

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。

QI152016-7.2.0

### はじめに

Quartus® II ソフトウェアは、デバイスの消費電力を全面的に最適化するパワー・ドリブン・コンパイルを提供します。パワー・ドリブン・コンパイルは、パワー・ドリブン・シンセシスとパワー・ドリブン配置配線を使用して、デザインの全消費電力を低減することに重点を置いています。この章では、パワー・ドリブン・コンパイルの機能とフローを詳細に説明し、またデザインにおける消費電力のさらなる低減を可能にする低消費電力デザイン手法についても説明します。この手法は、主に Arria™ GX、Stratix® および Cyclone® シリーズ、HardCopy® II デバイスをターゲットにしています。これらのデバイスは、動作時消費電力を劇的に低減し、性能を向上させる低誘電 (Low-k) 材を使用した最初の 90 nm FPGA です。Stratix III および Stratix II デバイスは、消費電力を最小限に抑えながら最大限の性能を得る、アダプティブ・ロジック・モジュール (ALM) と呼ばれる新しく、より効率的なロジック構造を内蔵しています。Cyclone III および Cyclone II デバイスは、低コスト FPGA における高性能と低消費電力の最適な融合を提供します。



Stratix III アーキテクチャについて詳しくは、「Stratix III デバイス・ハンドブック」を参照してください。

アルテラの Quartus II PowerPlay Power Analyzer は、高速かつ正確な消費電力の見積りを提供し、デザイン・プロセスでの作業を支援します。この章で説明するツールと手法を使用することによって、業界最先端の FPGA を活用しながら消費電力を最小限に抑えることができます。



PowerPlay Power Analyzer について詳しくは、「Quartus II ハンドブック Volume 3」の「PowerPlay による電力解析」の章を参照してください。

全消費電力は、I/O 電力、コア・スタティック電力、およびコア・ダイナミック電力から成ります。この章では、コア・ダイナミック電力と I/O 電力の低減に役立つデザイン最適化オプションおよびデザイン最適化手法を中心に説明します。これらの手法に加えて、Stratix III デバイスに対する追加の消費電力の最適化手法があります。これらの手法には、次のものがあります。

- 選択可能なコア電圧
- プログラマブル・パワー・テクノロジー
- デバイスのスピード・グレードの選択



Stratix III デバイスの消費電力の最適化手法について詳しくは、アプリケーション・ノート「AN 437: Stratix III FPGA の消費電力の最適化」を参照してください。

## 消費電力

ここでは、Stratix II および Cyclone II デバイスにおける消費電力の発生源について説明します。消費電力の発生源を理解することにより、デザインにおいて消費電力を低減する手法を改善することができます。

図 9-1 に、様々なデザインにおける Stratix II および Cyclone II デバイスの消費電力を示します。すべてのデザインは 200 MHz の固定クロック・レートで解析され、利用可能なリソースでの各ロジック・リソース利用率が表示されています。

図 9-1. 平均コア・ダイナミック消費電力

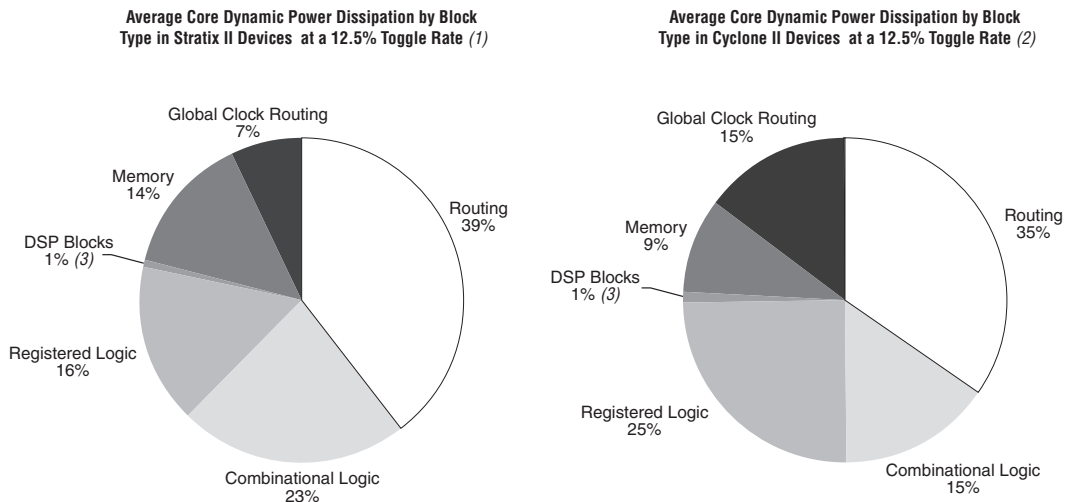


図 9-1 の注：

- (1) これらの結果は、112 種類のデザインを使用して得られました。
- (2) これらの結果は、93 種類のデザインを使用して得られました。
- (3) DSP ブロックを使用したデザインでは、DSP によって 5% のコア・ダイナミック電力が消費されました。

図 9-1 に示すように、Stratix II および Cyclone II デバイスの両方の配線で全体の消費電力のうちかなりの量が消費され、ロジック・ブロック、クロック・ブロック、および RAM ブロックで残りの電力が消費されています。

Stratix および Cyclone デバイス・ファミリでは、一連の可変長のカラムおよびロウ・インタコネクタ・ワイヤによって、ロジック・アレイ・ブロック (LAB)、メモリ・ブロック構造、およびデジタル信号処理 (DSP) ブロックまたは乗算器ブロックの間の信号の相互接続が提供されます。これらのインタコネクタが消費する電力は、デバイスの消費電力のうちで最も大きな部分を占めます。

消費電力のもう 1 つの発生源は、FPGA の組み合わせロジックです。ロジックの基本的なビルディング・ブロックは、Stratix III および Stratix II デバイスでは ALM で、Cyclone III および Cyclone II デバイスではロジック・エレメント (LE) です。



Stratix III、Stratix II、Cyclone III および Cyclone II デバイスの ALM および LE について詳しくは、それぞれのデバイス・ハンドブックを参照してください。

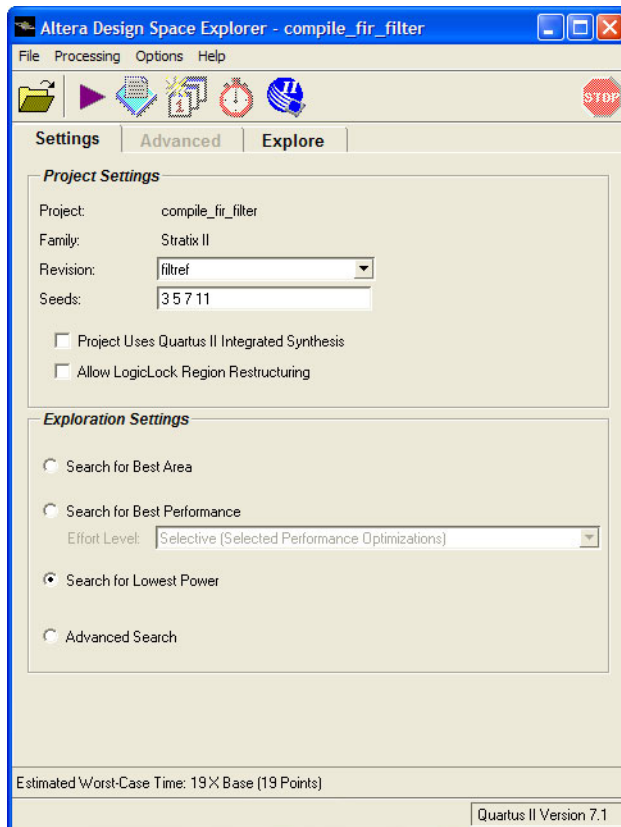
FPGA で電力を消費するその他の主要なものは、メモリ・リソースとクロック・リソースです。Stratix II デバイスは、TriMatrix メモリ・アーキテクチャを備えています。TriMatrix メモリには、512 ビットの M512 ブロック、4 K ビットの M4K ブロック、および 512 K ビットの M-RAM ブロックがあり、これらはそれぞれコンフィギュレーション可能で、多数の機能をサポートします。Stratix III TriMatrix オンチップ・メモリは、Stratix II FPGA の TriMatrix メモリをベースに強化されており、MLAB ブロック、M9K ブロック、および M144K ブロックの 3 種類のサイズのメモリ・ブロックで構成されています。Cyclone II デバイスは 4K ビットの M4K メモリ・ブロックを備えており、Cyclone III デバイスは 9K ビットの M9K メモリ・ブロックを備えています。

## Design Space Explorer

Design Space Explorer (DSE) は、Quartus II ソフトウェアに含まれている、シンプルで使いやすいデザイン最適化ユーティリティです。DSE は、消費電力の最適化、デザイン性能の向上、またはロジック利用率の向上を目標とするデザインに対して、最適な Quartus II ソフトウェアのオプションを検索し、報告します。DSE を使用して、この章で説明する手法を実装することができます。

図 9-2 に、DSE のユーザ・インタフェースを示します。**Settings** タブは、**Project Settings** と **Exploration Settings** に分かれています。

図 9-2. Design Space Explorer のユーザ・インタフェース



**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、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。この設定は、特にデザインの全熱電力を低減する異なるオプションを適用することに重点を置いています。DSE ウィンドウの **Advanced** タブを使用して、デザインに対して **Optimization Goal** オプションを設定することもできます。**Optimization Goal** リストで使用可能な設定の一覧から、消費電力の最適化など、デザイン最適化の目標を選択できます。DSE は次に、**Optimization Goal** リストから選択されたものを **Search for Lowest Power** で選択したものと併せて使用して、最良のコンパイル結果を決定します。

デフォルトでは、**Search for Lowest Power** オプションが選択されている場合、DSE が検索を実行するたびに、Quartus II PowerPlay Power Analyzer が実行されます。これは、デザインをデバッグし、電力要件と性能の最適化との間のトレードオフを決定するのに役立ちます。



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

## パワー・ ドリブン・ コンパイル

Quartus II の標準コンパイル・フローは、Analysis & Synthesis、フィッタ、アセンブラ、およびタイミング解析から成ります。パワー・ドリブン・コンパイルは、Analysis & Synthesis レベルおよびフィッタ・レベルで実行されます。パワー・ドリブン・コンパイルの設定は、**Analysis & Synthesis Settings** ページの **PowerPlay power optimization** リストと、**Fitter Settings** ページの **PowerPlay power optimization** に分かれています。以下の項では、Analysis & Synthesis レベルおよびフィッタ・レベルにおけるこれらの消費電力の最適化について説明します。

