

# Vivado Design Suite

## ユーザーガイド: 消費電力解析および最適化

UG907 (v2020.1) 2020 年 6 月 3 日

この資料は表記のバージョンの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。資料によっては英語版の更新に対応していないものがあります。日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

# 改訂履歴

次の表に、この文書の改訂履歴を示します。

セクション	改訂内容
2020 年 6 月 3 日 バージョン 2020.1	
資料全体	編集上のアップデートのみ。技術内容の変更なし。

# 目次

改訂履歴.....	2
第 1 章: ザイリンクス デバイスの消費電力.....	5
概要.....	5
消費電力の用語.....	5
ザイリンクス デバイスの電源.....	8
ザイリンクス デバイスの消費電力と全般的な設計プロセス.....	8
ザイリンクス消費電力見積もり、解析、最適化ツール.....	9
第 2 章: 消費電力の見積もり: 初期評価段階.....	12
概要.....	12
Xilinx Power Estimator を使用した正確なワースト ケース消費電力を見積もるための 7 つの手順.....	12
第 3 章: 消費電力の見積もり: Vivado Design Suite でのデザイン フロー段階.....	23
概要.....	23
消費電力見積もり.....	23
Vivado IDE での消費電力の見積もり.....	23
消費電力レポート用の HBM 設定.....	36
消費電力レポート用の GTM 設定.....	38
第 4 章: Vivado Design Suite での消費電力解析および最適化.....	41
概要.....	41
Vivado IDE での消費電力解析.....	41
消費電力最適化の機能.....	63
第 5 章: Vivado 消費電力レポートを使用した正確な消費電力解析.....	74
概要.....	74
第 6 章: 消費電力削減のためのヒントおよび手法.....	86
概要.....	86
システム レベルの消費電力削減.....	86
消費電力および温度の計測.....	87
デザイン レベルの消費電力削減.....	89
付録 A: その他のリソースおよび法的通知.....	95
ザイリンクス リソース.....	95
Documentation Navigator およびデザイン ハブ.....	95
参考資料.....	95

トレーニング リソース.....	96
お読みください: 重要な法的通知.....	96

# ザイリンクス デバイスの消費電力

## 概要

この章では、ボードに搭載されたザイリンクス デバイスの消費電力の説明に使用される用語を説明します。システム設計におけるデバイス開発についても説明し、設計フローの各段階でどのような操作が実行されるかを示します。また、消費電力の見積もり、解析、および最適化に使用されるザイリンクス ツールについても説明します。



**ビデオ:** Vivado® で消費電力を見積もる方法と、最も正確な見積もりを得るためのベスト プラクティスについては、[Vivado QuickTake ビデオ: Vivado での消費電力の見積もりと解析](#)を参照してください。



**ビデオ:** ザイリンクス デバイス消費電力に影響する要因および Vivado でデザインの消費電力を最小限に抑える方法について、Vivado の消費電力最適化を最大限に利用するアドバンス制御、ベスト プラクティスについては、[Vivado QuickTake ビデオ: Vivado を使用した消費電力の最適化](#)を参照してください。

## 消費電力の用語

このガイドでは、次の用語が使用されます。

- **デバイスのスタティック消費電力:** コンフィギュレーション後にデバイスが正常に動作するために必要な、接続されているすべての電圧レールと回路のトランジスタ リーク電流から発生する消費電力。通常、空のビットストリームをデバイスにプログラムすると計測できます。デバイスのスタティック消費電力は、プロセス、電圧、温度に左右されます。デバイスが動作していないときの、デバイス内のリーク電流を表します。
- **デザインの消費電力:** 入力データのパターンおよびデザインの内部動作により発生する、ユーザー デザインのダイナミック消費電力。瞬間的なものであり、各クロック サイクルで変わります。電圧レベル、使用されるロジック リソースおよび配線リソースによって左右されます。また、I/O 終端、クロック マネージャー、および使用時に電力を必要とするその他の回路からのスタティック電流も、これに含まれます。オフチップ デバイスに供給される電力は含まれません。
- **オンチップの総消費電力:** デバイス内の消費電力の合計で、デバイスのスタティック消費電力とデザインの消費電力を足したものです。熱電力とも呼ばれます。
- **オフチップ消費電力:** 電源からデバイスの電源ピンを介して I/O から供給され、外部ボード コンポーネントで消費される電流。デバイスから供給される電流は、通常 I/O 終端、LED、ほかのチップの I/O バッファなどのオフチップ コンポーネントで消費されるので、デバイスのジャンクション温度は上昇しません。

**注記:** 負のオフチップ消費電力は、外部ソースから供給され、デバイス内で消費される電流のことです。

- **パワーオン電流:** デバイスに最初に電源を投入したときに発生する過渡電流。各電圧電源、デバイスの構造、電源が公称電圧まで上昇する立ち上がり時間によって異なります。この電流は、温度、電源の投入シーケンスなど、デバイスの動作条件にも左右されます。アーキテクチャが改善されていたり、適切な電源投入シーケンスに従っている場合は、パワーオン電流は動作電流よりも小さくなるのが一般的です。

- ジャンクション温度 (°C): 動作中のデバイスの温度。通常デバイスを選択する際は、温度グレードを選択します。この温度グレードは、ザイリンクスがデバイスが仕様どおり正常に動作することを確認した温度範囲を定義します。動作条件が温度グレードを超えると、絶対最大温度未満であっても、デバイスの動作は保証されません。絶対最大温度を超えると、デバイスが破損する可能性があります。

ジャンクション温度 = 周囲温度 + (オンチップ消費電力合計 \* 外気に対する有効熱抵抗 (ΘJA))

- 周囲温度 (°C): システムの動作条件下におけるデバイス周囲の外気の温度。
- 外気に対する熱抵抗 (ΘJA (°C/W)): Theta-JA および JA と呼ばれ、デバイス シリコンから周囲の環境 (デバイスジャンクションから外気) までの電力損失を定義する係数です。シリコン チップの寸法から外気までのすべての要素に加え、パッケージ、PCB、ヒートシンク、エアフローなど、その間にあるものが影響します。発生した熱が環境に放熱される方向には、主に次の 2 つがありますが、外気に対する熱抵抗は、通常、熱抵抗と、この 2 つのかかわりを組み合わせています。
  - ダイから上方向 (ジャンクションから外気、ΘJA)。
  - ダイからボードを介して下方向 (ジャンクションからボード、ΘJB)。

