



この翻訳版ドキュメントのメンテナンスは終了しております。

この文書には、古いコンテンツや商標が含まれている場合があります。

最新情報につきましては、次のリンクから英語版の最新資料をご確認ください。

<https://www.intel.com/content/www/us/en/programmable/documentation/lit-index.html>

Please take note that this document is no longer being maintained. It may contain legacy content and trademarks which may be outdated.

Please refer to English version for latest update at

<https://www.intel.com/content/www/us/en/programmable/documentation/lit-index.html>

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。

2010年2月 v1.1

AN-592-1.1

このアプリケーション・ノートでは、使いやすいガイドラインを提供し、Cyclone® IV デザインで考慮すべき要素について説明します。アルテラは、デザイン・プロセスを通じてこのアプリケーション・ノートに記載されたガイドラインに従うことを推奨しています。アルテラの Cyclone IV デバイスは、ロジック、メモリ、デジタル信号処理 (DSP) のさまざまな組み合わせを最小の消費電力で提供します。Cyclone IV デバイスは、ディスプレイ、無線基盤設備、工業用イーサネット、放送コンバータ、およびチップ間ブリッジなどのコスト重視の量産向けアプリケーションに最適です。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 とシステムを計画することが不可欠です。

このアプリケーション・ノートでは、Cyclone IV デバイスのアーキテクチャ、およびデザインに使用する 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。製品に関するすべての詳細が含まれていません。また、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。

このアプリケーション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。表 1 では、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。



 このアプリケーション・ノートに述べたガイドラインに従ったことを確認するには、55 ページの「デザイン・チェックリスト」を参照してください。

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

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「システム仕様」	<ul style="list-style-type: none"> <li>■ デザイン仕様のプランニング</li> <li>■ IP の選択</li> </ul>
3 ページの「デバイスの選択」	<ul style="list-style-type: none"> <li>■ デバイス情報</li> <li>■ デバイス集積度、パッケージ、マイグレーション、およびスピード・グレードの決定</li> </ul>
7 ページの「早期システムおよびボード・プランニング」	<ul style="list-style-type: none"> <li>■ 早期消費電力見積もり</li> <li>■ コンフィギュレーション手法のプランニング</li> <li>■ オンチップ・デバッグのプランニング</li> </ul>
16 ページの「ボード・デザインにおけるピン接続の検討事項」	<ul style="list-style-type: none"> <li>■ パワーアップ</li> <li>■ 電源ピン、PLL の接続</li> <li>■ ディカップリング・コンデンサ</li> <li>■ コンフィギュレーション・ピン</li> <li>■ シグナル・インテグリティ</li> <li>■ ボード・レベル検証</li> </ul>

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

デザイン・フローのステージ	ガイドライン・トピック
25 ページの「I/O およびクロックのプランニング」	<ul style="list-style-type: none"> <li>■ ピン・アサインメント</li> <li>■ 早期ピン・プランニング</li> <li>■ I/O 機能および接続、メモリ・インタフェース</li> <li>■ クロックおよび PLL の選択</li> <li>■ 同時スイッチング・ノイズ (SSN)</li> </ul>
38 ページの「デザイン・エントリ」	<ul style="list-style-type: none"> <li>■ コーディング・スタイルおよびデザインに関する推奨事項</li> <li>■ SOPC Builder</li> <li>■ 階層デザインおよびチーム・ベース・デザインのプランニング</li> </ul>
45 ページの「デザインの実装、解析、最適化、および検証」	<ul style="list-style-type: none"> <li>■ 合成ツール、デバイス使用率、メッセージ</li> <li>■ タイミング制約および解析</li> <li>■ 面積およびタイミングの最適化</li> <li>■ コンパイル時間、検証</li> <li>■ 消費電力の解析および最適化</li> </ul>

 Cyclone IV デバイスのアーキテクチャについて詳しくは、アルテラ・ウェブサイトの資料: [Cyclone IV デバイスのセクション](#)を参照してください。Cyclone IV デバイスに関する最新の既知の問題については、[ナレッジ・データベース](#)を参照してください。

## システム仕様

Cyclone IV 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。デザイン・プロセスを開始するには、システムおよび FPGA に対して詳細なデザイン仕様を作成することによって、システムの他の部分に FPGA の入力および出力インタフェースを決定しなければなりません。

## 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、システムの定義、FPGA における I/O インタフェースの指定、異なるクロック・ドメインの識別、および基本デザイン機能のブロック図を含める詳細なデザイン仕様が必要です。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は、デザイン効率の向上に役立ちます。

1.  詳細なデザイン仕様を作成し、また、該当する場合はテスト・プランも作成。
2.  ブロック図を用いて、早期にクロック・ドメイン、クロック・リソース、および I/O インタフェースを計画。

チームがシステムの検証方法を知ること確保するために、機能検証プランを作成します。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。すべてのデザイン・インタフェースを検証する能力が必要な場合があります。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するには、FPGA デバイス内で **Nios®II** プロセッサと共に **UART** インタフェースを使用することによりできます。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、[14 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、共通のデザイン・ディレクトリ構造を使用することを検討してください。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。チーム・ベース・デザインについて詳しくは、[43 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。


## 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、ユーザー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 **QuartusII Programmer** がハードウェア評価中動作している必要があります。

- 3.  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。
- 4.  IP に **Open Core Plus 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。

 提供されている IP コアについて詳しくは、アルテラ・ウェブサイトの製品情報の [IP & リファレンス・デザインのページ](#)を参照してください。

## デバイスの選択

この項では、**Cyclone IV** デザイン・プロセスにおける最初のステップ、すなわちデザインの要件に最適なデバイス・ファミリー、デバイス集積度、機能、パッケージおよびスピード・グレードの選択について説明します。アルテラでは、**FPGA** マイグレーション・デバイスをターゲットにすることを推奨しています。これについて、この項でも説明されています。

- 5.  トランシーバ、I/O ピン数、LVDS チャネル、パッケージの種類、ロジック / メモリ / マルチプライヤの集積度、PLL、クロック配線、およびスピード・グレードに基づいて、デバイスを選択。

- 各デバイス集積度のロジック、メモリ・ブロック、乗算器、PLL、パッケージの種類および I/O ピン数について詳しくは、「Cyclone IV デバイス・ハンドブック Volume 1」の「*Cyclone IV デバイス・ファミリの概要*」の章を参照してください。

## デバイス・ファミリの種類および高速トランシーバ

Cyclone IV デバイス・ファミリは、異なるアプリケーションの要件に対応する 2 種類のタイプで提供されています。Cyclone IV GX デバイスは、最大 3.125 Gbps のデータ・レートで独立して動作できる全二重高速トランシーバを最大 8 個備えて、フィジカル・コーディング・サブレイヤ (PCS) およびフィジカル・メディア・アタッチメント (PMA) をサポートし、そして 1 個の PCI Express ハード IP ブロックを備えています。Cyclone IV E デバイスは最低消費電力に最適化され、最低のコストで高い性能を実現します。アプリケーションが 3.125 Gbps の高速トランシーバを要する場合は、GX デバイスを選択します。それ以外の場合は、E デバイスを選択して最低のコストで高い性能を得ます。

## ロジック、メモリ、および乗算器の集積度

Cyclone IV デバイスは、ロジック・エレメント (LE)、メモリおよび乗算器など、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。必要なロジック集積度を決定することは、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は少なくなります。6 ページの「*デバイスのパーティカル・マイグレーション*」で説明しているように、Cyclone IV デバイスは、柔軟性を提供するパーティカル・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います。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、それらのリソース使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。リソース使用率を確認して、デザインに適合するデバイスの集積度を決定します。コーディング・スタイル、デバイス・アーキテクチャ、および Quartus II ソフトウェアで使用される最適化オプションが、デザインのリソース使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。コンパイル済みのデザインにおけるリソース使用率の決定については、45 ページの「*デバイスのリソース使用率レポート*」を参照してください。

- アルテラ IP デザインのコンフィギュレーションにおけるリソース使用率の見積もりについては、アルテラ・ウェブサイトの「*オンライン資料：IP メガファンクション*」の項を参照してください。

### 6.□ 将来の開発およびデバッグ作業のために、予備リソースを保留。

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要件に最適なデバイスを選択します。43 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。また、14 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。

## I/O ピン数、LVDS チャンネルおよびパッケージの種類

Cyclone IV GX デバイスは、さまざまな I/O ピン数を持つ省スペースの Quad Flat Pack No Lead (QFN) パッケージおよび FineLine BGA (FBGA) パッケージで提供されます。Cyclone IV E デバイスは、さまざまな I/O ピン数を持つ省スペースの Enhanced Thin Quad Flat Pack (EQFP) および FineLine BGA パッケージで提供されます。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、アプリケーションに必要な I/O ピン数を決定します。

集積度とパッケージ・ピン数が高いほど、より多くのシリアライゼーションおよびデシリアライゼーション用の LVDS チャンネルが提供されます；デバイスの集積度とパッケージの組み合わせが十分な LVDS チャンネルを備えていることを確認します。

また、同時スイッチング・ノイズ (SSN) の問題、ピン配置ガイドライン、専用入力として使用されるピン、各 I/O バンクで使用可能な I/O 規格、ロウおよびコラム I/O バンク間における I/O 規格の差異、ロウとコラム I/O バンク間における速度の差異、およびパッケージ・マイグレーション・オプションなどのほかの要因は、必要 I/O ピンの数に影響を与えます。ピン位置の選択について詳しくは、16 ページの「ボード・デザインにおけるピン接続の検討事項」と 25 ページの「I/O およびクロックのプランニング」を参照してください。また、14 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。

## PLL およびクロック配線


Cyclone IV GX デバイスは、汎用 PLL (GPLL) および多用途 PLL (MPLL) の 2 種類の PLL を備えています。Cyclone IV E デバイスは、GPLL のみを備えています。FPGA フレブリックにおける汎用アプリケーション、または外部メモリ・インタフェースなどの周辺部品に GPLL を使用します。トランシーバ・ブロックのクロッキングに MPLL を使用します。MPLL をトランシーバのクロッキングに使用しない場合に、それらを汎用クロッキングに使用することができます。

Cyclone IV GX デバイスは、グローバル・クロック (GCLK) をドライブするための最大 12 本の専用クロック・ピン (CLK[15..4]) を備えています。Cyclone IV GX デバイスは、左側を除いてデバイスの各側で専用クロック・ピンを 4 本ずつサポートします。これらのクロック・ピンは、最大 30 個の GCLK をドライブできます。Cyclone IV E デバイスは、最大 20 個の GCLK をドライブできる最大 15 本の専用クロック・ピン (CLK[15..1]) を提供します。EP4CE6 と EP4CE10 デバイスを除くすべての Cyclone IV E デバイスでは、左側で専用クロック・ピンを 3 本サポートし、上側、右側および下側で専用クロック・ピンを 4 本ずつサポートします。EP4CE6 と EP4CE10 デバイスでは、左側において 3 本の専用クロック・ピンおよび右側において 4 本の専用クロック・ピンのみをサポートします。選択されたデバイス集積度とパッケージの組み合わせにデザインに必要な PLL およびクロック配線リソースが十分に含まれていることを確認します。GCLK リソースは一部の PLL に共用され、どの入力を利用可能であるかに影響を与えることがあります。クロック・ピンおよびグローバル配線リソースの情報や参考資料については、25 ページの「I/O およびクロックのプランニング」を参照してください。