### パワー・ドリブン・シンセシス

合成ネットリスト最適化は、コンパイル・フローの合成ステージで行われます。最適化手法によって合成ネットリストに変更が加えられ、面積、速度、または消費電力の最適化の選択に従ってデザインが最適化されます。ここでは、合成レベルでの消費電力の最適化手法について説明します。

**Analysis & Synthesis Settings** ページでは、ロジック合成オプションを指定することができます。**PowerPlay power optimization** オプションは、Arria GX、Stratix および Cyclone ファミリー、および MAX<sup>®</sup> II デバイスに使用できます (図 9-3)。

Quartus II ソフトウェアで合成レベルでの消費電力の最適化を実行するには、以下のステップを実行します。

1. Assignments メニューで、**Settings** をクリックします。**Settings** ダイアログ・ボックスが表示されます。
2. **Category** リストで、**Analysis & Synthesis** を選択します。**Analysis & Synthesis** ページが表示されます。
3. **PowerPlay power optimization** リストで、希望の設定を選択します。このオプションによって、Analysis & Synthesis がデザインの消費電力をどの程度最適化するかが決定されます。

図 9-3. Analysis & Synthesis Settings ページ

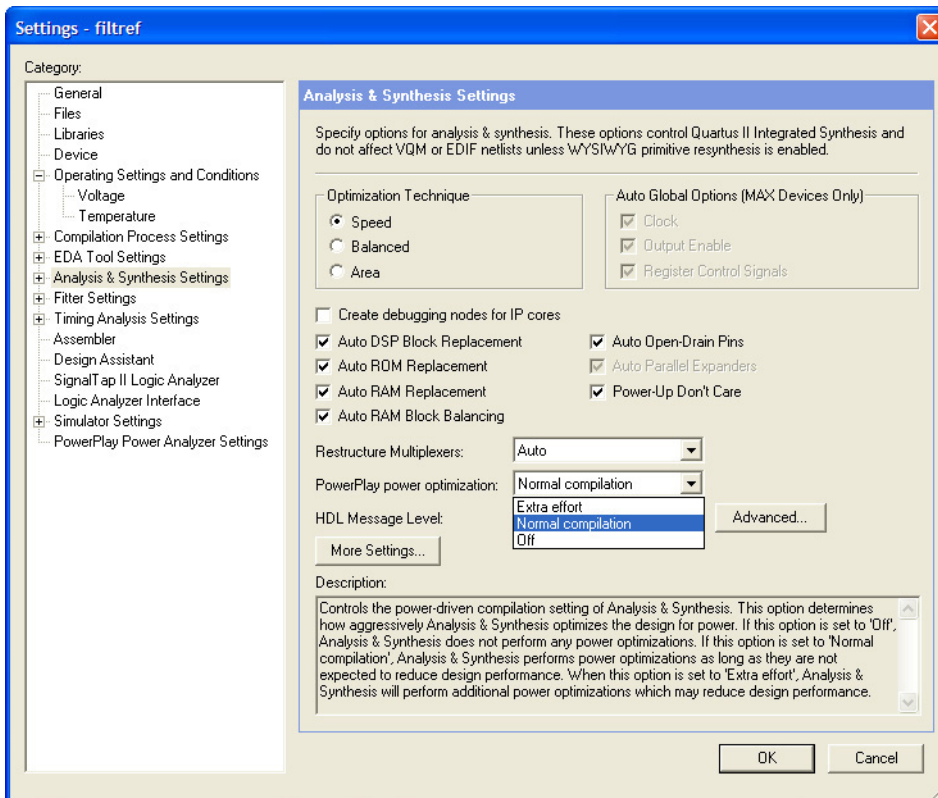


表 9-1 に、PowerPlay power optimization リストの設定を示します。これらの設定は、プロジェクト・レベルまたはエンティティ・レベルで適用できます。

表 9-1. 合成オプションでの消費電力の最適化	
設定	説明
Off	消費電力を最小化するためのネットリスト、配置、または配線の最適化は行われません。
Normal compilation (デフォルト)	消費電力の最適化によってデザイン性能が低下することが予測されない限り、消費電力の最適化を可能にします。
Extra effort	デザイン性能が低下する可能性がある追加の消費電力の最適化を実行することができます。

デフォルトでは、**Normal compilation** の設定はオンです。この設定では、合成中にメモリの最適化と消費電力を考慮したロジック・マッピングが実行されます。

メモリ・ブロックは、9-25 ページの「メモリの消費電力の低減」で説明するデザインの全動作時消費電力の大部分を占める可能性があります。各クロック・サイクル中にアクセスされるメモリ・ブロック数を最小限に抑えることによって、メモリ電力を大幅に低減できます。メモリの最適化は、ユーザ定義のリード/ライト・イネーブル信号を関連するリードおよびライト・クロック・イネーブル信号に変換するという最適化をすべてのメモリ・タイプで行います (図 9-4)。

図 9-4. メモリの変換

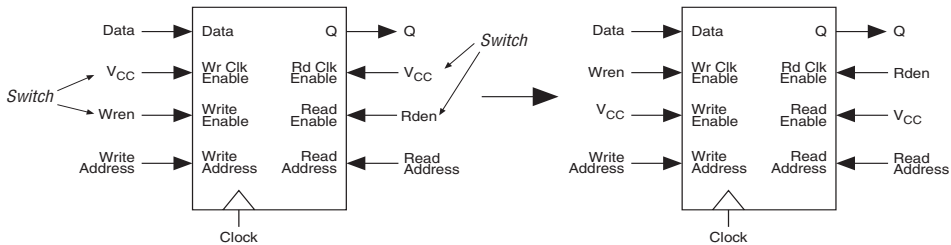


図 9-4 に、ライト・クロック・イネーブル信号およびリード・クロック・イネーブル信号が V<sub>CC</sub> に接続されており、各クロック・サイクル中にリードおよびライト・メモリの両方をアクティブにする、シングルなデュアル・ポート・メモリ・ブロックのデフォルト実装を示します。メモリの変換では、リード・イネーブル信号およびライト・イネーブル信号を対応するリード・クロック・イネーブル信号およびライト・クロック・イネーブル信号に効果的に移動します。この手法を使用すると、メモリ・ポートへのアクセスがないときにはメモリ・ポートがシャット・ダウンされます。これによって、デザインのメモリの消費電力が大幅に低減されます。クロック・イネーブル信号について詳しくは、9-25 ページの「メモリの消費電力の低減」を参照してください。Stratix III デバイスでは、メモリの変換は消費電力最適化オプションの **Normal compilation** 設定を選択してフィット・レベルで行われます。



Stratix III および Cyclone III デバイスでは、Read-During-Write 動作は、シングル・ポート RAM および双方向デュアル・ポート RAM の消費電力に著しく影響を与える可能性があります。最適化の際にリード・イネーブル信号を既存のライト・イネーブル信号（存在する場合）の反転に設定できるので、Read-During-Write パラメータを「Don't care」（HDL レベルで）に設定するのが最も適切です。これにより、RAM のコアをシャットダウンすることによって（すなわち、トグルしないで）、消費電力を大幅に節約できます。

**Normal compilation** の設定で行われるその他の消費電力の最適化は、消費電力を考慮したロジック・マッピングです。消費電力を考慮したロジック・マッピングでは、合成中にロジックを再配列してトグル率の高いネットをなくすことによって、消費電力を低減します。

**Extra effort** の設定では、**Normal compilation** の設定の機能とその他のメモリ最適化が実行され、アクセスされないメモリ・ブロックをシャット・ダウンすることによって、メモリ電力をさらに低減します。このレベルのメモリの最適化には、デザイン性能を低減することができる追加ロジックが必要になることがあります。

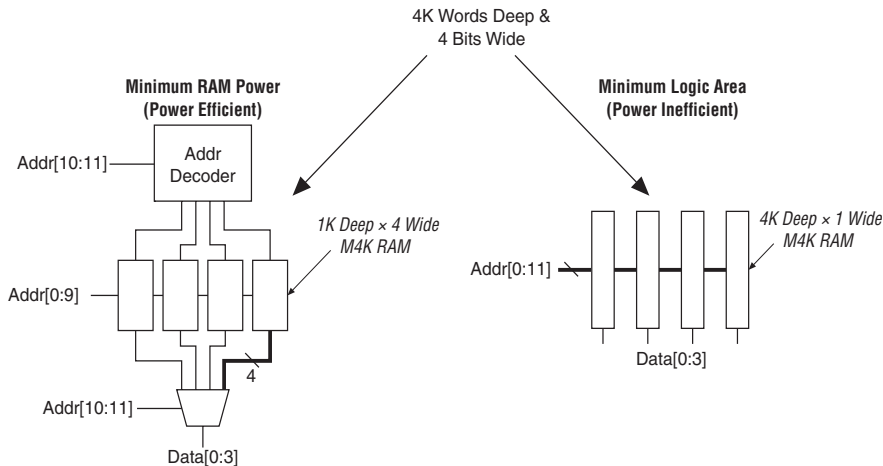
**Extra effort** の設定では、消費電力を考慮したメモリ・バランシングも実行されます。消費電力を考慮したメモリ・バランシングでは、メモリ実装に最良のメモリ・コンフィギュレーションが自動的に選択され、必要なメモリ・ブロック、デコーダ、およびマルチプレクサ回路の数を確定することによって、最適な低消費電力化を提供します。これまでにデザインのメモリ・ファンクションにターゲット・エンベデッド・メモリ・ブロックを指定していない場合、消費電力を考慮したバランサがメモリ実装時に自動的にそれらを選択します。

図 9-5 に、Stratix II デバイスで提供されている M4K メモリ・ブロックを使用した 2 つの異なるコンフィギュレーションの 4K×4（4K の深さ、4 ビット幅）メモリの実装例を示します。最小ロジック面積の実装では、4K×1 としてコンフィギュレーションされた M4K ブロックを使用します。この実装は、最小ロジック面積（0 個のロジック・セル）で最高速度を実現するため、Quartus II ソフトウェアでデフォルトになっています。ただし、この実装では、メモリ・アクセスが実行されるたびに 4 つの M4K ブロックのすべてがアクティブになり、その結果 RAM の消費電力が増大します。RAM 電力が最小の実装は、**PowerPlay power optimization** リストの **Extra effort** を選択することによって作成されます。この実装では、最適な低消費電力化を得るために自動的に 1K×4 としてコンフィギュレーションされた 4 つの M4K ブロックを使用します。altsyncram メガファンクションによってアドレス・デコーダが実行され、あるサイクルでアクティブにすべきブロックが、ユーザ・アドレス・ビットの最上位 2 ビットの状態に基づいて、4 つの M4K ブロックから選択さ

れます。altsyncram メガファンクションは、自動的にマルチプレクサを実装して、適切な M4K 出力を選択することにより下流ロジックに信号を供給します。この実装では、どのサイクルでもアクティブな M4K ブロックが1つしかないので、RAM 電力が低減されますが、余分なロジック・セルが必要なので、ロジック面積を要し、またデザイン性能にも影響を及ぼす可能性があります。