ザイリンクス デバイス パッケージの温度データは、[パッケージ温度データ クエリ ツール](#)を使用すると取得できます。次の図に、温度データ クエリの結果例を示します。

図 1: 温度データ クエリの結果例

## Package Thermal Data Query

Thermal data summary for XC7K325T-FF900

Device Family : Kintex 7  
Device Name : XC7K325T  
Package Name : FF900  
Unit : C/Watt

JA (Still Air) 9.7  
JA (250 LFM) 6.2  
JA (500 LFM) 5.4  
JA (750 LFM) 5  
JB 2.8  
JC 0.26

LFM = Linear Feet per Min

Note:

If you do not find a package or product in the list contact support. While the list is constantly reviewed to reflect current supported products, note that the availability of thermal data for a product in the Query results does not necessarily imply that the product is currently supported or will be supported in the future. Do NOT use the Query to check current device availability and support.

For a table view of all packages in a specific device family, please refer to the "Thermal Specifications" section of the *Packaging and Pinout Product Specification* document for that family.

熱抵抗の詳細は、『7 シリーズ FPGA パッケージおよびピン配置: 製品仕様』(UG475: [英語版](#)、[英語版](#))、『UltraScale および UltraScale+ FPGA パッケージおよびピン配置ユーザー ガイド』(UG575: [英語版](#)、[日本語版](#)) を参照してください。



**重要:** このユーザー ガイドに示す温度データは、デバイス/パッケージの比較のためにのみ示しています。これらの値を熱シミュレーションで使用しないでください。[ザイリンクス ウェブサイト](#)から提供されている熱モデルを使用してください。

## デバイス特性

- Advance: このステータスのデバイスのデータ モデルは、早期プロダクション デバイス ロットからのシミュレーション結果または計測結果に基づいています。このデータは通常、製品リリースから 1 年以内に入手可能となります。消費電力モデル データは比較的安定しており、余裕を持たせた設定ですが、実際の値は上下する可能性があります。この仕様のデータは、Preliminary および Production 仕様のデータほど正確ではありません。

- Preliminary: このステータスのデバイスのデータ モデルは、エンジニアリング サンプル シリコンの特性評価データに基づいています。この仕様では、デバイス ファブリック内にあるほぼすべてのブロックが特性評価されています。Advance と比較すると、より正確に消費電力値を見積もることができます。
- Production: ステータスがプロダクションになっているデバイスは、特定デバイス ファミリメンバーのプロダクション シリコンが十分に特性評価され、量産されても一定した電力データが得られることが確認されてから、リリースされています。この特性化データを使用したデバイス モデルは、この時点からは変更されません。

消費電力見積もりの正確性は、モデルに入力する情報によります。消費電力レポートでは、デバイス特定に基づいて次のモデルが使用されます。

- Advance: +/-30%
- Preliminary: +/-25%
- Production: +/-15%

これらのモデルの精度は Xilinx® Power Estimator と同じですが、消費電力レポートには、リソース設定および使用率、ネットのファンアウトおよびネットの長さなど、デザインの消費電力見積もりに影響するデザインの詳細が示されるので、より正確な見積もりが得られます。ただし、見積もりの精度が入力によることは変わらないので、信頼性レベルは重要です。信頼性レベルの低い Production 特性モデルを使用する場合は、評価してより正確な見積もりが得られるよう改善する必要があります。

**注記:** [Process] を [maximum] に設定する場合は、デバイスのスタティック消費電力がツールでレポートされる値を超えないようにする必要があります。

## 信号レート

エレメントのステートが秒ごとに変化 (High から Low、Low から High) する回数。ザイリンクス ツールでは、秒ごとの遷移数が百万単位 (Mtr/s) で示されます。たとえば、信号が 100 MHz (10 ns) クロックの 4 クロック サイクルごとに変化する場合、信号レートは  $1/(4 \times 10 \text{ ns}) = 25 \text{ Mtr/s}$  になります。

## トグル レート

トグル レートは、同期ロジック エレメントの出力がその入力に対して切り替わるレート (%) を示し、0 ~ 100% の範囲で表されます。トグル レートが 100% の場合、出力は各クロック サイクルごとに平均 1 回トグルします。たとえば、任意の周波数のクロックに対して信号が 4 クロック サイクルごとに遷移する場合、トグル レートは  $(1/4) \times 100 = 25\%$  です。



**重要:** クロック ネットのトグル レートは常に 200% なので、ネットは 1 サイクルで 2 回トグルします。



**ヒント:** 同期ネット (DDR ネット以外) は最大でもクロックごとに変化するのが理想的なので、最大トグル レートは 100% です。同期ネットにグリッチが発生しやすい場合は、[Signal Rate] を使用してスイッチング アクティビティを指定します。

クロックに同期しないネットやロジックなどの非同期エレメントでは、トグル レートは算出されません。Vivado® 消費電力ツールは、これらのタイプのエレメントに信号レートが使用されていると想定します。

デフォルトでは、デザインのプライマリ入力は特定のクロックには関連付けられません。set\_input\_delay 制約を使用して、クロックとプライマリ入力を関連付けてください。クロックを関連付けない場合、消費電力ツールでは、デスティネーション クロックまたはデザインで最速のクロックのいずれかを基に、トグル レートが計算されます。

## スタティック確率

スタティック確率は、エレメントが High (1'b1) に駆動されている時間の割合を定義します。有効な範囲は 0 ~ 1 です。たとえば、100 ns のうち信号が 1 になるのが 40 ns の場合、スタティック確率は  $40/100 = 0.4$  になります。



