック・ファミリから E エンハンスド・ファミリへのマイグレーションが可能で、バーティカル・マイグレーションのコストを抑えることができます。

デザインを別のデバイス集積度にマイグレーションするオプションが必要かどうか決定します。デザインが完成に近づいたら、考えられる将来のデバイス・マイグレーションに柔軟に対応するために、デバイスの集積度とパッケージを選択します。デザイン・サイクルの初期段階で、Quartus II ソフトウェアで潜在的なマイグレーション・オプションを指定する必要があります。マイグレーション・デバイスの選択は、デザインのピン配置に影響を与え、デザインが選択したデバイスに準拠することを保証します。マイグレーション・デバイスをデザイン・サイクルの後半で追加することができますが、新しいターゲット・デバイスに適合するようにピン・アサインメントをチェックする余分な作業が必要であり、さらにデザインやボード・レイアウトの変更が必要になる場合もあります。デザインがほぼ完成しマイグレーションが可能な状態となるデザイン・サイクルの後半よりも前半の方がこれらの問題に容易に対応できます。ターゲット・マイグレーション・デバイスを指定するには、Assignments メニューで、Device をクリックし、Settings ダイアログ・ボックスで Migration Devices を選択します。

28 ページの「FPGA ピン・アサインメントの作成」で説明するように、Quartus II Pin Planner の Pin Migration ビューは、現在選択されているデバイスとマイグレーション・デバイスで機能が異なるピンをハイライトします。

## スピード・グレード

デバイスのスピード・グレードは、デバイスのタイミング性能およびタイミング・クロージャ、また消費電力に影響を与えます。Stratix III デバイスは、-2、-3、および -4 の 3 つのスピード・グレードで提供されており、-2 が最も高速です。一般に、高速なデバイスほどコストも高くなります。デザインに必要なスピード・グレードを判断する 1 つの方法は、特定の I/O インタフェースでサポートされているクロック・レートを参照することです。



デバイスのスピード・グレードに応じてデバイスの異なるサイドの I/O ピンを使用するメモリ・インタフェースに対してサポートされるクロック・レートについては、「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスの外部メモリ・インタフェース」の章を参照してください。

一部の設計者は、プロトタイプ作成時に最も高速なスピード・グレードを使用してコンパイル時間を短縮し（短い時間でデザインを最適化し、タイミング要件を満たすことができる）、デザインがタイミング要件を満たした場合は、コストを低減するために生産段階で低速なスピード・グレードに移行します。

Quartus II ソフトウェアは、スピード・グレードごとに異なるタイミング・モデルを使用して、デザインを最適化および解析します。スピード・グレードの異なるデバイスに移行する場合、タイミング解析を実行して、タイミング性能の変化によって生じるタイミング違反がないことを確認する必要があります。

## 選択可能なコア電圧

選択可能なコア電圧では、標準 1.1 V コア電圧よりも多くのスタティックおよびダイナミック消費電力を削減できる 0.9 V のコア電圧を使用するオプションが得られます。性能の最大化が要求されるデザインではコア電圧を 1.1 V にし、消費電力の最小化が要求されるデザインでは 0.9 V を使用することができます。Quartus II ソフトウェアでデバイスやスピード・グレードとともにコア電圧を選択します。これは選択可能なコア電圧は、Stratix III ファブリック性能に影響するためです。Quartus II ソフトウェアは、選択したコア電圧に固有のタイミングおよびパワー・モデルを使用して、すべてのタイミング依存およびパワー依存解析と最適化を実行します。

低消費電力の電圧設定を使用する場合は、デバイスの集積度、パッケージ、およびスピード・グレードの組み合わせに対して 0.9 V オプションが使用可能なことを確認します。0.9 V オプションでは、スピード・グレードのサポートは制限されています。

## デバイス・ コンフィギュ レーションの プランニング

Stratix III デバイスは、SRAM セルをベースにしています。SRAM メモリは揮発性のため、電源が投入されるたびに Stratix III デバイスにコンフィギュレーション・データをダウンロードする必要があります。デバイス・コンフィギュレーション手法は、システム設計者やボード設計者がシステムにコンパニオン・デバイスを追加する必要があるかどうか判断できるように、初期段階で選択します。また、手法ごとに必要な接続が異なるため、ボード・レイアウトはプログラマブル・デバイスに使用するコンフィギュレーション手法に依存します。コンフィギュレーション・ピンに関連するボード・デザイン・ガイドラインについては、[18 ページ](#)の「[ボード・デザインの検討事項](#)」を参照してください。

さらに、Stratix III デバイスは、コンフィギュレーション・メモリ容量および時間を短縮するコンフィギュレーション・データの圧縮復元機能、ユーザのデザインを保護するデータ暗号化を使用したデザイン・セキュリティ、およびリアルタイム・リモート・システム・アップグレードを提供しています。これらのコンフィギュレーション機能のサポートは、選択したコンフィギュレーション手法によって異なります。また、Stratix III デバイスには、オプションのコンフィギュレーション・ピンとボードおよびシステム・デザインに必要なすべての情報を取得するために、事前に選択して Quartus II ソフトウェアの起動時に設定しなければならないコンフィギュレーション・オプションもあります。

この項では、次の内容について説明します。

- 「[コンフィギュレーション手法の選択](#)」
- 8 ページの「[コンフィギュレーション機能](#)」
- 9 ページの「[Quartus II のコンフィギュレーション設定](#)」
- 10 ページの「[コンフィギュレーション・ファイルおよびプログラミング・ファイル](#)」



コンフィギュレーションについて詳しくは、「[Stratix III デバイス・ハンドブック Volume 1](#)」の「[Stratix III デバイスのコンフィギュレーション](#)」の章を参照してください。詳細は、[コンフィギュレーション・センタ](#)を参照してください。このウェブ・ページでは、コンフィギュレーション問題のデバッグを支援するトラブルシュータ [JTAG Configuration & ISP Troubleshooter](#) および [FPGA Configuration Troubleshooter](#) へのリンクを提供しています。

## コンフィギュレーション手法の選択

Stratix III デバイスは、4 種類のコンフィギュレーション手法のいずれか 1 つを使用してコンフィギュレーションできます。

- ファースト・パッシブ・パラレル (FPP)
- ファースト・アクティブ・シリアル (AS)
- パッシブ・シリアル (PS)
- JTAG (Joint Test Action Group)

コンフィギュレーション手法を選択するには、Stratix III デバイスの MSEL ピンを High または Low にドライブします。



サポートされているコンフィギュレーション手法については、[Stratix III Device Configuration Center](#) を参照してください。Stratix III デバイスでサポートされるコンフィギュレーション手法、必要なコンフィギュレーション手法の実行方法、コンフィギュレーション電圧規格および POR 時間、および MSEL ピン設定など必要なすべてのオプション・ピン設定については、「[Stratix III デバイス・ハンドブック Volume 1](#)」の「[Stratix III デバイスのコンフィギュレーション](#)」の章を参照してください。

すべてのコンフィギュレーション手法では、コンフィギュレーション・デバイス、ダウンロード・ケーブル、または外部コントローラ (MAX®II デバイスまたはマイクロプロセッサなど) のいずれかを使用します。

## コンフィギュレーション・デバイス

FPP および PS コンフィギュレーション手法では、アルテラのエンハンスト・コンフィギュレーション・デバイス (EPC) を使用することができます。アルテラのシリアル・コンフィギュレーション・デバイス (EPCS) は、ファースト AS コンフィギュレーション手法で使用されます。コンフィギュレーション・デバイスがユーザの Stratix III デバイスのコンフィギュレーション・ビットストリーム・ファイル・サイズをサポートしているかどうか確認してください。大規模の Stratix III デバイスのコンフィギュレーション・ファイル・サイズを縮小するために、フラッシュ・メモリを用いて MAX II デバイスまたはマイクロプロセッサを使用するか、または圧縮機能を使用することができます。



エンハンスト・コンフィギュレーション・デバイスについて詳しくは、「エンハンスト・コンフィギュレーション・デバイス・データシート」および「コンフィギュレーション・ハンドブック Volume 2」の「アルテラのエンハンスト・コンフィギュレーション・デバイス」の章を参照してください。シリアル・コンフィギュレーション・デバイスについて詳しくは、「コンフィギュレーション・ハンドブック Volume 2」の「シリアル・コンフィギュレーション・デバイス・データシート」の章を参照してください。このデータシートには、Stratix III デバイスをサポートするシリアル・コンフィギュレーション・デバイスの一覧が含まれていません。圧縮されていないファイルのサイズが、特定のデバイスの集積度に対して大きすぎる場合、この一覧にファイル・サイズを縮小するための圧縮機能が含まれていることが示されます。

## ダウンロード・ケーブル

Quartus II プログラマは、アルテラのプログラミング・ダウンロード・ケーブルを介して PS または JTAG インタフェースを使用し、Stratix III デバイスのコンフィギュレーションを直接サポートします。アルテラのダウンロード・ケーブルで接続されたデバイスにデザインの変更を直接ダウンロードして、簡単にプロトタイプを作成して、複数のデザインの繰り返しをすばやく連続して実行できます。同じダウンロード・ケーブルを使用してボード上のコンフィギュレーション・デバイスをプログラムし、SignalTap II ロジック・アナライザなどのデバッグ・ツールを使用することができます。



アルテラの最新のダウンロード・ケーブルの使用方法については、以下の使用を参照してください。

- [ByteBlaster II Parallel Port Download Cable Data Sheet](#)
- [USB-Blaster II USB Port Download Cable Data Sheet](#)
- [EthernetBlaster Download Cable User Guide](#)

## MAX II パラレル・フラッシュ・ローダ

システムに既にコモン・フラッシュ・インタフェース (CFI) 対応フラッシュ・メモリが実装されている場合は、それを Stratix III デバイスのコンフィギュレーション・ストレージにも利用できます。MAX II デバイスでパラレル・フラッシュ・ローダ (PFL) 機能を使用することにより、JTAG インタフェースを通じて CFI フラッシュ・メモリ・デバイスをプログラムすることができます。また、フラッシュ・メモリ・デバイスから Stratix III デバイスへのコンフィギュレーションを制御するためのロジックを提供し、コンフィギュレーション・データのサイズを縮小するための圧縮をサポートします。この PFL 機能を使用して、PS および FPP 両方のコンフィギュレーション・モードがサポートされます。このコンフィギュレーション手法を選択する場合、システム・デザイン・サイクルの初期段階でサポートされているフラッシュ・デバイスのリストをチェックし、それに応じてプランニングする必要があります。



PFL について詳しくは、「AN 386: Using the Parallel Flash Loader with the Quartus II Software」を参照してください。

## コンフィギュレーション機能

この項では、Stratix III のコンフィギュレーション機能、およびこれらの機能がデータ圧縮、デザイン・セキュリティ、およびリモート・システム・アップグレードのデザイン・プロセスにどのように影響するかを説明します。

### データ圧縮

データ圧縮機能をイネーブルにすると、Quartus II ソフトウェアは圧縮されたコンフィギュレーション・データでコンフィギュレーション・ファイルを生成します。この圧縮ファイルは、コンフィギュレーション・デバイスまたはフラッシュ・メモリで必要な容量を低減し、Stratix III デバイスにビットストリームを送信するのに必要な時間を短縮します。Stratix III デバイスがコンフィギュレーション・ファイルを復元するのに必要な時間は、コンフィギュレーション・データをデバイスに送信するのに必要な時間よりも短くなっています。Stratix III デバイスは、FPP (MAX II デバイス / マイクロプロセッサ + Flash 使用時)、ファースト AS、および PS コンフィギュレーション手法で復元をサポートしています。Stratix III の復元機能は、FPP でエンハンスド・コンフィギュレーション・デバイスを使用する場合、または JTAG を使用してコンフィギュレーションする場合には使用できません。



データの圧縮について詳しくは、「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのコンフィギュレーション」の章を参照してください。

## コンフィギュレーション・ビットストリーム暗号化を使用したデザイン・セキュリティ

コンフィギュレーション・ビットストリームを暗号化するデザイン・セキュリティ機能は、複製、リバース・エンジニアリング、および改ざんから Stratix III デザインを保護します。デザイン・セキュリティ機能は、外部ホスト (MAX II デバイスやマイクロプロセッサ) でファースト・パッシブ・パラレル (FPP) コンフィギュレーション・モードを使用して、Stratix III FPGA をコンフィギュレーションするとき、あるいはファースト・アクティブ・シリアル (AS) またはパッシブ・シリアル (PS) コンフィギュレーション手法を使用するときに使用できます。デザイン・セキュリティ機能は、ファースト AS コンフィギュレーション・モードでのリモート・アップデートのときにも使用できます。エンハンスド・コンフィギュレーション・デバイスを使用した FPP 手法、または JTAG コンフィギュレーション手法で Stratix III デバイスをコンフィギュレーションする場合、デザイン・セキュリティ機能は使用できません。



コンフィギュレーション・ビットストリーム暗号化を使用したデザイン・セキュリティについて詳しくは、「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのデザイン・セキュリティ」の章を参照してください。セキュリティ・キーをプログラムするためのステップは、今後提供される Stratix III デザイン・セキュリティ・アプリケーション・ノートに記載される予定です。

## リモート・システム・アップグレード

Stratix III デバイスは、AS コンフィギュレーション手法でリモート・アップデートをサポートしています。AS コンフィギュレーション手法では、シリアル・コンフィギュレーション・デバイスのコンフィギュレーション・メモリ容量を節約する必要がある場合は、コンフィギュレーション・データのリアルタイム復元と共にリモート・アップデートを実行することができます。

`altremote_update` メガファンクションを使用して、またはリモート・システム・アップグレード・アトムをインスタンスして、リモート・システム・アップグレード・インタフェースを実装できます。



リモート・システム・アップグレード機能について詳しくは、「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのリモート・システム・アップグレード」の章を参照してください。`altremote_update` メガファンクションについて詳しくは、「`altremote_update` Megafunction User Guide」を参照してください。

## Quartus II のコンフィギュレーション設定

この項では、コンフィギュレーション・ファイルまたはプログラミング・ファイルを生成するためにコンパイル前に Quartus II ソフトウェアで設定できるいくつかのコンフィギュレーション・オプションについて説明します。これらの設定およびピンは、ボードおよびシステム・デザインに影響を与えます。

## オプションのコンフィギュレーション・ピン

**Device and Pin Options** ダイアログ・ボックスの **General** タブで、以下のオプションのコンフィギュレーション・ピンをイネーブルにすることができます。

- **CLKUSR ピン—Enable user-supplied start-up clock (CLKUSR)** オプションにより、内部オシレータまたは CLKUSR ピンに供給される外部クロックのどちらのクロック・ソースを初期化に使用するかを選択することができます。
- **INIT\_DONE ピン—INIT\_DONE ピンをモニタして、デバイスが初期化を完了してユーザ・モードになっているかどうかをチェックすることができます。** このピンは **Enable INIT\_DONE output** オプションでイネーブルにします。INIT\_DONE ピンはオープン・ドレイン出力で、V<sub>CCPGM</sub> への外部プルアップを必要とします。

## エラー発生後のコンフィギュレーションの再開

**Device and Pin Options** ダイアログ・ボックスの **General** タブで、**Auto-restart after configuration error** オプションをイネーブルにすることができます。このオプションでは、コンフィギュレーション・エラーが発生すると、デバイスは nSTATUS を Low にドライブします。これにより、デバイスは内部でリセットされます。デバイスは、リセット・タイムアウト期間を過ぎると nSTATUS ピンを解放します。nSTATUS ピンが、内部プルアップを提供する外部コンフィギュレーション・デバイスに接続されていない場合は、V<sub>CCPGM</sub> に 10 kΩ の外部プルアップ抵抗を介して接続させます。

## コンフィギュレーション・ファイルおよびプログラミング・ファイル

この項では、Stratix III デバイスおよびコンパニオン・デバイスに使用するコンフィギュレーションおよびプログラミング・ファイルに関連する情報を示します。



プログラミング・ファイル形式およびプログラマの使用については、「Quartus II ハンドブック Volume 2」の「Quartus II Programmer」の章を参照してください。

## プログラミング・ファイル

Stratix III のデータをコンフィギュレーション・デバイスに格納するために、追加ファイルを作成するかデフォルトの SRAM オブジェクト・ファイル (.sof) データを異なるファイル形式に変換し、コンフィギュレーション・デバイスにプログラムすることができます。**Device and Pin Options** ダイアログ・ボックスで、コンパイル中に追加プログラミング・ファイルを作成するようにソフトウェアを設定できます。SOF を変換するには、File メニューの **Convert Programming Files** をクリックします。コンフィギュレーション・デバイスをプログラムするために、プログラマ・オブジェクト・ファイル (.pof) および JTAG Indirect Configuration (.jic) ファイルは Quartus II Programmer で使用され、Hexout (.hexout)、ロウ・バイナリ・ファイル (.rbf)、表形式テキスト・ファイル (.ttf)、および RPD ファイルは他のプログラマで使用されます。

マルチ・デバイス・コンフィギュレーション・チェーンを使用するときは、**Convert Programming Files** ダイアログ・ボックスで、各デバイスの SOF を 1 つのコンフィギュレーション・ファイルに結合します。1 つのコンフィギュレーション・ファイルを設定するときには、コンフィギュレーション・ファイルとボード上のデバイスの順序を一致させてください。

## コンフィギュレーション・ファイル・サイズの見積もり

圧縮されていない RBF のサイズは、Stratix III デバイスの圧縮されていないコンフィギュレーション・ファイルのサイズとほぼ同じです。複数のデバイス・コンフィギュレーションに必要なストレージ容量を算出するには、各デバイスのファイル・サイズを加算してください。

ロウ・バイナリ・ファイルのサイズのデータは、デザインをコンパイルする前のファイル・サイズの見積もりにのみ使用してください。16 進 (.hex) フォーマットや TTF フォーマットなど、コンフィギュレーション・ファイル形式によってファイル・サイズが異なります。ただし、Quartus II ソフトウェアの特定のバージョンでは、同じデバイスを対象としたデザインの実圧縮コンフィギュレーション・ファイルのサイズは同じになります。

Stratix III デバイスは、圧縮されたコンフィギュレーション・ビットストリームを受信して、このデータをリアルタイムで復元することができるため、アクティブ・シリアルおよびパッシブ・シリアル方式に必要なメモリおよびコンフィギュレーション時間を低減します。圧縮を使用した場合、圧縮率はデザインに依存するため、ファイル・サイズはコンパイルするたびに変わる可能性があります。

## 早期システム・プランニング

Stratix III デバイスが搭載されたシステムでは、FPGA は通常システム全体において大きな役割を果たし、システム・デザインの他の部分に影響を与えます。Quartus II ソフトウェアでデザインを完成させる前に、デザイン・プロセスの早期段階で FPGA デバイス情報を提供することにより、システムおよびボード・デザインの早期プランニングが可能になります。詳細なデザイン仕様を作成することによって、デザイン・プロセスを開始することが重要です。IP (Intellectual Property)、オン・チップ・デバッグ手法など、システム・デザインに影響を与える FPGA の重要な側面について決定する必要があります。また、ソース・コードを作成する前、あるいはデザインの暫定版が存在するときには、早期消費電力見積もりも実行して PCB ボードおよびシステム設計者に情報を提供することができます。この項では、次の内容について説明します。

- 12 ページの「デザイン仕様の作成」
- 13 ページの「IP の選択」
- 13 ページの「オン・チップ・デバッグのプランニング」
- 16 ページの「早期消費電力見積もり」

### デザイン仕様の作成

ロジック・デザインを作成したりシステム・デザインを完成させる前に、デザインの詳細な仕様を確定する必要があります。仕様では、システムの動作を定義し、FPGA の I/O インタフェースを指定します。また、基本デザイン機能のブロック図も含めます。IP (Intellectual Property) ブロックを含めるための推奨事項については、「[IP の選択](#)」を参照してください。これらの仕様を作成することにより、デザイン効率が向上します。

また、この段階でテスト・プランを作成すると、DFT (Design-For-Testability) および DFM (Design-For-Manufacturability) も容易になります。例えば、ビルトイン・セルフ・テスト機能を実行して、インタフェースをドライブしますか？ その場合は、FPGA デバイス内で Nios II プロセッサとともに UART インタフェースを使用することができます。すべてのデザイン・インタフェースを検証する機能が必要な場合があります。システムに実装されたデバイスの解析およびデバッグに関連するガイドラインについては、[13 ページの「オン・チップ・デバッグのプランニング」](#)を参照してください。

デザインに複数の設計者が携わっている場合、この時点で共通のデザイン・ディレクトリ構造について検討することも得策です。これによって、デザインの統合ステージが容易になります。[49 ページの「階層およびチーム・ベース・デザインのプランニング」](#)に、チーム・ベースのデザインに関するより詳細な提案が記載されています。

## IP の選択

アルテラおよびアルテラのサードパーティ IP パートナは、アルテラのデバイスに最適化された標準 IP コアを多数提供しています。これらのパラメータ化された IP ブロックはデザインに容易に実装でき、システムの実装時間と検証時間が短縮されるため、ユーザは独自の価値の追加に集中することができます。IP の選択は、しばしばシステム・デザイン、特にシステム内の FPGA が他のデバイスとインタフェースする部分に影響を与えます。システム・デザイン内のどの I/O インタフェースまたは他のブロックが IP コアを使用して実装できるかを検討し、これらのコアを FPGA デザインに組み込むようプランニングします。