## 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。Cyclone IV GX デバイスは、6、7、および 8 の最大 3 つのスピード・グレードで提供されています (6 が最も高速です)。Cyclone IV E デバイスは、6、7、8 および 9 の最大 4 つのスピード・グレードで提供されています (6 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、「Cyclone IV デバイス・ハンドブック Volume 1」の「*External Memory Interfaces in Cyclone IV Devices*」の章を参照してください。

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・グレードに移行することが考えられます。

## デバイスのバーティカル・マイグレーション

Cyclone IV 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います。これで、専用入力ピン、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。この機能により、ボード上の FPGA を集積度が異なる別の Cyclone IV 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。

 マイグレーション・デバイスのリストについては、「Cyclone IV デバイス・ハンドブック Volume 1」の「*Cyclone IV デバイス・ファミリの概要*」の章の「Cyclone IV デバイス・ファミリのパッケージ」の表を参照してください。

### 7.□ バーティカル・デバイス・マイグレーションの可用性および要件を検討。

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を決定します。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ように、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、QuartusII 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ことが推奨されています。フィッタは、デザインが選択したデバイスに準拠することを保証するため、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ることがあります。

マイグレーション・デバイスをデザイン・サイクルの後半でも追加できますが、そうすると新しいターゲット・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。

25 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、QuartusII Pin Planner 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。

## 早期システムおよびボード・プランニング

Quartus II ソフトウェアでデザインを完成する前のデザイン・プロセスの早期に、FPGA に関するシステム情報を計画すべきです。早いプランニングは、FPGA チームが PCB ボード・デザイナーおよびシステム・デザイナーに早期情報を提供することを可能にします。この項では、次の内容について説明します。

- 「早期消費電力見積もり」
- 8 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」
- 14 ページの「オン・チップ・デバッグのプランニング」

### 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、計算されたジャンクション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ます。

8. □ ロジック・デザインが完成する前に冷却ソリューションと電源をプランニングするために、アルテラの Powerplay Early Power Estimator(EPE) スプレッドシートを使用して消費電力を見積もる。


FPGA デバイスの消費電力は、デザイン・ロジック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見積もりを困難にしています。EPE スプレッドシートでは、デザイン、動作周波数、トグル・レート、および環境条件を考慮するとき使用されるデバイスおよびデバイス・リソースに関する情報を処理することにより、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。EPE 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します。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、QuartusII 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。




EPE スプレッドシートには、Quartus II 生成の消費電力見積もりファイルまたは古いバージョンの **Early Power Estimator** から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 EPE スプレッドシートに手動で転送します。既存の **QuartusII** 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ことが推奨されています。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。

消費電力をもっと正確にチェックするために、デザインが完了するときに完全な消費電力解析を実行することが推奨されています。**QuartusII** 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。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、**QuartusII** シミュレータまたはサードパーティ・シミュレーション・ツールの **.vcd** (Verilog Value Change Dump ファイル) 出力ファイルで使用します。詳細は、51 ページの「消費電力の解析」を参照してください。

 **PowerPlay EPE** スプレッドシートおよびサポートされる各デバイス・ファミリのユーザーガイドについて詳しくは、アルテラ・ウェブサイトの **PowerPlay Early Power Estimators (EPE) & 消費電力解析** ページを参照してください。

 **PowerPlay EPE** スプレッドシートについて詳しくは、*「PowerPlay Early Power Estimator User Guide」* を参照してください。

 消費電力の見積りおよび解析について詳しくは、「QuartusII ハンドブック Volume 3」の *「PowerPlay Power Analysis」* の章を参照してください。

## デバイス・コンフィギュレーションのプランニング

Cyclone IV デバイスは揮発性である **SRAM** メモリをベースとするため、電源が投入されるたびに **Cyclone IV** 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。複数のコンフィギュレーション手法が必要か否かを考慮します（例えば、1つはデバッグまたはテスト用、もう1つは制作環境用）。デバイス・コンフィギュレーション手法をデザインの初期段階で選択することで、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、およびコンフィギュレーションのためのデバイス接続について詳しくは、16 ページの「**ボード・デザインにおけるピン接続の検討事項**」を参照してください。

さらに、**Cyclone IV** デバイスはコンフィギュレーション手法に応じて、アドバンス・コンフィギュレーション機能を提供します。また、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、**Cyclone IV** デバイスには、デザイン・プロセスの初期に選択して **QuartusII** ソフトウェアで設定しなければならないオプションのコンフィギュレーション・ピンおよびコンフィギュレーション・オプションも含まれています。

この項では、次の内容について説明します。

- 9 ページの「[コンフィギュレーション手法の選択](#)」
- 11 ページの「[コンフィギュレーション機能](#)」
- 13 ページの「[QuartusII のコンフィギュレーション設定](#)」

 コンフィギュレーションについて詳しくは、「Cyclone IV デバイス・ハンドブック Volume 1」の「[Configuration and Remote System Upgrades in Cyclone IV Devices](#)」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


Cyclone IV 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を使用してコンフィギュレーションできます。


- **Fast passive parallel (FPP)**— コンフィギュレーション・データをパラレルに Cyclone IV デバイスに供給するコントローラ。
- **Active parallel (AP)**— Cyclone IV デバイスは、パラレル・コンフィギュレーション・デバイスによってコンフィギュレーション・プロセスを制御し、コンフィギュレーション・データを取得します。
- **Active serial (AS)**— Cyclone IV デバイスは、シリアル・コンフィギュレーション (EPCS) デバイスによってコンフィギュレーション・プロセスを制御し、コンフィギュレーション・データを取得します。
- **Passive serial (PS)**— コンフィギュレーション・データをシリアルに Cyclone IV デバイスに供給するコントローラ。
- **JTAG**—Cyclone IV デバイスは、ダウンロード・ケーブルまたは MAX II デバイスあるいはフラッシュ・メモリ付きのマイクロプロセッサで、IEEE 1149.1 規格のインタフェースを通してコンフィギュレーションされます。

ボード上で Cyclone IV デバイスの MSEL ピンを特定の値にドライブすることにより、任意のコンフィギュレーション手法をイネーブ爾できます。

9.  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ニング。

 AP コンフィギュレーションは Cyclone IV E デバイスでのみサポートされています。


 FPP コンフィギュレーションは、EP4CGX30 (F484 パッケージのみ)、EP4CGX50、EP4CGX75、EP4CGX110、EP4CGX150 およびすべての Cyclone IV E デバイスでのみサポートされています。

 Cyclone IV GX デバイスが PCIe 100ms のウェイクアップ時間要件を満たすようにするためには、EP4CGX15、EP4CGX22、および (F484 パッケージ以外の) EP4CGX30 デバイスに PS コンフィギュレーション・モードを使用し、そして EP4CGX30 (F484 パッケージのみ)、EP4CGX50、EP4CGX75、EP4CGX110 および EP4CGX150 デバイスに FPP コンフィギュレーション・モードを使用しなければなりません。


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 (MAXII 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。

### シリアル・コンフィギュレーション・デバイス


AS コンフィギュレーション手法では、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) を使用します。シリアル・コンフィギュレーション・デバイスは、低コストでピン数の少ないコンフィギュレーション・ソリューションを提供します。


 シリアル・コンフィギュレーション・デバイスについては、「コンフィギュレーション・ハンドブック Volume 2」の *「Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet」* を参照してください。

シリアル・コンフィギュレーション・デバイスは、USB-Blaster™、EthernetBlaster、または ByteBlaster™ II ダウンロード・ケーブルによって QuartusII ソフトウェアでプログラムすることができます。また、APU (Altera Programming Unit) がサポートされている BP Microsystems や System General などのサードパーティ・プログラマ、または SRunner ソフトウェア・ドライバを搭載したマイクロプロセッサを使用してもかまいません。SRunner は、異なるエンベデッド・システムにフィットするよう簡単にカスタマイズ可能なエンベデッド・シリアル・コンフィギュレーション・デバイス・プログラミングを構築するコンフィギュレーション・ソフトウェア・ドライバです。

 SRunner については、*「AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming」* またはアルテラ・ウェブサイトの資料：*アプリケーション・ノート*のセクションにあるソース・コードを参照してください。


シリアル・コンフィギュレーション・デバイスは JTAG インタフェースを直接的にサポートしません。ただし、JTAG ダウンロード・ケーブルおよび QuartusII ソフトウェアの SFL (シリアル・フラッシュ・ローダ) 機能によりデバイスをプログラムすることができます。この機能は、FPGA を JTAG インタフェースとコンフィギュレーション・デバイス間のブリッジとして使用し、2つのデバイスが同一 JTAG インタフェースを使用することを可能にします。

 SFL ソリューションでは、コンフィギュレーション・デバイスをプログラムする前に FPGA をコンフィギュレーションする必要があるため、SFL ソリューションは標準の AS コンフィギュレーションより遅いです。

 SFL については、*「AN 370: Using the Serial FlashLoader with the Quartus II Software」* を参照してください。

## ダウンロード・ケーブル

Quartus II プログラマは、アルテラ・プログラミング・ケーブルを介して PS または JTAG インタフェースを使用し、Cyclone IV 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 エンベデッド・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。JTAG デバッグ・ツールについては、14 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。


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

- [ByteBlaster II Download Cable User Guide](#)
- [USB-Blaster Download Cable User Guide](#)
- [EthernetBlaster Communications Cable User Guide](#)

## MAX II パラレル・フラッシュ・ローダ

システムにコモン・フラッシュ・インタフェース (CFI) フラッシュ・メモリがある場合は、それを Cyclone IV デバイスのコンフィギュレーション・ストレージとしても使用できます。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 Cyclone IV 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。

10.  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。

 PFL については、[「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、Cyclone IV コンフィギュレーション機能およびそのデザイン・プロセスに対する影響について説明します。

11.  コンフィギュレーション手法とボードが、データの復元、リモート・システム・アップグレード、SEU (Single Event Upset) の緩和などの必要な機能をサポートしていることを確認。

 データの復元およびリモート・システム・アップグレードについては、「Cyclone IV デバイス・ハンドブック Volume 1」の [「Configuration and Remote System Upgrades in Cyclone IV Devices」](#) の章を参照してください。

 SEU の緩和については、「Cyclone IV デバイス・ハンドブック Volume 1」の [「SEU Mitigation in Cyclone IV Devices」](#) の章を参照してください。

## データ圧縮

データ圧縮機能をイネーブルにすると、QuartusII 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・ファイルを作成します。これらの圧縮ファイルは、コンフィギュレーション・デバイスまたはフラッシュ・メモリに必要な容量を低減し、Cyclone IV 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。Cyclone IV 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。

Cyclone IV デバイスは、復元機能を AS および PS コンフィギュレーション手法でサポートしています。AS または PS モードを使用してコンフィギュレーション・タイムを短縮する場合に Cyclone 圧縮復元機能を使用することが推奨されています。Cyclone IV 復元機能は、JTAG、AP、および FPP コンフィギュレーション手法では提供されていません。