ヒント: スタティック確率が 1 の場合、解析中エレメントが 1 に保持され、トグルしないことを示します (トグル/信号レート = 0)。スタティック確率が 0 の場合、解析中エレメントが 0 に保持され、トグルしないことを示します (トグル/信号レート = 0)。

## ザイリンクス デバイスの電源

ザイリンクス デバイスに電源を供給するには、複数の電源が必要です。個別の電源から、異なるデバイス リソースに必要な電力が供給されます。こうすることで、異なる電圧レベルでさまざまなリソースを動作させることができるため、ノイズや寄生効果に対して高い耐性を保ちつつ、パフォーマンスおよび信号強度を向上できます。

ザイリンクス デバイスで通常利用可能なロジック リソースについては、『Xilinx Power Estimator ユーザー ガイド』(UG440: [英語版](#)、[日本語版](#)) の第 1 章の表「FPGA リソースと電源」を参照してください。

## ザイリンクス デバイスの消費電力と全般的な設計プロセス

プロジェクト考案から完成まで、消費電力に影響する多数の要因を考慮する必要があります。ほかのすべての制約 (機能、パフォーマンス、コスト、およびタイム トゥ マーケット) を考慮しない場合、消費電力に関するタスクは次の 2 つに分類できます。

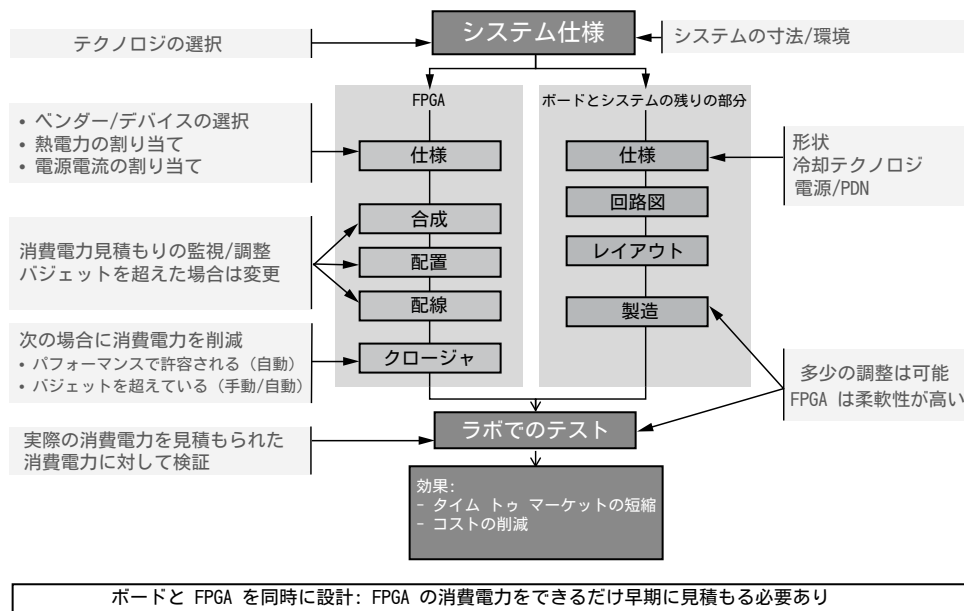
- 物理的: エンクロージャ、ボードの形状、電源、電源分配ネットワーク (PDN)、熱電力損失の構造。
- 論理的: エリア、パフォーマンス、I/O インターフェイスのシグナル インテグリティ。

次の章で、これらの 2 つの依存関係を説明します。物理的タスクはハードウェアに関する決定事項、論理的タスクはデザイン作成に関係しています。通常、ハードウェアの選択およびサイズ設定は、プロトタイプ ボードを構築する時間を確保するため、デザイン フローの初期段階で実行します。デバイスの機能の消費電力への影響は早期に見積もり可能で、デザイン ロジックが完成していくにつれ調整されます。次の図に、消費電力に関する決定ポイントをハイライトした典型的なシステム設計プロセスを示します。この図では、デバイスとその冷却パーツを選択する時点では、デバイス ロジックは完成していません。そのため、デバイス ロジックの消費電力要件を慎重に見積もる手法が必要になります。これらの手法については、次で説明します。

- [第 2 章: 消費電力の見積もり: 初期評価段階](#)
- [第 3 章: 消費電力の見積もり: Vivado Design Suite でのデザイン フロー段階](#)



図 2: FPGA の設計プロセスにおける消費電力



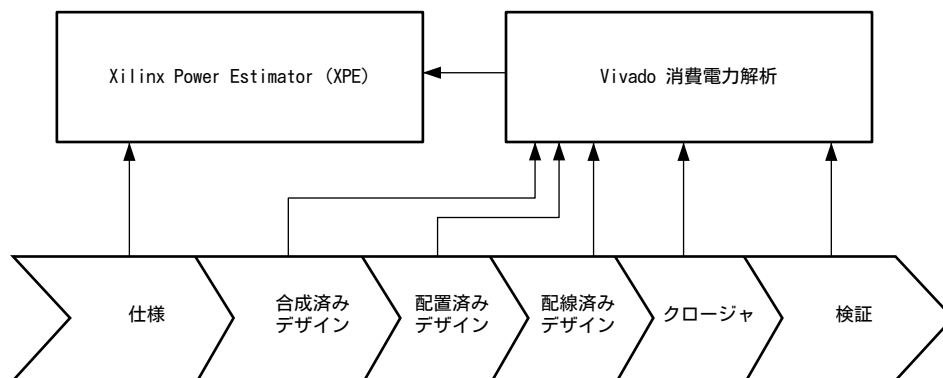
X12129-122719

この後のセクションで、設計プロセスに沿って消費電力を解析および削減する手法を説明していきます。

## ザイリンクス消費電力見積もり、解析、最適化ツール

ザイリンクスでは、FPGA の温度要件および電源要件をデザイン サイクルを通して評価できるソフトウェア ツールおよび資料を提供しています。次の図に、FPGA の各デザイン サイクルで利用できるツールを示します。ツールには、スタンドアロンのものとインプリメンテーション ツールに統合されているものがあり、設計プロセスの各段階で利用できる環境および情報に基づいて実行できます。効率よく解析できるように、すべてのツール間で情報を交換できるようになっています。