多くの IP コアで利用可能な OpenCore Plus 機能により、IP ライセンスを購入する前に FPGA をプログラムして、ハードウェアでデザインを検証することができます。評価版は、デザインを一定時間実行できる **Untethered** モード、または **Tethered** モードをサポートしています。**Tethered** モードでは、アルテラのシリアル JTAG (Joint Test Action Group) ケーブルをボードの JTAG ポートとホスト・コンピュータとの間に接続する必要があります。このホスト・コンピュータ上では **Quartus II Programmer** がハードウェア評価中動作している必要があります。**Tethered** モードを使用する場合は、ボード・デザインでこの動作モードがサポートされていることを確認します。



提供されている IP コアの説明は、アルテラ・ウェブサイトの製品情報の **IP (Intellectual Property)** ページを参照してください。

## オン・チップ・デバッグのプランニング

システムおよび設計者ごとに、最適なデバッグ・ツールは異なります。システム・ボード、**Quartus II** プロジェクト、およびデザインで適切なオプションをサポートできるように、デザイン・プロセスの早期段階でオン・チップ・デバッグ・オプションを評価します。プランニングによってデバッグに費やす時間を短縮でき、後で使用するデバッグ手法に合わせて変更を加える必要がなくなります。デバイスにおける内部信号および I/O ピンのアクセス性のために、デバッグ・ピンを追加するだけでは不十分な場合があります。まず、「**オン・チップ・デバッグ・ツール**」から希望のデバッグ・ツールを選択し、次に **15 ページ**の「**デバッグ・ツールのプランニング・ガイドライン**」を参照します。

### オン・チップ・デバッグ・ツール

検証ツールの **Quartus II** ポートフォリオには、以下のシステム・デバッグ機能が含まれます。

- **SignalProbe** インクリメンタル配線 — この機能は、元のデザインの配線に影響を与えることなく、内部信号をすばやく I/O ピンに配線することによってデザイン検証の効率を高めます。完全に配線されたデザインから開始して、デバッグ用の信号を選択し、以前予約した I/O ピンまたは現在未使用の I/O ピンに配線することができます。

- **SignalTap® II エンベデッド・ロジック・アナライザ**— このロジック・アナライザは、FPGA デバイスでデザインをフル・スピードで実行しながら、外部装置または追加 I/O ピンを使用しないで内部信号および I/O 信号の状態をプローブすることによって、FPGA デザインをデバッグするのに役立ちます。カスタム・トリガ条件ロジックを定義して、精度を向上させ、問題を特定する能力を改善します。SignalTap II エンベデッド・ロジック・アナライザは、デザインの内部ノードまたは I/O ピンの状態をキャプチャするために外部プローブやデザイン・ファイルへの変更を必要としません。キャプチャしたすべての信号データは、ユーザがデータを読み出して解析できるようになるまでデバイス・メモリに保存されます。この機能をインクリメンタル・コンパイル機能と併用することで、コンパイル時間を短縮することができます。
- **ロジック・アナライザ・インタフェース**— このインタフェースにより、内部 FPGA 信号を外部ロジック・アナライザに接続および送信して解析できます。これによって、外部ロジック・アナライザまたはミックスド・シグナル・オシロスコープの最新機能を活用することができます。この機能を使用して、多数の内部デバイス信号をデバッグのために少数の出力ピンに接続することができます。
- **In-System Memory Content Editor**— この機能は、JTAG インタフェースを介してイン・システム FPGA メモリおよび定数への読み出しおよび書き込みアクセスを提供し、システム内でデバイスが動作している間に、FPGA のメモリ内容および定数値への変更をより簡単にテストできるようにします。
- **イン・システム・ソースおよびプローブ**— この機能は、カスタマイズされたレジスタ・チェーンを設定して、ロジック・デザインに組み込まれたノードをドライブまたはサンプリングし、シンプルな仮想ステミュラスを提供し、組み込まれたノードの現在の値をキャプチャします。SignalTap II ロジック・アナライザを使用してトリガ条件を設定し、外部テスト装置を使用しないでデザインをエキササイズするための単純なテスト・ベクタを作成し、JTAG チェーンを使用してランタイム・コントロール信号をダイナミックに制御することができます。
- **Virtual JTAG メガファンクション**— `sld_virtual_jtag` メガファンクションを使用すれば、システム・レベルのデバッグのためのソフトウェアにおけるプロセッサ・ベースのデバッグ・ソリューションおよびデバッグ・ツールなど、独自のシステム・レベルのデバッグ基盤を構築することができます。このメガファンクションは、HDL コードで直接インスタンス化し、デバイスの JTAG インタフェースを使用して FPGA デザインの一部にアクセスするために、1 本または複数の透過通信チャネルを提供することができます。



これらのデバッグ・ツールについて詳しくは、「`sld_virtual_jtag` Megafunction User Guide」および「Quartus II ハンドブック Volume 3」の「Quick Design Debugging Using SignalProbe」、**「SignalTap II エンベデッド・ロジック・アナライザを使用したデザインのデバッグ」**、「外部ロジック・アナライザを使用したイン・システム・デバッグ」、「FPGA メモリおよび定数のインシステム・アップデート」、および「Design Debugging Using In-System Sources and Probes」を参照してください。

## デバッグ・ツールのプランニング・ガイドライン

いずれかのオン・チップ・デバッグ・ツールを使用する場合は、この項で説明するとおり、システム・ボード、Quartus II プロジェクト、およびデザインを開発するときにツールをプランします。

SignalTap II ロジック・アナライザは、同期インタフェースに最適です。非同期インタフェースをデバッグする場合は、最も高い精度で信号を表示するために SignalProbe または外部ロジック・アナライザの使用を検討します。

SignalTap II エンベデッド・ロジック・アナライザ、ロジック・アナライザ・インタフェース、In-System Memory Content Editor、イン・システム・ソースおよびプローブ、および Virtual JTAG メガファンクションはすべて、イン・システム・デバッグを実行するのに JTAG 接続を必要とします。デバッグに使用できる JTAG ポート付きのシステムおよびボードをプランします。

また、JTAG デバッグ機能では、JTAG ハブ・ロジックを実装するために少量の追加ロジック・リソースも必要になります。デザイン・サイクルの早期段階で適切な機能を設定すると、これらのデバイス・リソースを早期リソース見積もりに含めて、デバイスでロジックが過剰になるのを防ぐことができます。

SignalTap II エンベデッド・ロジック・アナライザは、システム動作中にデバイス・メモリを使用してデータをキャプチャします。このデバッグ手法を活用するのに十分なメモリ・リソースを確保するために、デバッグ時に使用するデバイス・メモリを予約することを検討します。

SignalTap II エンベデッド・ロジック・アナライザでインクリメンタル・デバッグを使用するには、**Full incremental compilation** オプションがオンになっていることを確認します。このオプションは、Quartus II ソフトウェア・バージョン 6.1 以降で作成したプロジェクトに対しては、デフォルトでオンに設定されますが、既存のプロジェクトでは自動的にオンにはなりません。インクリメンタル・コンパイルがオンになっていない場合は、デバッグ機能を追加するか、または SignalTap II 設定を変更するときは、デザイン全体を再コンパイルする必要があります。SignalTap II エンベデッド・ロジック・アナライザでインクリメンタル・コンパイルを使用することによって、デバッグに必要なコンパイル時間が大幅に短縮されます。

SignalProbe およびロジック・アナライザ・インタフェースには、デバッグ用の I/O ピンが必要です。デバッグ用の I/O ピンを予約しておき、後でデバッグ信号に対応するためにデザインやボードを変更しなくてすむようにします。ロジック・アナライザ・インタフェースでは、必要に応じて信号をデザイン I/O ピンでマルチプレクス化することができます。対象のボードで、デバッグ信号がシステムの動作に影響を与えないデバッグ・モードがサポートされていることを確認します。

外部ロジック・アナライザまたはミックスド・シグナル・オシロスコープを使用する場合は、アナライザ用のピン・ヘッドまたは Mictor コネクタを組み込みます。

カスタム・デバッグ・アプリケーションに Virtual JTAG メガファンクションを使用する場合は、これをインスタンス化し、デザイン・プロセスの一部として組み込む必要があります。

イン・システム・ソースおよびプローブ機能では、HDL コードでメガファンクションをインスタンス化することも必要です。さらに、オプションで SignalTap II エンベデッド・ロジック・アナライザをメガファンクションとしてインスタンス化し、それをデザインのノードに手動で接続して、タップされたノード名が合成時に変更されないようにすることができます。デバッグ・ブロックをインクリメント・コンパイル用の個別デザイン・パーティションとして追加して、再コンパイル時間を短縮することができます。

RAM や ROM ブロック、あるいは lpm\_constant メガファンクションに In-System Memory Content Editor を使用するには、MegaWizard® Plug-In Manager でメモリ・ブロックを作成するときに、**Allow In-System Memory Content Editor to capture and update content independently of the system clock** オプションをオンにします。

## 早期消費電力見積もり

FPGA の消費電力は、デザインの重要な検討事項です。適切な電力供給量を把握して、電源、電圧レギュレータ、デカップリング、ヒート・シンク、および冷却システムを設計するには、デバイスの消費電力を正確に見積もる必要があります。消費電力の見積もりと解析には、以下の2つの重要なプランニング要件があります。

- **熱プランニング:** 冷却ソリューションによって、デバイスで発生した熱を十分に放逸します。特に、計算された接合温度がデバイスの標準仕様の範囲内に収まる必要があります。
- **電源プランニング:** 電源は十分な電流を供給することによって、デバイスの動作をサポートしなければなりません。

FPGA デバイスの消費電力は、ロジック・デザインによって異なります。このことが、ボード仕様およびレイアウトの早期段階での消費電力の見積もりを困難にしています。アルテラの PowerPlay Early Power Estimator (EPE) スプレッドシートでは、デバイスおよびデザインで使用されるデバイス・リソース、そして動作周波数、トグル・レート、および環境面への配慮などに関する情報を処理して、デザインが完成する前に消費電力を見積もることができます。スプレッドシートを使用して、周囲温度やヒート・シンク、空気流量、ボードの熱モデルなどに関する情報を入力して、デバイスの接合温度を計算することができます。EPE は、次にデザインの消費電力、電流見積もり、熱解析を計算します。

既存のデザインがない場合は、デザインで使用するデバイス・リソース数を見積もって、その情報を手動で入力します。スプレッドシートの精度は、デバイス・リソースの入力と見積もりに依存します。この情報が（デザインの途中または完了後に）変更された場合は、消費電力見積もり結果の精度が低くなります。既存のデザインまたは部分的にコンパイルされたデザインがある場合は、Quartus II ソフトウェアの **Generate PowerPlay Early Power Estimator File** コマンドを使用して、スプレッドシートに入力することができます。

PowerPlay Early Power Estimator スプレッドシートには、消費電力見積もりファイルの情報を分解しスプレッドシートに転送する **Import Data** マクロがあります。マクロを使用しない場合は、データを **Early Power Estimator** スプレッドシートに手動で転送します。既存の **Quartus II** プロジェクトがフル・デザインの一部のみをカバーしている場合は、最終デザインで使用する追加リソースを手動で入力する必要があります。消費電力見積もりファイル情報をインポートした後、スプレッドシートを編集し、追加デバイス・リソースを追加するか、またはパラメータを調整することができます。

デザインが完成したら、**Quartus II** ソフトウェアで **PowerPlay Power Analyzer** ツールを使用して、デザインの消費電力を正確に見積もって、熱バジェットおよび電源バジェットに違反していないことを確認します。59 ページの「消費電力の解析」を参照してください。

サポートしている各デバイス・ファミリの **PowerPlay Early Power Estimator** スプレッドシートおよびユーザガイドは、アルテラ・ウェブサイトのサポートのデバイス・サポートのセクション ([www.altera.co.jp/support/devices/estimator/pow-powerplay.html](http://www.altera.co.jp/support/devices/estimator/pow-powerplay.html)) を参照してください。



**PowerPlay Early Power Estimator** スプレッドシートの使用について詳しくは、「**PowerPlay Early Power Estimator User Guide For Stratix III FPGAs**」を参照してください。消費電力の見積もりおよび解析について詳しくは、「**Quartus II** ハンドブック Volume 3」の「**PowerPlay** による電力解析」の章を参照してください。

## ボード・ デザインの 検討事項

Stratix III デバイスへのインタフェースの設計では、さまざまな要因がプリント基板 (PCB) のデザインに影響を与えます。この項では、次の内容の重要なガイドラインについて説明します。

- 「デバイスのパワーアップ」
- 19 ページの「電源ピン間の接続」
- 21 ページの「コンフィギュレーション・ピンの接続」
- 24 ページの「ボード関連の Quartus II 設定」
- 25 ページの「シグナル・インテグリティの検討事項」
- 27 ページの「ボード・レベル・シミュレーションおよびアドバンスド I/O タイミング解析」



ボード・デザインのガイドラインについて詳しくは、アルテラのボード・デザイン・ガイドライン・ソリューション・センタ ([www.altera.co.jp/support/devices/board/brd-index.html](http://www.altera.co.jp/support/devices/board/brd-index.html)) を参照してください。このソリューション・センタでは、設計者がアルテラのデバイスと他の要素を統合する高速 PCB を適切に実装できるよう支援するアプリケーション・ノートおよびその他の文書を提供しています。

### デバイスのパワーアップ

Stratix III デバイスは、外部デバイスを使用せずに、ホット・プラグインまたはホット・スワップとしても知られるホット・ソケット (活線挿抜) およびパワー・シーケンスをサポートします。ユーザは、動作中のシステム・バスやシステムに実装されたボードに影響を与えることなく、システムの動作中に Stratix III デバイスまたはボードをシステムに取り付けたり、取り外すことができます。ホット・ソケット機能によって、複数の電圧が混在するプリント基板 (PCB) 上で Stratix III デバイスを使用することができます。

このシステムは信号を電源投入前および投入時に、デバイスの I/O に入力することができます。V<sub>CCIO</sub>、V<sub>CC</sub>、V<sub>CCPGM</sub>、および V<sub>CCPD</sub> ピンは、任意のシーケンスでパワーアップまたはパワーダウンできます。各電源のランプ・アップおよびランプ・ダウン・レートの範囲は 50  $\mu$ s ~ 100 ms です。電源ランプは、モニタリングでなければなりません。ホット・ソケットの状態では、Stratix III デバイスの出力バッファはシステムのパワーアップまたはパワーダウン時にオフになります。また、Stratix III デバイスは、デバイスがコンフィギュレーションされて推奨動作条件下で動作するまで I/O をドライブしません。

CONF\_DONE、nCEO、および nSTATUS のコンフィギュレーション・ピンはコンフィギュレーション時に必要になるため、これらのピンでホット・ソケット回路を使用することはできません。これらのピンに期待される動作は、パワーアップおよびパワーダウン・シーケンス中にドライブ・アウトすることです。

Stratix III デバイスに電源が投入されたとき、電源が所定の時間（最大電源ランプ時間、 $t_{\text{RAMP}}$  として規定）内に推奨動作範囲に達した場合は、パワー・オン・リセット・イベントが発生します。Stratix III デバイスの最大電源ランプ時間は 100 ms、最小電源ランプ時間は 50  $\mu\text{s}$  です。パワー・オン・リセット（POR）回路は、 $V_{\text{CC}}$ 、 $V_{\text{CCL}}$ 、 $V_{\text{CCPD}}$ 、 $V_{\text{CCPGM}}$ 、および  $V_{\text{CCPT}}$  の各電源電圧レベルをモニタし、電源投入時にこれらの電源電圧レベルが安定するまで FPGA をリセット状態に維持します。デバイスがユーザ・モードになった後も、POR 回路はユーザ・モード中にブラウンアウト状態を継続してモニタします。Stratix III デバイスでは、12 ms または 100 ms の POR 標準遅延時間から選択することのできるピン選択オプション（PORSEL）が提供されています。いずれの場合も、外部コンポーネントを使用して nSTATUS ピンを Low にアサートすることによって、POR 時間を延長することができます。ボードが最大電源ランプ時間仕様を満たさない場合は、デバイスを適切にコンフィギュレーションしてユーザ・モードに入るには、POR 時間を延長する必要があります。



詳細については、「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのホット・ソケットおよびパワー・オン・リセット」の章を参照してください。

パワー・シーケンスは正しい動作を実現するための必須条件ではありませんが、マルチ・レール電源システムを設計するときには、長期間にわたるデバイスの信頼性低下を防止するために、各レールのパワーアップ・タイミングを考慮する必要があります。適切なシーケンスおよび電圧レギュレータ・デザインにより、デバイスの突入電流を低減することができます。



詳細については、「AN 448: Stratix III の消費電力管理デザイン・ガイド」を参照してください。

## 電源ピン間の接続

Stratix III の選択可能なコア電圧では、省電力 0.9 V コア電圧  $V_{\text{CCL}}$  または標準 1.1 V コア電圧  $V_{\text{CCL}}$  を使用するオプションが提供されます。いずれの場合も、PLL、周辺電源  $V_{\text{CC}}$  および  $V_{\text{CCD\_PLL}}$  には 1.1 V の電源が必要です。 $V_{\text{CC\_CLKIN}}$ 、 $V_{\text{CCA\_PLL}}$ 、 $V_{\text{CCPT}}$ 、および  $V_{\text{CCBAT}}$  の各電圧には、2.5 V の電源が必要です。 $V_{\text{CCPGM}}$  ピンは、1.8 V、2.5 V、3.0 V または 3.3 V でコンフィギュレーション・ピンに給電します。

I/O 電圧  $V_{CCIO}$  の接続はデザインの I/O 規格に依存し、1.2 V、1.5 V、1.8 V、2.5 V、3.0 V および 3.3 V をサポートします。デバイスの出力ピンは、 $V_{CCIO}$  レベルが I/O 規格の推奨動作範囲外にある場合は、I/O 規格の仕様を満たしません。 $V_{CCPD}$  ピンは、3.0 V  $V_{CCIO}$  の場合は 3.0 V、2.5 V  $V_{CCIO}$  の場合は 2.5 V に接続しなければなりません。電圧リファレンス (VREF) ピンは、特定の I/O 規格の電圧リファレンスとして機能します。VREF ピンは、主に電圧バイアスとして使用され、多くの電流をソースまたはシンクしません。この電流の最大値は、10  $\mu$ A で、特定の VREF ピンをリファレンス電圧として使用する I/O ピン数とは無関係です。電圧はレギュレータまたは抵抗ディバイダ・ネットワークによって生成することができます。I/O バンクの  $V_{CCIO}$  電圧および VREF ピンについて詳しくは、31 ページの「選択可能な I/O 規格と柔軟性の高い I/O バンク」を参照してください。

ボードを設計するときには、すべての電源ピンが正しく接続されているかどうかチェックし、ボード上の FPGA の電源ピンまたはその他の電源ピンの独自の要件を調べて、ボード上のどのデバイスが電源レールを共有できるかを判断します。



Stratix III デバイスに必要な電源電圧のリストおよび推奨動作条件については、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスの DC およびスイッチング特性」の章を参照してください。

Stratix III デバイスでは、アナログ回路と同様にリニア・レギュレータを使用して  $V_{CCA\_PLL}$  および  $V_{CCPT}$  電源ピンを駆動することが推奨されています。デジタル電圧レールは、効率またはコストを検討した上で、リニア・レギュレータまたはスイッチング・レギュレータによって駆動することができます。



パワー・シーケンスおよび電源分配アーキテクチャの詳細など、消費電力管理システムのデザインに関する推奨事項は、「AN 448: Stratix III の消費電力管理デザイン・ガイド」を参照してください。

### デカップリング・キャパシタ

電源の要件が高まるに伴って、電源全体のシグナル・インテグリティを向上させるためのボード・デカップリングがますます重要になっています。

Stratix III デバイスには、外部 PCB デカップリング・コンデンサとボルテージ・レギュレータ・モジュールでサポートできない高周波デカップリングを供給する、エンベデッド・オン・パッケージおよびオン・ダイ・デカップリング・コンデンサが搭載されています。これらの低インダクタンス・コンデンサは、電源ノイズを抑制して優れたシグナル・インテグリティ性能を達成し、また外部 PCB デカップリング・コンデンサの数を削減し、ボード・スペースの節約、コストの削減、PCB デザインの大幅な簡略化を実現します。アルテラの Stratix III デバイスのためのボード・デカップリング・ガイドラインが発行されたら、内容を確認します。

## PLL ボード・デザイン・ガイドライン

クロックおよび PLL 手法の設計について詳しくは、39 ページの「クロックおよび PLL の選択」および 40 ページの「PLL デザイン・ガイドライン」を参照してください。PLL には、デジタル・デバイスに組み込まれたアナログ・コンポーネントが搭載されています。以下のリストに、PLL を使用するボードの設計とジッタの低減についての検討事項を示します。

- デバイスですべての PLL を使用しない場合でも、すべての  $V_{CCA\_PLL}$  および  $V_{CCD\_PLL}$  電源ピンをそれぞれ 2.5 V および 1.1 V 電源に接続する。
- 電源から各  $V_{CCA\_PLL}$  ピンに厚い配線パターン（最低 20 mils）を走らせる。
- 絶縁されたリニア・レギュレータを使用して、 $V_{CCA\_PLL}$  を駆動する。
- すべての  $V_{CCD\_PLL}$  電源ピンをボード上で最もノイズの少ないデジタル電源に接続する。
- 各  $V_{CCA\_PLL}$  および  $V_{CCD\_PLL}$  ピンをデカップリング回路でフィルタする。
- 電源がボードに入る  $V_{CCA\_PLL}$  および  $V_{CCD\_PLL}$  にフェライト・ビーズとタンタル並列コンデンサを使用する。