## リモート・システム・アップグレード

リモート・システム・アップグレードは、経費のかかる製品回収を行わずに機能強化やバグ修正を行うことができ、製品の市場投入の短縮、製品寿命の延長、およびシステム・ダウンタイムの回避に役立ちます。Cyclone IV デバイスは専用のリモート・システム・アップグレード回路を備えています。Cyclone IV デバイスに実装されるソフト・ロジック (Nios II エンベデッド・プロセッサまたはユーザー・ロジック) は、遠隔地から新しいコンフィギュレーション・イメージをダウンロードし、それをコンフィギュレーション・メモリに格納し、さらに専用リモート・システム・アップグレード回路にリコンフィギュレーション・サイクルの開始を指示することもできます。

Cyclone IV デバイスは、シングル・デバイス AP および AS コンフィギュレーション手法でのみリモート・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、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。



ALTREMOTE\_UPDATE メガファンクションについては、[Remote Update Circuitry \(ALTREMOTE\\_UPDATE\) Megafunction User Guide](#) を参照してください。

## SEU の緩和および CRC エラー・チェック

Cyclone IV デバイスでは、CRC (Cyclic Redundancy Check) エラー検出機能の専用回路が組み込まれ、オプションで継続的かつ自動的に SEU をチェックすることができます。これで、Cyclone IV デバイスに格納されたコンフィギュレーション・データが正確であるかを確認することができ、システムにコンフィギュレーション・エラーの発生に警戒することもできます。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 ピンを使用してエラー発生を通知し、それを適切に対処するようにシステムを設計します。これらの CRC\_ERROR 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に、デザイン I/O ピンとして使用できます。



SEU 緩和機能は、1.2V のコア電圧を使用する Cyclone IV GX デバイスおよび Cyclone IV E デバイスに使用できます。

## QuartusII 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。ボードやシステムのデザインはこれらの設定およびピンに影響されるため、プランニング・ステージでこれらを考慮してください。**Device and Pin Options** ダイアログ・ボックスでの **General** タブでこれらのオプションを設定します。

### オプションのコンフィギュレーション・ピン

表 2 は、Cyclone IV デバイスに使用できるオプションのコンフィギュレーション・ピンのリストです。


表 2. オプションのコンフィギュレーション・ピン

コンフィギュレーション・ピン	説明
CLKUSR	<ul style="list-style-type: none"> <li>■ <b>Enable user-supplied start-up clock(CLKUSR)</b> オプションにより、内部発振器または 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。</li> <li>■ Cyclone IV デバイスでは、<b>AS</b> コンフィギュレーション時に、CLKUSR ピンを <b>DCLK</b> 用の外部クロック・ソースとして選択することができます。Quartus II ソフトウェアで、<b>Device and Pin Options</b> ダイアログ・ボックスの <b>Configuration</b> タブから、クロック・ソースのオプションを変更することができます。</li> </ul>
INIT_DONE	<ul style="list-style-type: none"> <li>■ デバイスが初期化を完了してユーザー・モードになっているかどうかをチェックするには、INIT_DONE ピンをモニタします。このピンは <b>Enable INIT_DONE output</b> オプションでイネーブルにします。INIT_DONE ピンはオープン・ドレイン出力であるため、INIT_DONE ピン所在の I/O バンクの <math>V_{CCIO}</math> 電源への外部 10K<math>\Omega</math> プルアップ抵抗を必要とします。</li> </ul>

12.  オプションの CLKUSR および INIT\_DONE コンフィギュレーション・ピンをサポートするようにボード・デザインを計画。

### エラー発生後のコンフィギュレーションの再開

**Auto-restart configuration after error** オプションがイネーブルされると、コンフィギュレーション・エラーの発生時にデバイスは nSTATUS を **Low** にドライブしてデバイスを内部でリセットします。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、nSTATUS ピン所在の I/O バンクの  $V_{CCIO}$  電源への外部 10K $\Omega$  プルアップ抵抗を必要とします。

-  アルテラのダウンロード・ケーブルでデザイン変更を直接にデバイスにダウンロードすると、Quartus II ソフトウェアは **Auto-restart configuration after error** オプションをディセーブルします。

13.  **Auto-restart configuration after error** オプションを使用するようにボード・デザインを計画。




## オン・チップ・デバッグのプランニング

オンチップ・デバッグはデザイン・フローにおけるオプションのステップです。システムおよび設計者によって、最適なデバッグ・ツールも異なります。システム・ボード、QuartusII プロジェクト、およびデザインが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。早期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手間を省くことができます。ただし、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、14 ページの「オン・チップ・デバッグ・ツール」で説明されるようにご希望のデバッグ・ツールを選択し、そして 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 QuartusII ポートフォリオには、以下のイン・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 — 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 ピンに配線します。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 ピンに配線します。
- **SignalTapII** エンベデッド・ロジック・アナライザ — 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 ピンを使用せずに内部信号および I/O 信号の状態をプローブします。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が改善されます。**SignalTapII** 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、ユーザー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。**SignalTapII** エンベデッド・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、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。
- **ロジック・アナライザ・インタフェース** — 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し、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もできます。
- **In-System Memory Content Editor - 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。
- **Virtual JTAG** メガファンクション — システム・レベル・デバッグのためのプロセッサ・ベース・デバッグ・ソリューションおよびソフトウェアによるデバッグ・ツールを含む、独自のシステム・レベルのデバッグ基盤を構築することを可能にします。**SLD\_VIRTUAL\_JTAG** メガファンクションを 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ン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、「*Virtual JTAG (SLD\_VIRTUAL\_JTAG) Megafunction User Guide*」および「Quartus II ハンドブック Volume 3」の「*Section IV. In-System Design Debugging*」の章を参照してください。このセクション概要ではデバッグ・ソリューションの選択について説明します。

14.  オン・チップ・デバッグ機能を活用して内部信号を解析し、高度なデバッグ手法を実行。

### デバッグ・ツールのプランニング・ガイドライン

オン・チップ・デバッグ・ツールを使用する場合は、次のチェックリストで説明するとおり、システム・ボード、QuartusII プロジェクト、およびデザインを開発するときにそのツールをプランニングします。


15.  メモリ要件、ロジック要件、I/O ピン接続、およびボード接続を計画するために、オン・チップ・デバッグ手法を早期段階で選択。
16.  SignalTapII 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、および Virtual JTAG メガファンクションを使用する場合は、デバッグに使用可能な JTAG 接続が付いたようにシステムおよびボードをプランニング。
17.  JTAG デバッグ機能の JTAG ハブ・ロジックを実装するための少量の追加ロジック・リソースをプランニング。
18.  SignalTap II エンベデッド・ロジック・アナライザでデバッグ作業をする場合は、デバイス・メモリ・リソースを保留してシステム動作中にデータをキャプチャ。
19.  SignalProbe またはロジック・アナライザ・インタフェースでデバッグ用の 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すむようにする。
20.  対象ボードで、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。
21.  外部ロジック・アナライザまたはミックスド・シグナル・オシロスコープに必要とするピン・ヘッダまたは Mictor コネクタを組み込む。
22.  デバッグ・ツールをインクリメントに使用してコンパイル時間を減少させるためには、デバッグ・ツールを修正する場合にデザインを再コンパイルする必要がなくなるよう、インクリメンタル・コンパイル機能をオンにすることを確認。
23.  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、これをデザイン・プロセスの一部として HDL コードにインスタンス化。
24.  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する。

## ボード・デザインにおけるピン接続の検討事項

Cyclone IV 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 PCB のデザインに影響を与えます。この項では、次の内容に関する重要なガイドラインについて説明します。

- 16 ページの「デバイスのパワーアップ」
- 17 ページの「電源ピンの接続および電源」
- 19 ページの「コンフィギュレーション・ピンの接続」
- 22 ページの「ボード関連の QuartusII 設定」
- 23 ページの「シグナル・インテグリティの検討事項」
- 24 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ト I/O タイミング解析」

25 ページの「I/O およびクロックのプランニング」ではボード・デザインに影響を与える FPGA の I/O 信号接続について説明します。

 ボード・デザインのガイドラインについて詳しくは、[Board Design Resource Center](#) を参照してください。このリソース・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が提供されています。

## デバイスのパワーアップ

Cyclone IV デバイスは外付け部品なしでホット・ソケットに対応します。Cyclone IV デバイスは、システム / ボードの通常動作を中断または妨害することなく、パワーアップ後のシステム・ボードに取り付けたり、取り外したりすることができます。

25.  パワーアップを考慮してボードを設計する。Cyclone IV 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。
26.  電圧電源のランプがモニタリングになるように設計。

パワーアップまたはパワーダウン前、またはその間に、デバイスに損傷を与えることなく、I/O ピンに信号を入力することができます。Cyclone IV デバイスは、VCCINT ピン、VCCA ピン、および VCCIO ピンのパワーアップまたはパワーダウンを任意のシーケンスでサポートし、システム・レベルのデザインを簡素化します。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 ~ 50 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、Cyclone IV 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、Cyclone IV 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。これらのピンが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・アウトすることは予期されています。

POR 回路は、電源投入後に電源電圧レベルが安定するまでシステム全体をリセット状態に維持します。電源投入後に、コンフィギュレーション・ピンを含む I/O バンクの  $V_{CCINT}$ 、 $V_{CCA}$ 、および  $V_{CCIO}$  がデバイスの POR トリップ・ポイントを上回るまで、デバイスは  $nSTATUS$  を解放しません。電源が切られた後、 $V_{CCINT}$  または  $V_{CCA}$  の電圧が POR トリップ・ポイントを下回る場合に、ブラウン・アウトが発生します。

MSEL ピンの設定に応じて、ファースト POR 時間またはスタンダード POR 時間を選択できます。ファースト・コンフィギュレーション時間の場合に、ファースト POR 時間は、 $3ms < T_{POR} < 9ms$  です。スタンダード POR 時間は、より低いパワー・ランプ・レートの  $50ms < T_{POR} < 200ms$  です。


27.  電源が安定するように POR 時間を設定。

Cyclone IV 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、 $t_{RAMP}$  として規定）内に推奨動作範囲に達した場合は、POR イベントが発生します。Cyclone IV デバイスでは、スタンダード POR 時間の最大電源ランプ時間は 50ms、ファースト POR 時間の最大電源ランプ時間は 3ms、そして最小電源ランプ時間は 50 $\mu$ s です。

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際、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ことが推奨されています。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。

 デバイスの信頼性およびホット・ソケット仕様に準拠することを確保するために、電源を接続する前にボード間の GND を接続することが強く推奨されています。


28.  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、電圧レギュレータおよびグランド接続を設計。


 詳細は、「Cyclone IV デバイス・ハンドブック Volume 1」の「*Power Requirements for Cyclone IV Devices*」の章を参照してください。

## 電源ピンの接続および電源


システムに必要な電源、および電源を共用可能な電圧入力を確認するために、電源ピン接続に関するガイドラインを参照してください。Cyclone IV GX のコア電圧  $V_{CCINT}$  は 1.2V で、Cyclone IV E のコア電圧は 1.2V または 1.0V です。

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。

 Cyclone IV 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、「Cyclone IV デバイス・ハンドブック Volume 1」の「*Power Requirements for Cyclone IV Devices*」の章を参照してください。

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。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 ピンについては、28 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。

 電源の種類および電源の共用または分離について詳しくは、「*Cyclone IV Device Family Pin Connection Guidelines*」を参照してください。