図 3: 設計プロセスで利用できる Vivado 消費電力見積もりおよび解析ツール



X12986-122719

## Xilinx Power Estimator (XPE)

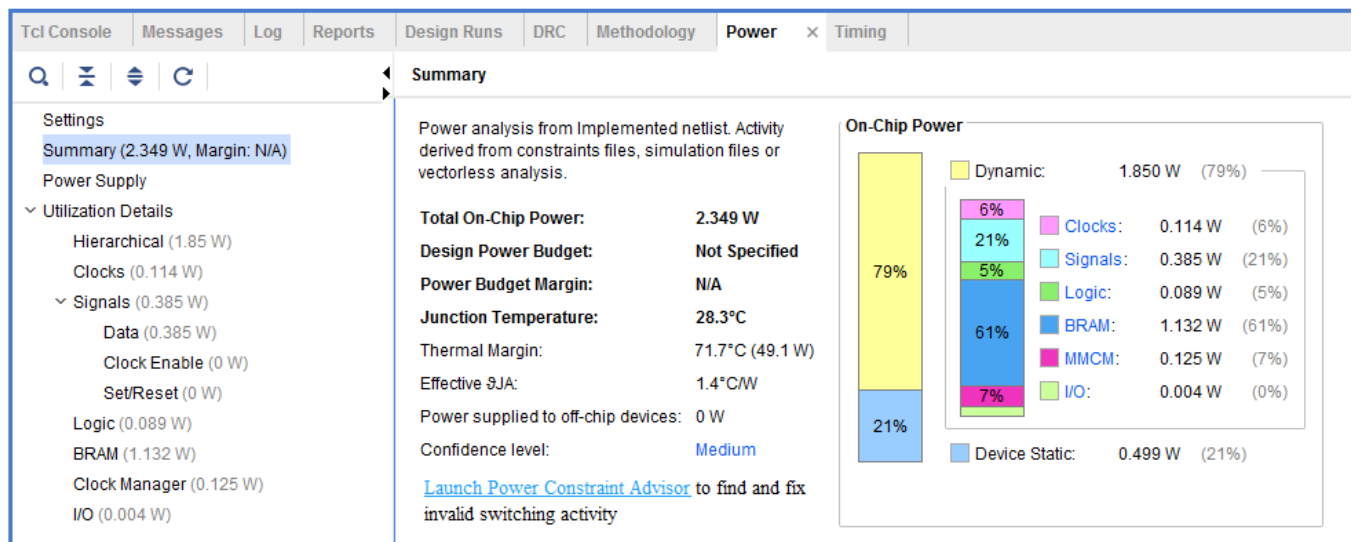
Xilinx® Power Estimator (XPE) は、通常プロジェクトの設計前とインプリメンテーション前の段階で使用されるスプレッドシート形式の消費電力見積もりツールです。XPE はアーキテクチャの評価およびデバイスの選択に利用でき、またアプリケーションに適切な電源や温度管理コンポーネントの選択にも便利です。XPE インターフェイスを使用して、ユーザーが、デザインのリソース使用数、アクティビティ レート、I/O 負荷などのさまざまな要因を指定すると、XPE がその情報をデバイス モデルと組み合わせて、消費電力の見積もり計算を実行します。

また、デザイン サイクル後半のインプリメンテーションや消費電力を決める最終段階に入っても、設計変更指示 (ECO) の消費電力へのインパクトを評価する場合などに、XPE はよく使用されます。複数のチームによりインプリメントされる大型デザインでは、プロジェクト リーダーが XPE を使用して各チームのモジュールの使用量およびアクティビティをインポートして総消費電力を監視し、制約が満たされるように消費電力バジェットを変更できます。Xilinx Power Estimator の使用方法の詳細は、『Xilinx Power Estimator ユーザー ガイド』(UG440: [英語版](#)、[日本語版](#)) を参照してください。

## Vivado 消費電力解析

Vivado® では、合成後、配置後、配線後など、フローのさまざまな段階で消費電力解析を実行できます。配線後には、インプリメント済みデザインから正確なロジック リソースおよび配線リソースが読み出されるので、最も正確に消費電力を見積もることができます。次の図に、消費電力レポートのサマリーと、クロック ドメイン、リソースの種類、デザイン階層など、さまざまなビューを示します。Vivado IDE で環境設定およびデザイン アクティビティを変更し、デザインの電源および熱電力の削減方法を検討できます。消費電力レポートからデザインにクロスプローブすることも可能で、消費電力の大きい階層やリソースを特定および評価するのに便利です。Vivado Design Suite のアーキテクチャ サポートは『Vivado Design Suite ユーザー ガイド: リリース ノート、インストール、およびライセンス』(UG973: [英語版](#)、[日本語版](#)) を参照してください。