## コンフィギュレーション・ピンの接続

コンフィギュレーション手法に応じて、異なるプルアップ / プルダウン抵抗あるいはシグナル・インテグリティ要件が適用される場合があります。また、一部のコンフィギュレーション・ピンには、未使用の場合には特定の要件があります。コンフィギュレーション・ピンは正しく接続することが非常に重要です。この項では、一般的な問題に対処するためのガイドラインを提供します。



専用および兼用コンフィギュレーション・ピンのリスト、および機能の説明と接続ガイドラインについては、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスのコンフィギュレーション」の章を参照してください。

## DCLK および TCK シグナル・インテグリティ

TCK トレースおよび / または DCLK トレースが、オーバーシュート、アンダーシュート、またはリングのないクリーンな信号を生成するようにしてください。ボードを設計するときは、クロック・ラインのレイアウトと同じ手法を使用して TCK トレースと DCLK トレースをレイアウトします。このモジュール・ベースの I/O バンク構造は、ピン使用効率を向上し、デバイスの移行を容易にします。DCLK 信号にノイズが多い場合、コンフィギュレーションが影響を受け、巡回冗長検査 (CRC) エラーが発生することがあります。デバイスのチェーンでは、チェーン内の TCK ピンまたは DCLK ピンのノイズによって、チェーン全体の JTAG プログラミングまたはコンフィギュレーションが失敗することがあります。



チェーン内のデバイスの接続について詳しくは、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスのコンフィギュレーション」の章を参照してください。

## JTAG ピン

JTAG インタフェースを使用しない場合、JTAG ピンを安定した電圧レベルに接続します。JTAG コンフィギュレーションは、他のすべてのコンフィギュレーション手法より優先されるため、これらのピンをコンフィギュレーション中にフローティング状態のままにしたり、トグルしないようにする必要があります。

JTAG インタフェースを使用する場合、この項のガイドラインに従ってください。

### JTAG ピンの接続

JTAG モードで動作するデバイスは、TDI、TDO、TMS、および TCK の 4 本の専用ピン、および TRST の 1 本のオプション・ピンを使用します。TCK ピンは内部ウィーク・プルダウン抵抗を備えていますが、TDI ピン、TMS ピン、および TRST ピンは内部ウィーク・プルアップ抵抗（標準 25 k $\Omega$ ）を備えています。JTAG 出力ピン TDO およびすべての JTAG 入力ピンは、2.5 V/3.0 V/3.3 V の V<sub>CCPD</sub> 電源で駆動します。すべての JTAG ピンは、LVTTLI/O 規格のみをサポートします。

デバイスの JTAG ピンを、ダウンロード・ケーブルのヘッダに正しく接続します。ピンの順番が逆にならないようにしてください。チェーン内に複数のデバイスがある場合、1 つのデバイスの TDO ピンをチェーン内の次のデバイスの TDI ピンに接続します。

デバイスがコンフィギュレーション、ユーザ・モード、またはパワーアップ中の場合、JTAG ピンのノイズによってデバイスが不定の状態または不定モードになることがあります。

パワーアップ時に JTAG ステート・マシンをディセーブルするには、TCK ピンを Low にプルダウンして、TCK 上に予期しない立ち上がりエッジが発生しないようにします。アルテラでは、抵抗を通して TCK ピンを Low に、TMS ピンを High に接続することを推奨しています。

1 k $\Omega$  抵抗を通して TRST を V<sub>CCPD</sub> に接続します。ピンをグラウンドに接続すると JTAG 回路がディセーブルされます。

### ダウンロード・ケーブルの動作電圧

ターゲット・ボードの 10 ピン・ヘッダからアルテラのダウンロード・ケーブルに供給される動作電圧により、ダウンロード・ケーブルの動作電圧レベルが決まります。すべての Stratix III デバイスの専用 JTAG ピンはバンク 1A に配置されており、V<sub>CCPD</sub> で駆動されます。ダウンロード・ケーブルはデバイスの JTAG ピンにインタフェースするため、ダウンロード・ケーブルの動作電圧と JTAG ピンの電圧が適合していることを確認してください。

V<sub>CCIO</sub> レベルが異なるデバイスを含む JTAG チェインでは、V<sub>CCIO</sub> レベルが高いデバイスは、V<sub>CCIO</sub> レベルが同じかそれ以下のデバイスをドライブしなければなりません。このデバイス構成では、チェーンの末端に 1 個のレベル・シフタが必要です。この配置が不可能な場合は、レベル・シフタをチェーンに追加する必要があります。



チェーン内のデバイスに複数の電圧を印加する JTAG チェインでの接続に関する推奨事項は、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスの IEEE 1149.1 (JTAG) バウンダリ・スキャン・テスト」の章を参照してください。

#### JTAG 信号のバッファリング

JTAG のシグナル・インテグリティ、特に TCK 信号のシグナル・インテグリティに応じて、JTAG チェインにバッファを追加しなければならない場合があります。これは TCK 信号が JTGA クロックであり、また最高速の JTAG 信号であるためです。アルテラでは、信号をコネクタでバッファすることを推奨しています。これは、ケーブルやボードのコネクタは適切な伝送ラインではなく、信号にノイズを発生させる傾向があるためです。このようなコネクタでの最初のバッファに続いて、チェーンが延長されたり、信号がボード・コネクタを通過する必要が生じた場合は必ずバッファを追加してください。

ケーブルが 3 個以上のデバイスをドライブする場合は、ケーブル・コネクタで JTAG 信号をバッファして、信号の劣化を防ぎます。これはボード・レイアウト、負荷、コネクタ、ジャンパ、ボード上のスイッチ類に依存します。JTAG 信号のインダクタンスまたはキャパシタンスに影響を与える要素がボードに追加されると、チェーンにバッファを追加する必要性が高まります。

並列にドライブされる TCK 信号と TMS 信号の場合は、各バッファでドライブされる負荷を 8 以下に抑える必要があります。ジャンパまたはスイッチをパスに追加した場合は、負荷の数を少なくします。

#### MSEL コンフィギュレーション・モード・ピン

コンフィギュレーション手法を選択するには、Stratix III デバイスの MSEL ピンを High または Low にドライブします。MSEL ピンは、それらのピンが存在するバンクの  $V_{CCPGM}$  電源で駆動されます。MSEL [2..0] ピンには、常にアクティブな 5 k $\Omega$  内部プルダウン抵抗があります。POR およびリコンフィギュレーション中、MSEL ピンがロジック Low およびロジック High と判定されるには、LVTTL VIL レベルおよび VIH レベルであることが必要です。不正なコンフィギュレーション手法の検出の問題を回避するために、適切な MSEL ピンを  $V_{CCPGM}$  または GND に接続してください。あるいは、テストまたはデバッグ中にコンフィギュレーション・モードを切り替えるために、各ピンを 0  $\Omega$  抵抗で、 $V_{CCPGM}$  または GND のいずれかに接続できるようにボードを設定します。MSEL ピンをマイクロプロセッサや他のデバイスでドライブしてはなりません。MSEL ピンをフローティング状態のままにしないでください。

#### その他のコンフィギュレーション・ピン

以下のピンを含むすべての専用および兼用コンフィギュレーション・ピンを正しく接続する必要があります。

V<sub>CCPD</sub> 専用電源ピンは、I/O プリドライバ、JTAG 入力および出力ピン、およびデザイン・セキュリティ回路を駆動するために使用します。このピンは、3.0 V I/O 規格の場合は 3.0 V、2.5 V 以下の I/O 規格の場合は 2.5 V に接続します。

nIO\_PULLUP ピンは、コンフィギュレーション実行前および実行中に、ユーザ I/O ピンおよび兼用 I/O ピン (DATA[7..0]、CLKUSR、INIT\_DONE、DEV\_OE、DEV\_CLRn、CRC\_ERROR) の内部プルアップ抵抗をオンまたはオフのいずれにするかを選択します。内部プルアップ抵抗をオフにするには、nIO-PULLUP を V<sub>CCPGM</sub> に直接接続するか 1 k $\Omega$  のプルアップ抵抗を使用し、オンにするには nIO-PULLUP を GND に直接接続します。

nCE チップ・イネーブル・ピンは、コンフィギュレーション実行中、初期化中、およびユーザ・モードでは Low に保持する必要があります。シングル・デバイス・コンフィギュレーションまたは JTAG プログラミングでは、nCE を Low に接続します。マルチ・デバイス・コンフィギュレーションでは、最初のデバイスの nCE は Low に接続し、nCEO ピンはチェーン内の次のデバイスの nCE ピンに接続します。

## ボード関連の Quartus II 設定

Quartus II ソフトウェアは、ボード設計時に考慮すべき FPGA I/O ピンに対するオプションを提供します。

### デバイス・ワイド出力イネーブル・ピン

Stratix III デバイスは、ユーザがデバイスの I/O のすべてのトライ・ステートを無効にできるオプションのチップ・ワイド出力イネーブルをサポートしています。この DEV\_OE ピンが Low にドライブされると、すべての I/O ピンはトライ・ステートになり、High にドライブされるとプログラムどおりに動作します。このチップ・ワイド出力イネーブルを使用するには、デザインをコンパイルする前に Quartus II ソフトウェアで、**Device & Pin Options** ダイアログ・ボックスの **General** タブにある **Enable device-wide output enable (DEV\_OE)** をオンにします。

## 未使用ピン

ボード・デザインを柔軟に行うために、Quartus II ソフトウェアで未使用ピンの状態を、トライ・ステートになる入力、グラウンドをドライブする出力、指定されていない信号をドライブする出力、バスホールドでトライ・ステートになる入力、あるいはウィーク・プルアップでトライ・ステートになる入力のいずれかに指定できます。シグナル・インテグリティを改善するには、未使用ピンをグラウンドをドライブする出力として設定し、それらをボード上のグラウンド・プレーンに直接接続します。これにより、短いリターン・パスを作成することによってインダクタンスを低減し、隣接する I/O 上のノイズを低減します。消費電力を低減するには、グラウンドをドライブするようクロック・ピンを設定し、未使用 I/O ピンをトライ・ステートになる入力として設定します。デザインに適切な設定を行うには、**Device & Pin Options** ダイアログ・ボックスの **Unused Pins** タブにある **Reserve all unused pins** に許容される 5 つの状態の 1 つを選択するか、Pin Planner で特定のピンに **Reserve Pin** アサインメントを適用します。

デザインをコンパイルするとき、Quartus II ソフトウェアはデバイス・ピンの接続方法を指定するためのピン・レポート・ファイル（.pin）を生成します。未使用 I/O ピンは、ソフトウェアで設定した未使用ピン・オプションに応じて、Quartus II レポート・ファイルでマークされます。GND\* として指定されるすべての I/O ピンは、デバイスのノイズ耐性を向上させるためにグラウンドに接続するか、未接続のままにすることができます。ボード上の RESERVED I/O ピンは、規定されていない信号をドライブするため、すべて未接続のままにしておきます。RESERVED I/O ピンが V<sub>CC</sub>、グラウンド、または別の信号ソースに接続されると、競合が起これ、デバイスの出力ドライバが損傷することがあります。RESERVED\_INPUT I/O ピンは、ボード上の High または Low 信号に接続でき、RESERVED\_INPUT\_WITH\_WEAK\_PULLUP ピンと RESERVED\_INPUT\_WITH\_BUS\_HOLD ピンは未接続のままにしておくことができます。

## シグナル・インテグリティの検討事項

この項では、電圧リファレンス・ピン、同時スイッチング・ノイズ、および I/O 終端に関連するボード・デザイン・ガイドラインについて説明します。



シグナル・インテグリティについて詳しくは、「Stratix III FPGA のシグナル・インテグリティ」ホワイトペーパーを参照してください。

## 電圧リファレンス・ピン

VREF 電圧リファレンス信号にノイズがないことが重要です。電圧リファレンス・ピンおよび I/O 規格について詳しくは、30 ページの「I/O 機能およびピン接続」を参照してください。VREF ピンの電圧変動は、入力のスレッショルド感度に影響を及ぼすことがあります。

## 同時スイッチング・ノイズ

(近接する) 多数のピンが同時に電圧レベルを変化させた場合、同時スイッチング・ノイズ (SSN) が問題になります。SSN で発生したノイズによって、ノイズ・マージンが減少し誤ったスイッチングが発生する可能性があります。SSN はデバイス・パッケージで一般的に発生する問題です。ノイズの低減に役立つボード・レイアウトの推奨事項については、アルテラのボード・デザイン・ガイドライン・ソリューション・センター ([www.altera.co.jp/support/devices/board/brd-index.html](http://www.altera.co.jp/support/devices/board/brd-index.html)) の PCB ガイドラインを参照してください。

デバイス近くのボード層で大きなバス信号が突発的に発生した場合に、SSN の低減に役立つことがあります。2 つの信号層が隣接している場合は、可能であれば配線パターンを直角に配線します。可能な場合、 $2\times \sim 3\times$  の配線パターン幅を分離して使用します。

## I/O 終端

電圧リファレンス形式の I/O 規格には、入力リファレンス電圧  $V_{REF}$  と、終端電圧  $V_{TT}$  の両方が必要です。受信デバイスのリファレンス電圧は、送信デバイスの終端電圧に追従します。電圧リファレンス形式の I/O 規格は、それぞれに固有の終端設定が必要です。例えば、SSTL2 規格では優れたノイズ・マージンを持つ信頼性の高い DDR メモリ・システムを作成するために、適切な抵抗性の信号終端方式が重要です。

シングル・エンド非電圧リファレンス形式の I/O 規格では終端は不要ですが、反射を抑え、シグナル・インテグリティを向上させるためにインピーダンス・マッチングが必要です。

Stratix III の直列および並列 On-Chip Termination では、外部コンポーネントが必要ないためデザインが簡潔になります。その代わりとして、外部プルアップ抵抗を使用して、SSTL や HSTL などの電圧リファレンス形式の I/O 規格を終端できます。

差動 I/O 規格は、通常はレシーバの 2 つの信号間に終端抵抗を必要とします。終端抵抗は、信号ラインの差動負荷インピーダンスと整合しなければなりません。Stratix III デバイスは、LVDS を使用するときにオプションの差動オンチップ抵抗を提供します。特定の専用クロック入力ペアは、差動終端をサポートしません。

On-Chip Termination 機能および制約については、30 ページの「I/O 機能およびピン接続」を参照してください。

## ボード・レベル・シミュレーションおよびアドバンスド I/O タイミング解析

I/O 信号がボード・セットアップにおいて確実にレシーバ・スレッショルド・レベルを満たすようにするには、IBIS モデルを使用するサードパーティ製のボード・レベル・シミュレーション・ツールでフル・ボード配線シミュレーションを実行します。

この機能を Quartus II ソフトウェアで使用できる場合、**Settings** ダイアログ・ボックスの **EDA Tool Settings** ページの **Board-level signal integrity analysis** で、**IBIS** を選択します。

IBIS モデルの一覧は、アルテラのウェブサイトにも適宜掲載されます。



このシミュレーション・フローについては、「Quartus II ハンドブック Volume 3」の「Signal Integrity with Third-Party Tools」の章を参照してください。

ボード・デザインに高速インタフェースを備えた FPGA デバイスを含める場合は、システムを適切に動作させる上で、シグナル・インテグリティとボード配線伝播遅延を把握することが不可欠です。特に高速デザインでは、I/O およびボード・プランニングの一部としてボード・レベル・タイミングを解析する必要があります。

Quartus II ソフトウェアでは、選択した I/O 規格のボード・トレース・モデルをコンフィギュレーションし、「ボードを考慮した」シグナル・インテグリティ・レポートを生成することができます。**Enable Advanced I/O Timing** がオンのとき、**TimeQuest** タイミング・アナライザは、システム・レベルでの信号動作を把握するために、I/O パッファ、パッケージ、およびボード・トレース・モデルに対するシミュレーション結果を使用して、より精度の高い I/O 遅延と特別なレポートを生成します。これらの高度なタイミング・レポートを参考にして、I/O アサインメントとボード・デザインを変更し、タイミングおよびシグナル・インテグリティを向上させることができます。

## I/O および クロックの プランニング

Stratix III デバイスの多数の I/O ピンおよび高度なクロック管理機能では、I/O リソースおよびクロック・リソースのプランニングおよび割り当ては重要なタスクです。利用可能な I/O リソースを効率的にプランニングして利用率を高め、シグナル・インテグリティに関連する問題を防止するために、さまざまな検討事項を理解することが重要です。FPGA デザインの性能にとって、優れたクロック管理システムも不可欠です。Stratix III デバイスは、完全なクロック管理ソリューションを実現する階層クロック構造および最先端の機能を備えた複数の PLL を提供します。

FPGA の I/O およびクロック接続は、システムの他の部分およびボード・デザインに影響与えるため、これらの接続は、デザイン・サイクルの早期段階で計画することが重要です。

この項では、以下の項目について説明します。

- 28 ページの「FPGA ピン・アサインメントの作成」
- 29 ページの「早期ピン・プランニングおよび I/O アサインメントの解析」
- 30 ページの「I/O 機能およびピン接続」
- 39 ページの「クロックおよび PLL の選択」
- 40 ページの「PLL デザイン・ガイドライン」
- 43 ページの「同時スイッチング・ノイズ」

## FPGA ピン・アサインメントの作成

Quartus II Pin Plannerを使用してピン・アサインメントを実施します。Assignments メニューで、**Pin Planner** をクリックします。Pin Planner GUI を使用すると、I/O プランニング・プロセスを通じて、I/O バンク、VREF グループ、および差動ピンの組み合わせを識別できます。特定のピンを検索するには、Pin Planner スプレッドシート・インタフェースで右クリックし、**Pin Finder** をクリックします。

マイグレーション・デバイスを選択すると、4 ページの「デバイスのパーティカル・マイグレーション」で説明したとおり、Pin Migration View で現在選択しているデバイスとマイグレーション・デバイスで機能が変化するピンがハイライトされます。

デザイン・フローで通常スプレッドシートを使用する場合は、Microsoft Excel スプレッドシートを Quartus II ソフトウェアにインポートして I/O プランニング・プロセスを開始する選択肢があります。さらに、すべてのピンが割り当てられている場合は、I/O アサインメントを含むカンマ区切り値 (.csv) ファイルをスプレッドシート用にエクスポートすることもできます。

Quartus II ソフトウェアでデザインをコンパイルする場合、フィッタにおける I/O アサインメントの解析により、アサインメントがすべてのデバイス要件を満足することが検証され、問題があればメッセージが生成されます。

その後、ピン位置に関する情報を PCB 設計者に渡すことができます。特にピン配置を変更する必要がある場合は、デザインが配置されたボード上で正しく機能するように、Quartus II ソフトウェアと回路図ツールおよびボード・レイアウト・ツールとの間でピン・アサインメントを一致させることが重要です。Pin Planner は、特定の PCB デザイン EDA ツールと密接に統合され、これらのツールからピン位置の変更を読み出して提案された変更をチェックすることができます。デザインをコンパイルすると、Quartus II ソフトウェアにより PIN ファイルが生成されます。このファイルを使用して、ボード回路図で各ピンが正しく接続されていることを確認することができます。

ピンを割り当てるときには、30 ページの「I/O 機能およびピン接続」の説明に従って、ターゲットとなるピンが確実に適切な I/O 規格およびその他の必要な I/O 機能をサポートするようにしてください。デバイスのトップおよびボトムまたはレフトおよびライトの特定の I/O バンクは、異なる I/O 規格と電圧レベルをサポートします。Pin Planner で、I/O 規格を割り当て、I/O 関連の設定を行うことができます。クロックやグローバル・コントロール信号などの信号には、39 ページの「クロックおよび PLL の選択」で説明するとおり、必ず正しい専用ピン入力を使用してください。



Pin Planner を使用した I/O 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「I/O 管理」の章を参照してください。Quartus II ソフトウェアとサードパーティの EDA ツール間での I/O 情報の受け渡しについて詳しくは、「Quartus II ハンドブック Volume 2」の「Mentor Graphics PCB Design Tools Support」および「Cadence PCB Design Tools Support」の章を参照してください。

## 早期ピン・プランニングおよび I/O アサインメントの解析

多くのデザイン環境において、FPGA 設計者は、トップレベル I/O ピンを早期にプランニングして、ボード設計者が PCB のデザインおよびレイアウトの開発を開始できるようにすることを望みます。FPGA デバイスの I/O 機能とボード・レイアウト・ガイドラインは、ピン位置およびその他のタイプのアサインメントに影響を及ぼします。ボード・デザイン・チームが FPGA のピン配置を指定する場合は、できるだけ早期に FPGA 配置配線ソフトウェアでピン位置を確認して、ボード・デザインを変更する必要が生じないようにすることが非常に重要です。

Quartus II Pin Planner により、28 ページの「FPGA ピン・アサインメントの作成」で説明するとおり、I/O ピン・アサインメントのプランニング、アサインメント、および検証を簡単に行うことができます。Quartus II の **Start I/O Assignment Analysis** コマンドにより、ターゲットの FPGA アーキテクチャでピン位置とピン・アサインメントがサポートされていることがチェックされます。チェックには、リファレンス電圧ピンの使用、ピン配置アサインメント、および I/O 規格の混在が含まれます。I/O アサインメントの解析を使用して、デザイン・プロセスを通じて作成または変更した I/O 関連アサインメントを検証することができます。

FPGA のピン・プランニングを早期に開始することにより、早期のボード・レイアウトに対する信頼が高まり、エラーが生じる可能性が低くなり、デザインの全体的な「Time-To-Market」が短縮されます。ソース・コードを設計する前に、Quartus II Pin Planner を使用してアルテラ FPGA 用の暫定的なピン配置を作成することができます。