アクセスするメモリ数を減らすことによる低消費電力化と、デコーダおよびマルチプレクサ・ロジックの追加によって生じる電力消費との間にトレードオフが生じます。Quartus II ソフトウェアは、自動的に低消費電力化とコストとのバランスを取りながら、各論理 RAM に対する消費電力が最も低いコンフィギュレーションを選択します。

図 9-5. 複数の M4K ブロックを使用した 4K × 4 のメモリの実装



メモリ最適化オプションは、Settings ダイアログ・ボックスの **Default Parameters** ページにある Low\_Power\_Mode パラメータによって制御することもできます。このパラメータの設定値は、None、Auto、および ALL です。None は、PowerPlay power optimization リストの Off の設定に対応します。Auto と ALL は、それぞれ Normal compilation と Extra effort の設定に対応します。PowerPlay による消費電力の最適化は、コンパイラ・ベースまたは個々のエンティティに対して適用可能です。Low\_Power\_Mode パラメータは、メモリでの消費電力最適化のための Optimize Power for Synthesis オプションよりも常に優先されます。

MAXIMUM\_DEPTH パラメータを手動で設定して、低消費電力のための最適化を行うようにメモリをコンフィギュレーションすることもできます。この手法は、消費電力を考慮したメモリ・バランスと同じですが、**PowerPlay power optimization** リストの **Extra effort** の設定のように自動的ではなく、手動で行います。メガファンクションのインスタンスまたは MegaWizard® Plug-In Manager で、メモリ・モジュールに対する MAXIMUM\_DEPTH パラメータを手動で設定して、9-25 ページの「メモリの消費電力の低減」で説明する消費電力の最適化を行うことができます。MAXIMUM\_DEPTH パラメータは、メモリ最適化での消費電力最適化のための Optimize Power for Synthesis オプションよりも常に優先されます。

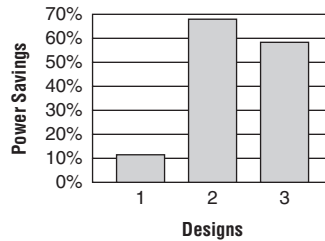
### Stratix II デバイスのパワー・ドリブン・シンセシスの実験

この Stratix II デバイスでの実験では、3つのデザインを Quartus II ソフトウェアで、**PowerPlay power optimization** リストの **Normal compilation** および **Extra effort** の設定を使用してコンパイルします。フィッタのデフォルトの設定は、**Normal compilation** です。表 9-2 に、パワー・ドリブン・シンセシスの実験で使用されたリソースを示します。

デザイン名	設定	ALUT 数	レジスタ数	メモリ・ビット数
デザイン 1	Normal compilation	8,941	9,150	293,856
	Extra effort	8,954	9,151	293,856
デザイン 2	Normal compilation	28,169	12,148	1,009,920
	Extra effort	28,817	12,297	1,009,920
デザイン 3	Normal compilation	5,376	2,809	153,864
	Extra effort	5,559	2,813	153,864

PowerPlay Power Analyzer は、ゲート・レベルのシミュレーション出力ファイルを使用して、消費電力を見積ります。図 9-6 は、パワー・ドリブン・シンセシスによって、Stratix II デバイスのメモリの消費電力が 68% 減少することを示しています。

図 9-6. パワー・ドリブン・シンセシスを使用した Stratix II デバイスのメモリ・ブロックの低消費電力化



## パワー・ドリブン・フィッタ

**Fitter Settings** ページでは、フィッティングのためのオプションを指定することができます (図 9-7)。**PowerPlay power optimization** オプションは、Arria GX、Stratix III、Stratix II、Stratix II GX、Cyclone III、Cyclone II、HardCopy II、および MAX II デバイスに使用できます。

フィッタ・レベルで消費電力の最適化を実行するには、以下のステップを実行します。

1. Assignments メニューで、**Settings** をクリックします。**Settings** ダイアログ・ボックスが表示されます。
2. **Category** リストで、**Fitter Settings** を選択します。**Fitter Settings** ページが表示されます。
3. **PowerPlay power optimization** リストで、希望の設定を選択します。このオプションによって、フィッタがデザインをどの程度消費電力について積極的に最適化するかが決定されます。

図 9-7. Fitter Settings ウィンドウ

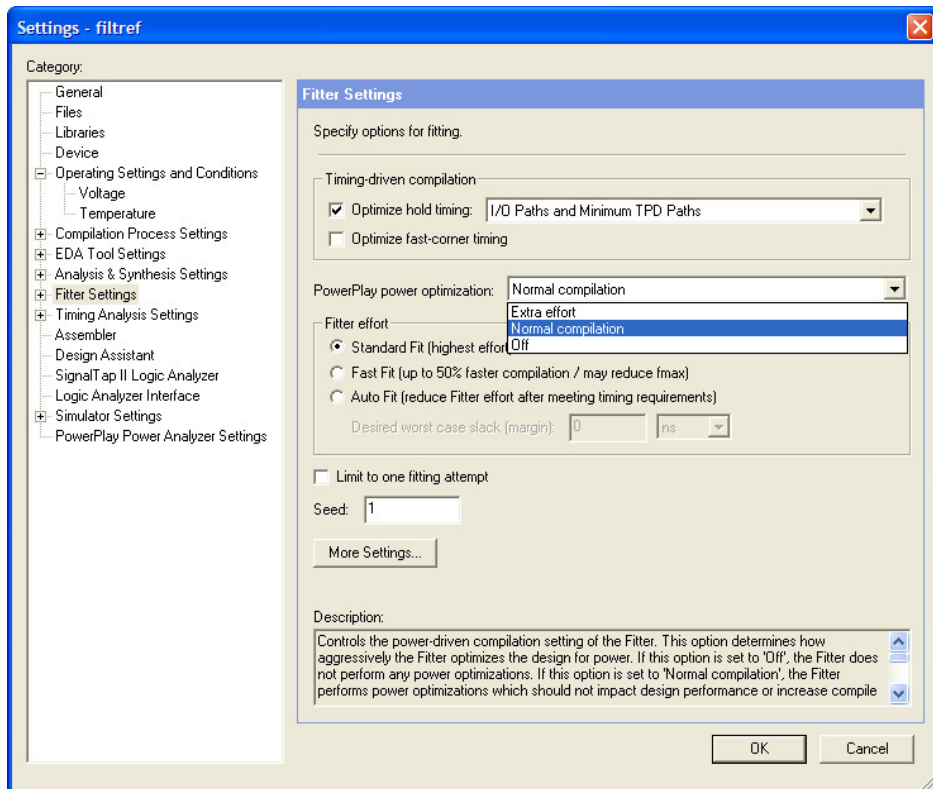


表 9-3 に、**PowerPlay power optimization** リストの設定を示します。これらの設定はプロジェクト全体でのみ適用可能です。フィッターの **Extra effort** の設定では、デザインを消費電力について最適化するための多大な労力が必要であり、またコンパイル時間が長くなることがあります。

表 9-3. パワー・ドリブン・フィッタのオプション

設定	説明
Off	消費電力を最小化するためのネットリスト、配置、または配線の最適化は行われません。
Normal compilation (デフォルト)	消費電力の最適化によってデザイン性能が低下することが予測されない限り、消費電力の最適化を可能にします。
Extra effort	デザイン性能が低下する可能性がある追加の消費電力の最適化を実行することができます。

**Normal compilation** の設定は、デフォルトで選択され、ユーザの DSP ファンクションに対する電力効率の高い DSP ブロックのコンフィギュレーションを作成することによって、DSP の最適化を実行します。Stratix III デバイスの場合、この設定はデザインに入力されたタイミング制約に基づいており、プログラマブル・パワー・テクノロジーでタイルを高速モードまたは低消費電力モードとしてコンフィギュレーションできます。Fitter PowerPlay power optimization オプションに **OFF** 設定が選択されているときでも、プログラマブル・パワー・テクノロジーは常に **ON** になっています。タイルとは、高速モードまたは低消費電力モードで動作するようにコンフィギュレーション可能な LAB と MLAB ペア (LAB と MLAB に関連する隣接する配線を含む) の組み合わせのことです。このレベルの消費電力最適化は、フィッティング、タイミング結果、またはコンパイル時間には影響を与えません。また、Stratix III デバイスの場合、この設定によって、9-5 ページの「パワー・ドリブン・シンセシス」に説明するとおり、メモリ変換が可能になります。



Stratix III デバイスの消費電力の最適化について詳しくは、アプリケーション・ノート「AN 437: Stratix III FPGA の消費電力の最適化」を参照してください。

**Extra effort** の設定では、フィッティング中に **Normal compilation** の設定の機能およびその他の配置配線の最適化が実行され、デザインを消費電力について完全に最適化します。フィッタは、配置中にトグル率の高いネットをローカライズするためにロジックを効率的に接近させることによって、またキャパシタンスの低い経路を使用することによって、タイミング要求値が満たされた後でも、特別な労力を使用して消費電力を最小限に抑えます。ただし、この労力のためにコンパイル時間が長くなることがあります。

**Extra effort** の設定では、フィッタを誘導する Signal Activity ファイル (.saf) または Verilog Value Change Dump ファイル (.vcd) を使用して、デザインの信号動作に基づき、デザインを消費電力について完全に最適化します。フィッティング中の最良の消費電力最適化は、最も正確な信

号動作情報を使用することによって得られます。フィッティング後の完全なネットリストの（タイミング）シミュレーションから得られる信号動作は、最高の精度を提供します。これは供給される入力ベクタが標準的なデザイン動作を表している場合、すべてのノードの動作が実際のデザインの振る舞いを反映するからです。（シミュレーションまたはその他の供給源からの）Signal Activity ファイルがない場合、Quartus II ソフトウェアはアサインメント、クロック・アサインメント、およびベクタレス見積り値（PowerPlay Power Analyzer Tool の設定値）を使用して、信号動作を予測します。この情報は、フィッティング中にデザインを消費電力について最適化するために使用されます。



フィッティング中に（Value Change Dump ファイルまたは SAF からの）信号動作を使用するのは、フィッタ・オプション用の **PowerPlay power optimization** リストにある **Extra effort** の設定だけです。デザインの信号動作の計算には、**Settings** ダイアログ・ボックスの **PowerPlay Power Analyzer Settings** ページで行われる設定値が使用されます。



Signal Activity ファイルおよび Verilog Value Change Dump ファイルの詳細および作成方法については、「Quartus II ハンドブック Volume 3」の「PowerPlay による電力解析」の章を参照してください。