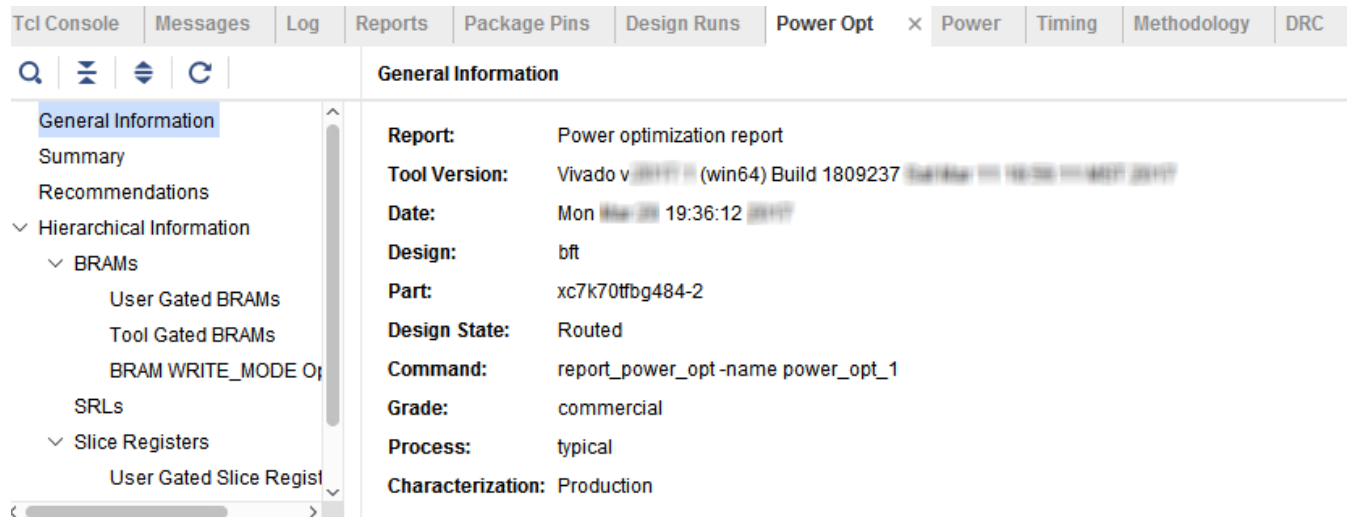
図 4: Vivado 消費電力解析



## Vivado の消費電力最適化

Vivado® デザイン ツールでは、さまざまな消費電力の最適化が提供されており、デザインのダイナミック消費電力を最大 30% 削減できます。これらの最適化は、複雑な ASIC のクロック ゲーティングの同等のテクニックを使用して、デザインの機能に影響を与えずにスイッチング アクティビティを最小限に抑えます。消費電力最適化の適用範囲は、デザイン全体か、選択した部分のみを選択できます。Vivado では、Vivado IDE または Tcl コマンドを使用して消費電力最適化を実行できます。

図 5: Vivado での消費電力最適化レポート



General Information	
Report:	Power optimization report
Tool Version:	Vivado v2019.2 (win64) Build 1809237
Date:	Mon May 20 19:36:12 2019
Design:	bft
Part:	xc7k70tfgb484-2
Design State:	Routed
Command:	report_power_opt -name power_opt_1
Grade:	commercial
Process:	typical
Characterization:	Production

# 消費電力の見積もり: 初期評価段階

---

## 概要

この章では、デザイン サイクルの初期段階でデザインの消費電力を評価する方法を説明します。この段階では、Xilinx<sup>®</sup> Power Estimator を使用します。初期評価が終了している場合は、デザイン サイクルの後の段階でデザインの消費電力を評価する方法を説明する次の章に進んでください。この段階では、消費電力見積もりを自動的に簡単に実行できる Vivado<sup>®</sup> Design Suite を使用します。