### デカップリング・コンデンサ

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。

高周波デカップリングを供給するために、Cyclone IV デバイスは、オン・パッケージおよびオン・ダイのデカップリング・コンデンサを内蔵しています。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。

アルテラは、ボード・レベル PDN をグラフィカルに最適化できる使いやすい PDN (電源分配ネットワーク) デザイン・ツールを開発しました。ボード・レベル PDN の目的は電源を分配し、電圧レギュレータ・モジュール (VRM) から FPGA 電源に電流をリターンし、そして最適なトランシーバ・シグナル・インテグリティおよび FPGA 性能をサポートします。

電源ごとに、バルク・デカップリング・コンデンサおよびセラミック・デカップリング・コンデンサのネットワークを選択しなければなりません。SPICE シミュレーションは回路のシミュレーションに使用できるため、PDN デザイン・ツールは最適なコストと性能トレードオフを実現するデカップリング・コンデンサの数を決定する高速、正確かつインタラクティブな方法を提供しています。

 PDN デザインおよび最適化プロセスについて詳しくは、「*Power Delivery Network (PDN) Tool User Guide*」を参照してください。また、PDN ツールもダウンロードできます。

29.  PDN ツールを使用して電源分配ネットリストおよびデカップリング・コンデンサをプランニング。

## 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、34 ページの「クロックおよび PLL の選択」および 36 ページの「PLL 機能のガイドライン」を参照してください。PLL はデジタル・デバイスに組み込まれたアナログ・コンポーネントを搭載しているため、以下のチェック・リストに、PLL を使用する電源システムの設計およびジッタの低減についての検討事項を示します。


- 30.  ノイズを低減するために、デザインですべての PLL を使用しない場合でも、すべての PLL 電源ピンを接続 ( $V_{CCA}$  を 2.5V に、 $V_{CCD\_PLL}$  を 1.2 V または 1.0 V に)。
- 31.  電源から各 PLL 電源ピンに厚い配線パターン (最低 20 ミル) を走らせる。
- 32.  すべての PLL デジタル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。
- 33.  フェライト・ビーズを使用して、PLL 電源をデジタル電源から絶縁。

 PLL 電源に関するボード・デザイン・ガイドラインについて詳しくは、Board Design Resource Center での「General Board Design Considerations/Guidelines」の項を参照してください。

## 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、異なるプルアップまたは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・ピンを正しく接続しなければなりません。この項では、コンフィギュレーション・ピン接続の一般的な問題に対処するためのガイドラインについて説明します。

- 34.  すべてのコンフィギュレーション・ピンおよびプルアップ / プルダウン抵抗はコンフィギュレーション手法に対して正しく設定されることを確認。

 専用コンフィギュレーション・ピンと兼用コンフィギュレーション・ピンの一覧、およびその機能と接続のガイドラインの説明については、「Cyclone IV デバイス・ハンドブック Volume 1」の「*Configuration and Remote System Upgrades in Cyclone IV Devices*」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 トレースまたは DCLK トレース、またはその両方は、オーバーシュート、アンダーシュート、またはリングングのないクリーンな信号を生成するようにしてなければなりません。

- 35.  DCLK および TCLK コンフィギュレーション・ピンを、ノイズが発生しないように設計。

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。オーバーシュート、アンダーシュート、リングング、または TCK 信号上の他のノイズはコンフィギュレーションに影響を与える可能性があります。DCLK 信号にノイズが多い場合、AP、AS、PS、または FPP コンフィギュレーションが影響を受け、CRC エラーが発生することがあります。デバイスのチェインでは、チェイン内の TCK ピンまたは DCLK ピンのノイズによって、チェイン全体の 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。



-  チェイン内のデバイスの接続について詳しくは、「Cyclone IV デバイス・ハンドブック Volume 1」の「*Configuration and Remote System Upgrades in Cyclone IV Devices*」の章を参照してください。

### JTAG ピン

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、JTAG インタフェースを使用しない時は、JTAG 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。

36.  JTAG ピンが未使用の場合は、安定した電圧レベルに接続。

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 Cyclone IV デバイスは、4 本の専用ピン (TDI、TDO、TMS、および TCK) 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て、TDI ピンおよび TMS 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。すべての JTAG ピンは、これらのピンが存在する I/O バンクの V<sub>CCIO</sub> 電源で駆動されます。

37.  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。

チェイン内に複数のデバイスがある場合、1 つのデバイスの TDO ピンをチェイ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、ユーザー・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。


38.  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 ピンを抵抗によって 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する。
39.  TMS を抵抗を介して High にプルアップ。

### 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。

40.  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。

電圧が異なるデバイスを含む JTAG チェインでは、電圧の高いデバイスは、電圧が同じかそれ以下である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、チェインの末端に 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・シフタをチェインに追加する必要があります。

-  チェイ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ついて詳しくは、「Cyclone IV デバイス・ハンドブック Volume 1」の「*JTAG Boundary Scan Testing for Cyclone IV Devices*」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 JTAG クロックであり、また最高速の JTAG 信号であるためです。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ります。そのため、信号をコネクタでバッファすることが推奨されています。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ある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。ただし、信号劣化の防止は、ボード・レイアウト、負荷、ボード上のコネクタ、ジャンパ、およびスイッチに依存します。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追加されると、チェーンにバッファが追加される可能性が高くなります。

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。

- 41.  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。
- 42.  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。

### 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、Cyclone IV デバイスの MSEL ピンを High または Low にドライブします。MSEL ピンの選択にかかわらず、JTAG コンフィギュレーションは常に使用できます。MSEL ピンは、 $V_{CCINT}$  電源で駆動されます。MSEL ピンには、常にアクティブな  $9K\Omega$  内部プルダウン抵抗があります。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、MSEL ピンを  $V_{CCA}$  および GND にプルアップ抵抗またはプルダウン抵抗なしで接続してください。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。

- 43.  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。テストまたはデバッグ作業中に各コンフィギュレーション・モードに切り換えす柔軟性を確保するために、各ピンを  $0\Omega$  抵抗で  $V_{CCA}$  または GND に接続するようにボードをセットアップ。

### その他のコンフィギュレーション・ピン

nCE 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。

- 44.  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、nCE チップ・イネーブルを Low に保持。

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 QuartusII 設定

QuartusII ソフトウェアは、ボード設計時に考慮しなければならない FPGA I/O ピンに対するオプションを提供します。Quartus II プロジェクトの作成時にこれらのオプションが正しく設定されることを確認し、ボード・デザインの際に機能性をプランニングします。

### デバイス・ワイド出力イネーブル・ピン

Cyclone IV デバイスは、ユーザーがデバイスの 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・ピン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・ワイドの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、QuartusII ソフトウェアにおいて、**Device & Pin Options** ダイアログ・ボックスの **General** タブでの **Enable device-wide output enable (DEV\_OE)** をオンにします。

45.  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。

### 未使用ピン

ボード・デザインを柔軟に行うために、QuartusII ソフトウェアで未使用ピンの状態を以下のいずれかに指定します。

- 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
- ウィーク・プルアップでトライ・ステートになる入力



デバイスの出力ドライバが損傷することがあるため、出力がグラウンドをドライブする未使用ピンを  $V_{CC}$  または別の信号ソースに接続してはいけません。

46.  未使用 I/O ピンの予約状態を指定。

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。未使用ピンの予約状態を指定した場合、それらのピンをボード上のほかのデバイスに接続してはいけません。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** オプションに許容される 5 つの状態の 1 つを選択するか、または Pin Planner で特定のピンに **Reserve Pin** アサインメントを適用します。

47.  QuartusII ソフトウェアで生成された .pin ファイル (Pin-Put 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。

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 .pin ファイルを生成します。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  $V_{CC}$ 、グラウンド、または別の信号ソースに接続されると、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。

### 電圧リファレンス・ピン


$V_{REF}$  ピンの電圧変動は、入力のスレッシュホールド感度に影響を及ぼすことがあります。

48.  ノイズが発生しないように  $V_{REF}$  ピンを設計。

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、27 ページの「I/O 機能およびピン接続」を参照してください。

### 同時スイッチング・ノイズ (SSN)

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。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、Board Design Resource Center の PCB ガイドラインを参照してください。

例としては、以下の項目を参照。

49.  ボード層上のデバイスに近い大きなバス信号を分割してクロストークを低減。  
50.  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。配線パターン幅の 2 ~ 3 倍の間隔を使用。

### 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 (double data rate) メモリ・システムを作成するためには、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。

Cyclone IV の直列 On-Chip Termination( $R_S$  OCT) 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。その代替りとして、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。

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。

On-Chip Termination (OCT) の機能および制限について詳しくは、27 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。


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。

 このシミュレーション・フローについて詳しくは、「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ことが推奨されています。

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。

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 オプションがオンになると、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 バッファ、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 解析用のボード・トレース・モデルについて詳しくは、「Quartus II ハンドブック Volume 2」の「I/O Management」の章を参照してください。

## I/O およびクロックのプランニング

Cyclone IV デバイスの多数の I/O ピンおよび高度なクロック管理とトランシーバ機能では、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使用率を高め、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。

FPGA の I/O 接続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、これらの接続をデザイン・サイクルの早期段階で計画してください。

この項では、以下の内容について説明します。

- 25 ページの「FPGA ピン・アサインメントの作成」
- 26 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」
- 27 ページの「I/O 機能およびピン接続」
- 34 ページの「クロックおよび PLL の選択」
- 36 ページの「PLL 機能のガイドライン」
- 37 ページの「クロック・コントロール・ブロック」
- 37 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 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、 $V_{REF}$  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、6 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。

54.  Quartus II Pin Planner を使用してピン・アサインメントを作成。

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 **QuartusII** 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。

QuartusII 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。

55.  QuartusII Fitter メッセージとレポートをピン・アサインメントのサイン・オフに使用。



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必要があります。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。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が確認できます。

56. □ QuartusII 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。

 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、「QuartusII ハンドブック Volume 2」の「*I/O Management*」の章を参照してください。QuartusII 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、「QuartusII 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・デザインが変更が生じないようにすることが非常に重要です。

QuartusII Pin Planner により、25 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。QuartusII 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、デザインの市場投入に要する時間が短縮されます。ソース・コードを設計する前に、QuartusII 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。

57. □ I/O Assignment Analysis を使用して、**Create Top-Level Design File** コマンドを使用して、デザインの完成前に I/O アサインメントをチェック。

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 とインタフェースで接続し、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。ピン配置ルールに影響を与えることができる PLL、LVDS、およびトランシーバ・ブロックを入力します。


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て **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。

前項で説明しているように、プランニングが完了したら、ピン暫定位置に関する情報を PCB 設計者に渡すことができます。デザインが完成したら、**QuartusII 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 アサインメントおよび解析について詳しくは、「**QuartusII ハンドブック Volume 2**」の「**I/O Management**」の章を参照してください。

## I/O 機能およびピン接続

Cyclone IV 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、広帯域幅および普及しているインタフェースをサポート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。

 ピン接続について詳しくは、「**Cyclone IV Device Family Pin Connection Guidelines**」を参照してください。

### I/O 信号の種類

Cyclone IV 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、電圧リファレンス形式の信号は、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。ただし、リファレンス電圧源  $V_{TT}$  に終端コンポーネントを追加する必要があります。