システム開発者は通常、デザイン・プロセスの早期に、標準 I/O インタフェース（メモリ、バス・インタフェースなど）、デザインで使用する IP コア、およびシステム要件で定義されるその他の I/O 関連アサインメントに関する情報を持っています。Pin Planner の **Create/Import Megafunction** 機能は、MegaWizard Plug-In Manager とインタフェースし、I/O インタフェースを使用するカスタム・メガファンクションおよび IP コアの作成とインポートを可能にします。オプションがピン配置ルールに影響を及ぼすので、PLL および LVDS ブロックをダイナミック・フェーズ・アラインメント（DPA）などのオプションを含めて入力します。できるだけ多くの I/O 関連情報を入力したら、**Create Top-Level Design File** コマンドを使用して、トップレベル・デザインのネットリスト・ファイルを生成します。I/O 解析の結果を使用して、ピン・アサインメントまたは IP パラメータを変更し、I/O インタフェースがデザイン要件を満たし、Quartus II ソフトウェアのピン・チェックに合格するまで、チェック・プロセスを繰り返すことができます。

プランニングが完了したら、ピン位置に関する情報を PCB 設計者に渡すことができます。Pin Planner は、特定の PCB デザイン EDA ツールと密接に統合され、これらのツールからピン位置の変更を読み出して提案された変更をチェックすることができます。特にピン配置を変更する必要がある場合は、デザインが配置されたボード上で正しく機能するように、Quartus II ソフトウェアと回路図ツールおよびボード・レイアウト・ツールとの間でピン・アサインメントを一致させることが重要です。



I/O 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「I/O 管理」の章を参照してください。

デザインが完成したら、Quartus II Fitter により生成されるレポートとメッセージを使用して、ピン・アサインメントの最終的なサインオフを行います。

## I/O 機能およびピン接続

Stratix III I/O は、使いやすさと迅速なシステム統合を実現すると同時に、広帯域幅を提供するように設計されています。バーティカル・マイグレーションが可能な共通バンク構造を持つ独立したモジュール・ベースの I/O バンクにより、高速 I/O の効率と柔軟性が高まります。この項では、I/O 機能とピン接続に関するガイドラインを提供します。デバイスの I/O バンクでの異なる I/O 信号の種類と I/O 規格のサポート、およびデザインで使用可能なその他の I/O 機能について説明します。さらに、メモリ・インタフェース、パッド配置ガイドライン、および特殊ピンの接続に関する情報も提供します。

### I/O 信号の種類

Stratix III デバイスは、シングル・エンド、電圧リファレンス形式のシングル・エンド、および差動 I/O 規格などの I/O 規格を幅広くサポートします。この項では、信号の種類に関する一般的なガイドラインを提供します。

シングル・エンド I/O 信号は、シンプルなレール・トゥ・レール・インタフェースを提供します。その速度は、大きな電圧振幅とノイズの制限を受けます。シングル・エンド I/O は、システム内の反射によって悪影響が生じない限り、終端は必要ありません。

電圧リファレンス形式の信号は、複数のピンで同時に電圧レベルを変化させる同時スイッチング出力 (SSO) (例えば、外部メモリ・インタフェースのデータおよびアドレス・バス) の影響を軽減します。また、電圧振幅の低減によりロジック遷移レートを改善し、終端要件に起因する反射によって発生するノイズを抑制します。唯一の欠点は、リファレンス電圧源 VTT に終端コンポーネントが追加されることです。

差動信号は、隣接する追加の反転データ信号とのペアで使用することにより、シングル・エンドおよび電圧リファレンス形式の信号のインタフェース性能障壁をなくします。また、この信号はクリーンなリファレンス電圧を必要としません。これは、低い振幅電圧とコモン・モード・ノイズ除去機能によるノイズ耐性によって実現できます。この実装の検討事項には、サンプリング・クロックを生成する専用 PLL の使用、および反転ペアと非反転ペア間の位相差をなくするためのトレース長の一致が含まれます。

Stratix III デバイスの I/O ピンはペアで編成されており、差動規格をサポートします。各 I/O ピン・ペアは、差動入力動作のみをサポートする特定のクロック・ピンを除き、差動入力動作または差動出力動作をサポート可能です。デザインのソース・コードでは、1 本のピンだけが差動ペアとなるように定義し、そのペアのポジティブ・エンドにピン・アサインメントを行います。差動 I/O 規格を指定すると、Quartus II ソフトウェアは自動的に対応するネガティブ・ピンを配置します。

### 選択可能な I/O 規格と柔軟性の高い I/O バンク

Stratix III デバイスの I/O ピンは、モジュラー I/O バンクと呼ばれるグループで編成されます。I/O バンク数は、デバイス集積度に応じて 16 ~ 24 バンクになります。小さなデバイスからより大きなデバイスへの移行時に、バンク・サイズは同じか増加しますが、決して減少することはありません。適切な I/O 規格をサポートする I/O バンクにピンを配置してください。

ボードは、各バンクに対して 1 つの  $V_{CCIO}$  電圧レベルを供給する必要があります。各 I/O バンクは、特定のバンクの  $V_{CCIO}$  ピンでパワーアップされ、他の I/O バンクの  $V_{CCIO}$  からは独立しています。1 つの I/O バンクが、 $V_{CCIO}$  と同じ電圧でドライブする出力信号をサポートします。I/O バンクは、異なる差動 I/O 規格が割り当てられたいかなる数の入力信号でも同時にサポートできます。

電圧リファレンス形式の I/O 規格に対応するために、Stratix III デバイスの各 I/O バンクは、共通の VREF バスに電源を供給する複数の VREF ピンをサポートします。VREF ピンを、バンク内で I/O 規格に対応する正しい電圧に設定してください。各 I/O バンクが任意の時点で持つことができるのは、1 つの  $V_{CCIO}$  電圧レベルと 1 つの VREF 電圧レベルだけです。VREF ピンは、電圧基準として使用されていない場合に汎用 I/O ピンとして使用することができないので、 $V_{CCIO}$  または GND に接続しなければなりません。

シングル・エンド規格または差動規格に対応する I/O バンクは、すべての電圧リファレンス形式の規格が同じ VREF 設定を使用している限り、電圧リファレンス形式の規格をサポートできます。性能上の理由により、電圧リファレンス形式の入力規格は、電源として独自の  $V_{CCPD}$  レベルを使用します。したがって、任意の  $V_{CCIO}$  電圧を持つバンクに電圧リファレンス形式の入力信号を配置することができます。電圧リファレンス形式の双方向信号および出力信号は、I/O バンクの  $V_{CCIO}$  電圧レベルでドライブ・アウトしなければなりません。

LVDS I/O ピンに対するシングル・エンド I/O ピンの配置は制限されています。シングル・エンド出力と LVDS I/O を分離する必要がある I/O ピンの数を規定するピン配置ルールに従ってください。Quartus II Fitter は、コンパイル中にこれらのガイドラインが満足されていることを検証します。



詳細については、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスの I/O 機能」の章を参照してください。Stratix III デバイスの I/O 規格とサポートされている I/O 規格に対応する電圧レベル、そして入力および出力の  $V_{CCIO}$ 、 $V_{CCPD}$ 、VREF、およびボード VTT の標準値について説明した項を参照してください。さらに、Stratix III デバイスの I/O バンクの図を参照してください。これには各 I/O バンクの位置と各バンクでサポートする機能が示されています。各バンク内の I/O 数を示す図は、各デバイス集積度に固有のバンク情報を提供します。各バンク内で組み合わせ可能な I/O 規格については、I/O バンクの制約について説明した項を参照してください。また、LVDS に関する制約について詳しくは、I/O 配置ガイドラインについて説明した項を参照してください。



各 I/O 規格の電気的特性について詳しくは、「Stratix III デバイス・ハンドブック Volume 2」の「Stratix III デバイスの DC およびスイッチング特性」の章を参照してください。

## メモリ・インタフェース

デバイスのすべてのサイドのすべての I/O バンクは、外部メモリ・デバイスとインタフェースするために、専用レジスタおよび位相シフト回路を備えています。Stratix III デバイスは、DDR3、DDR2、DDR SDRAM、および RLDRAM II、QDRII/QDRII + SRAM とのインタフェースをサポートします。

セルフ・キャリプレート・メガファンクション (ALTMEMPHY) は、Stratix III デバイスの I/O 構造を活用するように最適化されています。ALTMEMPHY メガファンクションでは、外部メモリ・インタフェース機能を設定し、システムに最適な物理インタフェース (PHY) の設定を支援します。アルテラおよびサードパーティのメモリ・インタフェース用 IP コアもこのメガファンクションを使用しています。

DQS データ・ストロープ・ピンは、メモリ規格に応じて、双方向シングル・エンド信号、単方向差動信号、双方向差動信号、または単方向コンプリメンタリ信号になります。単方向リード・データ・ストロープまたはクロックを Stratix III DQS ピンに接続し、単方向ライト・データ・ストロープまたはクロックに使用可能な DQ または DQS ピン (データ・ピンと同じ I/O バンク内) を使用します。Stratix III デバイスは、差動リード・データ・ストロープ / クロック動作の差動入力バッファ、およびコンプリメンタリ・リード・データ・ストロープ / クロック動作の各 CQn データ・ストロープ・ピンのための独立した DQS ロジック・ブロックを提供します。

DQ データ・ピンは双方向または単方向信号になります。単方向読み出しデータ信号を Stratix III の DQ ピンに接続し、単方向書き込みデータ信号を DQ ピンの別のグループに接続します。定義済みの DQ グループを使用して、DQ ピンをデータ・バス内の信号に割り当てます。

Stratix III デバイスでは、DQS ピンおよび DQ ピンの位置は固定されています。メモリ・インタフェース回路は、どの Stratix III I/O バンクでも使用できます。レフトおよびライトの I/O ピンは LVDS I/O 規格をサポートするために高いピン・キャパシタンスを備えているため、トップおよびボトム of the I/O バンクが提供する性能は、レフト・サイドおよびライト・サイドの I/O バンクが提供する性能よりも高くなります。

DQS 位相シフト回路は、DLL (Delay-Locked Loop) を使用して DQS/CQn ピンに必要なクロック周期をダイナミックに測定します。各 DLL のリファレンス・クロックは、PLL 出力クロックまたは DLL のいずれかのサイドに位置する 2 本の専用クロック入力ピンから供給されます。

アルテラの IP を使用してデバイス内に複数のメモリ・インタフェースを設計する場合は、1 つのメモリ・インタフェースを一度だけ設計してそれを繰り返しインスタンス化する方法を取らずに、必ず各インスタンス用の固有インタフェースを生成して、良好な結果が得られるようにしてください。



Stratix III デバイスの外部メモリ・デバイスとの接続について詳しくは、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスの外部メモリ・インタフェース」の章を参照してください。関連情報については、アルテラ・ウェブサイトのメモリ・ソリューション・センター ([www.altera.co.jp/technology/memory/mem-index.jsp](http://www.altera.co.jp/technology/memory/mem-index.jsp)) を参照してください。特定のメモリ・インタフェースに関連する具体的なガイドラインについては、該当するアプリケーション・ノートを参照してください。

## パッド配置ガイドライン

バンクの  $V_{CCIO}$  電源は、バンク内のスイッチング出力からのノイズに敏感です。 $V_{CCIO}$  電源で許容ノイズ・レベルを維持するために、差動パッドに対するシングル・エンド I/O パッドの配置に制約があります。Quartus II ソフトウェアは、これらの制約を自動的にチェックします。

バンク内にシングル・エンド電圧リファレンス形式の入力があると、Quartus II ソフトウェアは VREF パッドおよび電源ペア ( $V_{CCIO}$  と GND) に対する出力の配置に関する制約を自動的にチェックします。この制約は  $V_{CCIO}$  電源で許容ノイズ・レベルを維持し、VREF レールのシフトによる出力スイッチング・ノイズを防止するためのものです。

隣接するパッケージ・ピンと隣接するボンド・パッドは対応しません。I/O ピンの位置をプランニングする際に、Quartus II Pin Planner の Pad View でピンのボンド・パッドを検索することができます。

特定のアプリケーションでは、Quartus II ソフトウェアでの制約チェックを緩和することができます。例えば、トグルしないシングル・エンド・ピンがある場合、それを差動ピンの近くに安全に配置することができます。Quartus II ソフトウェアにこのような状態を知らせるには、Assignment Editor でそのピンに対して 0 Mhz Toggle Rate アサインメントを作成します。

もう 1 つの設定は **Output Enable Group** アサインメントで、これは特に外部メモリ・インタフェースで役立ちます。ロジック・オプションにより、出力イネーブル・グループ番号が割り当てられ、デザイン内のどのピンが同時にドライブ・インおよびドライブ・アウトするかを指定できます。このため、デザインは、電圧リファレンス形式の入力があるときに VREF グループからドライブ・アウトする最大ピン数の要件に違反しません。このオプションは、フィッタが VREF グループ内の出力イネーブル・グループのピンを検出できない場合 (例えば、出力イネーブルがステート・マシンまたは複雑なロジックから送られる場合) に使用します。このオプションは、DQ および DQS I/O ピンの出力イネーブル・グループ番号を指定するのに、また DM ピンが書き込み動作中のみ使用される場合に、専用 DDR SDRAM インタフェースで使用するのに便利です。個々のコントローラがそれぞれに、異なる出力イネーブル・グループ番号を持っている必要があります。

## 兼用ピンおよび特殊ピンの接続

Stratix III デバイスは、兼用コンフィギュレーション・ピンによる I/O の柔軟性を可能にします。兼用コンフィギュレーション・ピンは、デバイス・コンフィギュレーションの完了後は汎用 I/O として使用できます。Device and Pin Options ダイアログ・ボックスの **Dual-Purpose Pins** タブで、各兼用ピンに必要な設定を選択します。これらのピンは、コンフィギュレーション手法に応じて、通常の I/O ピン、トライ・ステート入力、グラウンドをドライブする出力、または規定されていない信号をドライブする出力として予約することができます。

また、グローバル・クロック・ネットワークにドライブする専用クロック入力ピンも、クロック・ピンとして使用されていない場合は、汎用入力ピンとして使用できます。クロック入力ピンには専用 I/O レジスタがないので、クロック入力を汎用入力として使用するとき、I/OレジスタはALMベースのレジスタを使用します。

デバイス・ワイドのリセットおよびクリア・ピンは、イネーブルされていない場合はデザイン I/O として使用できます。詳細については、24 ページの「デバイス・ワイド出力イネーブル・ピン」および 47 ページの「レジスタ・パワーアップ・レベルとコントロール信号」を参照してください。

Stratix III デバイスには CRC エラー検出機能用の専用回路が組み込まれており、オプションで継続的かつ自動的に SEU (Single Event Upset) の有無をチェックすることができます。SEU 緩和機能を活用するには、CRC エラー検出用の適切なメガファンクションを使用します。CRC\_ERROR ピンまたは CRITICAL\_ERROR ピンを使用してエラーを通知します。これらのピンは、CRC ファンクションに対してイネーブルされていない場合は、デザイン I/O として使用できます。



SEU の緩和について詳しくは、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスにおける SEU の緩和」の章を参照してください。

## Stratix III の I/O 機能

Stratix III の双方向 I/O エLEMENT (IOE) は、他のデバイスとのインタフェースを支援する多数の機能をサポートしており、PCB の複雑さとコストを低減します。表 2 に、Stratix III の I/O 機能、使用方法に関する情報とデザイン検討事項、およびこれらの機能に関する詳細情報の参照先を示します。

特長	使用方法	ガイドラインおよび詳細情報
MultiVolt I/O インタフェース	この機能により、すべてのパッケージが異なる電源電圧のシステムとインタフェースできます。V <sub>CCIO</sub> ピンは要求される出力のレベルに応じて、1.2 V、1.5 V、1.8 V、2.5 V、3.0 V または 3.3 V のいずれかの電源に接続することができます。出力レベルは電源と同じ電圧のシステムと互換性があります。V <sub>CCPD</sub> 電源ピンは、3.3 V V <sub>CCIO</sub> の場合は 3.3 V、3.0 V V <sub>CCIO</sub> の場合は 3.0 V、その他の I/O 電圧の場合は 2.5 V に接続しなければなりません。	MultiVolt I/O のサポートの概要、および 3.3 V/3.0 V LVTTTL/LVC MOS の追加ガイドラインについては、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスの I/O 機能」の章を参照してください。
専用 DPA サポート付き高速差動 I/O	速度が最大 1.25 Gbps の特定の差動 I/O インタコネクト規格用の専用回路です。ダイナミック・フェーズ・アライメント (DPA) により、ビット・エラーが最小になり、PCB レイアウトとタイミング管理が簡略化され、チャネル間およびチャネル・クロック間スキューが排除されます。	「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスの DPA を使用した高速差動 I/O インタフェース」の章を参照してください。DPA を使用する場合は、必ずこの機能をイネーブルして、デザインでデバイスのライト・サイドおよびレフト・サイドの正しい PLL が使用されるようにしてください。
プログラマブル・ドライブ強度	特定の I/O 規格に使用できるプログラマブル・ドライブ強度コントロール機能です。長い伝送線路またはレガシー・バックプレーンに起因する高い信号減衰の影響を緩和できます。ドライブ強度を高くすると、I/O 性能は向上しますが、インタフェースのノイズも増大するため、ドライブ強度コントロール機能を使用してノイズを管理できます。	出力バッファ・ドライブ強度が十分に高いが、I/O 規格の電圧スレッシュホールド・パラメータに違反する過剰なオーバーシュートやアンダーシュートは生じないことを確認してください。アルテラでは、特定のアプリケーションに対する正しいドライブ強度設定を決定するために、IBIS または SPICE シミュレーションを実行することを推奨しています。規格および設定の一覧は、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスの I/O 機能」の章を参照してください。

表 2. Stratix III の I/O 機能 (2 / 3)

特長	使用方法	ガイドラインおよび詳細情報
プログラマブル・スルー・レート・コントロール	低ノイズ性能または高速性能が得られるように、各ピンをコンフィギュレーションします。高速スルー・レートを指定した場合は、高速転送が行われます。より高速なスルー・レートを使用して、メモリ・インタフェース・アプリケーションで得られるタイミング・マージン、または出力ピンに高い容量性負荷がある場合のタイミング・マージンを改善することができます。低速スルー・レートを指定した場合、システム・ノイズの低減には役立ちますが、立ち上がりおよび立ち下がりエッジに一定の遅延が追加されます。スルー・レートを使用して、SSN を低減することができます。	低速スルー・レートを使用する場合は、インタフェースが性能要件を満たしていることを確認してください。アルテラでは、特定のアプリケーションに対する正しいスルー・レート設定を決定するために、IBIS または SPICE シミュレーションを実行することを推奨しています。
プログラマブル IOE 遅延	プログラマブル遅延チェーンは、デスキュー回路として使用できます。ピンから入力レジスタまでに異なる入力遅延値を使用してセットアップ時間を変更するか、出力レジスタから出力ピンまでの遅延値を使用してクロック出力時間を変更し、バスのすべてのビットが同じ遅延でデバイスに入出力されるようにします。	遅延の仕様については、「Stratix III デバイス・ハンドブック Volume 2」の「Stratix III デバイスの DC およびスイッチング特性」の章を参照してください。
プログラマブル出力バッファ遅延	シングル・エンド出力バッファの遅延チェーンは、出力バッファの立ち上がりエッジおよび立ち下がりエッジ遅延を独立して制御することができます。	遅延を使用して、出力バッファのデューティ・サイクルを調整し、チャンネル間スキューを補正し、意図的にチャンネル間スキューを導入することによって SSO ノイズを低減し、高速メモリ・インタフェースのタイミング・マージンを改善することができます。
オープン・ドレイン出力	オープン・ドレインとしてコンフィギュレーションした場合、出力のロジック値は high-Z または 0 のいずれかです。システム内の複数のデバイスでアサートできるシステム・レベルのコントロール信号で使用されます。	一般に、ロジック High を供給するには外部プルアップ抵抗が必要です。

表 2. Stratix III の I/O 機能 (3 / 3)