---

## Xilinx Power Estimator を使用した正確なワースト ケース消費電力を見積もるための 7 つの手順

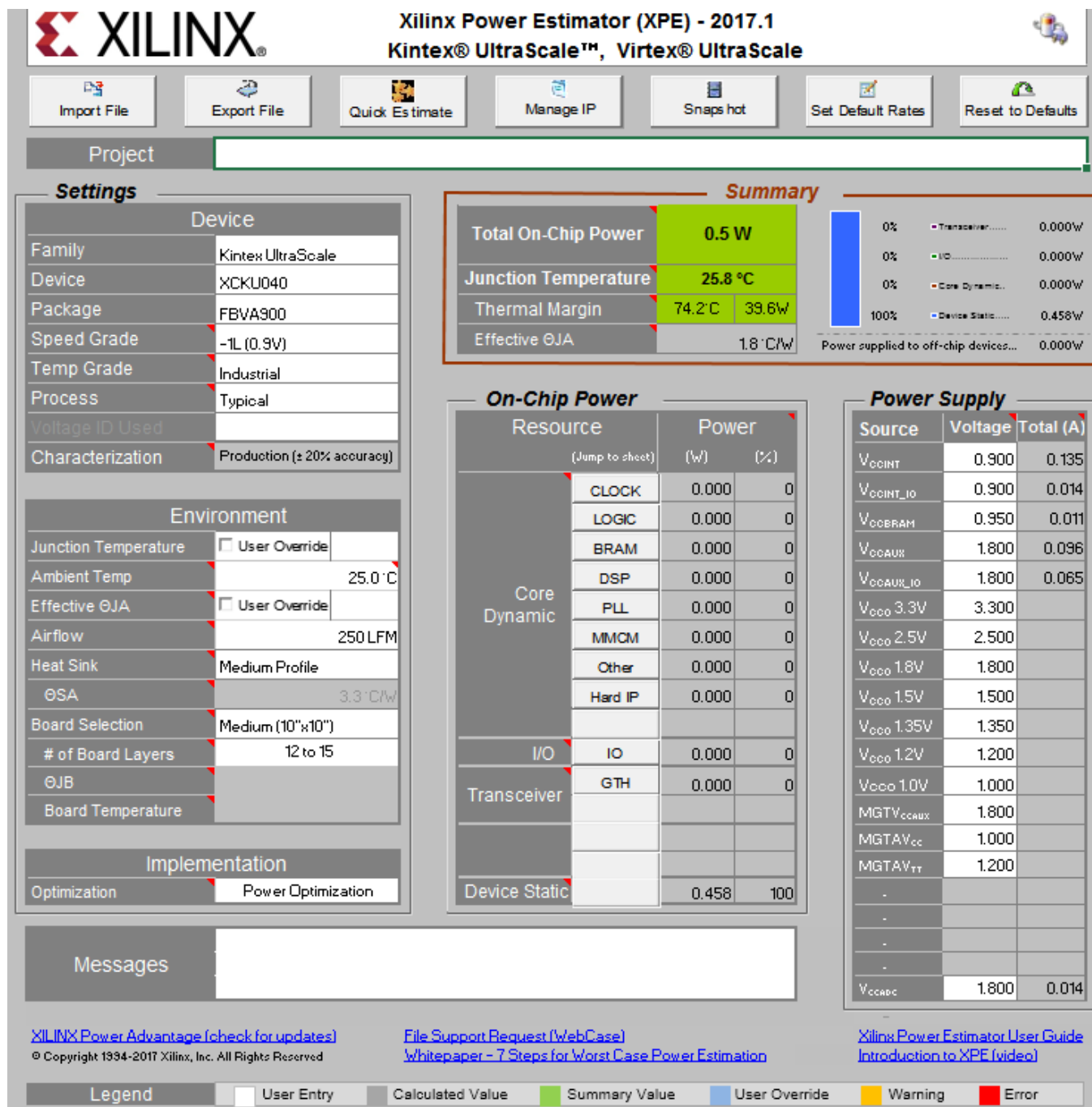
### 消費電力バジェット

この段階では、アプリケーションにとってサイリンクス デバイスが最も効率のよい技術であることが既に決定されています。ここでは、求められる機能、パフォーマンス、コスト、および消費電力に合うベンダー、ファミリ、およびパッケージを選択します。消費電力に関しては、ロジックがまだ 1 つも開発されていない状態でデバイスの総消費電力を見積もる必要があります。総消費電力要件を理解しておく、電力分配および冷却仕様を定義するのに役立ちます。通常、次の点を考慮に入れます。

- 電源はいくつ必要か
- 各電源で使用される電力はどれくらいか
- 吸収されたエネルギーでどれくらいの熱が生成するか

Xilinx<sup>®</sup> Power Estimator を使用すると、これらの質問に対する答えを得ることができます。Xilinx Power Estimator は、デバイス ロジックとデバイスがはんだ付けされるプリント回路基板を同時に開発するのに役立ちます。これにより、必要なマージンを理解し、インプリメント後にシステムを指定範囲内で動作させることができるようになります。次の図に、Xilinx Power Estimator インターフェイスを示します。

図 6: Xilinx Power Estimator (XPE) の消費電力情報サマリ



## Xilinx Power Estimator (XPE) での消費電力見積もり

ザイリンクス デバイス デザインでは、正常に機能する信頼性の高いシステムを作成するため、消費電力仕様および冷却仕様を正しく設定する必要があります。たいていの場合、これらの温度仕様および消費電力仕様は PCB 設計の前に設定する必要があります。ザイリンクス デバイスは柔軟性が高いため、システム デザインまたは PCB 製造の前にデバイス デザインが完了していなかったり、始まってさえいないことがよくあります。デバイス デザインではビットストリーム、クロッキング、およびデバイスのデータフローによって温度特性および電力特性が大きく変化するもので、これは課題となります。

電力または温度システムを過小設計すると、デバイスが仕様範囲外で動作することがあり、デバイスが目的のパフォーマンスで動作しなかったり、深刻な問題が発生したりする可能性があります。電力システムを過剰設計した場合、通常問題はそれほど深刻ではありませんが、不必要にコストが高くなったり、全体的なデバイス デザインが複雑になったりすることがあります。デザインが完了する前に消費電力を見積もるのは簡単なことではありません。

次の手順では、消費電力解析に焦点が置かれています。複数の消費電力の最適化を解析中に試したり適用したりでき、消費電力の大幅な削減につながる可能性があります。消費電力最適化手法については、次の章で説明します。

### 手順 1: ターゲット デバイス用の最新版 Xilinx Power Estimator の入手

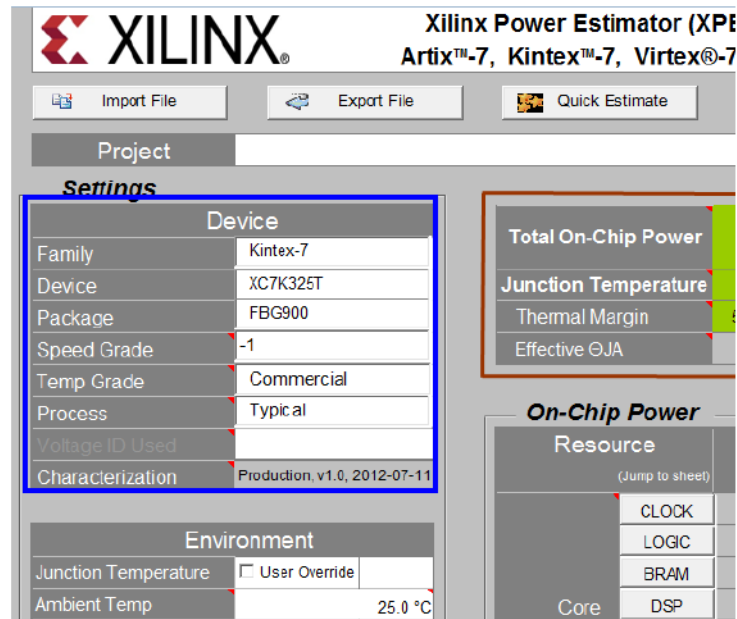
最新版の Xilinx® Power Estimator ツールを使用することが重要です。消費電力情報は、最新の消費電力モデルおよび特性評価データに基づいて随時アップデートされます。

最新バージョンの XPE は、ザイリンクス ウェブサイトの [XPE ダウンロード](#) ページから入手できます。デザイン プロセス中このウェブサイトにとどきアクセスして、新しいバージョンが入手可能であるかどうかを確認してください。新しいバージョンが入手可能な場合は、新しいバージョンの [Summary] シートで [Import File] ボタンをクリックすると、以前のバージョンからデータをインポートできます。XPE を最新の状態にしておくことにより、消費電力解析に常に最新の消費電力情報が使用されます。

### 手順 2: [Summary] シートへのデバイス情報の入力

[Summary] シートの [Device] セクションの各フィールドを正しく設定してください。各設定が、特にスタティック消費電力およびクロッキング消費電力の最終的な値に大きく影響する可能性があります。