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、差動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。

Cyclone IV 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・アサインメントを作成します。差動 I/O 規格を指定すると、QuartusII 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。

- 58.  システム要件に応じて信号の種類をプランニング。
- 59.  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。

### 選択可能な I/O 規格と柔軟性の高い I/O バンク

Cyclone IV デバイスの I/O ピンは I/O バンクにまとめられ、各バンクには独立したパワー・バスがあります。デバイスの各 I/O ピンは 1 個の I/O バンクに関連付けられます。Cyclone IV E デバイスは 8 個の I/O バンクを備えています。Cyclone IV GX デバイスは最大 10 個の I/O バンクおよび 1 個のコンフィギュレーション・バンクを備えています。Cyclone IV GX デバイスのコンフィギュレーション・バンクには、セカンダリ・コンフィギュレーション・プログラミング機能を備えた I/O ピンが 3 本あります。コンフィギュレーション・モードに使用されていない場合は、これらのピンを通常のユーザ I/O ピンとして使用できます。

- 60.  各 I/O ピンに最適な信号タイプおよび I/O 規格を選択。
- 61.  ターゲットの I/O バンクで適切な I/O 規格がサポートされていることを確認。


カラム I/O バンクでのみサポートされる HSTL-12 Class II 以外、すべてのシングル・エンド I/O 規格がサポートされます。すべての差動 I/O 規格は、上側、下側および右側の I/O バンクでサポートされます。カラム I/O バンクのみでサポートされる HSTL-12 Class II は例外です。Cyclone IV GX デバイスの左側全体には、高速トランシーバ・インタフェース・アプリケーション用の専用高速トランシーバがあります。

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、34 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。

- 62.  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。
- 63.  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。
- 64.  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  $V_{REF}$  電圧レベルを使用するようになっていることを確認。

ボードは、バンク内の各  $V_{CCIO}$  ピンに給電するために、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  $V_{CCIO}$  ピンで給電され、他の I/O バンクの  $V_{CCIO}$  電源からは独立しています。1 つの I/O バンクが、 $V_{CCIO}$  電源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。ただし、電圧リファレンス入力に対してはいくつかの例外があります。


電圧リファレンス規格は、任意の数のシングル・エンド規格または差動規格を使用して I/O バンクでサポートされますが、これらの規格が同じ  $V_{REF}$  および  $V_{CCIO}$  値を使用している場合に限られます。例えば、Cyclone IV デバイスに SSTL-2 と SSTL-18 の両方を実装する場合、これらの規格を使用する I/O ピンは（異なる  $V_{REF}$  値を必要とするため）互いに異なるバンクに存在しなければなりません。ただし、 $V_{CCIO}$  を 2.5V に設定し、 $V_{REF}$  を 1.25V に設定した場合、同じ I/O バンクで SSTL-2 と 2.5V LVCMOS をサポートできます。


  $V_{REF}$  ピンが通常の I/O として使用されているときは、通常のユーザー I/O ピンよりもピン・キャパシタンスが高くなります。これは、ピンが入力および出力として使用されている場合は影響を及ぼします。

  $V_{REF}$  ピンのキャパシタンスについては、「Cyclone IV デバイス・ハンドブック Volume 3」の「*Cyclone IV Device Data Sheet*」の章を参照してください。 $V_{REF}$  グループの識別については、「Cyclone IV デバイス・ファミリー用のピンアウト・ファイル」または Quartus II Pin Planner ツールを参照してください。

65.  LVDS およびトランシーバ機能の I/O バンク・サポートをチェック。

異なる I/O バンクは LVDS 信号に対するサポートが異なり、Cyclone IV トランシーバ・バンクは追加のサポートが含まれています。

 LVDS I/O 規格で使用可能なチャンネル数については、「Cyclone IV デバイス・ハンドブック Volume 1」の「*I/O Feature in Cyclone IV Devices*」の章を参照してください。トランシーバ・バンクに関する機能については、「Cyclone IV デバイス・ハンドブック Volume 2」の「*Cyclone IV Transceivers Architecture*」の章を参照してください。

 I/O については、「Cyclone IV デバイス・ハンドブック Volume 2」の「*I/O Feature in Cyclone IV Devices*」の章を参照してください。各 I/O バンクの位置と各バンクでサポートする機能を示す Cyclone IV デバイスの I/O バンクの図を参照してください。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。Cyclone IV デバイスを使用して LVTTTL および LVCMOS 入力をデザインするときには、I/O バンクのガイドラインの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は、「Cyclone IV デバイス・ハンドブック Volume 3」の「*Cyclone IV Device Data Sheet*」の章を参照してください。

### 差動 I/O ピンの配置ガイドライン

差動 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。コンパイル後に、Quartus II ソフトウェアは、コンパイル時に Quartus II ソフトウェアがチェックしたガイドラインをまとめるフィッター・レポートを生成します。

66.  LVDS I/O 付近のピン配置に対して、ガイドラインに従うことを注意してください。

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  $V_{REF}$  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、 $V_{REF}$  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。

## メモリ・インタフェース

Cyclone IV デバイスは、小型モジュール・ベース I/O バンク構造のバンクにより、幅広い外部メモリ・インタフェースに迅速かつ容易に適合する効率的なアーキテクチャを提供します。Cyclone IV デバイスは、DDR2 SDRAM、DDR SDRAM、QDR II SRAM などの既存および新しい DDR メモリ規格をサポートします。Cyclone IV デバイスは、上側、下側および右側 I/O バンクにおいて外部メモリをサポートします。

67. □ 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や制約に従う。

セルフ・キャリブレート・メガファンクション (ALTMEMPHY) は、Cyclone IV デバイスの I/O 構造および Quartus II TimeQuest タイミング・アナライザ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でき、そしてシステムに最も確実な動作周波数のある最適な物理インタフェース (PHY) を設定するのに役立ちます。アルテラのメモリ・コントローラ MegaCore ファンクションを使用する場合、ALTMEMPHY メガファンクションがインスタンス化されます。

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。

68. □ メモリ・インタフェースに専用 DQ/DQS ピンおよび DQ グループを使用。

Cyclone IV デバイスでは、データ・ストロブ DQS ピンおよびデータ DQ ピンの位置は固定されています。デバイス・ピンアウトをデザインする前に、これらまたはほかのメモリ関連の信号の接続について詳細や重要な制約を把握するために、メモリ・インタフェース・ガイドラインを参照してください。

異なるメモリ規格がサポートされている最大クロック・レート、およびピン配置における制約を含む、Cyclone IV デバイスの外部メモリ・デバイスへの接続について詳しくは、「Cyclone IV デバイス・ハンドブック Volume 1」の「*External Memory Interfaces in Cyclone IV Devices*」の章を参照してください。関連情報については、アルテラ・ウェブサイトの外部メモリ・ソリューション・センターを参照してください。

Cyclone IV PLL について詳しくは、「Cyclone IV デバイス・ハンドブック」の「*Clock Networks and PLLs in Cyclone IV Devices*」を参照してください。ALTMEMPHY メガファンクションについて詳しくは、「外部 DDR メモリ PHY インタフェース・メガファンクション・ユーザーガイド (ALTMEMPHY)」を参照してください。



## 兼用ピンおよび特殊ピンの接続

Cyclone IV 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、GCLK 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 LE ベースのレジスタを使用します。

デバイス・ワイド・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、デザインの I/O として使用できます。詳細は、22 ページの「デバイス・ワイド出力イネーブル・ピン」および 40 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。

- 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。

## Cyclone IV デバイスの I/O 機能

Cyclone IV デバイスの IOE は、I/O ピンにさまざまなプログラム可能な機能を提供しています。これらの機能は I/O 使用の柔軟性を向上し、プルアップ抵抗やダイオードなど、外部ディスクリート・コンポーネントの使用を削減する代替手段を提供します。表 3 に、Cyclone IV デバイスの 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。

表 3. Cyclone IV I/O 機能 ( 1 / 4 )

機能	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。VCCIO ピンは要求される出力のレベルに応じて、1.5V、1.8V、2.5V、3.0V、または 3.3V 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。	MultiVolt I/O サポートの概要、サポートされる I/O 規格の一覧、および $V_{CCIO}$ 、 $V_{REF}$ の出力とボード終端電圧 ( $V_{TT}$ ) の標準値については、「Cyclone IV デバイス・ハンドブック Volume 1」の「 <i>I/O Features in Cyclone IV Devices</i> 」の章を参照してください。
3.3V I/O インタフェース	Cyclone IV I/O バッファは 3.3 V I/O 規格をシステム内のトランスミッタまたはレシーバとしてサポートします。Cyclone IV $V_{CCIO}$ 電圧が 3.3V によって駆動されると、出力 High 電圧 ( $V_{OH}$ )、出力 Low 電圧 ( $V_{OL}$ )、入力 High 電圧 ( $V_{IH}$ )、および入力 Low 電圧 ( $V_{IL}$ ) のレベルは、3.3V I/O 規格の仕様を満たします。	デバイスの信頼性と適切な動作を実現するには、3.3V I/O システムとインタフェースで接続するときに、Cyclone IV デバイスの絶対最大定格に違反しないように確認する必要があります。 アルテラでは、オーバーシュート電圧およびアンダーシュート電圧がガイドライン範囲内にあることを確認するように、IBIS シミュレーションを実行することを推奨しています。詳細は、「Cyclone IV デバイス・ハンドブック Volume 1」の「 <i>I/O Features in Cyclone IV Devices</i> 」の章を参照してください。



表 3. Cyclone IV I/O 機能 ( 2 / 4 )

機能	使用方法	ガイドラインおよび詳細情報
プログラマブル出力電流強度	<p>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。これにより、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。</p>	<p>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。</p> <p>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、「Cyclone IV デバイス・ハンドブック Volume 1」の「<i>I/O Features in Cyclone IV Devices</i>」の章を参照してください。</p>
プログラマブル・スルーレート・コントロール	<p>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、高性能システムに対応した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。スルーレートを調整して、SSN を低減することができます。</p>	<p>低速スルーレートをを使用する場合は、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。</p> <p>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。</p>
プログラマブル IOE 遅延	<p>プログラマブル IOE 遅延はゼロ・ホールド・タイムの維持、セットアップ・タイムの最小化、clock-to-output 時間の向上を実現できます。遅延をデスクュー回路として使用して、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ことができます。</p>	<p>Quartus II Compiler は、このような遅延をプログラムして、0 ns のホールド・タイムを維持しながら、セットアップ・タイムを自動的に最小限に抑えます。プログラマブル遅延は、出力レジスタのレジスタからピンまでの遅延を増加させることもできます。</p> <p>遅延の仕様について詳しくは、「Cyclone IV デバイス・ハンドブック Volume 3」の「<i>Cyclone IV Device Data Sheet</i>」の章を参照してください。</p> <p>入力および出力ピンの遅延の設定方法について詳しくは、「Quartus II ハンドブック Volume 2」の「<i>Area and Timing Optimization</i>」の章を参照してください。</p>

表 3. Cyclone IV I/O 機能 ( 3 / 4 )