特長	使用方法	ガイドラインおよび詳細情報
バス・ホールド	約 7 k $\Omega$ の公称抵抗値 (RBH) を持つ抵抗を使用して、次の入力信号が現れるまで I/O ピンの信号を最後にドライブされた状態にウィーク状態で保持します。この機能により、バスがトライ・ステートになったとき、信号レベルを保持するための外部プルアップ抵抗またはプルダウン抵抗が不要になります。バス・ホールド回路は、ノイズによって高周波数スイッチングが予定外に発生しそうな場合、ドライブされていないピンを入力スレッシュホールド電圧から離します。	バス・ホールド機能がイネーブルにされている場合、プログラマブル・プルアップ・オプションは使用することができません。I/O ピンが差動信号用にコンフィギュレーションされているときは、バス・ホールド機能をディセーブルにしてください。各 V <sub>CCIO</sub> 電圧レベルのための、この抵抗を通してドライブされる特定の持続電流、および次にドライブされる入力レベルの識別に使用されるオーバードライブ電流については、「Stratix III デバイス・ハンドブック Volume 2」の「Stratix III デバイスの DC およびスイッチング特性」の章を参照してください。
プログラマブル・プルアップ抵抗	プルアップ抵抗 (通常 25 k $\Omega$ ) は、ユーザーモードのときに I/O を V <sub>CCIO</sub> レベルにウィーク状態で保持します。オープン・ドレイン出力と併用して、外部プルアップ抵抗を不要にすることができます。	プログラマブル・プルアップ・オプションがイネーブルされている場合、バス・ホールド機能は使用することができません。
PCI クランプ・ダイオード	PCI/PCI-X I/O 規格インタフェースでの過剰なオーバーシュート電圧からピンを保護するために使用できます。	—
On-Chip Termination (OCT)	ドライバ・インピーダンス・マッチングは、I/O ドライバに伝送ラインのインピーダンスと厳密にマッチングする制御された出力インピーダンスを提供して、反射を大幅に低減します。キャリブレーション付きまたはキャリブレーションなし直列 On-Chip Termination (チップ内終端) (RS)、キャリブレーション付き並列 On-Chip Termination (RT)、シングル・エンド I/O 規格用のダイナミック直列および並列終端、差動 LVDS I/O 規格用の差動 On-Chip Termination (RD) に対するサポートが提供されています。8 個または 10 個の OCT キャリブレーション・ブロックのいずれかを持つ Stratix III I/O バンクをキャリブレーションできます。	異なる I/O 規格が同じ V <sub>CCIO</sub> 電源電圧を使用する場合、それらの I/O 規格に対して、同じ I/O バンクで OCT RS および RT がサポートされます。I/O バンクの各 I/O は、OCT RS、プログラマブル・ドライブ強度、または OCT RT をサポートするために、独立してコンフィギュレーションすることができます。同じ I/O バッファに対して、OCT RS とプログラマブル・ドライブ強度の両方をコンフィギュレーションすることはできません。差動 On-Chip Termination (RD) はロウ I/O バンクでのみ使用でき、コラム I/O バンクは OCT RD をサポートしません。Stratix III デバイスのロウ I/O バンクの専用クロック入力ペア CLK1p、CLK1n、CLK3p、CLK3n、CLK8p、CLK8n、CLK10p、および CLK10n は、RD 終端をサポートしません。この機能のサポートおよび実装について詳しくは、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスの I/O 機能」の章を参照してください。

## クロックおよび PLL の選択

クロック方式のプランニングでの最初のステージは、システムのクロック要件を決定することです。デバイスの使用可能なクロック・リソースを理解し、それに応じてデザインのクロック方式をプランニングします。タイミング性能に関する要求、および特定のクロックによってドライブされるロジック量を考慮してください。

Stratix III デバイスは、低スキュー、高ファンアウトの専用配線ネットワークを提供します。これらのネットワークは、デバイス内で最大 220 個の固有のクロック・ドメインを提供し、デバイスの 1 つのエリアにつき最大 67 個の固有のクロック・ソースを許容する階層的なクロック構造で編成されています。デバイスごとに最大 12 個の PLL があり、PLL ごとに最大 10 個の個別にプログラム可能な出力があります。16 本の専用差動グローバル・クロック入力ピンまたは 32 のシングル・エンド・クロック入力を使用することができます。

デザインへのクロック信号の入力には、専用クロック・ピンおよび専用配線を使用することが重要です。専用クロック・ピンはクロック・ネットワークを直接ドライブし、他の I/O ピンと比較して、スキューが確実に低減されます。ファンアウトの大きい信号のスキューを低減しながら遅延を予測可能なものにするには、専用配線ネットワークを使用します。また、クロック・ピンとクロック・ネットワークを使用して、非同期リセットなどのコントロール信号をドライブすることも可能です。

特定のクロック入力は、特定の低スキュー配線ネットワークをドライブできる特定の PLL に接続します。各 PLL に対するグローバル・リソースの可用性と各クロック入力ピンに対する PLL の可用性を解析します。

以下の説明を使用して、デザインのクロック信号にどのクロック・ネットワークが適しているかを判断してください。

- グローバル・クロック (GCLK) ネットワークは、デバイス全体でドライブ可能であり、デバイス・ロジックの低スキュー・クロック・ソースとして働きます。このクロック領域は、他のクロック領域に比べて遅延が最大になりますが、デバイス内のすべてのディスティネーションに信号を配信することができます。このオプションは、グローバル・リセット / クリア信号の配線、またはデバイス全体のクロックの配線に適しています。
- リージョナル・クロック・ネットワーク (RCLK) はそれ自身がドライブするエリアにのみ属します。RCLK ネットワークは、デバイスの 1 つのエリア内に含まれるロジックに対して、最小のクロック遅延とスキューを実現します。
- I/O エレメント (IOE) と内部ロジックは、GCLK および RCLK をドライブして、内部で生成されるグローバルまたはリージョナル・クロック、および同期クリアまたは非同期クリアやクロック・イネーブルなど、その他の高ファンアウト・コントロール信号を生成することもできます。

- PLLは、内部で生成されたGCLKやRCLKからはドライブできません。PLLへの入力クロックは、専用のクロック入力ピン、ピンまたは PLL から供給される GCLK/RCLK のみを使用する必要があります。
- ペリフェラル・クロック (PCLK) ネットワークは、Stratix III デバイスの周辺からドライブされる個々のクロック・ネットワークの集合です。DPA ブロック、水平方向の I/O ピン、および内部ロジックからのクロック出力により、PCLK ネットワークをドライブできます。これらの PCLK は、GCLK および RCLK ネットワークと比べるとスキューが高く、汎用配線の代わりに使用して、Stratix III デバイスの内外に信号をドライブできます。



これらの機能について詳しくは、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスのクロック・ネットワークおよび PLL」の章を参照してください。

システムがターゲット・デバイスで使用可能な数を超えるクロック信号またはコントロール信号を必要とする場合は、専用クロック・リソースを節約することができるケースを検討します。特にクロック遅延およびクロック・スキューがデザインの性能に重大な影響を及ぼさない部分での、高ファンアウト信号および低周波数信号について検討してください。Quartus II Assignment Editor で **Global Signal** アサインメントを使用して、グローバル配線のタイプを選択するか、アサインメントを **Off** に設定して、信号がいかなるグローバル配線リソースも使用しないように指定します。

## PLL デザイン・ガイドライン

システム要件に基づき、FPGA デザインに必要なクロック周波数と、FPGA で使用可能な入力周波数を定義します。これらの仕様を使用して、PLL 方式を決定します。Quartus II MegaWizard Plug-In Manager を使用して、altpll メガファンクション用の設定を入力し、結果をチェックして特定の機能や入力 / 出力周波数を特定の PLL に実装できるかどうかを確認します。

### クロック・コントロール・ブロック

グローバルおよびリージョナル・クロック・ネットワークごとに、専用のクロック・コントロール・ブロックがあります。コントロール・ブロックは、以下の機能をサポートしています。

- クロック・ソースの選択 (グローバル・クロックに対してはダイナミック選択)
- グローバル・クロックのマルチプレキシング
- クロックのパワーダウン (スタティックまたはダイナミック・クロック・イネーブルまたはディセーブル付き)

デザインで組み合わせロジックを使用せずに、異なるクロック入力信号またはパワーダウン・クロック・ネットワークを選択して消費電力を低減する場合は、これらの機能を使用します。Stratix III デバイスでは、クロック・イネーブル信号は PLL 出力カウンタ・レベルではなく、クロック・ネットワーク・レベルでサポートされているので、PLL が使用されていないときでもクロックをオフにすることができます。



altclkctrl メガファンクションを使用したクロック・コントロール・ブロックの設定方法については、「altclkctrl Megafunction User Guide」を参照してください。

### PLL の特長

Stratix III PLL は、デバイス・クロック管理、外部システム・クロック管理、および高速 I/O インタフェースのための強力なクロック管理および合成機能を提供します。固有のジッタのフィルタリング、および通倍、分周、ダイナミック位相シフト・リコンフィギュレーションに対する精細な制御により、今日の高速アプリケーションに要求される高い精度を提供します。Stratix III デバイスの PLL は豊富な機能を備えており、クロック・スイッチオーバー、ダイナミックな位相シフト、PLL リコンフィギュレーション、およびリコンフィギュレーション可能な帯域幅など、先進的な機能をサポートしています。Stratix III の PLL は、外部フィードバック・モード、スペクトラム拡散トラッキング、およびポスト・スケール・カウンタのカスケード接続もサポートしています。Stratix III のすべての PLL は、サポートされる機能が多少異なるだけの同じコア・アナログ構造を持っています。PLL デザインをプランニングするときは、以下の追加機能のいくつかを使用することができます。



PLL の設計、および altpll メガファンクションを使用して、この項で説明した機能を活用する方法については、altpll Megafunction User Guide を参照してください。

#### クロック・フィードバック・モード

Stratix III の PLL は、最大でソース同期モード、非補償モード、ノーマル・モード、ゼロ遅延バッファ (ZDB) モード、外部フィードバック・モード、および LVDS 補正の 6 種類のクロック・フィードバック・モードをサポートします。各モードは異なるクロック・ネットワークおよび遅延を補償しているため、クロックは異なる方法で揃えられます。アプリケーションに合わせて正しいフィードバック・モードを選択してください。

#### クロック出力

クロック出力を専用クロック出力ピンまたは専用クロック・ネットワークに接続することができます。デバイスのトップおよびボトム of the PLL は、レフトおよびライト PLL よりも多くの専用クロック出力を提供します。

### クロック・スイッチオーバー

クロック・スイッチオーバー機能により、PLL は2つの基準入力クロックを切り換えることができます。この機能はクロック冗長性の目的で、あるいは前のクロックが停止した場合に冗長クロックがオンになるシステムのようなデュアル・クロック・ドメイン・アプリケーションに使用します。クロックがトグルしていないとき、またはユーザ・コントロール信号 (clkswitch) をベースにしている場合、デザインはクロック・スイッチオーバーを自動的に実行できます。

### ダイナミック・リコンフィギュレーション

PLL 設定をリコンフィギュレーションして、Stratix III デバイス全体をリコンフィギュレーションしないで、出力クロック周波数と PLL 帯域幅を更新し、リアルタイムで位相シフトできます。PLL をリアルタイムでリコンフィギュレーションする機能は、複数の周波数で動作するアプリケーションで有用です。またプロトタイプ環境でも、PLL の出力周波数をスイープし、出力クロック位相をダイナミックに調整できるため有益です。また、この機能を使用すると、PLL 出力クロック位相シフトを変更して、クロック - 出力 ( $t_{CO}$ ) 遅延を調整することも可能です。この手法により、新しい PLL 設定でコンフィギュレーション・ファイルを再生成する必要がなくなります。

Altpll メガファンクションで、ダイナミック・リコンフィギュレーション機能を有効にします。次に、altpll\_reconfig メガファンクションをインスタンス化することで、この機能をより使いやすくすることができます。



altpll\_reconfig MegaWizard Plug-In Manager について詳しくは、「altpll\_reconfig Megafunction Users Guide」を参照してください。

### PLL コントロール信号

PLL の演算および再同期を観察および制御する場合は、以下の3つの信号を使用できます。

- **areset**—各PLLに対するリセットまたは再同期化入力です。PLL入力クロックと出力クロック間の正しい位相関係を保証するために、PLLがロックを失うたびにareset信号をアサートする必要があります。ロック損失状態で自動的にリセットする(セルフ・リセット)ようにPLLを設定できます。PLLへの入力クロックがトグルしていなかったり、または電源投入時に不安定な場合、入力クロックが安定し、仕様範囲内に収まった後、areset信号をアサートします。ダイナミック・リコンフィギュレーションまたはクロック・スイッチオーバー機能をイネーブルする場合は、このコントロール・ピンをイネーブルする必要があります。
- **locked**—各PLLのこの出力ピンは、PLLが基準クロックにロックし、PLLクロック出力が目的の位相および周波数で動作していることを示します。デバイスのコア・ロジックでこの信号をモニタできます。

- `pfdena`—この信号は最後にロックされた周波数を維持するために使用しますが、多少の長期的ドリフトが発生して周波数は低くなります。入力クロックがディセーブルされている場合、システムで PLL からの特定のクロック周波数が必要な場合は、システムがシャット・ダウンする前に現在の設定を保存する時間が確保されるよう、このピンをイネーブルする必要があります。

## 同時スイッチング・ノイズ

SSN は、同時に電圧レベルを変化させるピン（近接して）が多すぎると問題になります。可能性のある同時スイッチング・ノイズの問題についてデザインを解析し、以下の推奨事項を検討してください。

- 可能であれば、同時に電圧レベルを切り換えるピン数を減らします。
- スwitching速度が高いI/Oには低いドライブ強度を使用します。デフォルトのドライブ強度設定は、デザインで要求されるドライブ強度設定よりも高い場合があります。36 ページの「Stratix III の I/O 機能」を参照してください。
- スwitching速度が高いI/Oには、差動I/O規格と低電圧規格を使用します。
- 各バンク内で同時にスitchingする出力数を減らします。可能であれば、出力ピンを複数のバンクに分散させます。
- バンク使用率が100%を十分に下回っている場合は、スitchingI/Oをバンク全体に均等に分散させて領域内のアグレッサの数を減らして、SSN を低減します。
- 同時にスitchingするピンを、SSN に敏感な入力ピンから分離します。
- 重要なクロック・コントロール信号および非同期コントロール信号は、大きなスitching・バスから離して、グランド信号の近くに配置します。
- PLL電源ピンから1本または2本分離れたI/Oピンを、スitching速度が速い、またはドライブ強度が高いピンに使用しないようにします。

スタグガード出力遅延を使用して、出力信号を経時的にシフトするか、調整可能なスルー・レート設定を使用します。36 ページの「Stratix III の I/O 機能」を参照してください。

## デザイン および コンパイル

デザインのソース・コードを作成し、デバイスの選択およびタイミング要求を含む制約を適用した後、合成ツールでコードを処理し、それをデバイス・アーキテクチャのエレメントにマップします。次に、Quartus II Fitter は、配置配線を実行して、特定のデバイス・リソース内にデザイン・エレメントを実装します。この項では、デザインおよびコンパイル・フローのこれらのステージに関するガイドラインを提供します。

## 合成ツールの選択

Quartus II ソフトウェアは、アルテラ・ハードウェア記述言語（AHDL）や回路図によるデザイン入力だけでなく Verilog HDL および VHDL も完全にサポートする、高度な使いやすい合成機能を備えています。また、業界をリードするサードパーティ EDA 合成ツールを使用して、Verilog または VHDL デザインを合成し、次に結果として得られる出力ネットリスト・ファイルを Quartus II ソフトウェアでコンパイルすることもできます。New Project Wizard または Settings ダイアログ・ボックスの **EDA Tools Settings** ページで任意のサードパーティ合成ツールを指定して、合成ネットリストに適した Library Mapping ファイルを使用します。

合成ツールが異なると、得られる結果が異なる可能性があります。アプリケーションに対して最高の性能を発揮するツールを選択したい場合は、アプリケーションおよびコーディング・スタイルに対応した標準的なデザインを合成し、結果を比較することによってツールを実験することができます。正確なタイミング解析およびロジック利用率の結果を得るために、必ず Quartus II ソフトウェアで配置配線を実行してください。

ツール・ベンダは絶えず新機能を追加し、ツールの問題を修正し、アルテラ・デバイスに対する性能を向上させているため、最新バージョンのサードパーティ合成ツールを使用するようにしてください。

合成ツールによっては、Quartus II プロジェクトを作成し、EDA ツールの設定、デバイスの選択、および合成プロジェクトで指定したタイミング要求などの制約を渡す機能を提供するものもあります。配置配線のために Quartus II プロジェクトを設定するときに、この機能を使用して時間を節約することができます。



サポートされている合成ツールについて詳しくは、「Quartus II ハンドブック Volume 1」の「合成」セクションの該当する章を参照してください。Quartus II リリース・ノートには、Quartus II ソフトウェアの当該バージョンで正式にサポートされている各合成ツールのバージョンが記載されています。

## デザインの推奨事項および HDL コーディング・スタイル

複雑な FPGA デザインの開発では、適切なデザイン手法およびコーディング・スタイルが、デバイスのタイミング性能、ロジック利用率、およびシステムの信頼性にきわめて大きな影響を与えます。最良の合成結果およびフィッティング結果が得られるよう、アルテラの推奨事項を順守してください。

## デザインの推奨事項

一貫したデザイン目標の達成のために、同期デザイン方式を使用します。その他のデザイン手法に関する問題としては、デバイスにおける伝播遅延への依存、不完全なタイミング解析、およびグリッチが発生する可能性があります。同期デザインでは、クロック信号がすべてのイベントをトリガします。同期デザインは、すべてのレジスタのタイミング要求が満たされている限り、すべてのプロセス、電圧、および温度（PVT）条件に対して、予測可能かつ信頼性を確保した形で動作します。異なるデバイス・ファミリまたはスピード・グレードを容易に同期デザインのターゲットにすることができます。

クロック信号は、デザインのタイミング精度、性能、および信頼性に大きな影響を及ぼすので、特に注意してください。クロック信号に関連する問題は、デザインにおいて機能上の問題およびタイミング問題を引き起こす可能性があります。最良の結果を得るために、専用クロック・ピンと専用クロック配線を使用します。クロックの反転、通倍、および分周には、デバイスの PLL を使用します。クロックの多重化およびゲーティングには、組み合わせロジックの代わりに専用のクロック・コントロール・ブロック、または PLL クロック・スイッチオーバー機能を使用します。40 ページの「PLL デザイン・ガイドライン」を参照してください。内部生成クロック信号を使用する必要がある場合は、グリッチを低減するためにコントロール信号として使用される組み合わせロジックの出力をラッチします。例えば、組み合わせロジックを使用してクロックを分周する場合、デバイダ回路をクロックするのに使用したクロック信号で最終ステージをクロックします。

Quartus II ソフトウェアのデザイン・アシスタントは、デザイン・フローの早期におけるデザイン問題のチェックを可能にするデザイン・ルール・チェック・ツールです。デザイン・アシスタントは、デザインがアルテラ推奨のデザイン・ガイドラインまたはデザイン・ルールを順守しているかどうかをチェックします。デザイン・アシスタントを実行するには、Processing メニューで Start をポイントして、**Start Design Assistant** をクリックします。デザイン・アシスタントがコンパイル時に自動的に実行されるように設定するには、Settings ダイアログ・ボックスで **Run Design Assistant during compilation** をオンにします。また、サードパーティの「リント」ツールを使用してコーディング・スタイルをチェックすることもできます。



デザイン推奨事項およびデザイン・アシスタントの使用について詳しくは、「Quartus II ハンドブック Volume 1」の「Design Guidelines for Altera Devices and the Quartus II Design Assistant」の章を参照してください。業界紙を参照して、複数のクロック・デザインに関する詳しい情報を入手することもできます。適切な解析を行うには、以下を参照してください。

[www.sunburst-design.com/papers/CummingsSNUG2001SJ\\_AsyncClk.pdf](http://www.sunburst-design.com/papers/CummingsSNUG2001SJ_AsyncClk.pdf)

## メガファンクションの使用

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節約することができます。さらに、アルテラが提供するメガファンクションは、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。メガファンクションのサイズを拡大 / 縮小し、パラメータでさまざまなオプションを設定することができます。メガファンクションには、LPM (Library of Parameterized Modules) メガファンクションおよびアルテラ・デバイス特有のメガファンクションが含まれます。13 ページの「IP の選択」で説明したように、アルテラおよびサードパーティの IP およびリファレンス・デザインを活用して、デザイン時間を節約することもできます。

Quartus II MegaWizard Plug-In Manager は、メガファンクションをカスタマイズするための簡単なユーザ・インタフェースを備えています。すべてのポートとパラメータを正しく設定するために、ウィザードを使用してメガファンクションのパラメータを構築または変更する必要があります。



メガファンクションについて詳しくは、Quartus II Help またはアルテラ・ウェブサイトの「[オンライン資料 : ユーザガイド](#)」ページのメガファンクション・ユーザガイドを参照してください。

## 推奨 HDL コーディング・スタイル

HDL コーディング・スタイルは、プログラマブル・ロジック・デザインの結果の品質 (QoR) に大きな影響を与える可能性があります。合成ツールは、ロジック利用率と性能の両方について HDL コードを最適化し、設定とアサインメントで制御することができます。ただし、合成ツールはデザインの目的や意図に関する情報は持っていないので、最良のデザイン最適化を行うには人間の判断が必要になる場合もあります。最適な合成結果が得られるように、アルテラの推奨コーディング・スタイルを使用してください。



メガファンクションを推論し、メモリやデジタル信号処理 (DSP) ブロックなどの専用デバイス・ハードウェアをターゲットとする場合のコーディング・ガイドラインを含む、特定の HDL コーディング例と推奨事項については、「Quartus II ハンドブック Volume 1」の「推奨 HDL コーディング・スタイル」の章を参照してください。その他のツール特有のガイドラインについては、合成ツールのドキュメントを参照してください。

## レジスタ・パワーアップ・レベルとコントロール信号

Stratix III デバイスはオプションのチップ・ワイドのリセットをサポートしており、このリセットによってメモリ・ブロックのレジスタを含む（ただし、メモリ内容そのものは含まない）、デバイスのすべてのレジスタのクリアをすべて無効にすることができます。この DEV\_CLRn ピンが Low にドライブされると、すべてのレジスタがクリアされ 0 にリセットされます。以下のパラグラフでは、合成により NOT-gate push-back と呼ばれる最適化が実行され、その場合に影響を受けたレジスタが DEV\_CLRn が Low にドライブされるときに 1 にプリセットされているかのように振舞うときの状況について説明します。DEV\_CLRn ピンが High にドライブされると、すべてのレジスタはプログラムされたとおりに動作します。このチップ・ワイドのリセットを使用するには、デザインをコンパイルする前に、**Device & Pin Options** ダイアログ・ボックスの **General** タブで Quartus II ソフトウェアの **Enable device-wide reset (DEV\_CLRn)** をオンにします。