図 7: デバイス情報 - 7 シリーズ デバイスの [Summary] シート



[Device] セクションに次の情報を入力します。

- [Family] および [Device]: ファミリーまたはデバイスを正しく選択しないと、クロックのデザイン消費電力など、デバイスおよびデザインの消費電力を正確に見積もることができません。また、使用可能なデバイス リソースも正しくレポートされません。
- [Package]: パッケージの選択は、デバイスの放熱、そして最終的なジャンクション温度に影響します。ジャンクション温度が適切でない場合、デバイスの正しいスタティック消費電力を得られません。
- [Speed Grade] (既知の場合): デザインの要件に最も適したスピード グレードを選択します。一部のデバイス ファミリーでは、スピード グレードによって消費電力仕様が異なります。
- [Temp Grade]: デバイスに適した温度グレードを選択します。通常は [Commercial] または [Industrial] です。設定によっては、スタティック消費電力仕様が異なるデバイスもあります。これを正しく設定すると、選択したデバイスのジャンクション温度範囲が表示されます。
- [Process]: ワースト ケース解析では、[Maximum] に設定することをお勧めします。デフォルト設定は [Typical] で、スタティックに計測した場合に近い結果が得られますが、[Maximum] に変更すると、消費電力仕様がワースト ケースの値に変更されます。
- [Voltage ID Used]: Voltage ID (VID) 電圧は、デバイスがパフォーマンスの仕様を満たしながら動作可能な最小 VCCINT 電圧です。この電圧はデバイスの製造時にテストされており、その値はデバイスの DNA (デバイス識別子) eFUSE レジスタにプログラムされています。VID 機能をデザインで有効にしてデバイスをこの VID 電圧で動作させると、公称電圧での動作時と比較してスタティック消費電力を大幅に削減できます。

**注記:** このオプションは Virtex®-7 で [Speed Grade] を [-1]、[Temp Grade] を [Commercial]、[Process] を [Maximum] に設定した場合にのみ有効になります。

### 手順 3: [Summary] シートへの環境情報の入力

[Summary] シートの [Environment] セクションで適切な環境条件を設定します。



図 8: 環境情報 - 7 シリーズ デバイスのサマリ シート

Process	Typical
Voltage ID Used	
Characterization	Production, v1.0, 2012-07-11

Environment	
Junction Temperature	<input type="checkbox"/> User Override
Ambient Temp	25.0 °C
Effective $\Theta_{JA}$	<input type="checkbox"/> User Override
Airflow	250 LFM
Heat Sink	Medium Profile
$\Theta_{SA}$	3.3 °C/W
Board Selection	Medium (10"x10")
# of Board Layers	12 to 15
$\Theta_{JB}$	
Board Temperature	

Implementation	
Optimization	Power Optimization

On-Chip Power	
Resource	
(Jump to sheet)	
Core Dynamic	CLOCK
	LOGIC
	BRAM
	DSP
	PLL
	MMCM
	Other
I/O	PCIE
	IO
Transceiver	GTX
Device Static	

[Environment] セクションに次の情報を入力します。

- [Junction Temperature] (°C): デバイスのジャンクション温度を指定します。ワースト ケース解析では、[User Override] オプションを使用して、この値をデバイスの温度グレードに基づく TJ (最大) に設定します。
- [Ambient Temp] (°C): デバイス デザインを含むエンクロージャ内で達する可能性のある最大温度を指定します。この設定と、エアフローやその他の放熱経路 (ヒートシンクなど) により、ジャンクション温度が正確に算出され、デバイスのスタティック消費電力がより正確に算出されます。
- [Effective  $\Theta_{JA}$ ] (°C/W): カスタム  $\Theta_{JA}$  (通常熱モデルから求める) の値を指定します。周辺温度および有効  $\Theta_{JA}$  に熱シミュレーションからの値を設定すると、より正確な見積もりが得られます。
- [Airflow] (LFM): チップのエアフローは、LFM (リニア フィート/分) で計測されます。LFM は、CFM (立方フィート/分) で表されたファンの出力を通気経路の断面で除算して求めることができます。デバイスまたはファン (あるいはその両方) の配置が、デバイス上の通気、さらには放熱に影響する可能性があります。このパラメーターのデフォルト値は 250 LFM です。エアフローなし (空気が静止した状態) でデバイスを動作させる場合、デフォルトの 250 LFM を 0 LFM に変更する必要があります。
- [Heat Sink] (使用する場合): ヒートシンクを使用しており、詳細な放熱情報がない場合は、使用しているヒートシンクのタイプに合ったプロファイルを選択します。このパラメーターをほかのパラメーターと一緒に使用すると、有効  $\Theta_{JB}$  を計算しやすくなり、より正確なジャンクション温度および静止消費電力が計算されます。ソケットの設計および構造によっては、ヒートシンクとして機能するものもあります。
- [Board Selection] および [# of Board Layers]: ボードのおおよそのサイズとスタックを選択すると、ボード自体の熱伝導性が考慮されるので、有効な  $\Theta_{JB}$  を計算できます。
- [ $\Theta_{JB}$ ]: ボードおよびシステムのより正確な熱モデルがある場合は、FPGA からの放熱量を指定するために [ $\Theta_{JB}$ ] (プリント回路基板の熱抵抗) を指定する必要があります。

[ $\Theta_{JB}$ ] に正確なカスタム値を指定すると、ジャンクション温度が正確に見積もられるため、デバイスのスタティック消費電力が正確に算出されます。



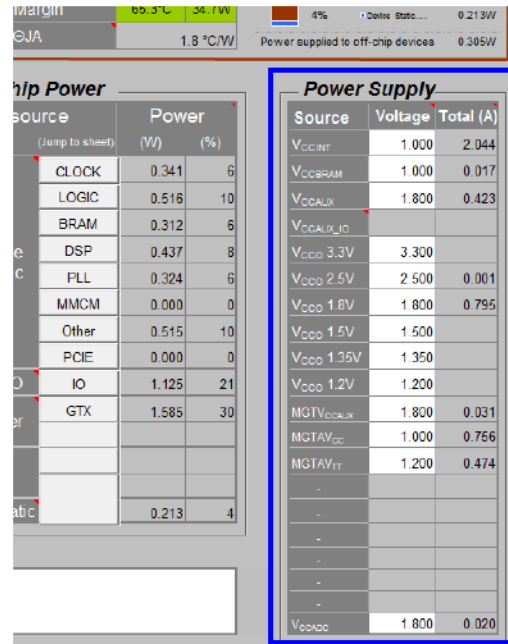
**重要:** [ $\Theta_{JB}$ ] にカスタム値を指定するには、[Board selection] を [Custom] に設定する必要があります。カスタム [ $\Theta_{JB}$ ] を指定する場合は、正確な消費電力を算出するため、[Board temperature] も指定する必要があります。