機能	使用方法	ガイドラインおよび詳細情報
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。	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。
バス・ホールド	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ョルド電圧から離します。	バス・ホールド機能がイネーブルの場合、プログラマブル・プルアップ・オプションは使用できません。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。  この抵抗を通してドライブされる具体的な保持電流、および次にドライブされる各 $V_{CCIO}$ 電圧レベルの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詳しくは、「Cyclone IV デバイス・ハンドブック Volume 3」の「 <i>Cyclone IV Device Data Sheet</i> 」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	プルアップ抵抗は、ユーザー・モードのときに I/O を $V_{CCIO}$ 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の場合、バス・ホールド機能は使用できません。
PCI クランプ・ダイオード	3.3V LVTTTL、3.3V LVCMOS、3.0V LVTTTL、3.0V LVCMOS、2.5V LVTTTL/LVCMOS、PCI、および PCI-X の各 I/O 規格での過剰なオーバershoot 電圧からピンを保護するために使用できます。	入力 I/O 規格が 3.3V LVTTTL、3.3V LVCMOS、3.0V LVTTTL、3.0V LVCMOS、2.5V LVTTTL/LVCMOS、PCI、または PCI-X の場合、PCI クランプ・ダイオードは Quartus II ソフトウェアによってデフォルトでイネーブルされます。

表 3. Cyclone IV I/O 機能 ( 4 / 4 )

機能	使用方法	ガイドラインおよび詳細情報
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。OCT は、信号品質の維持、ボード・スペースの節約、外部コンポーネント・コストの低減を実現します。Cyclone IV デバイスは、キャリブレーション付きとキャリブレーションなしの両方の $R_S$ OCT をサポートします。Cyclone IV デバイスは、 $25\Omega$ および $50\Omega$ の直列終端値を提供しています。	直列 On-Chip Termination は、すべての I/O バンクでサポートされます。特定の I/O バンクで直列 On-Chip Termination をイネーブルにできるように、 $V_{CCIO}$ と $V_{REF}$ はすべての I/O ピンで互換性がなければなりません。異なる $R_S$ 値をサポートする I/O 規格は、 $V_{CCIO}$ と $V_{REF}$ が競合しない限り、同一 I/O バンクに存在することが可能です。 詳細は、「Cyclone IV デバイス・ハンドブック Volume 1」の「 <i>I/O Features in Cyclone IV Devices</i> 」の章を参照してください。
プログラマブル・プリエンファシスおよび VOD	プリエンファシスは、出力信号の高周波成分の振幅を大きくして、伝送線路における周波数依存減衰を補償するのに役立ちます。	詳細は、「Cyclone IV デバイス・ハンドブック Volume 1」の「 <i>Cyclone IV Device I/O Features</i> 」の章を参照してください。

詳細については、次のチェックリストを検討して、該当するドキュメントを参照してください。

- 70.  I/O インタフェースに役立つ利用可能なデバイス I/O 機能をチェック。例えば、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、プログラマブル・プリエンファシス、および差動出力電圧 ( $V_{OD}$ )。
- 71.  ボード・スペースを節約するように On-Chip Termination (チップ内終端) の使用を考慮。
- 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。

## 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。

Cyclone IV デバイスは、GCLK ネットワークをドライブできる専用クロック・ピン (CLK[15..4]) を最大 12 本備えています。Cyclone IV GX デバイスは、デバイスの各サイドで 4 本の専用クロック・ピンをサポートします。これらのクロック・ピンは、最大 30 の GCLK をドライブできます。Cyclone IV E デバイスには、最大 20 の GCLK をドライブできる専用クロック・ピン (CLK[15..1]) を最大 12 本備えています。Cyclone IV E デバイスは、デバイスの左側で 3 本の専用クロック・ピン、そしてデバイスの上側、右側、および下側で専用クロック・ピンを 4 本サポートします。EP4CE6 および EP4CE10 デバイスは、左側で 3 本の専用クロック・ピンおよび右側で 4 本の専用クロック・ピンのみをサポートします。

Cyclone IV GX デバイスは、GPLL および MPLL の 2 種類の PLL を提供します。Cyclone IV E デバイスは、GPLL のみ提供します。GPLL は、外部メモリ・インターフェイスなど、FPGA ファブリックおよび周辺部品における汎用アプリケーションにのみ使用されます。MPLL はトランシーバ・ブロックのクロッキングに使用されます。トランシーバのクロッキングに使用されない場合に、MPLL は汎用クロッキングに使用可能です。Cyclone IV デバイスは、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する最大 8 個の PLL を備えています。

 各デバイス集積度で使用可能な GCLK、GPLL、および MPLL の数について詳しくは、「Cyclone IV デバイス・ハンドブック Volume 1」の「*Cyclone IV デバイス・ファミリの概要*」の章を参照してください。

73.  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。

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。専用配線ネットワークを使用して、遅延を予測可能にし、ファンアウトの大きい信号のスキューを最小限に抑えます。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。


74.  デバイスの PLL をクロック管理に使用。

特定のクロック入力、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。

- 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・リセット信号およびクリア信号の配線、またはデバイス全体のクロックの配線に適しています。
- IOE と内部ロジックは、GCLK をドライブして、内部で生成される GCLK およびその他の高ファンアウト・コントロール信号（同期クリア、非同期クリアやクロック・イネーブルなど）を生成することもできます。
- PLL は、内部で生成された GCLK によってドライブできません。PLL の入力クロックは、ピンでドライブされる専用 GCLK によって供給する専用クロック入力ピンからでなければなりません。あるいは、クロック・コントロール・ブロックが他の PLL またはピンでドライブされる専用 GCLK によって供給される場合にはクロック・コントロール・ブロックから供給されます。

75.  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。

 これらの機能およびクロック接続について詳しくは、「Cyclone IV デバイス・ハンドブック Volume 1」の「*Clock Networks and PLLs in Cyclone IV Devices*」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、低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。QuartusII 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、またはアサインメントを **Off** に設定して、信号がグローバル配線リソースを使用しないように指定します。

## PLL 機能のガイドライン


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。QuartusII 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。

76.  PLL 機能をイネーブルにし、MegaWizard Plug-In Manager の設定をチェック。

 PLL におけるタイミング制約の設定について詳しくは、[「AN 471: High-Performance FPGA PLL Analysis with TimeQuest」](#) を参照してください。

Cyclone IV 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。すべての Cyclone IV PLL は、いくつかの汎用クロック管理機能をサポートしています。PLL デザインをプランニングするときは、以下の機能を使用します。

 PLL の機能について詳しくは、「Cyclone IV デバイス・ハンドブック」の [「Clock Networks and PLLs in Cyclone IV Devices」](#) の章を参照してください。

 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詳しくは、[「Phase-Locked Loops \(ALTPLL\) 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

Cyclone IV PLL は、次の 5 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。

- ソース・シンクロナス・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
- 確定的レイテンシ補償モード

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。

77.  正しい PLL フィードバック補正モードを選択することを確認。

## 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、あるいはデュアル・クロック・ドメイン・アプリケーションに使用します（例えば、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）。クロックがトグルしていないとき、またはユーザー・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。

## クロック・コントロール・ブロック

Cyclone IV デバイスでは、専用クロック入力ピン、PLL カウンタ出力、兼用クロック I/O ピン、および内部ロジックはすべて、各 GCLK のクロック・コントロール・ブロックに信号を供給できます。また、クロック・コントロール・ブロックからの出力が対応する GCLK に供給されます。GCLK は、クロック・コントロール・ブロック入力が別の PLL または専用クロック入力ピンの出力である場合は、PLL 入力をドライブできます。デバイス周辺の各側に 5 個または 6 個のクロック・コントロール・ブロックがあり、デバイスの集積度によって、1 つの Cyclone IV GX デバイスは最大 30 個までのクロック・コントロール・ブロックを備えています。各 Cyclone IV E デバイスは最大 20 個のクロック・コントロール・ブロックを備えています。

コントロール・ブロックは以下の 2 つの機能を備えています。

- ダイナミックな GCLK のクロック・ソースの選択 (DPCLK または CDPCLK および内部ロジック入力には適用しません)
- グローバル・クロック・ネットワークのパワーダウン (ダイナミック・イネーブル/ディセーブル)

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。Cyclone IV 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。QuartusII ソフトウェアの設定を使用して、クロック・ソースをスタティックに選択したり、マルチプレクサ選択入力をドライブする内部ロジックを使用して、クロック・ソースをダイナミックに選択することができます。

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。

-  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、「*Clock Control Block (ALTCLKCTRL) Megafunction User Guide*」を参照してください。

## 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。I/O 接続およびクロック接続をプランニングする際、次のチェック・リストを検討してください。

79.  デザインを解析して、SSN 問題が発生する可能性があるかどうか調べる。
80.  可能であれば同時に電圧を切り換えるピン数を減らす。
81.  スイッチング速度が高い I/O には、差動 I/O 規格と低電圧規格を使用。



- 82.  スイッチング速度が高い 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。
- 83.  各バンク内の同時スイッチング出力ピン数を減らす。可能であれば、出力ピンを複数のバンクに分散させる。
- 84.  スイッチング I/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（バンク使用率が 100% を十分に下回っている場合）。
- 85.  同時にスイッチングするピンを、SSN に敏感な入力ピンから分離。
- 86.  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、大きなスイッチング・バスから離して、グランド信号の近くに配置。
- 87.  PLL 電源ピンから 1 本または 2 本分離した I/O ピンを、スイッチング速度が速い、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。
- 88.  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。

使用可能な機能について詳しくは、31 ページの「Cyclone IV デバイスの I/O 機能」を参照してください。

## デザイン・エントリ

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、ロジック使用率、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。また、メガファンクションおよび SOPC Builder は FPGA のデザインにも役立ちます。また、デザインのプランや作成のほかに、階層またはチーム・ベースに基づくデザインを計画して、デザインの生産率を向上させます。

## デザインの推奨事項

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。すべてのレジスタのタイミング要件が満たされていると、同期デザインはすべてのプロセス、電圧、および温度（PVT）条件で、予測可能かつ信頼性の高い方法で動作します。同期デザインでは、簡単に異なるデバイス・ファミリーやスピード・グレードをターゲットにすることができます。

- 89.  同期デザイン手法を使用。クロック信号に注意。

非同期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。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、逡倍、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。詳細は、19

ページの「PLL ボード・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、ディ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。

90. □ Quartus II Design Assistant を使用して、デザインの信頼性をチェックする。

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「lint」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。


 デザインの推奨事項および Design Assistant 機能の使用については、「Quartus II ハンドブック Volume 1」の「*Design Recommendations 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。複数の非同期クロックを使用したデザインに対して適切な解析を実行するには、「*Synthesis and Scripting Techniques for Designing Multi-Asynchronous Clock Designs*」を参照してください。

## 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できます。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。3 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。

QuartusII 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ユーザー・インタフェースを備えています。アルテラでは、すべてのポートとパラメータを正しく設定するために、MegaWizard Plug-In Manager 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ことを推奨しています。

91. □ MegaWizard Plug-In Manager でメガファンクションを使用。

 メガファンクションについては詳しくは、QuartusII Help またはアルテラ・ウェブサイトのオンライン資料：ユーザーガイドページのメガファンクション・ユーザー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。最適な合成結果が得られるように、アルテラ推奨のコーディング・スタイルを使用してください。メモリ・ファンクションおよびデジタル・システム処理 (DSP) ファンクションを設計する場合、デバイスのアーキテクチャを理解すれば、専用のロジック・ブロックのサイズやコンフィギュレーションを活用できます。