Stratix III の各ロジック・アレイ・ブロック (LAB) には、各 ALM に対するレジスタ・コントロール信号をドライブするための専用ロジックも内蔵されています。このコントロール信号には、3 本のクロック、3 本のクロック・イネーブル、2 本の非同期クリア、同期クリア、および同期ロードが含まれます。LAB 内で信号が共有されるため、レジスタ・コントロール信号によりレジスタを LAB 内に実装する方法が制限されます。コントロール信号がデバイス・アーキテクチャ内の専用コントロール信号を使用することが重要です。したがって、場合によっては、デザインで使用する異なるコントロール信号の数を制限する必要があります。



LAB および ALM アーキテクチャについて詳しくは、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスのロジック・アレイ・ブロックおよびアダプティブ・ロジック・モジュール」の章を参照してください。

リセットがアサートされたときにコントロール信号を使用できない可能性がある場合は、通常非同期リセットを使用してロジックをリセットします。推奨されるリセット・アーキテクチャでは、リセット信号を非同期でアサートし、かつ同期してデアサートすることができます。リセット信号のソースは、グローバル配線リソースに直接接続できるレジスタの非同期ポートに接続されます。同期デアサーションにより、すべてのステート・マシンとレジスタを同時に起動することができます。同期デアサーションは、フリップ・フロップのアクティブ・クロック・エッジまたはその付近で非同期リセット信号がリリースされると、フリップ・フロップの出力が準安定の不定状態になる可能性も回避します。



業界紙を参照して、リセット・デザインに関する詳しい情報を入手することができます。リセット・アーキテクチャの適切な解析を行うには、以下を参照してください。

[www.sunburst-design.com/papers/CummingsSNUG2003Boston\\_Resets.pdf](http://www.sunburst-design.com/papers/CummingsSNUG2003Boston_Resets.pdf)

Quartus II の合成は、デフォルトにより **Power-Up Don't Care** と呼ばれるロジック・オプションをイネーブルします。このオプションでは、デザインがデバイス・アーキテクチャのパワーアップ状態に依存しないとみなして、ソフトウェアが **High** にスタックしたレジスタを削除することができます。他の合成ツールが同様の仮定を使用することもあります。

設計者は通常デザインには、必ずしもパワーアップ時ではなく、リセット後にすべてのレジスタを適切な値に強制する明示的なリセット信号を使用します。非同期リセットによってボードが安全な状態で動作できるようデザインを作成できます。その後で、リセットをアクティブにしてデザインを立ち上げることができます。こうすれば、デバイスのパワーアップ状態に依存する必要はありません。

デザインに特定のパワーアップ条件を強制したい場合は、合成ツールで使用可能な合成オプションを使用します。Quartus II の合成では、**Assignment Editor** で、またはソース・コードの `altera_attribute` アサインメントにより、**Power-Up Level** ロジック・オプションを適用することができます。

合成ツールによっては、ソース・コードでラッチされた信号のデフォルト値または初期値を読み出し、この動作をデバイスに実装することも可能です。例えば、Quartus II 合成機能は、レジスタにラッチされた信号の HDL デフォルト値および初期値を **Power-Up Level** 設定に変換します。このようにすれば、合成された動作は、機能シミュレーション中に HDL コードのパワーアップ状態に適合します。



**Power-Up Level** オプションおよび `altera_attribute` アサインメントについては、「Quartus II ハンドブック Volume 1」の「Quartus II Integrated Synthesis」の章で説明されています。

デバイス・コア内のレジスタは、物理的なデバイス・アーキテクチャにおいて常に **Low** (0) ロジック・レベルでパワーアップします。**High** のパワーアップ・レベルすなわち 0 以外のリセット値（プリセット信号と呼ばれることが多い）を指定した場合、合成ツールは通常、レジスタで使用可能なクリア信号を使用し、**NOT-gate push-back** と呼ぶ最適化を実行します。**NOT-gate push back** は、レジスタの入力および出力にインバータを追加します。レジスタのハードウェアは実際にパワーアップして **Low** にリセットされますが、レジスタ出力は反転するので、すべてのディスティネーションで得られる結果は **High** のロジック値になります。合成で **NOT-gate push back** 最適化が実行された場合、レジスタはリセット状態またはパワーアップ状態での **High** (1) ロジック・レベルのように動作します。通常のデータ・パスでは信号が 2 回反転するので、通常のレジスタ動作が影響を受けることはありません。この最適化はデザインのフィッティングまたは性能に悪影響を与えませんが、オンチップ検証中にレジスタからタップを取り出す場合、またはシミュレーション中にそれを表示する場合は、正しい値が得られるよう、出力の反転後に信号をチェックする必要があることに留意してください。

**Low** にリセットされたレジスタに対して **High** のパワーアップ・レベルを割り当てた場合、または **High** にプリセットされたレジスタに対して低いパワーアップ・レベルを割り当てた場合、合成ツールは **NOT-gate push back** 最適化手法を使用することができず、パワーアップ条件を無視する可能性があります。

リセット信号とプリセット信号の両方を同じレジスタに適用してはなりません。このタイプのロジックを実装するために、合成ツールは、レジスタへの異なるパス間で遅延が異なるためグリッチを発生しやすいロジックとラッチでコントロールをエミュレートします。また、これらのレジスタに対するパワーアップ値は未定義です。

## 階層およびチーム・ベース・デザインのプランニング

Quartus II のインクリメンタル・コンパイル機能は、デザインのロジックを変更した場合、変更されていないロジックの結果と性能を維持するため、ユーザはより多くのデザインの繰り返しを実行し、より効率的にタイミング・クロージャを達成することができます。インクリメンタル・コンパイル・フローでは、システム開発者は大規模なデザインを個別に設計可能な小さなパーティションに分割します。チーム・デザイン環境では、チーム・メンバが個々にパーティションで作業を行うことができるため、デザイン・プロセスが簡素化され、コンパイル時間が短縮されます。これらの利点は大規模な Stratix III デザインにとって重要です。

Quartus II インクリメンタル・コンパイルでのコンパイル時間の節約と性能の維持を活用する場合は、デザイン・サイクルの最初からインクリメンタル・コンパイル・フローをプランします。適切な分割とフロアプラン・デザインは、低いレベルのデザイン・ブロックがトップレベルのデザイン要件を満たすのに役立ち、トップレベル・デザインの統合とタイミングの検証に費やされる時間を短縮します。



Quartus II ソフトウェアでのインクリメンタル・コンパイル・フローの使用の詳細、およびデザイン・パーティションとデザイン・フロアプランの作成のための重要なガイドラインについては、「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。

### デザイン・パーティションのプランニング

FPGA のデザインを分割するには、パーティションが統合されたときに最良の結果が得られるよう、また各パーティションがデバイス内の他のパーティションに対して適切に配置されるようにプランする必要があります。

デザイン・パーティションを作成して全体的な結果の品質を改善するには、アルテラの推奨事項を順守してください。例えば、パーティションの I/O 境界を登録することにより、個別に最適化可能な 1 つのパーティション内のクリティカル・タイミング・パスを維持します。デザイン・パーティションが指定された場合は、インクリメンタル・コンパイル・アドバイザーを使用して、パーティションがアルテラの推奨事項を確実に満たすようにします。

サードパーティ合成ツールを使用する場合は、合成ツールでデザイン・パーティションごとに個別の VQM または EDIF ネットリストを作成します。必要に応じて、合成ツール内で個別にプロジェクトを作成して、ツールが各パーティションを個々に合成し、個別の出力ネットリスト・ファイルを生成するようにします。Quartus II インクリメンタル・コンパイルのサポートについては、合成ツールのドキュメントを参照してください。

#### タイミング・バジェット

設計者が個々のブロックを開発する前にタイミング・バジェットを決定することにより、特にチーム・ベースのデザインにおいて、システム統合時にタイミング問題が発生する可能性が低減されます。下位レベルのパーティションを個別に最適化される場合は、パーティション間にまたがる未登録のパスは完全なパスとして最適化されません。ソフトウェアが各パーティションの入力および出力ロジックを正しく最適化するように、手動で何らかのタイミング・バジェットを実行する必要があります。パーティションをまたぐそれぞれの未登録のタイミング・パスについて、各パーティションの対応する I/O パスでタイミング・アサインメントを行い、パスの両端を予定タイミング遅延に制限します。ソフトウェアは、パスを適切に最適化して、接続の各部分に対するタイミング・バジェットを割り当てたときに、パスがトップレベル・デザイン要求に適合するようにします。

#### リソース・バランシング

リソース利用率をプランニングし、バランスを取ることが重要になる場合がよくあります。インクリメンタル・コンパイルを実行する場合、各パーティションは別々に合成され、他のパーティションで使用されるリソースに関するデータは含まれません。したがって、合成により個々のパーティションでデバイス・リソースが過剰に使用される可能性があり、パーティションがマージされたときにデザインがターゲット・デバイスに適合しなくなる場合があります。パーティションがトップレベルで結合されるときには、競合するリソースが使用される場合もあります。デバイス・リソースには、ロジック、メモリ、乗算器だけでなく、PLL やクロックなどのグローバル配線信号も含まれます。デザイン・パーティション間でリソース利用率のバランスを取り、それを割り当てることによって、すべてのパーティションが統合されるときにリソース競合の問題を回避することができます。

#### ボトムアップおよびチーム・ベース・フローにおけるプランニング

ボトムアップ・デザイン・フローでは、システム開発者が下位レベルのブロックの設計者に対して、各パーティションが適切なデバイス・リソースを使用するように指導することが重要です。デザインが個別に開発されるため、各下位レベルの設計者は、全体的なデザインまたは各自のパーティションが他のパーティションとどのように接続されるのかについて情報を持っていません。このような情報の欠如は、システム統合時に問題を引き起こす可能性があります。下位レベル・パーティションの設計者がデザインに着手する前に、ピン位置、物理的制約、およびタイミング要求を含むトップレベル・プロジェクト情報を設計者に伝達する必要があります。

システム開発者は、トップレベルでデザイン・パーティションをプランニングし、Quartus II インクリメンタル・コンパイルを使用して、自動的に生成されたスクリプトを介して下位レベルの設計者に情報を伝達することができます。Quartus II ソフトウェアの **Generate bottom-up design partition scripts** オプションは、トップレベル・プロジェクト情報の下位レベル・モジュールへの転送プロセスを自動化します。このソフトウェアには、トップレベル・デザインでプロジェクト情報を管理するための、プロジェクト・マネージャ・インタフェースが備わっています。

### デザイン・フロアプランの作成

インクリメンタル・コンパイルをフルに活用するために、デザイン・フロアプランを作成してデザイン・パーティション間の競合を回避し、各パーティションが他のパーティションに対して適切に配置されるようにすることができます。各パーティションに対して異なるロケーション・アサインメントを作成した場合、ロケーションの競合は発生しません。また、デザイン・フロアプランは、ほとんどのリソースが要求済みであるデバイス・エリアで、フィットにデザインの一部を配置または再配置するよう指示する状況を回避するのに役立ちます。フロアプラン・アサインメントがない場合、この状況はコンパイル時間の増大や結果の品質低下につながることがあります。

Quartus II Chip Planner を使って、各デザイン・パーティションに対する LogicLock 領域アサインメントを使用したデザイン・フロアプランを作成することができます。トップレベル・デザインの基本的なデザインの枠組みがあれば、領域間の接続を表示し、チップ上のフィジカル・タイミング遅延を見積もり、デザイン・フロアプランで領域を移動させることができます。完全なデザインをコンパイル済みの場合は、ロジック配置を表示し、配線が密集する領域を特定してフロアプラン・アサインメントを改善することもできます。



デザイン・フロアプランでの配置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。

## SOPC Builder

SOPC Builder は、プロセッサ、ペリフェラル、およびメモリで構成されるシステムを作成するための強力なシステム開発ツールです。SOPC Builder により、従来の手動による統合手法を使用する場合よりもはるかに短い時間で、完全な System-On-a-Programmable-Chip (SOPC) を定義および生成できます。SOPC Builder を使用し、GUI でシステム・コンポーネントを指定すると、SOPC Builder は自動的にインタコネクタ・ロジックを生成します。SOPC Builder は、システムのすべてのコンポーネントを定義する HDL ファイルと、すべてのコンポーネントをまとめて接続するトップレベルの HDL デザイン・ファイルを出力します。

SOPC Builder は、一般に Nios® II プロセッサをベースとするシステムを作成するためのツールとして使用されます。ただし、SOPC Builder は、プロセッサの有無に関係なく任意の SOPC デザインを作成するための汎用ツールです。SOPC Builder コンポーネントは、コンポーネントの物理的接続に Avalon インタフェースを使用しており、ユーザは SOPC Builder を使用して Avalon インタフェースを備えた任意の論理デバイス（オンチップまたはオフチップ）を接続できます。Avalon Memory-Mapped インタフェースではアドレス・マップド・リード/ライト・プロトコルが使用されており、これによりマスタ・コンポーネントを接続するための柔軟なトポロジーで任意のスレーブ・コンポーネントを読み書きできます。Avalon Streaming インタフェースは、ソース・ポートとシンク・ポートを使ってデータを送受信するストリーミング・コンポーネント間のポイント・ツー・ポイント接続を可能にする高速、双方向システム・インタコネクトです。

SOPC Builder は、ハードウェア生成ツールとしての役割に加え、システム・シミュレーションおよびエンベデッド・ソフトウェア作成の出発点としての役割も果たします。SOPC Builder は、ソフトウェアを簡単に記述できるようにする機能、およびシステム・シミュレーションを高速化する機能を提供します。



生産性の向上のための SOPC Builder の使用については、「Quartus II ハンドブック Volume 4」の「SOPC Builder」の章を参照してください。

## タイミング・クロージャおよび検証

完成した論理デザインをコンパイルした後、デバイス利用率をチェックし、デザインがタイミング要求を満足していることを検証します。コンパイル中に生成されるメッセージを解析して、潜在的な問題がないかチェックします。必要に応じて、Quartus II ソフトウェアを使用して、デザインのリソース利用率を最適化してタイミング・クロージャを達成し、変更されていないデザイン・ブロックの性能を維持し、コンパイル時間を短縮して将来の繰返しに備えることができます。また、シミュレーションまたはフォーマル検証によってデザインの機能性を検証することもできます。この項では、コンパイル・フローのこれらのステージに関するガイドラインを提供します。

### デバイスのリソース利用率レポート

コンパイル後、デバイスのリソース利用率情報に目を通して、将来ロジックを追加したり、その他のデザインの変更を行うことにより、フィッティングに支障を来す恐れがあるか否かを判断します。コンパイルの結果、no-fit エラーが生じた場合、リソース利用率情報は、デザインのフィッティングの問題の解析を可能にするために重要です。

リソース使用量を求める場合は、コンパイル・レポートの **Flow Summary** セクションを参照してください。この項では、組み合わせおよびメモリ・アダプティブ・ルックアップ・テーブル (ALUT) 数およびデバイス・レジスタ数について概説します。これらの数は、リソース利用率を比較または見積る場合に、デザインのサイズを適切に表します。さらに、既存の接続またはロジックの使用により使用不可能なリソースの見積りを含めた総ロジック利用率も示されています。

Stratix III デバイスに関しては、ロジック利用率が低いデバイスの ALM 利用率が考えられる最低の ALM 利用率であるわけではありません。また、ほぼ 100% とレポートされたデザインでも、スペースが残されており、追加のロジックを収容できる場合もあります。フィッタは、ロジックを 1 個の ALM 内に配置できる場合でも、異なる ALM 内の ALUT を使用し、それにより最良のタイミングおよび配線性の結果を達成することができます。これらの結果が達成される場合、ロジックがデバイス全体に分散されている場合があります。デバイスの使用率が高くなると、フィッタは 1 個の ALM 内に一緒に配置できるロジックを自動的にサーチします。

Compilation レポートの **Fitter** セクションの **Resource Section** の下にあるレポートを表示すると、詳細なリソース情報を入手できます。**Fitter Resource Usage Summary** レポートは、ロジック利用率情報を分析し、完全に使用されている ALM および部分的に使用されている ALM の数を表示し、各タイプのメモリ・ブロック内のビット数を含むその他のリソース情報を提供します。また、コンパイル中に実行された一部の最適化について説明するレポートもあります。例えば、Quartus II の統合合成機能を使用している場合、**Analysis & Synthesis** セクションの **Optimization Results** フォルダの下にあるレポートは、合成中に削除されたレジスタを含む情報について説明します。このレポートは、部分的に完成したデザインについてデバイスのリソース利用率を見積る際に、デザインの他の部分との接続の欠落が原因でレジスタが削除されていないことを確認するのに役立ちます。

## Quartus II メッセージ

コンパイル・フローの各ステージで、情報メモ、ワーニング、クリティカル・ワーニングを含むメッセージが生成されます。これらのメッセージに目を通して、デザインの問題がないかチェックします。あらゆるワーニング・メッセージの意味を確実に理解し、必要に応じてデザインまたは設定を変更してください。Quartus II のユーザ・インタフェースでは、メッセージ・ウィンドウのタブを使用して特定のタイプのメッセージのみを見ることができ、ユーザは、あるメッセージがユーザからのアクションを必要としないと判断した場合、そのメッセージを抑制することができます。



メッセージおよびメッセージの抑制について詳しくは、「Quartus II ハンドブック Volume 2」の「Managing Quartus II Projects」の章を参照してください。

## タイミング制約およびタイミング解析

FPGA デザイン・フローでは、正確なタイミング制約により、タイミング・ドリブン・シンスシス・ソフトウェアと配置配線ソフトウェアは最適な結果を得ることができます。タイミング制約は、デザインが確実にタイミング要求、すなわちデバイスが正しく動作するために満足されなければならない実際のデザイン要件を満足するために重要です。最終的なプログラム済みのデバイスは、タイミング・パスが十分に解析および検証されて要件が満足されない限り、予測と異なる動作をする可能性があります。

Quartus II ソフトウェアには、デザインのすべてのロジックのタイミング性能を検証する強力な ASIC スタイルのタイミング解析ツールである TimeQuest タイミング・アナライザが含まれています。TimeQuest タイミング・アナライザは、業界標準の Synopsys Design Constraints (SDC) フォーマットのタイミング制約をサポートし、インタラクティブなタイミング・レポート付きの使いやすい GUI を備えています。TimeQuest タイミング・アナライザは、高速ソース・シンクロナス・インタフェースおよびクロック多重化デザイン構造を制約するのに理想的です。(Quartus II ソフトウェアには、レガシー・デザイン用に、異なるデザイン制約とレポートを使用するクラシック・タイミング・アナライザも含まれています。Stratix III デザインには、TimeQuest タイミング・アナライザを使用します。)

このソフトウェアは、業界標準の Synopsys 社の PrimeTime ソフトウェアでのスタティック・タイミング解析もサポートします。New Project Wizard または **Settings** ダイアログ・ボックスの **EDA Tools Settings** ページでツールを指定して、必要なタイミングネットリストを生成します。

包括的なスタティック・タイミング解析には、レジスタ間パス、I/O パス、および非同期リセット・パスの解析が含まれます。デザイン内のすべてのクロックの周波数と関係を指定することが重要です。外部デバイスまたは外部ボードのタイミング・パラメータを指定するには、入力および出力遅延制約を使用します。システムの意図が正確に反映されるように、外部インタフェース用コンポーネントに対する正確なタイミング要求を指定します。タイミング・アナライザは、システム全体に対してスタティック・タイミング解析を実行し、データ要求時間、データ到達時間、およびクロック到達時間を使用して、回路性能を検証し、発生する可能性があるタイミング違反を検出します。タイミング・アナライザは、デザインが正しく機能するために満足する必要があるタイミング関係を決定します。

Stratix III デバイスにデータが供給されるときに、入力 I/O 時間に違反しないようにします。report\_datasheet コマンドを使用して、デザイン全体の I/O タイミング特性を要約したデータシート・レポートを生成することができます。



タイミング解析について詳しくは、「Quartus II ハンドブック Volume 3」の「TimeQuest Timing Analyzer」または「Synopsys PrimeTime Support」の章を参照してください。

### 推奨されるタイミング最適化およびタイミング解析のアサインメント

この項で説明するアサインメントと設定は、どのデザインでもデフォルトではソフトウェアでオンになりませんが、Stratix III デバイスのデザインのように大規模なデザインにとって重要です。

**Settings** ダイアログ・ボックスの **Fitter Settings** ページで、**Optimize fast-corner timing** をオンにします。このオプションがオンのとき、デザインは Fast Timing プロセス・コーナおよび動作条件、そして Slow Timing コーナにおけるタイミング要求を満足するように最適化されます。したがって、このオプションをオンにすると、プロセス、温度、および電圧のばらつきに対してより堅牢なデザイン実装を作成するのに役立ちます。

デザインでホールド・タイムを満足できない問題がある場合、ゲート制御クロックがクロック・ラインに遅延を追加しているなど、デザイン上の問題が生じている可能性があります。そうでない場合は、**Optimize hold timing** オプションを **All paths** に設定して、レジスタ間のホールド・タイム・パスを最適化します。

**Settings** ダイアログ・ボックスの **Timing Analysis Settings** の下にある **TimeQuest Timing Analyzer** ページで **Enable multicorner timing analysis** をオンにするか、**TimeQuest** に対して `--multicorner` コマンド・ライン・オプションを使用します。このオプションは、**TimeQuest** タイミング・アナライザにデザインを解析し、低速および高速コーナについてのスラック・レポートを生成するよう指示します。