### Stratix II デバイス用のパワー・ドリブン・フィッタの実験

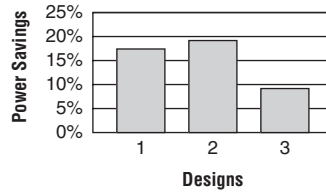
この Stratix II デバイスでの実験では、PowerPlay power optimization リストのフィッタにおける **Normal compilation** および **Extra effort** の設定を使用して、3つのデザインを Quartus II ソフトウェアでコンパイルします。Analysis and Synthesis のデフォルトの設定は、**Normal compilation** です。

表 9-4 に、パワー・ドリブン・フィッタの実験で使用されたリソースを示します。

表 9-4. Stratix II デバイスのパワー・ドリブン・フィッタの実験で使用されたデザイン		
デザイン名	ALUT (Normal Compilation)	ALUT (Extra Effort)
デザイン 1	21,435	21,363
デザイン 2	19,035	18,970
デザイン 3	5,335	5,328

PowerPlay Power Analyzer は、ゲート・レベルのシミュレーション出力ファイルを使用して、消費電力を見積ります。図 9-8 は、パワー・ドリブン・フィッタ手法によって、Stratix II デバイスのメモリの消費電力が 19% 減少することを示しています。

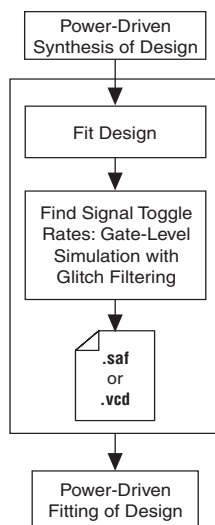
図 9-8. パワー・ドリブン・フィッタを使用した Stratix II デバイスの低消費電力化



## 推奨される パワー・ ドリブン・ コンパイルの フロー

図 9-9 に、コンパイル時にデザインを消費電力について完全に最適化するための推奨デザイン・フローを示します。このフローでは、パワー・ドリブン・シンセシスおよびパワー・ドリブン・フィッタのオプションを使用します。平均すると、コア・ダイナミック電力は、合成およびフィッティングを **Extra Effort** に設定すると、パワー・ドリブン・コンパイル用の合成およびフィッタ・オプションでオフに設定した場合と比較して、16% 低減することができます。

図 9-9. 推奨されるパワー・ドリブン・コンパイルのフロー



## エリア・ドリブン・シンセシス

合成時にタイミング最適化や遅延最適化を行うよりも、使用するロジック・ブロック数が少ないため、面積最適化を行う方が消費電力を節約します。使用するロジックが少なくと通常、スイッチング動作が少なくなることを意味します。

## Stratix II デバイスのエリア・ドリブン・シンセシスの実験

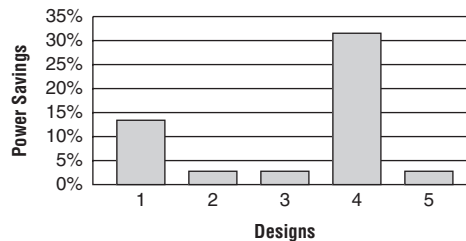
この Stratix II デバイスでの実験では、5つのデザインを Quartus II ソフトウェアを使用して、2通りの方法でコンパイルします。最初に、デザインが面積について最適化されコンパイルされます。次に、同じデザインが速度について最適化されコンパイルされます。合成およびフィッティングに対する消費電力最適化の設定は、**Off** に設定されています。

表 9-5 に、エリア・ドリブン・シンセシスの実験における ALUT の使用方法を示します。

デザイン名	ALUT (エリア・マッピング)	ALUT (速度マッピング)
デザイン 1	5,682	8,553
デザイン 2	16,986	17,783
デザイン 3	36,554	36,312
デザイン 4	4,717	5,820
デザイン 5	15,947	15,978

PowerPlay Power Analyzer は、ゲート・レベルのシミュレーション出力ファイルを使用して、消費電力を見積ります。図 9-10 は、エリア・ドリブン手法によって、Stratix II デバイスの消費電力が 31% 減少することを示しています。

図 9-10. エリア・ドリブン・シンセシスを使用した Stratix II デバイスの低消費電力化



### Cyclone II デバイスのエリア・ドリブン・シンセシスの実験

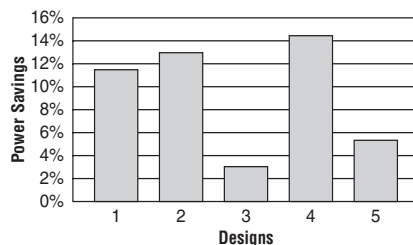
この Cyclone II デバイスでの実験では、5 つのデザインを Quartus II ソフトウェアを使用して、2 通りの方法でコンパイルします。最初に、デザインが面積について最適化されコンパイルされます。次に、同じデザインが速度について最適化されコンパイルされます。

表 9-6に、エリア・ドリブン・シンセシスの実験でのLEの使用を示します。

デザイン名	LE (エリア・マッピング)	LE (速度マッピング)
デザイン 1	13,020	16,429
デザイン 2	13,317	13,636
デザイン 3	5,384	5,690
デザイン 4	33,640	40,008
デザイン 5	21,409	22,988

PowerPlay Power Analyzer は、ゲート・レベルのシミュレーション出力ファイルを使用して、消費電力を見積ります。図 9-11 は、エリア・ドリブン手法によって、Cyclone II デバイスの消費電力が 15% 減少することを示しています。

図 9-11. エリア・ドリブン・シンセシスを使用した Cyclone III および Cyclone II デバイスの低消費電力化

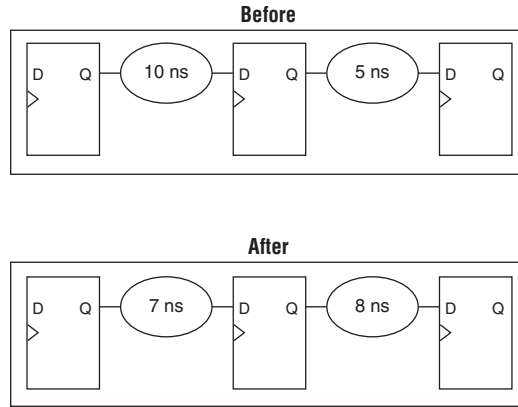


## ゲート・レベル・レジスタのリタイミング

ゲート・レベル・レジスタのリタイミングを使用して、回路のスイッチング動作を低減することもできます。リタイミングによって、デザインの機能性を変更することなく、組み合わせブロックをまたがってレジスタの入れ替えが行われます。Quartus II ソフトウェアの **Perform gate-level register retiming** オプションでは、タイミングのバランスを図るために組み合わせロジック全体でレジスタを移動させることができ、ソフトウェアはクリティカルなタイミング・パスとクリティカルでないタイミング・パスとの間の遅延をトレードオフすることができます。

リタイミングでは、パイプライン化の場合よりも使用するレジスタが少なくて済みます。図 9-12 に、ゲート・レベル・レジスタのリタイミングの例を示します。ここでは、組み合わせロジックと相対的にレジスタを移動させることによって 10 ns のクリティカルな遅延が低減され、データの深度とスイッチング動作が低減されます。

図 9-12. ゲート・レベル・レジスタのリタイミング



ゲート・レベル・レジスタのリタイミングによって、ゲート・レベルでの変更が行われます。サードパーティ合成ツールからの素子レベルのネットリストを使用している場合、**Perform WYSIWYG primitive resynthesis** オプションを選択して、素子レベル・プリミティブのゲートへのマッピングを解除し（これによってレジスタのリタイミングの実行が可能になります）、ゲートをアルテラ・プリミティブにマッピングし直さなければなりません。Quartus II の統合合成機能を使用する場合には、リタイミングは合成中にアルテラ・プリミティブにマップされる前に行われます。



レジスタのリタイミングについて詳しくは、「Quartus II ハンドブック Volume 2」の「Netlist Optimizations and Physical Synthesis」の章を参照してください。

### Stratix II デバイスのゲート・レベル・レジスタのリタイミングの実験

この Stratix II デバイスでの実験では、3つのデザインを Quartus II ソフトウェアを使用して、2通りの方法でコンパイルします。最初に、サードパーティ合成ツールからのネットリストがコンパイルされます。次に、

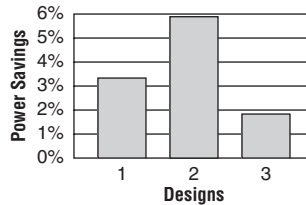
**Perform WYSIWYG primitive resynthesis** オプションおよび **Perform gate-level register retiming** オプションを選択した後で、同じネットリストがコンパイルされます。

表 9-7 に、リソースの使用結果を示します。

デザイン名	WYSIWYG および レジスタ・ リタイミング	ALUT 数	レジスタ数	DSP ブロック数	メモリ
デザイン 1	×	2,051	691	0	16
	○	1,882	731	0	16
デザイン 2	×	123,909	40,070	0	0
	○	95,593	39,816	0	0
デザイン 3	×	6,354	6,019	64	3,584
	○	7,496	5,970	64	3,584

PowerPlay Power Analyzer は、ゲート・レベルのシミュレーション出力ファイルを使用して、消費電力を見積ります。図 9-13 は、WYSIWYG リマッピングとゲート・レベル・レジスタのリタイミングの組み合わせによって、Stratix II デバイスで消費電力がほぼ 6% 減少することを示しています。

図 9-13. リタイミングを使用した Stratix II デバイスの低消費電力化



### Cyclone II デバイスのゲート・レベル・レジスタのリタイミングの実験

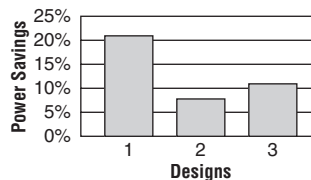
この Cyclone II デバイスでの実験では、5つのデザインを Quartus II ソフトウェアを使用して、2通りの方法でコンパイルします。最初に、サードパーティ合成ツールからのネットリストがコンパイルされます。次に、**Perform WYSIWYG primitive resynthesis** オプションおよび **Perform gate-level register retiming** オプションを選択することにより、同じネットリストがコンパイルされます。

表 9-8 に、リソースの使用結果を示します。

デザイン名	WYSIWYG および レジスタ・ リタイミング	LE 数	レジスタ数	乗算器 ブロック数	メモリ
デザイン 1	×	385	137	0	0
	○	278	143	0	0
デザイン 2	×	14,758	1,683	0	0
	○	13,079	1,683	0	0
デザイン 3	×	31,727	29,097	96	3,120
	○	27,038	24,272	96	3,120

PowerPlay Power Analyzer は、ゲート・レベルのシミュレーション出力ファイルを使用して、消費電力を見積ります。図 9-14 は、WYSIWYG リマッピングとゲート・レベル・レジスタのリタイミングの組み合わせによって、Cyclone II デバイスで消費電力が 21% 減少することを示しています。