92. □ メモリ・ブロックや DSP 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。

 具体的な 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「*Recommended HDL Coding Styles*」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。Quartus II ソフトウェアでは、テキスト・エディタの右クリック・メニューから選択できる Language Templates の HDL の例を使用できます。


## レジスタ・パワーアップ・レベルとコントロール信号

Cyclone IV 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。

次の項では、合成により NOT gate push back と呼ばれる最適化が実行された場合について説明します。NOT gate push back では、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 High にプリセットされているように動作します。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 QuartusII ソフトウェアの **Enable device-wide reset (DEV\_CLRn)** をオンにします。

93. □ 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。

Cyclone IV デバイスの各ロジック・アレイ・ブロック (LAB) には、各 LE 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を内蔵されています。このコントロール信号には、2 本のクロック、2 本のクロック・イネーブル、2 本の非同期クリア、1 本の同期クリア、および 1 本の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。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。

 LE および LAB アーキテクチャについては、「Cyclone IV デバイス・ハンドブック Volume 1」の「*Logic Elements and Logic Array Blocks in Cyclone IV Devices*」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が使用できない場合は、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、かつ同期してディ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。同期ディ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。同期ディアサーションは、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て、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。



リセット・デザインに関する詳しい情報については、業界紙を参照してください。リセット・アーキテクチャの適切な解析を行うには、*「Asynchronous & Synchronous Reset Design Techniques - Part Deux」* を参照してください。

デフォルトでは、QuartusII の合成は **Power-Up Dont 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する場合は、合成ツールで使用可能な合成オプションを使用します。QuartusII 統合合成では、**Assignment Editor**、**Tcl** アサインメント、またはソース・コードに `altera_attribute` アサインメントを作成することにより、**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、QuartusII 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、「QuartusII 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場合が多い）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ります。

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。


94. □ 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。

リセット信号とプリセット信号を同じレジスタに実装するために、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。

## 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 は、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るオプションのツールです。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ต์・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 NiosII 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、ユーザーは 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な双方向システム・インタコネクต์です。

-  Avalon インタフェースについて詳しくは、[「Avalon Interface Specifications」](#) マニュアルを参照してください。

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。

95. □ 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。




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

## 階層およびチーム・ベース・デザインのプランニング

QuartusII 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、ユーザー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。デザインの分割はオプションですが、これらの利点は大規模な Cyclone IV デザインにとって重要です。

QuartusII 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・フローをプランニングします。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・フローの使用について詳しくは、「Quartus II ハンドブック Volume 1」の「*Quartus II Incremental Compilation for Hierarchical and Team-Based Design*」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ニング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 境界をレジスタで受けることにより、クリティカル・タイミング・パスを個別に最適化可能な 1 つのパーティション内に維持できます。デザイン・パーティションを指定する場合は、**Incremental Compilation Advisor** 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。

これにより、ソフトウェアが別々で各ブロックの変更を自動的に検出できます。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 ファイル (.vqm) または EDIF (.edf) ネットリスト・ファイルを作成します。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。デザインに階層を使用することにより、パーティション作成時により多くの柔軟性を提供します。デザイン・ロジックを階層ツリーの「葉」として扱います；すなわち、階層のトップ・レベルではロジックの数を最小にして、ロジックは低いレベルのデザイン・ブロックに含まれるべきです。

- 96.  インクリメンタル・コンパイル実行のために、ソース・コードのセットアップおよびデザインの分割に関する推奨事項に従う；デザイン・フローの早期段階でプランニング。
- 97.  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・バジレットの作成とリソース・バランシングを実行。



 デザイン・パーティションの作成について詳しくは、「Quartus II ハンドブック Volume 1」の「*Best Practices for Incremental Compilation Partitions and Floorplan Assignments*」を参照してください。

### 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必要があります。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。QuartusII 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使用済みであるデバイス・エリアで、Fitter にデザインの一部分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。トップダウン・フローにおけるタイミング・クリティカルなパーティションに対しては、フロアプラン・アサインメントが推奨されています。

98. □ 必要に応じて、インクリメンタル・コンパイル用の各パーティションに対してデザイン・フロアプランを作成。

QuartusII 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・フロアプラン作成について詳しくは、「Quartus II ハンドブック Volume 1」の「*Best Practices for Incremental Compilation Partitions and Floorplan Assignments*」の章を参照してください。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、「*Analyzing and Optimizing the Design Floorplan*」の章を参照してください。

## デザインの実装、解析、最適化、および検証

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・アーキテクチャの要素にマップします。次に、**QuartusII 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・要素を実装します。必要に応じて、**QuartusII** ソフトウェアを使用して、デザインのリソース使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。

### 合成ツールの選択


**QuartusII** 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 HDL** 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 **QuartusII** 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。

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。

99. □ 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。

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するには、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します。正確なタイミング解析およびロジック使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。

合成ツールによっては、**QuartusII** 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します。

 サポートされている合成ツールについて詳しくは、「**QuartusII** ハンドブック Volume 1」の **[Synthesis]** セクションの該当する章を参照してください。**QuartusII** リリース・ノートには、**QuartusII** 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。

### デバイスのリソース使用率レポート

**Quartus II** ソフトウェアでコンパイルした後、デバイスのリソース使用率情報をレビューし、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。コンパイルの結果に no-fit エラーが生じた場合、リソース使用率情報は、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ために、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使用率を示すコンパイル・レポートの **Flow Summary** の項を参照してください。

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、ロジック使用率情報を分析し、ロジック・エレメントの使用率を表示し、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、QuartusII 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。このレポートを使用して、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使用率を見積り、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します。

100. □ コンパイル後にリソースの利用および最適化に関するレポートを確認。

## QuartusII 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。QuartusII のユーザー・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ます。ユーザーは、あるメッセージがユーザー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。

101. □ すべての Quartus II メッセージ、特にワーニング・メッセージまたはエラー・メッセージを確認。



メッセージおよびメッセージの抑制について詳しくは、「QuartusII 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・シンセシス・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。QuartusII 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するため、正しいスピード・グレードを求めるためにタイミング解析を実行する必要があります。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、制約、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。

QuartusII 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。(QuartusII 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。Cyclone IV デザインに 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。

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・レポートを生成します。

- 102.  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。
- 103.  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。
- 104.  Cyclone IV デバイスにデータが提供される時に、入力 I/O タイムに違反していないことを確認。

 タイミング解析について詳しくは、「QuartusII ハンドブック Volume3」の *「Quartus II TimeQuest Timing Analyzer」* および *「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、Cyclone IV 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。

- 105.  **Settings** ダイアログ・ボックスの **Fitter Settings** ページで、**Optimize multi-corner timing** をオンにする。

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。

- 106.  **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 multi-corner timing analysis** をオンにするか、または TimeQuest に対して `--multicorner` コマンド・ライン・オプションを使用。


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ー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。

- 107.  create\_clock、create\_generated\_clock を使用してデザイン内のすべてのクロックの周波数と関係を指定。
- 108.  set\_input\_delay、set\_output\_delay を使用して、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。
- 109.  derive\_pll\_clocks を使用して 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平行変換係数に対して、マルチサイクル関係を指定。
- 110.  derive\_clock\_uncertainty を使用し、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。
- 111.  check\_timing を使用して、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。

## 面積およびタイミングの最適化

この項では、面積（またはリソース使用率）およびタイミング性能の最適化を支援するために QuartusII ソフトウェアで提供される機能のいくつかを説明します。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、「QuartusII ハンドブック Volume 2」の [「Area and Timing Optimization」](#) 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ります。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%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。

- 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。

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 QoR 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。



- 詳細は、「QuartusII 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) ユーティリティは、デザインに最適な QuartusII ソフトウェア設定の集合を検索するプロセスを自動化にします。**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。消費電力の最適化に関心がある場合は、52 ページの「消費電力の最適化」を参照してください。

- 詳細は、「QuartusII 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選択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。

- 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 QuartusII 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。

## 性能の維持および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用して、ロジックをデザインの変更されていない部分に維持し、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。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。

- 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。

ガイドラインおよび参考文献については、43 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。

- 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。

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、QuartusII 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。QuartusII 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。デフォルトで、このオプションは **Use all available processors** に設定されます。パラレル・コンパイルはデフォルトでオンになります。

- 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。



Compilation Time Advisor は、デザインのコンパイル時間を短縮するための設定選択について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。

 詳細は、「QuartusII ハンドブック Volume 2」の「*Area and Timing Optimization*」の章を参照してください。


## シミュレーション

QuartusII 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、**TimeQuest** 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、トップレベル・デザインに対してデザイン・フローの最後にタイミング・シミュレーションを実行します。

アルテラでは、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る **ModelSim®-Altera** シミュレータが用意されています。また、**QuartusII EDA Netlist Writer** は、**Synopsys** 社の **VCS** や **Cadence** 社の **NC-Sim**、**Aldec** の **Active-HDL** 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。

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。

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、**QuartusII** ソフトウェア・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。[QuartusII ソフトウェアのリリース・ノート](#)には、**QuartusII** 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 **QuartusII** 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。テストベンチを生成するには、**Processing** メニューで **Start** をポイントし、**Start Testbench Template Writer** をクリックします。

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

## 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。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。

QuartusII ソフトウェアのリリース・ノートには、QuartusII 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。

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。

120.  フォーマル検証を使用する場合、サポートおよびデザインの制限を確認。

 フォーマル検証のフローについては、「Quartus II ハンドブック Volume 3」の *[Formal Verification]* セクションの該当する章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、7 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、QuartusII 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。

121.  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.  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・アクティビティを生成。

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、またはユーザー定義のデフォルト・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、QuartusII シミュレータまたはサードパーティ・シミュレーション・ツールの **.vcd** 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。

123.  電力解析の正しい動作条件を指定。

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** メニューで、**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は、「QuartusII 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、「QuartusII ハンドブック Volume 3」の *「Quartus II Simulator」* の章を参照してください。

## 消費電力の最適化

Cyclone IV デバイスは、特許を取得しているアーキテクチャの低消費電力技術を駆使して、消費電力を最小に抑えながら高い性能を提供します。

Cyclone IV 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手法およびソフトウェア手法を使用してデザインを最適化します。

QuartusII 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。

- 124.  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。



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、「QuartusII ハンドブック Volume 2」の *「Power Optimization」* の章を参照してください。

### 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、ダイナミック消費電力の大きな部分に関係します。QuartusII 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されると、クロック・ネットワークから信号が供給されるすべてのロジックがオフ状態になり、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、「*Clock Control Block (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。QuartusII 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・ポートにクロック・ゲーティングまたはクロック・イネーブル信号を使用して、メモリ・クロッキング・イベント数を低減することです。リード・イネーブル信号を使用して、必要な場合にのみリード動作が実行されるようにします。例えば、デザインで Read-During-Write 動作が不要な場合は、ライト動作中にリード・イネーブル信号をディアサートして消費電力を削減することができます。QuartusII ソフトウェアは、自動的に未使用メンバー・ブロックを低消費電力モードに切り替えてスタティック消費電力を削減します。

### 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。

LVTTTL 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 などの抵抗終端 I/O 規格を使用します。出力負荷電圧振幅は、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 EPE 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。