**TimeQuest SDC** 制約ファイルでは、デザインに適用可能であれば、以下の推奨される制約を使用してください。

- `create_clock`, `create_generated_clock`: デザインのすべてのクロックの周波数と関係を指定します。
- `derive_pll_clocks`: PLL メガファンクションでの設定に従って、すべての PLL 出力に対して生成されるクロックを作成します。LVDS トランスミッタまたはレシーバのバラレル変換係数に対して、マルチサイクル関係を指定します。
- `set_input_delay`, `set_output_delay`: 外部デバイスまたは外部ボードのタイミング・パラメータを指定します。
- `derive_clock_uncertainty`: インター・クロック、イントラ・クロック、および I/O インタフェースの不確実性を自動的に適用します。
- `check_timing`: 制約の欠落を含め、デザインまたは適用された制約に関する問題に関するレポートを生成します。

## 面積およびタイミングの最適化

この項では、面積（またはリソース利用率）およびタイミング性能の最適化を支援するために **Quartus II** ソフトウェアで提供される機能のいくつかを重点的に説明します。タイミング解析でデザイン要件に適合しなかったことがレポートされた場合は、タイミング・クロージャを達成するように、デザインまたは設定を変更して、デザインを再コンパイルする必要があります。コンパイルの結果、`no-fit` メッセージが生成された場合は、配置配線を成功させるために変更を行う必要があります。



追加の最適化機能については、「**Quartus II** ハンドブック Volume 2」の「面積およびタイミングの最適化」の章を参照してください。

Early Timing Estimation 機能を使用して、ソフトウェアが完全な配置配線を実行する前に、デザインのタイミング結果を見積もることができます。解析と合成を実行した後、Processing メニューで **Start** をポイントし、**Start Early Timing Estimate** をクリックして、最初のコンパイル結果を生成します。この機能を使用すると、フル・コンパイルを実行した場合よりもタイミングの見積もり速度が最大 45 倍速くなります。フィットは完全には最適化されず、配線もされません。したがって、タイミング解析レポートは見積もりにすぎません。見積もられた遅延は、最終的なタイミング結果と比較した場合、平均でフル・コンパイルで達成される遅延の 11% 以内です。

フィジカル・シンセシス最適化によって、ネットリストに対して配置固有の変更が行われ、特定のアルテラ・デバイスの結果が改善されます。**Physical synthesis for performance** オプションまたは **Physical synthesis for fitting** オプションを指定することができます。これらのオプションを指定すると通常、コンパイル時間は大幅に増えますが、プッシュボタン最適化によって結果の品質が大きく改善されます。これらのオプションをオンにした場合は、デザインの結果が改善されたことを確認してください。デザインのタイミング要求を満足するのにこれらのオプションが必要ない場合は、これらをオフにしてコンパイル時間を短縮します。



詳しくは、「Quartus II ハンドブック Volume 2」の「Netlist Optimizations and Physical Synthesis」の章を参照してください。

Design Space Explorer (DSE) は、デザインに最適な Quartus II ソフトウェア設定の集合を検索するプロセスを自動化するユーティリティです。**Exploration Settings** の下にある **Search for Best Performance** および **Search for Best Area** オプションでは、複数のコンパイルと併せて、デザインの性能または面積の改善を目標とする定義済み Exploration Space を使用します。さらに、DSE ウィンドウの **Advanced** タブを使用して、**Optimization Goal** を **Optimize for Speed** または **Optimize for Area** に設定することもできます。



詳しくは、「Quartus II ハンドブック Volume 2」の「Design Space Explorer」の章を参照してください。

Optimization Advisor は、デザインを最適化する設定を作成するためのガイドを提供します。Tools メニューで **Advisors** をポイントし、**Resource Optimization Advisor** または **Timing Optimization Advisor** をクリックします。オプションを評価し、要件に最も適した設定を選択します。

## 性能の維持

インクリメンタル・コンパイル機能を使用して、デザインの変更されていない部分を維持し、それによって性能を維持すると共に、より効率的にタイミング・クロージャに達成することができます。ガイドラインおよび参考文献については、[49 ページの「階層およびチーム・ベース・デザインのプランニング」](#)を参照してください。

## コンパイル時間の短縮

インクリメンタル・コンパイル機能を使ってデザインを変更するときは、デザインのイタレーション時間を平均 60% 短縮できます。ガイドラインおよび参考文献については、49 ページの「階層およびチーム・ベース・デザインのプランニング」を参照してください。

デザインをコンパイルするのに複数のプロセッサを使用できる場合、Quartus II ソフトウェアは、複数のアルゴリズムをパラレルに実行して、複数のプロセッサを活用しながらコンパイル時間を短縮することができます。Quartus II のコンパイルに使用可能なプロセッサの数を設定するには、**Settings** ダイアログ・ボックスの **Compilation Process Settings** ページで、**Maximum processors allows for parallel compilation** を指定します。プロセッサ数のデフォルト値は 1 ですが、この値はパラレル・コンパイルをディセーブルにします。

Compilation Time Advisor は、デザインのコンパイル時間を短縮する設定を作成するためのガイドを提供します。Tools メニューで **Advisors** をポイントし、**Compilation Time Advisor** をクリックします。これらの手法のいくつかを使ってコンパイル時間を短縮すると、結果の総合的な品質が低下する可能性があります。



詳しくは、「Quartus II ハンドブック Volume 2」の「面積およびタイミングの最適化」の章を参照してください。

## シミュレーション

Quartus II ソフトウェアは、機能シミュレーションとゲート・レベルのタイミング・シミュレーションの両方をサポートします。デザイン・フローの初めに機能シミュレーションを実行して、各デザイン・ブロックのデザイン機能性または論理動作をチェックします。デザインを完全にコンパイルする必要はなく、タイミング情報が含まれていない機能シミュレーション・ネットリストを生成することができます。タイミング・シミュレーションは、タイミング・アナライザで生成された、異なるデバイス・ブロックの遅延や配置配線情報を含むタイミング・ネットリストを使用します。デザインがターゲット・デバイスで確実に動作するように、デザイン・フローの最後にトップレベル・デザインに対してタイミング・シミュレーションを実行することができます。

内蔵の Quartus II シミュレータを使って、素早く簡単にシミュレーションを実行できます。Quartus II シミュレータでデザインをデバッグするために、ブ레이크ポイントを設定して特定の出力状態をキャプチャしたり、最新のシミュレーション結果と過去の結果を比較することができます。アルテラは、さらに Quartus II ライセンス・サブスクリプション付きの ModelSim-Altera シミュレータを提供しています。これにより、最先端のテストベンチ機能およびその他の機能を活用することができます。また、Quartus II EDA Netlist Writer は、Synopsys 社の VCS や Cadence 社の NC-Sim などのその他のサードパーティ・シミュレーション・ツールをサポートするために、タイミング・ネットリスト・ファイルを生成できます。Settings ダイアログ・ボックスの EDA Tools Settings ページでシミュレーション・ツールを指定して、適切な出力シミュレーション・ネットリストを生成します。

サードパーティ製シミュレーション・ツールを使用する場合は、Quartus II バージョンでサポートされているソフトウェア・バージョンを使用してください。Quartus II リリース・ノートには、Quartus II ソフトウェアの特定のバージョンで正式にサポートされている各シミュレーション・ツールのバージョンが記載されています。ライブラリはバージョンによって異なることがあり、使用するシミュレーション・ネットリストとの間に不一致が生じる可能性があるため、使用中の Quartus II ソフトウェアに付属しているモデル・ライブラリを使用してください。




シミュレーション・ツールのフローについて詳しくは、「Quartus II ハンドブック Volume 3」の「シミュレーション」セクションの該当する章を参照してください。

## フォーマル検証

Quartus II ソフトウェアは、フォーマル検証フローをサポートしています。必要なフォーマル検証フローがデザインおよびデザインのコンパイル・ステージに影響を及ぼすか否かについて検討してください。

フォーマル検証フローを使用すると、レジスタのリタイミングなどの特定のロジック最適化をオフにし、階層ブロックが強制的に維持され、それによって最適化が制約されるので、性能結果に影響を及ぼす可能性があります。フォーマル検証ではメモリ・ブロックをブラック・ボックスとして扱います。したがって、メモリを別の階層ブロックに保持して、他のロジックが検証のためにブラック・ボックスに取り込まれないようにすることが最良の策です。デザインが制限されるその他の制約もあります。詳しくは、ドキュメントを参照してください。デザインにフォーマル検証が重要な場合は、デザイン・フローの初期段階で制限と制約をプランニングする方が、後半で変更を行うよりも簡単です。

Quartus II リリース・ノートには、Quartus II ソフトウェアの特定のバージョンで正式にサポートされている各フォーマル検証ツールのバージョンが記載されています。Settings ダイアログ・ボックスの EDA Tools Settings ページでフォーマル検証ツールを指定して、適切な出力ネットリストを生成します。




## 消費電力の 解析および 最適化

フォーマル検証のフローについて詳しくは、「Quartus II ハンドブック Volume 3」の「フォーマル検証」セクションの該当する章を参照してください。

Stratix III デバイスは、最先端のプロセスと回路技術を主要回路およびアーキテクチャ技術革新と併用して、低消費電力と高性能を達成しています。Quartus II ソフトウェアは、選択したコア電圧（0.9 V または 1.1 V）に固有のパワー・モデルを使用して、デザインを最適化および解析します。プログラマブル・パワー・テクノロジー機能により、デザイン要件に応じて各プログラマブル LAB、DSP ブロック、およびメモリ・ブロックで高速または低消費電力を達成することができます。Quartus II ソフトウェアは、非クリティカル・デザイン・パス上の過度のスラックを自動的に活用して、クリティカル・パスの高性能を維持したまま消費電力を最小限に抑えます。

この項で説明するように、このソフトウェアは消費電力をさらに低減するために追加の消費電力最適化手法を提供します。この項ではまた、早期デザイン・コンセプトからデザイン実装まで、デバイスの消費電力と放熱性を見積もることができる消費電力解析ツール、および熱管理用の Stratix III 温度検知ダイオードの使用方法について説明します。



消費電力の低減に役立つアーキテクチャ機能、およびデザインの最適化手法については、アプリケーション・ノート「AN 437: Stratix III FPGA の消費電力の最適化」を参照してください。

### 消費電力の解析

デザインの完成前に、16 ページの「早期消費電力見積もり」の説明に従って、スプレッドシートを使用して消費電力を見積もります。デザインをコンパイルしたら、Quartus II PowerPlay Power Analyzer を使用して消費電力と放熱性を解析し、デザインが電源バジェットおよび熱バジェットに違反していないことを確認します。

PowerPlay Power Analyzer を使用するには、デザインをコンパイルし（デザイン・リソース、配置配線、および I/O 規格に関する情報を提供するため）、シグナル・アクティビティ・データ（トグル・レートおよびスタティック確率）を提供する必要があります。シグナル・アクティビティ・データは、シミュレーション結果、またはユーザ定義のデフォルト・トグル・レートおよびベクタなし見積もりから得ることができます。解析に使用されるシグナル・アクティビティは、実際の動作を反映したものでなければなりません。消費電力を最も正確に見積もるには、ゲート・レベル・シミュレーションの結果を、Quartus II Simulator またはサードパーティ・シミュレーション・ツールの VCD 出力ファイルで使用します。シミュレーション動作には、機能検証で頻繁に使用される例外的なケースではなく、実際の期間における標準的な入力ベクタを含める必要があります。良い結果を得るには、シミュレータの推奨設定（グリッチ・フィルタリングなど）を使用します。

また、コア電圧、デバイスの電力特性、周囲温度およびジャンクション温度、冷却ソリューション、およびボードの熱モデルなどの動作条件も指定する必要があります。**Settings** ダイアログ・ボックスの **Operating Conditions** ページで適切な設定を選択します。

ダイナミック消費電力、スタティック消費電力、および I/O 熱消費電力を計算するには、**Processing** メニューから **PowerPlay Power Analyzer Tool** をクリックします。このツールでは、解析に使用されるシグナル・アクティビティの集計、そしてシグナル・アクティビティのためのデータ・ソースの総合的な品質を反映する信頼性指標も提供されます。レポートは、提供されるデータに基づく消費電力の見積もりであり、電力仕様ではありません。常にデバイスのデータ・シートを参照してください。



消費電力解析、シグナル・アクティビティ情報を作成するためのシミュレーション設定の推奨事項について詳しくは、「Quartus II ハンドブック Volume 3」の「PowerPlay Power Analyzer」の章を参照してください。Signal Activity ファイル (.saf) およびその作成方法については、「Quartus II ハンドブック Volume 3」の「Quartus II Simulator」の章を参照してください。消費電力スプレッドシートの作成を含む消費電力管理システム・デザインの推奨事項およびガイドラインについては、「AN 448: Stratix III の消費電力管理デザイン・ガイド」を参照してください。

## 消費電力の最適化

Stratix III デバイスのダイナミック消費電力を削減するために、さまざまなデザインおよびソフトウェア手法を使用してデザインを最適化することができます。

Quartus II ソフトウェアの消費電力最適化は、正確な電力解析の結果に依存します。前の項のガイドラインを使用して、Quartus II ソフトウェアがデザインの動作および条件に対して正しく電力利用を最適化するようにします。

### デバイスおよびデザインの消費電力最適化手法

この項では、消費電力を低減可能ないくつかのデザイン手法について説明します。これらの手法の結果はデザインごとに異なります。



消費電力を低減するための詳細および追加デザイン手法については、「Quartus II ハンドブック Volume 2」の「消費電力の最適化」の章を参照してください。

#### デバイスのスピード・グレード

デザインに高性能モードを必要とする多数のクリティカル・タイミング・パスが含まれているときには、より高速なスピード・グレード・デバイス（利用可能な場合）を使用して消費電力を低減することができます。高速デバイスでは、ソフトウェアでより多くのデバイス・タイルが低消費電力モードを使用するように設定できます。

### クロック消費電力の管理

クロックは、高いスイッチング動作と長いパスのために、ダイナミック消費電力の大きな部分に関係します。Quartus II ソフトウェアは、ダウンストリームのレジスタに供給する必要があるクロック・ネットワーク部のみをイネーブルにすることによって、クロック配線の消費電力を自動的に最適化します。また、クロック・コントロール・ブロックを使用してクロック・ネットワークをダイナミックにイネーブルまたはディセーブルすることもできます。クロック・ネットワークがパワーダウンすると、そのクロック・ネットワークで供給されるすべてのロジックはトグルしないため、デバイスの全体的な消費電力が減少します。



クロック・コントロール・ブロックの使用方法については、「[altclkctrl Megafunction User Guide](#)」を参照してください。

クロック・ツリー全体をディセーブルしないで LAB ワイド・クロックの消費電力を低減するには、LAB ワイド・クロック・イネーブル信号を使用して LAB ワイド・クロックをゲートします。Quartus II ソフトウェアは、レジスタ・レベルのクロック・イネーブル信号を自動的に LAB レベルに昇格させます。

### メモリの低消費電力化

メモリの消費電力を低減する鍵は、メモリ・クロッキング・イベント数を低減することです。61 ページの「[クロック消費電力の管理](#)」で説明するクロック・ゲーティングまたはメモリ・ポートのクロック・イネーブル信号を使用することができます。

### I/O 消費電力のガイドライン

I/O バッファのダイナミック消費電力は、合計負荷キャパシタンスに比例するため、キャパシタンスが低いと消費電力が減少します。

LVTTL や LVCMOS などの非終端 I/O 規格のレール・トゥ・レール出力振幅は、 $V_{CCIO}$  電源電圧と同じです。ダイナミック消費電力は電圧を 2 乗した値に比例するため、低電圧 I/O 規格を使用してダイナミック消費電力を低減します。これらの I/O 規格は、ほとんどスタティック電力を消費しません。

ダイナミック消費電力は出力遷移周波数にも比例するため、高周波数アプリケーションの場合は、SSTL などの抵抗で終端された I/O 規格を使用します。出力負荷電圧振幅は、一部のバイアス・ポイント周辺では  $V_{CCIO}$  より小さいため、ダイナミック消費電力は同様の条件下では非終端 I/O の場合よりも少なくなります。

抵抗終端 I/O 規格では、電流が継続的に終端ネットワークにドライブされるため、多くのスタティック電力が消費されます。抵抗終端された I/O 規格を使用する場合は、スタティック消費電力を低減するために、速度および波形要件を満たす最小ドライブ強度を使用してください。外部デバイスで使用される電力は PowerPlay での計算には含まれないため、別にシステム消費電力の計算に含める必要があります。

## Quartus II の消費電力の最適化のテクニック

Quartus II ソフトウェアは、コア・ダイナミック消費電力を低減する消費電力が最適化された合成とフィッティングを提供します。パワー・ドリブン・コンパイルは、Stratix III シリコンのプロゲラマブル・パワー・テクノロジーと連携して動作します。デフォルトの設定は、**Normal compilation** です。デザインの達成可能な最大性能に影響を与える追加の消費電力最適化には、**Extra effort** を選択することができます。**Settings** ダイアログ・ボックスの **Analysis and Synthesis Settings** ページおよび **Fitter Settings** ページで、**PowerPlay power optimization** をクリックします。

使用するロジック・ブロック数が減少すると通常、スイッチング動作も低下するため、デザインの面積を最適化すれば消費電力が低下します。高性能モードを必要とする代わりに、低消費電力タイルを使用してデザインの多くを配置できるので、デザイン・ソース・コードを修正して性能を最適化しても消費電力低減できます。DSE および Power Optimization Advisor を使用して、消費電力を低減するための追加推奨事項を提供することができます。



パワー・ドリブン・コンパイルおよび Power Optimization Advisor について詳しくは、「Quartus II ハンドブック Volume 2」の「消費電力の最適化」の章を参照してください。

### DSE

DSE は、デザインに最適な Quartus II ソフトウェア設定の集合を検索するプロセスを自動化するユーティリティです。**Exploration Settings** の下にある **Search for Lowest Power** (最低消費電力を検索する) オプションでは、複数のコンパイルを使用してデザインの全体的な消費電力の改善を目標とする、定義済み Exploration Space を使用します。DSE ウィンドウの **Advanced** タブを使用して、**Optimize for Power** に対して **Optimization Goal** オプションを設定することもできます。



詳しくは、「Quartus II ハンドブック Volume 2」の「Design Space Explorer」の章を参照してください。

### Power Optimization Advisor

Quartus II ソフトウェアには、現在のデザイン・プロジェクトの設定およびアサインメントに基づく、消費電力の最適化に関する具体的なアドバイスや推奨事項を提供する Power Optimization Advisor 機能があります。Tools メニューで Advisors をポイントし、**Power Optimization Advisor** をクリックします。推奨される変更のいずれかを行った後、デザインを再コンパイルし、Power Play Power Analyzer を実行して消費電力の変化をチェックします。

## 熱管理

デザインの消費電力を最適化し、消費電力解析の最終的な数値が出た後、ボードの冷却ソリューションがデバイスの消費電力で生成された熱を放逸させるのに十分かどうかチェックします。熱管理にとっては、ジャンクション温度を計算または測定することが非常に重要です。従来、ジャンクション温度は、周囲温度またはケース温度、ジャンクションから周囲まで ( $\theta_{JA}$ ) の熱抵抗、またはジャンクションからケースまで ( $\theta_{JC}$ ) の熱抵抗、およびデバイスの消費電力を使用して計算されていました。

Stratix III デバイスは、温度検知ダイオード (TSD) を備えているため、デバイスのジャンクション温度を自身でモニタし、FPGA への空気流量の制御などの作業のために、外部回路を付加して使用できます。

TSD を回路使用する場合は、それをデザインに含める必要があります。また、検知ダイオードを外部温度センサに接続する場合、外部ピンを正しく接続する必要があります。



これらの機能について詳しくは、「Stratix III デバイス・ハンドブック」の「Stratix III デバイスのプログラマブル・パワーおよび温度検知ダイオード」の章を参照してください。

## まとめ

本資料のデザイン・ガイドラインでは、高集積、高性能 Stratix III デザインで考慮すべき重要な要素について説明します。良好な結果を達成し、一般的な問題を回避してデザインの生産性を向上させるには、デザイン・プロセス全体を通じてアルテラの推奨事項に従うことが重要です。67 ページの「デザイン・チェックリスト」を使用して、Stratix III デザインを完成させる前に、すべてのガイドラインを再検討したことを確認することができます。