図 9-14. リタイミングを使用した Cyclone II デバイスの低消費電力化



## デザイン・ ガイドライン

FPGA デザインの実装時に、低消費電力デザイン手法を使用して、消費電力を低減することができます。この項では、デザイン全体の消費電力に影響を与える Stratix III、Stratix II、Cyclone III、および Cyclone II デバイスのデザイン手法について詳細に説明します。これらの手法による結果は、デザインによって異なる場合があります。

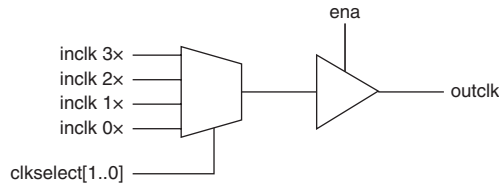
### クロック消費電力の管理

クロックは、高速スイッチング動作およびロング・パスに起因するダイナミック消費電力の重要な要素です。9-2 ページの図 9-1 は、グローバル・クロック配線の占める消費電力が Stratix II デバイスでは平均 7%、Cyclone II デバイスでは平均 5% であることを示しています。実際のクロック関連の消費電力は、ロジック、メモリ、および DSP または乗算器ブロック内のローカル・クロック分配における消費電力が対応するブロックの消費電力に含まれているため、これよりも高くなります。

クロック配線の消費電力は、Quartus II ソフトウェアで自動的に最適化されますが、Quartus II ソフトウェアはダウンストリーム・レジスタに供給する必要があるクロック・ネットワークのこれらの部分をイネーブルするだけです。不必要なクロックをゲートすることによって、さらに消費電力を低減することができます。クロック・ゲーティング・ロジックを構築することは可能ですが、ALM または LE を使用して FPGA 内でグリッチのないクロックを生成するのは困難なため、アルテラではこの方法は推奨していません。

Arria GX、Stratix III、Stratix II、Cyclone III、および Cyclone II デバイスは、イネーブル信号を含むクロック・コントロール・ブロックを使用しています。クロック・コントロール・ブロックは、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したり、クロック・ネットワークをドライブする複数のソースをダイナミックに切り換えることができるクロック・バッファです。Quartus II MegaWizard Plug-In Manager を使用して、altclkctrl メガファンクションでこのクロック・コントロール・ブロックを作成できます。Arria GX、Stratix III、Stratix II、Cyclone III、および Cyclone II デバイスは、グローバル・クロック・ネットワークにクロック・コントロール・ブロックを提供します。さらに、Stratix III および Stratix II デバイスはリージョンナル・クロック・ネットワーク用のクロック・コントロール・ブロックを備えています。ダイナミック・クロック・イネーブル機能により、内部ロジックでクロック・ネットワークを制御できます。クロック・ネットワークがパワーダウンすると、クロック・ネットワークで供給されるすべてのロジックがトリップを中止し、デバイスの全消費電力が減少します。図 9-15 に、4 入力クロック・コントロールのブロック図を示します。

図 9-15. クロック・コントロールのブロック図



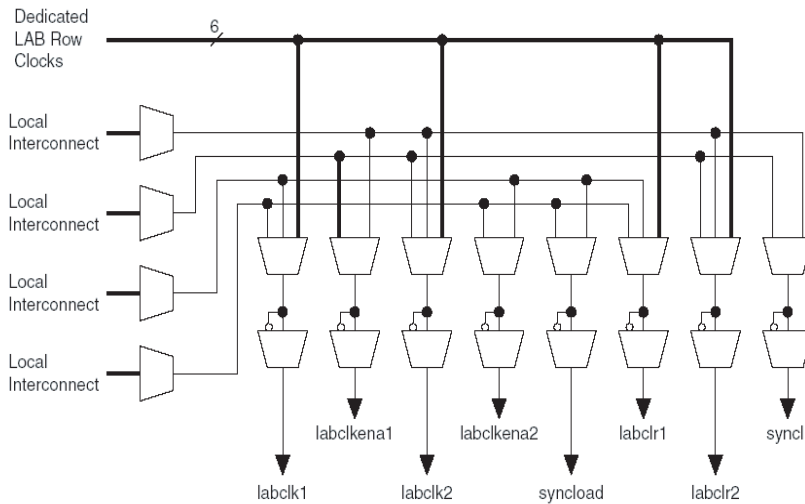
イネーブル信号は、クロック信号に印加されてからグローバル配線に分配されます。したがって、イネーブル信号は大きなタイミング・スラック（グローバル配線遅延以上）を持つことができます。あるいはクロック信号の  $f_{MAX}$  を低減できます。



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

クロックの消費電力に関するもう1つの重要な要素として、LAB内のレジスタにクロックを分配するLABクロックがあります。LABクロックの消費電力は、クロック全体の消費電力の大部分を占める場合があります。例えば、図 9-16 に示すように、Cyclone III および Cyclone II デバイスでは、各LABは2本のクロック信号と2本のクロック・イネーブル信号を使用することができます。各LABのクロック信号とクロック・イネーブル信号はリンクされています。例えば、labclk1 信号を使用する特定のLABのLEは、labclkenal 信号も使用します。

図 9-16. LAB ワイド・コントロール信号



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・イネーブルを使用して LAB ワイド・クロックをゲートします。Quartus II ソフトウェアは、レジスタ・レベルのクロック・イネーブル信号を自動的に LAB レベルに昇格させます。共通クロックおよびクロック・イネーブルを共有する LAB 内のすべてのレジスタは、共有ゲート・クロックで制御されます。これらのクロック・イネーブルを利用するには、レジスタ化されたロジックの関連 HDL コードでクロック・イネーブル構造を使用します。

### LAB ワイド・クロック・イネーブルの例

この VHDL コードでは LAB ワイド・クロック・イネーブルを使用します。このクロック・ゲーティング・ロジックは、自動的に LAB レベル・クロック・イネーブル信号に変換されます。

```

IF clk'event AND clock = '1' THEN
    IF logic_is_enabled = '1' THEN
        reg <= value;
    ELSE
        reg <= reg;
    END IF;
END IF;

```



LAB ワイド・コントロール信号について詳しくは、各デバイスのハンドブックの「Stratix II アーキテクチャ」、「Cyclone III アーキテクチャ」、または「Cyclone II アーキテクチャ」の章を参照してください。

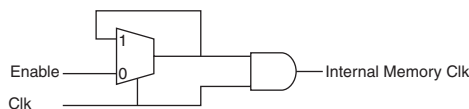
## メモリの消費電力の低減

FPGA デバイスのメモリ・ブロックは、標準的なコア・ダイナミック消費電力の大部分を表すことができます。メモリは、標準的な Stratix II デバイス・デザインのコア・ダイナミック消費電力の 14%、Cyclone II デバイス・デザインの 9% を表します。メモリ・ブロックの消費電力の大部分はクロック・レートに関係しており、データ・ラインおよびアドレス・ラインでのトグル・レートの影響を受けないという点で、デバイス内の他の多くのブロックとは異なります。

メモリ・ブロックがクロックされると、ブロック内で一連のタイムド・イベントが発生して、読み出しまたは書き込みが実行されます。クロックにより制御される回路は、アドレスまたはデータがあるサイクルから次のサイクルに変化したかどうかに関係なく、同じ電力量を消費します。したがって、入力データとアドレス・バスのトグル・レートはメモリの消費電力に影響を与えません。

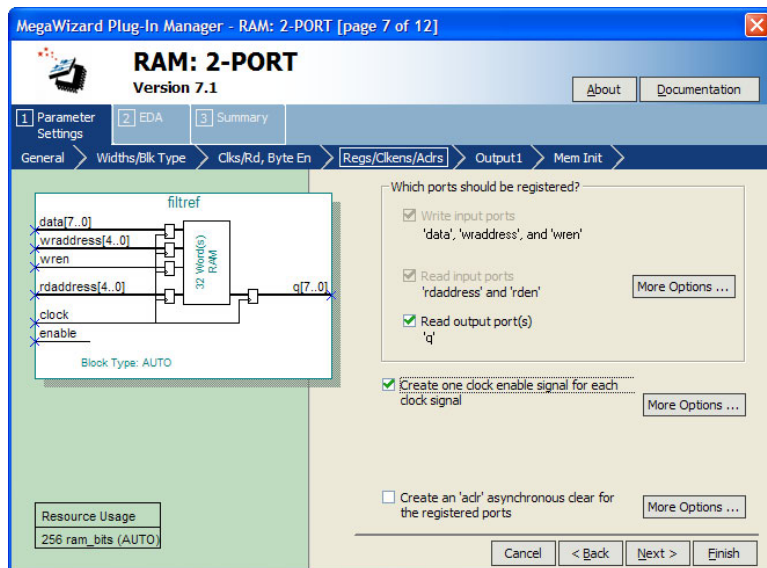
メモリの消費電力を低減する鍵は、メモリ・クロッキング・イベント数を低減することです。これは、[9-22 ページの「クロック消費電力の管理」](#)で説明したクロック・ネットワーク・ワイドのゲーティング、またはメモリ・ポート上のクロック・イネーブル信号をメモリ単位で使用することによって実現できます。[図 9-17](#)に、メモリ・ブロックの内部クロックのロジック図を示します。メモリで適切なイネーブル信号を使用して、クロックをゲートする代わりにクロック・イネーブル信号を利用します。

図 9-17. メモリ・クロック・イネーブル信号



クロック・イネーブル信号を使用して必要な場合にのみメモリをイネーブルにし、それ以外の時間はシャット・ダウンしてメモリ全体の消費電力を低減します。メモリ・ブロック・ファンクションを生成する場合、Quartus II の MegaWizard Plug-In Manager を使用し、**Clock enable signal** オプションで適切なポートを選択して、これらのイネーブル信号を作成することができます ([図 9-18](#))。