### QuartusII の消費電力の最適化のテクニック

QuartusII 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。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。

また、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、デザインのソース・コードを面積に対して最適化すれば消費電力が低下します。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します。

- 
 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、「QuartusII ハンドブック Volume 2」の「*Power Optimization*」の章を参照してください。

### DSE

DSE ユーティリティは、デザインに最適な QuartusII ソフトウェア設定の集合を検索するプロセスを自動化にします。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。さらに、DSE ウィンドウの **Advanced** タブを使用して、**Optimization Goal** を **Optimize for Power** に設定することもできます。

- 
 詳細は、「QuartusII 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

QuartusII 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。

## 改訂履歴

表 4 に、このアプリケーション・ノートの改訂履歴を示します。

表 4. 改訂履歴

日付	バージョン	変更内容
2010 年 2 月	1.1	<ul style="list-style-type: none"> <li>■ 「デバイス・ファミリの種類および高速トランシーバ」の項にタイトルを変更</li> <li>■ 「デバイスの選択」、「I/O ピン数、LVDS チャネルおよびパッケージの種類」、「PLL およびクロック配線」、「デバイス・コンフィギュレーションのプランニング」、「クロックおよび PLL の選択」、および 「クロック・コントロール・ブロック」の項を更新</li> </ul>
2009 年 11 月	1.0	初版

## 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です。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。

### Done N/A デザイン・チェックリスト ( 1 / 7 )

1.   詳細なデザイン仕様を作成し、また、該当する場合はテスト・プランも作成。
2.   ブロック図を用いて、早期にクロック・ドメイン、クロック・リソース、および I/O インタフェースを計画。
3.   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。
4.   IP に **Open Core PlusTethered** 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。
5.   トランシーバ、I/O ピン数、LVDS チャネル、パッケージの種類、ロジック / メモリ / マルチプライアの集積度、PLL、クロック配線、およびスピード・グレードに基づいて、デバイスを選択。
6.   将来の開発およびデバッグ作業のために、予備リソースを保留。
7.   パーティカル・デバイス・マイグレーションの可用性および要件を検討。
8.   ロジック・デザインが完成する前に冷却ソリューションと電源をプランニングするために、アルテラの **Powerplay Early Power Estimator(EPE)** スプレッドシートを使用して消費電力を見積もる。
9.   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ニング。
10.   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。
11.   コンフィギュレーション手法とボードが、データの復元、リモート・システム・アップグレード、SEU (Single Event Upset) の緩和などの必要な機能をサポートしていることを確認。
12.   オプションの CLKUSR および INIT\_DONE コンフィギュレーション・ピンをサポートするようにボード・デザインを計画。
13.   **Auto-restart configuration after error** オプションを使用するようにボード・デザインを計画。
14.   オン・チップ・デバッグ機能を活用して内部信号を解析し、高度なデバッグ手法を実行。
15.   メモリ要件、ロジック要件、I/O ピン接続、およびボード接続を計画するために、オン・チップ・デバッグ手法を早期段階で選択。
16.   **SignalTapII** 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**、および **Virtual JTAG** メガファンクションを使用する場合は、デバッグに使用可能な JTAG 接続が付いたようにシステムおよびボードをプランニング。
17.   JTAG デバッグ機能の JTAG ハブ・ロジックを実装するための少量の追加ロジック・リソースをプランニング。



Done N/A デザイン・チェックリスト ( 2 / 7 )

18.   **SignalTap II** エンベデッド・ロジック・アナライザでデバッグ作業をする場合は、デバイス・メモリ・リソースを保留してシステム動作中にデータをキャプチャ。
19.   **SignalProbe** またはロジック・アナライザ・インタフェースでデバッグ用の 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する。
20.   対象ボードで、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。
21.   外部ロジック・アナライザまたはミックスド・シグナル・オシロスコープに必要とするピン・ヘッダまたは **Mictor** コネクタを組み込む。
22.   デバッグ・ツールをインクリメントに使用してコンパイル時間を減少させるためには、デバッグ・ツールを修正する場合にデザインを再コンパイルする必要がなくなるよう、インクリメンタル・コンパイル機能をオンにすることを確認。
23.   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、これをデザイン・プロセスの一部として HDL コードにインスタンス化。
24.   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する。
25.   パワーアップを考慮してボードを設計する。**Cyclone IV** 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。
26.   電圧電源のランプがモノトニックになるように設計。
27.   電源が安定するように **POR** 時間を設定。
28.   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、電圧レギュレータおよびグラウンド接続を設計。
29.   **PDN** ツールを使用して電源分配ネットリストおよびデカップリング・コンデンサをプランニング。
30.   ノイズを低減するために、デザインですべての **PLL** を使用しない場合でも、すべての **PLL** 電源ピンを接続 ( $V_{CCA}$  を 2.5V に、 $V_{CCD\_PLL}$  を 1.2 V または 1.0 V に)。
31.   電源から各 **PLL** 電源ピンに厚い配線パターン (最低 20 ミル) を走らせる。
32.   すべての **PLL** デジタル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。
33.   フェライト・ビーズを使用して、**PLL** 電源をデジタル電源から絶縁。
34.   すべてのコンフィギュレーション・ピンおよびプルアップ / プルダウン抵抗はコンフィギュレーション手法に対して正しく設定されることを確認。
35.   **DCLK** および **TCLK** コンフィギュレーション・ピンを、ノイズが発生しないように設計。
36.   **JTAG** ピンが未使用の場合は、安定した電圧レベルに接続。

### Done N/A デザイン・チェックリスト ( 3 / 7 )

- 37.   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。
- 38.   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 ピンを抵抗によって 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する。
- 39.   TMS を抵抗を介して High にプルアップ。
- 40.   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。
- 41.   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。
- 42.   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。
- 43.   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。テストまたはデバッグ作業中に各コンフィギュレーション・モードに切り換えす柔軟性を確保するために、各ピンを 0 Ω 抵抗で VCCA または GND に接続するようにボードをセットアップ。
- 44.   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、nCE チップ・イネーブルを Low に保持。
- 45.   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。
- 46.   未使用 I/O ピンの予約状態を指定。
- 47.   QuartusII ソフトウェアで生成された .pin ファイル (Pin-Put 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。
- 48.   ノイズが発生ないように VREF ピンを設計。
- 49.   ボード層上のデバイスに近い大きなバス信号を分割してクロストークを低減。
- 50.   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。配線パターン幅の 2 ~ 3 倍の間隔を使用。
- 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。
- 54.   Quartus II Pin Planner を使用してピン・アサインメントを作成。
- 55.   QuartusII Fitter メッセージとレポートをピン・アサインメントのサイン・オフに使用。
- 56.   QuartusII 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。

Done N/A デザイン・チェックリスト ( 4 / 7 )

- 57.   I/O Assignment Analysis を使用して、Create Top-Level Design File コマンドを使用して、デザインの完成前に I/O アサインメントをチェック。
- 58.   システム要件に応じて信号の種類をプランニング。
- 59.   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。
- 60.   各 I/O ピンに最適な信号タイプおよび I/O 規格を選択。
- 61.   ターゲットの I/O バンクで適切な I/O 規格がサポートされていることを確認。
- 62.   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。
- 63.   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。
- 64.   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  $V_{REF}$  電圧レベルを使用するようになっていることを確認。
- 65.   LVDS およびトランシーバ機能の I/O バンク・サポートをチェック。
- 66.   LVDS I/O 付近のピン配置に対して、ガイドラインに従うことを注意してください。
- 67.   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や制約に従う。
- 68.   メモリ・インタフェースに専用 DQ/DQS ピンおよび DQ グループを使用。
- 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.   I/O インタフェースに役立つ利用可能なデバイス I/O 機能をチェック。例えば、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、プログラマブル・プリエンファシス、および差動出力電圧 ( $V_{OD}$ )。
- 71.   ボード・スペースを節約するように On-Chip Termination（チップ内終端）の使用を考慮。
- 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。
- 73.   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。
- 74.   デバイスの PLL をクロック管理に使用。
- 75.   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。
- 76.   PLL 機能をイネーブルにし、MegaWizard Plug-In Manager の設定をチェック。
- 77.   正しい PLL フィードバック補正モードを選択することを確認。
- 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.   デザインを解析して、SSN 問題が発生する可能性があるかどうか調べる。

### Done N/A デザイン・チェックリスト ( 5 / 7 )

- 80.   可能であれば同時に電圧を切り換えるピン数を減らす。
- 81.   スイッチング速度が高い I/O には、差動 I/O 規格と低電圧規格を使用。
- 82.   スイッチング速度が高い 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る。
- 83.   各バンク内の同時スイッチング出力ピン数を減らす。可能であれば、出力ピンを複数のバンクに分散させる。
- 84.   スイッチング I/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 (バンク使用率が 100% を十分に下回っている場合)。
- 85.   同時にスイッチングするピンを、SSN に敏感な入力ピンから分離。
- 86.   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、大きなスイッチング・バスから離して、グラウンド信号の近くに配置。
- 87.   PLL 電源ピンから 1 本または 2 本分離れた I/O ピンを、スイッチング速度が速い、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。
- 88.   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。
- 89.   同期デザイン手法を使用。クロック信号に注意。
- 90.   Quartus II Design Assistant を使用して、デザインの信頼性をチェックする。
- 91.   MegaWizard Plug-In Manager でメガファンクションを使用。
- 92.   メモリ・ブロックや DSP 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。
- 93.   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。
- 94.   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。
- 95.   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。
- 96.   インクリメンタル・コンパイル実行のために、ソース・コードのセットアップおよびデザインの分割に関する推奨事項に従う ; デザイン・フローの早期段階でプランニング。
- 97.   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グ・バジェットの作成とリソース・バランシングを実行。
- 98.   必要に応じて、インクリメンタル・コンパイル用の各パーティションに対してデザイン・フロアプランを作成。
- 99.   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。
- 100.   コンパイル後にリソースの利用および最適化に関するレポートを確認。

Done N/A デザイン・チェックリスト ( 6 / 7 )

- 101.   すべての Quartus II メッセージ、特にワーニング・メッセージまたはエラー・メッセージを確認。
- 102.   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。
- 103.   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。
- 104.   Cyclone IV 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。
- 105.   Settings ダイアログ・ボックスの Fitter Settings ページで、Optimize multi-corner timing をオンにする。
- 106.   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 multi-corner timing analysis をオンにするか、または TimeQuest に対して --multicorner コマンド・ライン・オプションを使用。
- 107.   create\_clock、create\_generated\_clock を使用してデザイン内のすべてのクロックの周波数と関係を指定。
- 108.   set\_input\_delay、set\_output\_delay を使用して、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。
- 109.   derive\_pll\_clocks を使用して 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平行変換係数に対して、マルチサイクル関係を指定。
- 110.   derive\_clock\_uncertainty を使用し、てインター・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。
- 111.   check\_timing を使用して、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。
- 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 QuartusII 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。



### Done N/A デザイン・チェックリスト ( 7 / 7 )

- 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。
- 120.   フォーマル検証を使用する場合、サポートおよびデザインの制限を確認。
- 121.   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.   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・アクティビティを生成。
- 123.   電力解析の正しい動作条件を指定。
- 124.   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。



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

Copyright © 2010 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



I.S. EN ISO 9001