## 参考資料

このアプリケーション・ノートでは、以下のドキュメントを参照しています。

- 「AN 386: Using the Parallel Flash Loader with the Quartus II Software」
- 「AN 437: Stratix III FPGA の消費電力の最適化」
- 「AN 448: Stratix III の消費電力管理デザイン・ガイド」
- 「altclkctrl Megafunction User Guide」
- 「altpll Megafunction User Guide」
- 「altpll\_reconfig Megafunction Users Guide」
- 「altremote\_update Megafunction User Guide」
- 「Quartus II ハンドブック Volume 2」の「Analyzing and Optimizing the Design Floorplan」の章
- 「Quartus II ハンドブック Volume 2」の「Area and Timing Optimization」の章
- ByteBlaster II Download Cable User Guide
- 「Quartus II ハンドブック Volume 2」の「Cadence PCB Design Tools Support」の章
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのクロック・ネットワークおよび PLL」の章
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのコンフィギュレーション」の章
- 「Stratix III デバイス・ハンドブック Volume 2」の「Stratix III デバイスの DC およびスイッチング特性」の章
- 「Quartus II ハンドブック Volume 3」の「SignalTap II エンベデッド・ロジック・アナライザを使用したデザインのデバッグ」の章
- 「Quartus II ハンドブック Volume 3」の「Design Debugging Using In-System Sources and Probes」の章
- 「Quartus II ハンドブック Volume 1」の「Design Recommendations for Altera Devices and the Quartus II Design Assistant」の章
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのデザイン・セキュリティ」の章
- 「Quartus II ハンドブック Volume 2」の「デザイン・スペース・エクスプローラ」の章
- 「コンフィギュレーション・ハンドブック Volume 2」の「エンハンスド・コンフィギュレーション・デバイス・データシート」の章
- EthernetBlaster Download Cable User Guide
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスの外部メモリ・インタフェース」の章
- 「Quartus II ハンドブック Volume 3」の「セクション VI. フォーマル検証」
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのホット・ソケットおよびパワー・オン・リセット」の章
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスの IEEE 1149.1 (JTAG) バウンダリ・スキャン・テスト」の章
- 「Quartus II ハンドブック Volume 2」の「I/O 管理」の章
- 「Quartus II ハンドブック Volume 3」の「外部ロジック・アナライザを使用したイン・システム・デバッグ」の章

- 「Quartus II ハンドブック Volume 3」の「In-System Updating of Memory and Constants」の章
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのロジック・アレイ・ブロックおよびアダプティブ・ロジック・モジュール」の章
- 「Quartus II ハンドブック Volume 2」の「Managing Quartus II Projects」の章
- 「Quartus II ハンドブック Volume 2」の「Mentor Graphics PCB Design Tools Support」の章
- 「Quartus II ハンドブック Volume 2」の「Netlist Optimizations and Physical Synthesis」の章
- 「Quartus II ハンドブック Volume 2」の「消費電力の最適化」の章
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのプログラマブル・パワーおよび温度検知ダイオード」の章
- 「Stratix III FPGA 用 PowerPlay Early Power Estimator ユーザガイド」
- 「Quartus II ハンドブック Volume 3」の「PowerPlay による電力解析」の章
- 「Quartus II ハンドブック Volume 1」の「Quartus II Incremental Compilation for Hierarchical and Team-Based Design」の章
- 「Quartus II ハンドブック Volume 3」の「Quartus II Programmer」の章
- 「Quartus II ハンドブック Volume 3」の「Quartus II Simulator」の章
- 「Quartus II ハンドブック Volume 3」の「Quick Design Debugging Using SignalProbe」の章
- 「Quartus II ハンドブック Volume 1」の「Recommended HDL Coding Styles」の章
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのリモート・システム・アップグレード」の章
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスにおける SEU の緩和」の章
- 「コンフィギュレーション・ハンドブック Volume 2」の「シリアル・コンフィギュレーション・デバイス・データシート」の章
- 「Quartus II ハンドブック Volume 3」の「Signal Integrity with Third-Party Tools」の章
- 「Quartus II ハンドブック Volume 3」の「セクション I. シミュレーション」
- sld\_virtual\_jtag Megafunction User Guide
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイス・ファミリの概要」の章
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスの I/O 機能」の章
- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのデザイン・セキュリティ」の章
- 「Stratix III FPGA シグナル・インテグリティ」ホワイトペーパー
- セクション III. 「Quartus II ハンドブック Volume 1」の「合成」
- 「Quartus II ハンドブック Volume 3」の「Quartus II TimeQuest Timing Analyzer」の章
- 「Quartus II ハンドブック Volume 3」の「Synopsys PrimeTime Support」の章
- 「USB-Blaster II USB ダウンロード・ケーブル・ユーザガイド」

- 「コンフィギュレーション・ハンドブック Volume 2」の「アルテラ・エンハンスド・コンフィギュレーション・デバイス」の章
- 「Quartus II ハンドブック Volume 4」の「Volume 4: SOPC Builder」

# デザイン・ チェックリスト

このチェックリストは、本資料で説明したガイドラインの要約を提供します。このチェックリストを使用して、デザインの各ステージでガイドラインに従っていることを確認してください。

プロジェクト名:
日付:

## 2 ページの「デバイスの選択」

	Done (確認済)	N/A (該当なし)	
1	<input type="checkbox"/>	<input type="checkbox"/>	ロジック/メモリ/マルチプライヤの集積度、PLL の I/O ピン数、パッケージの種類、および将来の開発のための予備リソースなどに基づいて、デバイスを選択。
2	<input type="checkbox"/>	<input type="checkbox"/>	パーティカル・デバイス・マイグレーションを検討。
3	<input type="checkbox"/>	<input type="checkbox"/>	スピード・グレードおよび選択可能なコア電圧の可用性を検討。

## 5 ページの「デバイス・コンフィギュレーションのプランニング」

	Done	N/A	
4	<input type="checkbox"/>	<input type="checkbox"/>	コンフィギュレーション手法を選択し、コンパニオン・デバイスとボード接続をプラン。
5	<input type="checkbox"/>	<input type="checkbox"/>	コンフィギュレーション・デバイスがユーザの Stratix III デバイスのコンフィギュレーション・ビットストリーム・ファイル・サイズをサポートしているかどうかを確認。
6	<input type="checkbox"/>	<input type="checkbox"/>	コンフィギュレーション手法が、データの復元、デザイン・セキュリティ、およびリモート・アップグレードなどの必要な機能をサポートしていることを確認。
7	<input type="checkbox"/>	<input type="checkbox"/>	PFL にフラッシュ・デバイスを使用する場合は、サポートされているデバイスのリストをチェック。
8	<input type="checkbox"/>	<input type="checkbox"/>	オプションのコンフィギュレーション・ピン CLKUSR および INIT_DONE を必要に応じてイネーブルにする。
9	<input type="checkbox"/>	<input type="checkbox"/>	Auto-restart after configuration error オプションを使用。
10	<input type="checkbox"/>	<input type="checkbox"/>	ファイルを別のフォーマットで生成するかコンフィギュレーション・チェーンのために Convert Programming Files を使用。

## 12 ページの「早期システム・プランニング」

	Done	N/A	
11	<input type="checkbox"/>	<input type="checkbox"/>	詳細なデザイン仕様を作成し、該当する場合はテスト・プランを作成。
12	<input type="checkbox"/>	<input type="checkbox"/>	システム・デザイン、特に I/O インタフェースに影響を与える IP を選択。
13	<input type="checkbox"/>	<input type="checkbox"/>	オン・チップ・デバッグ機能を活用して内部信号を解析し、高性能デバッグ手法を実行。
14	<input type="checkbox"/>	<input type="checkbox"/>	オン・チップ・デバッグ手法を早期段階で選択して、メモリおよびロジック要件、I/O ピン接続、およびボード接続をプラン。
15	<input type="checkbox"/>	<input type="checkbox"/>	OpenCore Plus (Tethered モード) または JTAG デバッグ機能を使用する場合は、ボードを JTAG 接続用に設計。
16	<input type="checkbox"/>	<input type="checkbox"/>	デバッグ機能で使用する追加のデバイス・リソース用のスペースをプラン。
17	<input type="checkbox"/>	<input type="checkbox"/>	デバッグ用 I/O ピンを予約。
18	<input type="checkbox"/>	<input type="checkbox"/>	ボードがデバッグ・モードをサポートしていることを確認し、必要に応じてヘッダまたはコネクタを組み込む。
19	<input type="checkbox"/>	<input type="checkbox"/>	インクリメンタル・デバッグのためにインクリメンタル・コンパイルがオンになっていることを確認。
20	<input type="checkbox"/>	<input type="checkbox"/>	必要な場合、デバッグ・メガファンクションをデザインに組み込む。
21	<input type="checkbox"/>	<input type="checkbox"/>	冷却ソリューションと電源をプランするために、Early Power Estimator スプレッドシートを使用して消費電力を見積もる。

## 18 ページの「ボード・デザインの検討事項」

	Done	N/A	
22	<input type="checkbox"/>	<input type="checkbox"/>	電圧電源のランプがモノトニックになるように設計。
23	<input type="checkbox"/>	<input type="checkbox"/>	パワーアップを考慮してボードを設計する。Stratix III の出力バッファは、デバイスがコンフィギュレーションされ、コンフィギュレーション・ピンがドライブ・アウトするまでトライステートになる。
24	<input type="checkbox"/>	<input type="checkbox"/>	電源が安定するように POR 時間を設定。

- 25   最高のデバイス信頼性が得られるように、パワー・シーケンスと電圧レギュレータを設計。
- 26   0.9 V または 1.1 V コア、1.1 V PLL および周辺、1.8 V、2.5 V、3.0 V、3.3 V のすべての電源ピンを正しく接続。
- 27   各バンクの I/O 規格をサポートするために、 $V_{CCIO}$  ピンを接続。
- 28   3.3 V  $V_{CCIO}$  に対しては  $V_{CCPD}$  ピンを 3.3 V に接続、3.0 V  $V_{CCIO}$  に対しては、 $V_{CCPD}$  ピンを 3.0 V に接続し、その他の I/O 電圧に対しては 2.5 V を接続。
- 29   電圧リファレンス形式の規格の場合は VREF ピンを接続し、デザインでノイズが発生しないようにする。
- 30   デザインで PLL を使用しない場合でも、すべての PLL 電源ピンを接続。
- 31   リニア・レギュレータを使用して、アナログ回路 ( $V_{CCA\_PLL}$  および  $V_{CCPT}$ ) を駆動。
- 32   電源から各  $V_{CCA\_PLL}$  ピンに厚い配線パターン (最低 20mils) を走らせる。
- 33   すべての  $V_{CCD\_PLL}$  電源ピンをボード上で最もノイズの少ないデジタル電源に接続する。
- 34   各  $V_{CCA\_PLL}$  および  $V_{CCD\_PLL}$  ピンをデカップリング回路でフィルタする。
- 35   電源がボードに入る  $V_{CCA\_PLL}$  および  $V_{CCD\_PLL}$  にフェライト・ビーズとタンタル並列コンデンサを使用。
- 36   コンフィギュレーション・ピンの DCLK および TCLK ピンにノイズが発生ないように設計。
- 37   JTAG ピンが未使用の場合は、安定した電圧レベルに接続。
- 38   JTAG ピンをダウンロード・ケーブルのヘッダに正しく接続する。ピンの順番が逆にならないようにする。
- 39   1 k $\Omega$  抵抗を通して TRST を  $V_{CCPD}$  に接続。
- 40   抵抗を通して TCK をプルダウンし、TMS をプルアップ。
- 41   ダウンロード・ケーブルと JTAG ピンの電圧互換性を確認。
- 42   特にコネクタに対して、またはケーブルが 4 個以上のデバイスをドライブする場合は、推奨事項に従って JTAG 信号をバッファ。

- 43   デバイスがコンフィギュレーション・チェーンにある場合は、チェーン内のすべてのデバイスが正しく接続されていることを確認。
- 44   MSEL ピンをフローティング状態にしないで、コンフィギュレーション手法を選択するように接続する。柔軟性を得るために、各ピンを 0- $\Omega$  抵抗で、V<sub>CCPGM</sub> または GND のいずれかに接続できるようにボードを設定。
- 45   nIO\_PULLUP を正しく接続して、内部プルアップ抵抗を設定。
- 46   nCE チップ・イネーブルを、コンフィギュレーション実行中、初期化中、およびユーザーモードでは Low に保持。
- 47   必要に応じて、デバイス・ワイド出力イネーブル・オプションをオンにする。
- 48   未使用 I/O ピンの予約状態を指定。
- 49   Quartus II ソフトウェアで生成された PIN ファイルで、ピン接続を慎重にチェックする。RESERVED ピンは接続しない。
- 50   ボード設計時に SSN を考慮する。ボード層上のデバイスに近い大きなバス信号を分割し、2つの信号層が隣接する場合は配線パターンを直交させて配線し、配線パターン幅の 2×～3× の距離を取る。
- 51   選択した I/O 規格、特に電圧リファレンス形式の規格の I/O 終端およびインピーダンス・マッチングをチェック。
- 52   IBIS モデル（使用可能な場合）を使用して、ボード・レベルでシミュレーションを実行。
- 53   Quartus II アドバンスド I/O タイミング解析のためのボード・トレース・モデルをコンフィギュレーションする。

### 27 ページの [I/O およびクロックのプランニング]

- |    | Done                     | N/A                      |  |
|----|--------------------------|--------------------------|--|
| 54 | <input type="checkbox"/> | <input type="checkbox"/> | Quartus II Pin Planner を使用してピン・アサインメントを実施。                     |
| 55 | <input type="checkbox"/> | <input type="checkbox"/> | ピン・アサインメントのサイン・オフのためのフル・コンパイルが完了した後、Quartus II Fitter レポートを使用。 |
| 56 | <input type="checkbox"/> | <input type="checkbox"/> | Quartus II ピン・アサインメントが回路図およびボード・レイアウト・ツールのアサインメントに適合することを検証。   |

- 57   I/O Assignment Analysis を使用して、Top-Level Design File コマンドを作成し、デザインの完成前に I/O アサインメントをチェック。
- 58   各 I/O ピンに最適な信号タイプおよび I/O 規格を選択。
- 59   ターゲットの I/O バンクで適切な I/O 規格サポートがサポートされていることを確認。
- 60   ソフトウェアで差動ピン・ペアのネガティブ・ピンの位置を割り当てる。
- 61   同じ電圧レベルを共有する I/O ピンを同じ I/O バンク内に配置。
- 62   各 I/O バンクのすべての出力信号が、バンクの  $V_{CCIO}$  電圧レベルでドライブ・アウトするようになっていることを確認。
- 63   各 I/O バンクのすべての電圧リファレンス形式の信号が、バンクの VREF 電圧レベルを使用するようになっていることを確認。
- 64   LVDS ピンの配置についてのガイドラインに従う。
- 65   各メモリ・インタフェースの ALTMEMPHY メガファンクション（または IP コア）を使用して、該当するドキュメントの接続ガイドラインに従う。
- 66   メモリ・インタフェースに専用 DQ ピンおよび DQ グループを使用。
- 67   Pin Planner Pad View を使用して、パッド配置ガイドラインに従うよう支援。
- 68   パッド配置ガイドラインに従うために必要な場合は、適切な I/O 関連アサインメントを設定。
- 69   兼用ピンの設定を行い、これらのピンを通常の I/O として使用する際の制約をチェック。
- 70   電流強度、スルー・レート、I/O 遅延、オープン・ドレイン、バス・ホールド、プログラマブル・プルアップ抵抗、PCI クランプ・ダイオードなど、他のデバイスとのインタフェースに役立つ利用可能なデバイスの I/O 機能をチェック。
- 71   On-Chip Termination（チップ内終端）を使用して、ボード・スペースを節約。
- 72   必要な終端方法がすべてのピン位置でサポートされていることをチェックする（コラム I/O バンクおよび特定の差動クロック・ペアは、RD 終端をサポートしない）。
- 73   DPA を使用する場合は、必ずこの機能をイネーブルして、デザインでデバイスの右側および左側の正しい PLL が使用されるようにする。
- 74   クロックおよびグローバル・コントロール信号に、正しい専用クロック・ピンおよび配線信号を使用。

- 75   デバイスの PLL をクロック管理に使用。
- 76   各 PLL およびクロック・ピンの入力および出力配線接続を解析。PLL 入力が専用クロック・ピンまたは別の PLL から供給されることを確認。
- 77   PLL 機能をイネーブルにし、MegaWizard Plug-In Manager の設定をチェック。
- 78   クロックの選択とパワーダウンにクロック・コントロール・ブロックを使用。
- 79   PLLが必要なクロック出力数を提供し、専用クロック出力ピンを使用していることをチェック。
- 80   デザインに最適な PLL 補正モードを使用。
- 81   デザインを解析して、同時スイッチング・ノイズ問題が発生する可能性があるかどうか調べる。
- 82   可能であれば同時に電圧を切り換えるピン数を減らす。
- 83   スwitching速度が高いI/Oには、低いドライブ強度、差動I/O規格、および低電圧規格を使用。
- 84   各バンク内の同時スイッチング出力ピン数を減らす。
- 85   スwitchingする I/O をバンク全体に均一に分散させる。
- 86   同時にスswitchingするピンを、SSN に敏感な入力ピンから分離する。
- 87   重要なクロック・コントロール信号および非同期コントロール信号は、大きなスswitching・バスから離して、グランド信号の近くに配置。
- 88   PLL 電源ピンから 1 本または 2 本分離れた I/O ピンを、スswitching速度が速い、またはドライブ強度が高いピンに使用しないようにする。
- 89   スタッガード出力遅延を使用して、出力信号を経時的にシフトするか、調整可能なスルー・レート設定を使用。

### 43 ページの「デザインおよびコンパイル」

Done N/A

- 90   サードパーティ合成ツールを指定し、サポートされている正しいバージョンを使用。
- 91   同期デザイン手法を使用。クロック信号に注意。

- 92   デザインの信頼性をチェックするために Quartus II Design Assistant を使用。
- 93   MegaWizard Plug-In Manager でメガファンクションを使用。
- 94   特にメモリ・ブロックなどのデバイス専用ロジックを推測する場合は、推奨コーディング・スタイルに従う。
- 95   必要に応じて、すべてのレジスタをクリアするためにチップ・ワイド・リセットを使用。
- 96   レジスタ・パワーアップおよびコントロール信号に使用できるリソースを検討する。リセット信号とプリセット信号の両方を 1 個のレジスタに適用しない。
- 97   インクリメンタル・コンパイル実行のためにデザインを分割し、デザイン・フローの早期段階でプランするという推奨事項に従う。
- 98   特にチーム・ベース・フローで、最高の結果を達成するためにパーティション間でタイミング・バジェットの作成とリソース・バランシングを実行。
- 99   最高の結果を得るために、各パーティションについてデザイン・フロアプランを作成。
- 100   システム・デザインおよびプロセッサ・デザインに SOPC Builder を活用。

## 52 ページの「タイミング・クロージャおよび検証」

- |     | Done                     | N/A                      |   |
|-----|--------------------------|--------------------------|---|
| 101 | <input type="checkbox"/> | <input type="checkbox"/> | コンパイル後にリソース利用に関するレポートを確認。   |
| 102 | <input type="checkbox"/> | <input type="checkbox"/> | すべての Quartus II Messages、特にワーニング・メッセージを確認。  |
| 103 | <input type="checkbox"/> | <input type="checkbox"/> | すべてのクロック信号および I/O 遅延を含めて、タイミング制約が完全かつ正確であることを確認。                                    |
| 104 | <input type="checkbox"/> | <input type="checkbox"/> | Optimize fast-corner timing オプションをオンにする。  |
| 105 | <input type="checkbox"/> | <input type="checkbox"/> | <b>Optimize hold timing to All paths</b> を設定（ホールド・タイム問題を発生させる可能性があるデザイン問題をチェックした後）。 |
| 106 | <input type="checkbox"/> | <input type="checkbox"/> | <b>Enable multicorner timing analysis</b> をオンにする。                                   |
| 107 | <input type="checkbox"/> | <input type="checkbox"/> | derive_pll_clocks を使用して、PLL 用に生成されたクロックを作成。   |

- 108   `derive_clock_uncertainty` を使用して、不確実性を適用。
- 109   `check_timing` を使用して、制約問題をレポート。
- 110   コンパイル後に TimeQuest タイミング・アナライザ・レポートを調べて、タイミング違反がないことを確認。
- 111   Stratix III デバイスにデータが提供されるときに、入力 I/O タイムに違反していないことを確認。
- 112   フル・コンパイル実行前にタイミングの見積もりが必要な場合は、Early Timing Estimation を実行。
- 113   タイミング・クロージャを達成したり、リソース利用を改善するために Quartus II の最適化機能を使用。
- 114   Timing and Area Optimization Advisors を使用して最適な設定を提案。
- 115   デザインで変更されていないブロックの性能を保持し、コンパイル時間を短縮するためにインクリメンタル・コンパイルを使用。
- 116   コンパイルに複数のプロセッサを使用できる場合は、パラレル・コンパイルを設定。
- 117   Compilation Time Advisor を使用してコンパイル時間を短縮する設定を提案。
- 118   サードパーティ・シミュレーション・ツールを指定し、サポートされている正しいバージョンとシミュレーション・モデルを使用。
- 119   サードパーティ・フォーマル検証ツールを指定し、サポートされている正しいバージョンを使用。

### 59 ページの「消費電力の解析および最適化」

- |     | Done                     | N/A                      |   |
|-----|--------------------------|--------------------------|---|
| 120 | <input type="checkbox"/> | <input type="checkbox"/> | デザインの完成前に、Early Power Estimator スプレッドシートで消費電力を見積もる。                   |
| 121 | <input type="checkbox"/> | <input type="checkbox"/> | コンパイル終了後、PowerPlay Power Analyzer で消費電力および放熱性を解析。                     |
| 122 | <input type="checkbox"/> | <input type="checkbox"/> | 正確な消費電力解析結果を得るために、できればゲート・レベル・シミュレーション VCD ファイルで正確な標準シグナル・アクティビティを提供。 |
| 123 | <input type="checkbox"/> | <input type="checkbox"/> | 正しい動作条件を指定。   |
| 124 | <input type="checkbox"/> | <input type="checkbox"/> | 必要に応じて、デザインの消費電力を最適化するために、推奨デザイン手法および Quartus II オプションを使用。            |

- 125   Power Optimization Advisor を使用して最適化設定を提案。
- 126   デザインで温度検知ダイオードを設定して、熱管理のためのデバイス・ジャンクション温度を測定。