図 9-18. MegaWizard Plug-In Manager RAM 2-Port Clock Enable Signal で選択可能なオプション

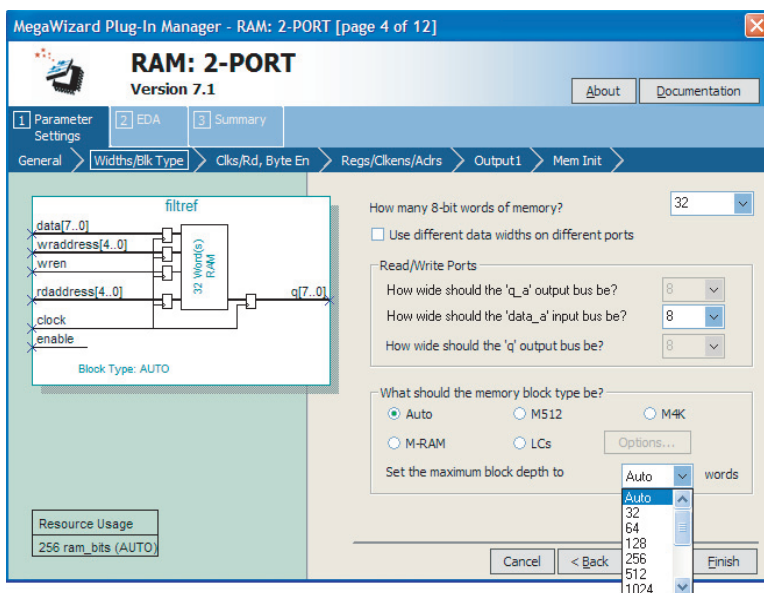


例えば、200 MHz で動作する ROM モードの 32 ビット幅 M4K メモリ・ブロックを搭載したデザインを検討します。このブロックは約 4 サイクルごとに出力すればよいと仮定すると、このメモリ・ブロックはダウンストリーム・ロジックの要求に応じて、8.45 mW のダイナミック電力を消費します。少数のコントロール・ロジックを追加して、関連するサイクルでのみメモリ・ブロックのリード・クロック・イネーブル信号を生成すると、消費電力は 75% 削減されて 2.15 mW になります。

メモリ・メガファンクションで、MAXIMUM\_DEPTH パラメータを使用して、Stratix III、Stratix II、Cyclone III および Cyclone II デバイスの消費電力を削減することができますが、この手法によりメモリの実装に必要な LE 数が増加し、デザインの性能に影響を与える可能性があります。

メモリ・モジュールの MAXIMUM\_DEPTH パラメータを、メガファンクションのインスタンス化または MegaWizard Plug-In Manager に手動で設定することができます (図 9-19)。Quartus II ソフトウェアは、9-5 ページの「パワー・ドリブン・コンパイル」で説明したとおり、最適な消費電力を得るために最良のデザイン・メモリ・コンフィギュレーションを自動的に選択します。

図 9-19. MegaWizard Plug-In Manager RAM 2-Port Maximum Depth で選択可能なオプション



### メモリの低消費電力化の例

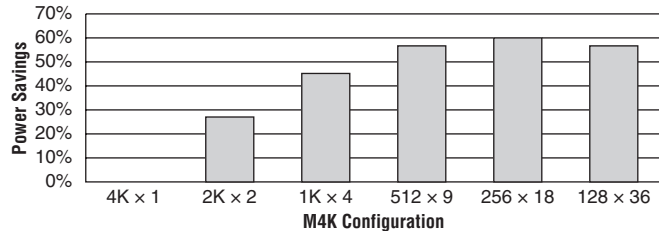
表 9-9 に、複数の M4K ブロックを使用して Stratix II EP2S15 デバイスに実装した 4K × 36 シンプル・デュアル・ポート・メモリの消費電力の測定値を示します。M4K ブロックは、実装ごとに異なるメモリの深度でコンフィギュレーションされます。

表 9-9. 複数の M4K ブロックを使用して実装された 4K × 36 シンプル・デュアル・ポート・メモリ

M4K の コンフィギュレーション	M4K ブロック数	ALUT 数
4K × 1 (デフォルト設定)	36	0
2K × 2	36	40
1K × 4	36	62
512 × 9	32	143
256 × 18	32	302
128 × 36	32	633

図 9-20 に、MAXIMUM\_DEPTH パラメータを使用した消費電力の削減量を示します。すべての実装において、読み出しデータが必要なタイミングを示すユーザ提供のリード・イネーブル信号が存在します。この省電力手法を使用して、消費電力を最大 60% 削減できます。

図 9-20. MAXIMUM\_DEPTH パラメータを使用した消費電力の削減



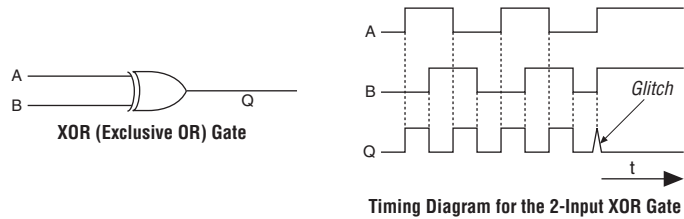
アドレス指定されない M4K ブロックは、デコードされたアドレス・ビットとリード・イネーブル信号の組み合わせを使用してシャット・オフできるため、メモリの深度が浅くなるとメモリのダイナミック消費電力が減少します。深度 128 のメモリ・ブロックの場合、追加 LE の消費電力が深度の浅いメモリ・ブロックを使用して得られた消費電力の削減量を上回ります。メモリ・ブロックと関連 LE の消費電力は、メモリ・コンフィギュレーションに依存します。

## パイプライン化およびリタイミング

多数のグリッチを含むデザインでは、高速スイッチング動作のために消費電力が増加します。グリッチによって、組み合わせロジックの出力で不要かつ予測不能な一時的なロジックの変化が発生します。グリッチは通常、入力信号のタイミングにミスマッチがある場合に発生し、それにより伝播遅延が不均等になります。

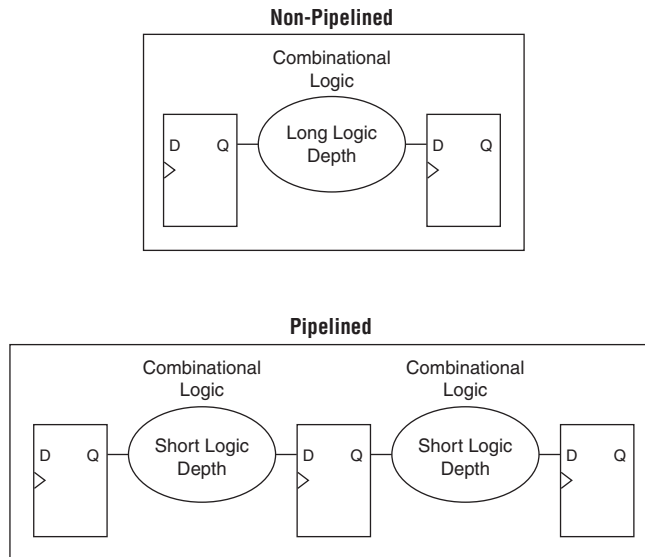
例えば、2 入力 XOR ゲートの 1 入力 が 1 から 0 に変化し、しばらくして、他の入力 が 0 から 1 に変化すると仮定します。状態の遷移中に、両方の入力が一瞬 1 (High) になり、その結果 XOR ゲートの出力が 0 (Low) になります。続いて、次の入力遷移が起こると、XOR ゲートの出力は 1 (High) になります。図 9-21 に示すように、信号の遷移中は、出力が安定する前にグリッチが発生します。このグリッチは後続のロジックを伝播して、不必要なスイッチング動作を引き起こし、消費電力を増加させることがあります。演算回路または巡回冗長チェック (CRC) 回路など、多数の XOR ファンクションを持つ回路では、レジスタ間にいくつかのレベルの組み合わせロジックが存在する場合は、多数のグリッチが発生する傾向があります。

図 9-21. 出力にグリッチを生じる XOR ゲート



パイプライン化により、長い組み合わせパスにフリップ・フロップを挿入して、デザインのグリッチを低減することができます。フリップ・フロップによって、グリッチが組み合わせパスを通して伝播しなくなります。したがって、パイプライン化された回路ではグリッチが減少する傾向があります。パイプライン化には一般に回路のレイテンシ（最初の結果を得るまでのクロック・サイクル数）が増加する一方で、より高いクロック速度動作が可能になるという別の利点があります。図 9-22 に、パイプライン化を適用して長い組み合わせパスを分割する例を示します。

図 9-22. パイプライン化の例



パイプライン化は、スイッチング動作を低減し、それによって組み合わせロジックの消費電力を削減するため、グリッチが発生しやすい演算システムにとって非常に有用です。また、パイプライン化によりレジスタ

間のロジック・レベル数を減らすことによって、高速動作が可能になります。パイプライン化手法の欠点として、デザインにグリッチが多く存在しない場合は、パイプライン化によって不要なレジスタが追加され、消費電力が増加することが挙げられます。また、リソースの使用率も増加します。

### Stratix II デバイスのパイプライン化の実験

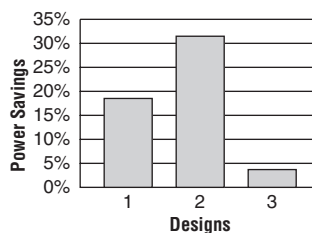
この実験では、Stratix II デバイスにパイプライン化したものと、パイプライン化していないものの3つのデザインを実装しています。これら3つのデザインは、大きなグリッチが発生する可能性がある（XOR ファンクションに基づく）演算を多用します。

表 9-10に実験で使用したデザインでのリソースの使用状況を示します。

デザイン名	パイプライン化	ALUT 数	レジスタ数
乗算器 (デザイン 1)	×	9,726	448
	○	9,772	1,109
累算乗算器 (デザイン 2)	×	13,719	1,120
	○	14,007	2,260
FIR フィルタ (デザイン 3)	○ (レベル 1)	1,048	949
	○ (レベル 2)	932	929

PowerPlay Power Analyzer は、ゲート・レベルのシミュレーション出力ファイルを使用して、消費電力を見積ります。図 9-23 は、パイプライン化によって Stratix II デバイスの消費電力が 31% 減少することを示しています。

図 9-23. パイプライン化による Stratix II デバイスの低消費電力化



## Cyclone II デバイスのパイプライン化の実験

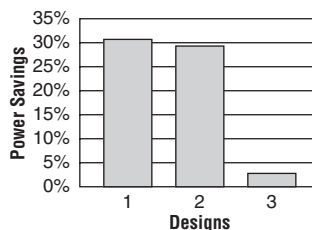
この実験では、Cyclone II デバイスにパイプライン化したものと、パイプライン化していない3つのデザインを実装しています。これら3つのデザインは、大きなグリッチが発生する可能性がある（XOR ファンクションに基づく）演算を多用します。

表 9-11に実験で使用したデザインでのリソースの使用状況を示します。

デザイン名	パイプライン化	LE 数	レジスタ数
累算乗算器（デザイン 1）	×	6,870	320
	○	13,071	3,719
加算器（デザイン 2）	×	7,392	1,076
	○	7,343	752
分周器（デザイン 3）	×	6,659	320
	○	6,735	520

PowerPlay Power Analyzer は、ゲート・レベルのシミュレーション出力ファイルを使用して、消費電力を見積ります。図 9-24 は、パイプライン化によって Cyclone II デバイスの消費電力が31%減少することを示しています。

図 9-24. パイプライン化による Cyclone II デバイスの低消費電力化



## アーキテクチャの最適化

特定のデバイスのアーキテクチャ機能を利用することによって、デザイン・レベルでアーキテクチャの最適化を使用できます。これらの機能には、メモリまたは演算関連ファンクションを実行するために、専用メモリおよび FPGA デバイスで利用可能な DSP または乗算器ブロックが含

まれます。これらのブロックを LUT の代わりに使用して、消費電力を削減することができます。例えば、LE レジスタからシフト・レジスタを構築する代わりに、RAM ベースの FIFO バッファから大きなシフト・レジスタを構築することができます。

Stratix デバイス・ファミリでは、TriMatrix メモリ・アーキテクチャで小、中、および大容量メモリを効率的にターゲットにすることができます。各 TriMatrix メモリ・ブロックは、特定ファンクション用に最適化されています。Stratix II デバイスの M512 メモリ・ブロックは、小さな FIFO バッファ、DSP、およびクロック・ドメイン転送アプリケーションを実装するのに役立ちます。M512 メモリ・ブロックは、一部の競合 FPGA の分散メモリ構造よりも電力効率に優れています。M4K メモリ・ブロックは、プロセッサ・コードの格納、大きなルック・アップ・テーブルの実装、および大容量メモリ・アプリケーションなど、多彩なアプリケーション用バッファの実装に使用されます。M-RAM ブロックは、大量のデータをチップ上に格納する必要があるアプリケーションに有用です。これらのメモリ・ブロックを効率的に使用すると、デザインの消費電力を大幅に低減することができます。

Cyclone デバイス・ファミリは、RAM、FIFO バッファ、ROM などの多様なメモリ・ファンクションを提供するコンフィギュレーション可能な M4K および M9K メモリ・ブロックを搭載しています。



DSP およびメモリ・ブロックの効率的な使用方法については、「Quartus II ハンドブック Volume 2」の「面積およびタイミングの最適化」の章を参照してください。

### Stratix II デバイスのアーキテクチャ最適化の実験

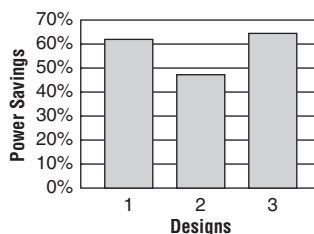
この実験では、Stratix II デバイスに 3 つのデザインを別々の方法で実装し、専用ブロックの消費電力低減機能を示します。最初の 2 つのデザインは、ロジック・エレメントと DSP ブロックを使用しています。3 番目のデザインは M4K および M-RAM ブロックを使用しています。3 番目のデザインでは、大容量メモリ・アプリケーションの場合、M-RAM ブロックの方が M4K ブロックよりも電力効率に優れていることがわかります。この実験では、合成およびフィッティングの消費電力最適化オプションをオフにしています。

表 9-12 に、関連するリソースの使用結果を示します。

デザイン名	実装	ALUT 数	レジスタ数	DSP ブロック数	メモリ
デザイン 1	通常の実装	9,726	448	0	0
	専用リソースの 実装	1,124	448	121	0
デザイン 2	通常の実装	13,719	1,120	0	0
	専用リソースの 実装	2,880	896	212	0
デザイン 3	M4K	286	228	0	1,835,008 (M4K)
	M-RAM	224	224	0	1,835,008 (M-RAM)

PowerPlay Power Analyzer は、ゲート・レベルのシミュレーション出力ファイルを使用して、消費電力を見積ります。図 9-25 は、アーキテクチャ最適化手法により、Stratix II デバイスの消費電力が 60% 減少することを示しています。

図 9-25. 専用ブロックを使用した Stratix II デバイスの低消費電力化



### Cyclone II のアーキテクチャ最適化の実験

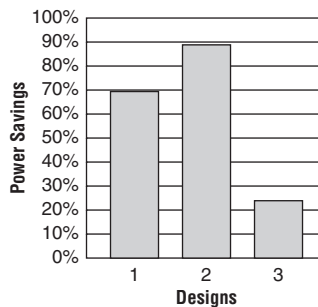
この実験では、Cyclone II デバイスに 3 つのデザインを別々の方法で実装し、専用ブロックの消費電力低減機能を示します。最初の 2 つのデザインは、LE および乗算器ブロックを使用しています。3 番目のデザインは LE および M4K ブロックを使用しています。

表 9-13 に、関連するリソースの使用結果を示します。

デザイン名	実装	LE 数	レジスタ数	乗算器 ブロック数	メモリ
デザイン 1	通常の実装	6,870	320	0	0
	専用リソースの実装	1,130	320	49	0
デザイン 2	通常の実装	7,343	752	0	0
	専用リソースの実装	1,401	608	44	0
デザイン 3	通常の実装	1,550	1,265	0	0
	M4K	72	72	0	1,152

PowerPlay Power Analyzer は、ゲート・レベルのシミュレーション出力ファイルを使用して、消費電力を見積ります。図 9-26 は、アーキテクチャ最適化手法により、Cyclone II デバイスの消費電力が 88% 減少することを示しています。

図 9-26. 専用ブロックを使用した Cyclone II デバイスの低消費電力化



### I/O 消費電力のガイドライン

LVTL および LVC MOS などの非終端 I/O 規格には、レール・トゥ・レール出力振幅が規定されています。出力ピンのロジック High 信号とロジック Low 信号間の電圧差は、 $V_{CCIO}$  電源電圧と同じです。出力ピンの容量性負荷が分かっている場合、I/O バッファでのダイナミック消費電力は以下のように計算することができます。

$$P = 0.5 \times F \times C \times V^2$$

この式では、 $F$  は出力遷移周波数、 $C$  は切り替わる全負荷キャパシタンスを表します。 $V$  は、 $V_{CCIO}$  電源電圧と等しくなります。 $V_{CCIO}$  は二次依存であるため、低電圧規格でのダイナミック消費電力は大幅に減少します。また、低いピン・キャパシタンスも I/O 消費電力を考慮する上で重要な要素です。ハードウェアおよびシミュレーション・データは、Stratix II デバイスの I/O ピンのピン・キャパシタンスは最も近い競合 FPGA のピン・キャパシタンスの半分であることを示しています。Cyclone II デバイスの消費電力は、競合する低コスト 90 nm FPGA に比べ 20% 低くなっています。

TTL (Transistor-to-Transistor Logic) I/O バッファはスタティック電力をほとんど消費しません。その結果、LVTTTL または LVC MOS 出力の全消費電力は、負荷およびスイッチング周波数に大きく依存します。

SSTL や HSTL などの抵抗終端 I/O 規格を使用する場合、出力負荷電圧はバイアス点周辺でわずかに振幅します。 $V$  が実際の負荷電圧振幅の場合は、同じダイナミック消費電力等式を使用します。この電圧は  $V_{CCIO}$  よりかはるかに小さいため、同様の条件下ではダイナミック消費電力は非終端 I/O の場合よりも小さくなります。I/O バッファは抵抗終端ネットワークに絶えず電流をドライブしているため、これらの抵抗終端 I/O 規格は大きなスタティック (周波数に依存しない) 消費電力を放散します。ただし、これらの I/O 規格のダイナミック消費電力が低い場合、高周波数アプリケーションでは LVC MOS または LVTTTL よりも全消費電力が低くなります。速度および波形要件を満たす最低ドライブ強度の I/O 設定を使用して、抵抗終端規格を使用したときの I/O 消費電力を最小限に抑えます。

未使用の I/O バンクを最も低い 1.2 V の  $V_{CCIO}$  電圧に接続することによって、少量のスタティック消費電力を節約することができます。

表 9-14 に、Stratix II デバイス用の各種 I/O 規格を使用した出力で消費される全供給電流および熱消費電力を示します。これらの数値は、10 pF の容量性負荷の接続時に 200 MHz でクロックされるランダム・データを送信する I/O ピンに対するものです。

表 9-14. Stratix II デバイスの各種 I/O 規格における I/O 消費電力

規格	V <sub>CCIO</sub> 電源からの 全供給電流 (mA)	チップ内 全熱消費電力 (mW)
3.3-V LVTTTL	2.42	9.87
2.5-V LVCMOS	1.9	6.69
1.8-V LVCMOS	1.34	4.18
1.5-V LVCMOS	1.18	3.58
3.3-V PCI	2.47	10.23
SSTL-2 Class I	6.07	4.42
SSTL-2 Class II	10.72	5.1
SSTL-18 Class I	5.33	3.28
SSTL-18 Class II	8.56	4.06
HSTL-15 Class I	6.06	3.49
HSTL-15 Class II	11.08	4.87
HSTL-18 Class I	6.87	4.09
HSTL-18 Class II	12.33	5.82

このコンフィギュレーションの場合、非終端規格の消費電力は一般に I/O 規格の消費電力より少なくなりますが、そうでない場合もあります。周波数または容量性負荷が増加すると、非終端出力の消費電力は終端出力の消費電力よりも速く増加します。



I/O 規格について詳しくは、「Stratix II デバイス・ハンドブック Volume 2」の「Stratix II デバイスおよび Stratix II GX デバイスの選択可能な I/O 規格」の章、または「Cyclone II デバイス・ハンドブック Volume 2」の「Cyclone II デバイスの選択可能な I/O 規格」の章を参照してください。

I/O 消費電力を計算するとき、PowerPlay Power Analyzer は、**Device & Pin Options** ダイアログ・ボックスの **Capacitive Loading** タブの I/O 規格に設定されたデフォルトの容量性負荷を使用します。**Enable Advanced I/O Timing** がオンになっている場合、I/O 消費電力は **Device & Pin Options** ダイアログ・ボックスの **Board Trace Model** タブまたは **Pin Planner** の **Board Trace Model** ビューで定義されるとおり、近端容量、伝送ライン分散容量、および遠端容量の和として計算される等価負荷を使用して測定されます。ボード・トレース・モデルで定義されるその他のコンポーネントは、消費電力の測定には考慮されません。



Advanced I/O タイミングの使用およびボード・トレース・モデルのコンフィギュレーションについて詳しくは、「Quartus II ハンドブック Volume 2」の「I/O 管理」の章を参照してください。

## Power Optimization Advisor

Quartus II ソフトウェアには、現在のデザイン・プロジェクトの設定およびアサインメントに基づく、消費電力の最適化に関する具体的なアドバイスや推奨事項を提供する Power Optimization Advisor 機能があります。この章で取り上げる提案の多くは、Power Optimization Advisor でカバーされています。以下の例では、Power Optimization Advisor によるデザインの消費電力低減方法を示しています。

### Power Optimization Advisor の例

デザインをコンパイルした後、PowerPlay Power Analyzer を実行してデザインの消費電力を求め、デザインで電力が消費される箇所を確認します。この情報に基づいて、Power Optimization Advisor を実行してデザインで消費電力を低減できる推奨事項を実施することができます。図 9-27 に、消費電力が十分に最適化されていないデザインをコンパイルした後の Power Optimization Advisor を示します。

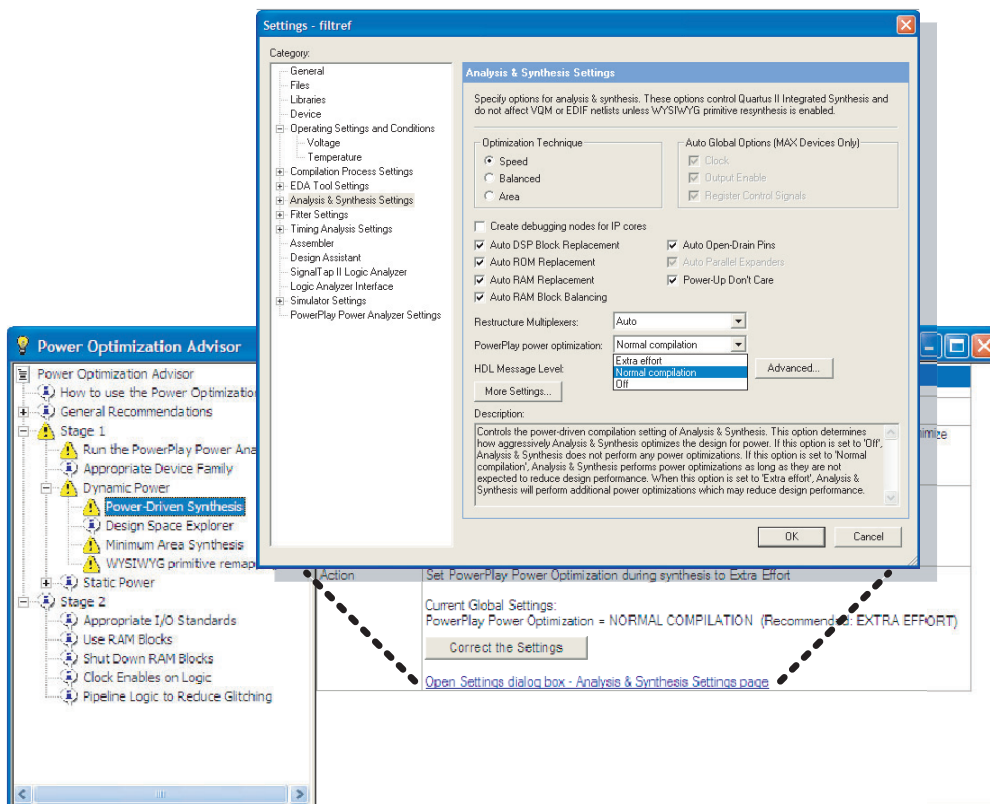
図 9-27. Power Optimization Advisor

Power-Driven Synthesis	
Recommendation	Set PowerPlay Power Optimization during synthesis to Extra Effort.
Description	Extra effort power-driven synthesis will choose logic and RAM implementations which minimize design dynamic power. <a href="#">More Info</a>
Summary	The following areas will be affected by the recommended changes: <ul style="list-style-type: none"> <li>- Delay may increase (fmax may decrease)</li> <li>- Logic element usage may increase</li> <li>= Compilation time is unaffected</li> </ul>
Action	Set PowerPlay Power Optimization during synthesis to Extra Effort  No action is needed for this recommendation. The recommended setting has been made.  Current Global Settings: PowerPlay Power Optimization = EXTRA EFFORT (Recommended: EXTRA EFFORT)  <a href="#">Correct the Settings</a>  <a href="#">Open Settings dialog box - Analysis &amp; Synthesis Settings page</a>

Power Optimization Advisor にデザインの消費電力の低減を可能にする推奨事項が表示されます。推奨事項は、推奨設定を適用する順序を示す複数のステージに分かれています。最初のステージは、簡単に実装可能で、デザインの消費電力の低減にきわめて効果的な CAS 設定オプションを主に示します。アイコンは現在のプロジェクトで推奨される各設定が行われているかどうかを示します。図 9-27 のステージ 1 のチェックマーク・アイコンは、すでに実装されている推奨事項を示します。警告アイコンは、このコンパイルでは満たされていない推奨事項を示します。情報アイコンは一般的な提案を示します。各推奨事項には説明、推奨事項の及ぼす影響の要約、および適切な設定を行うために必要なアクションが含まれています。

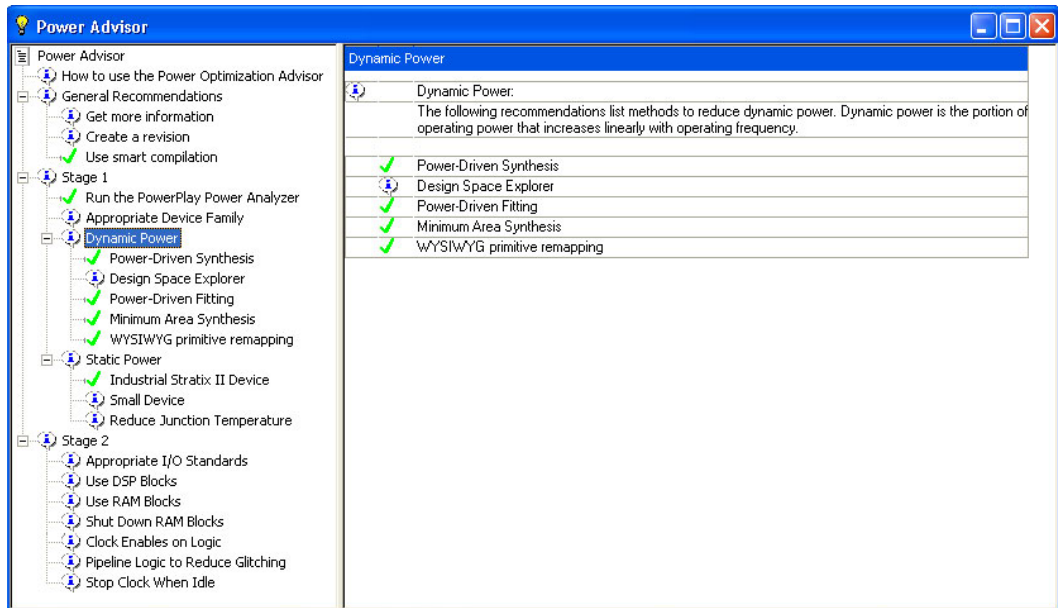
各推奨事項から Quartus II ユーザ・インタフェースの適切な位置へのリンクがあり、Power-Driven Synthesis 設定などの設定を変更することができます。Power-Driven Synthesis 設定は、Open Settings ダイアログ・ボックスの Analysis & Synthesis Settings ページ (図 9-28) をクリックすると変更できます。Analysis & Synthesis Settings ページを選択すると、Setting ダイアログ・ボックスが表示されます。ここで、PowerPlay power optimization 設定を変更できます。

図 9-28. Analysis &amp; Synthesis Settings ページ



推奨される変更を行った後、デザインを再コンパイルします。Power Optimization Advisor は、推奨事項が正常に実施されたことを緑色のチェックマークで示します (図 9-29)。PowerPlay Power Analyzer を使用して、デザインの消費電力の結果を検証できます。

図 9-29. Power Optimization Advisor の推奨事項の実施



ステージ 2 に表示される推奨事項は、一般にステージ 1 のような CAD の設定変更ではなく、デザインの変更に關するものです。これらの推奨事項を使用して、さらにデザインの消費電力を低減することができます。アルテラは、最初にステージ 1 の推奨事項を実施し、次にステージ 2 の推奨事項を実施することを推奨します。

### まとめ

微細プロセス・テクノロジーの組み合わせにより、低誘電 (Low-k) 材および低電源電圧を使用することで、最新の FPGA のダイナミック消費電力を大幅に低減できます。ダイナミック消費電力をさらに低減するには、この章に記載したデザインの推奨事項を使用して、リソースの利用率を最適化し、消費電力を最小限に抑えます。

## 参考資料

この章では以下のドキュメントを参照しています。

- 「altclkctrl Megafunction User Guide」
- 「AN 437: Stratix III FPGA の消費電力の最適化」
- 「Cyclone III デバイス・ハンドブック」の「Cyclone III デバイス・ファミリの概要」の章
- 「Cyclone II デバイス・ハンドブック」の「Cyclone II アーキテクチャ」の章
- 「Quartus II ハンドブック Volume 2」の「デザイン・スペース・エクスプローラ」の章
- 「Quartus II ハンドブック Volume 2」の「I/O 管理」の章
- 「Quartus II ハンドブック Volume 2」の「Netlist Optimizations and Physical Synthesis」の章
- 「Quartus II ハンドブック Volume 3」の「PowerPlay による電力解析」の章
- 「Stratix II デバイス・ハンドブック Volume 1」の「Stratix II アーキテクチャ」の章
- 「Stratix II デバイス・ハンドブック Volume 2」の「Stratix II デバイスおよび Stratix II GX デバイスの選択可能な I/O 規格」の章
- 「Cyclone II デバイス・ハンドブック」の「Cyclone II デバイスの選択可能な I/O 規格」の章
- 「Cyclone II デバイス・ハンドブック」の「Cyclone II デバイスの選択可能な I/O 規格」の章
- 「Quartus II ハンドブック Volume 2」の「面積およびタイミングの最適化」の章
- 「Stratix III デバイス・ハンドブック」
- 「Stratix II デバイス・ハンドブック」

## 改訂履歴

表 9-15 に、本資料の改訂履歴を示します。

表 9-15. 改訂履歴		
日付 & ドキュメント・バージョン	変更内容	概要
2007 年 10 月 v7.2.0	9-41 ページの「参考資料」を再編成。	—
2007 年 5 月 v7.1.0	<ul style="list-style-type: none"> <li>● 消費電力の項を更新。</li> <li>● パワー・ドリブン・コンパイルの項を更新。</li> <li>● 参考資料の項を追加。</li> </ul>	Quartus II ソフトウェア・バージョン 7.1 向けに図を更新。Arria GX デバイスのサポートを追加。
2007 年 3 月 v7.0.0	Cyclone III の情報を含むテキストのマイナーな編集。 図 9-2 に新しいスクリーンショットを追加。	Cyclone III デバイスのサポート情報を追加。
2006 年 11 月 v6.1.0	ソフトウェアの GUI の変更に対応するよう図を更新。	Stratix III デバイスのサポート情報を追加。 Quartus II の新しい拡張ユーザ機能によりプロシージャを変更。
2006 年 5 月 v6.0.0	Quartus II ソフトウェア・バージョン 6.0.0 のための更新： デバイスのサポートを更新 マルチ VCD/SAF サポート情報を追加 低消費電力化された値を更新	—
2005 年 10 月 v5.1.0	Volume 1: Stratix II Low Power Design Techniques の 7 章を 9 章に変更。	—