## 手順 4: すべての電源に対してワースト ケースの電源電圧を設定

デフォルトでは、デバイスの各電圧レールは公称値に設定されます。正確な消費電力見積もりを得るには、デバイスでのワースト ケースの値 (最大電圧値) を指定する必要があります。これは通常、各レールへの電源およびレギュレータの公称出力値と許容誤差を使用して算出されます。特に未調整の電源で IR (電圧) が大きく低下する可能性がある場合は、電圧降下を最大電圧の計算に含める必要があります。一部の  $V_{CCO}$  または MGT 電源を使用しない場合は、これらの電源の値はデフォルトのままにします。

図 9: 7 シリーズ デバイスの [Summary] シートの電源情報



Source	Voltage	Total (A)
V <sub>CCINT</sub>	1.000	2.044
V <sub>CCSRAM</sub>	1.000	0.017
V <sub>CCALX</sub>	1.800	0.423
V <sub>CCALX_IO</sub>		
V <sub>CCO</sub> 3.3V	3.300	
V <sub>CCO</sub> 2.5V	2.500	0.001
V <sub>CCO</sub> 1.8V	1.800	0.795
V <sub>CCO</sub> 1.5V	1.500	
V <sub>CCO</sub> 1.35V	1.350	
V <sub>CCO</sub> 1.2V	1.200	
MGT <sub>VCCALX</sub>	1.800	0.031
MGT <sub>AVCC</sub>	1.000	0.756
MGT <sub>AVTT</sub>	1.200	0.474
V <sub>CCOCC</sub>	1.800	0.020

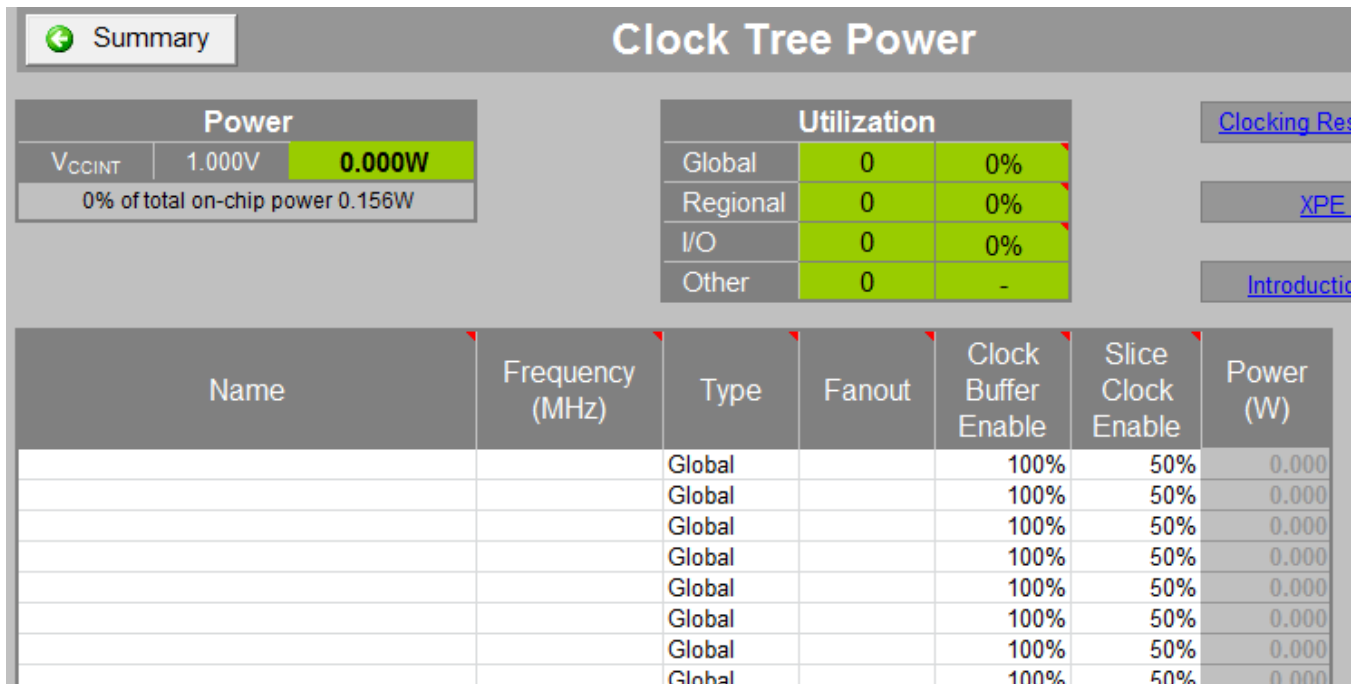
## 手順 5: クロックおよびリソース情報の入力

デザインを Vivado® ツールで既に実行している場合や、デザインの以前のリビジョンがあり解析の開始点として使用できる場合は、デザインからの XPower エクスポート ファイル (.xpe) を XPE にインポートしてリソース情報を入力できます。これを実行するには、[Summary] シートで [Import File] ボタンをクリックします。Vivado XPE インポート ファイルを読み込んだ場合でも、データが正しいことを確認してください。インポートした情報は、完全な情報ではなく、情報を入力するための開始点と考えてください。各リソースのタブを確認し、必要に応じてデザインで使用されるリソースを入力します。

**注記:** XPE では、消費電力の数値のセルは小数点以下 3 桁まで (例: 0.000) 表示されるよう設定されています。小数点以下 3 桁への丸めは、Microsoft Excel の動作に基づいて実行されます。1 mW 未満の値は、0.000W として表示されます。セルをコピーして [User] シートに貼り付けると、精度が調整された実際の値が表示されます。

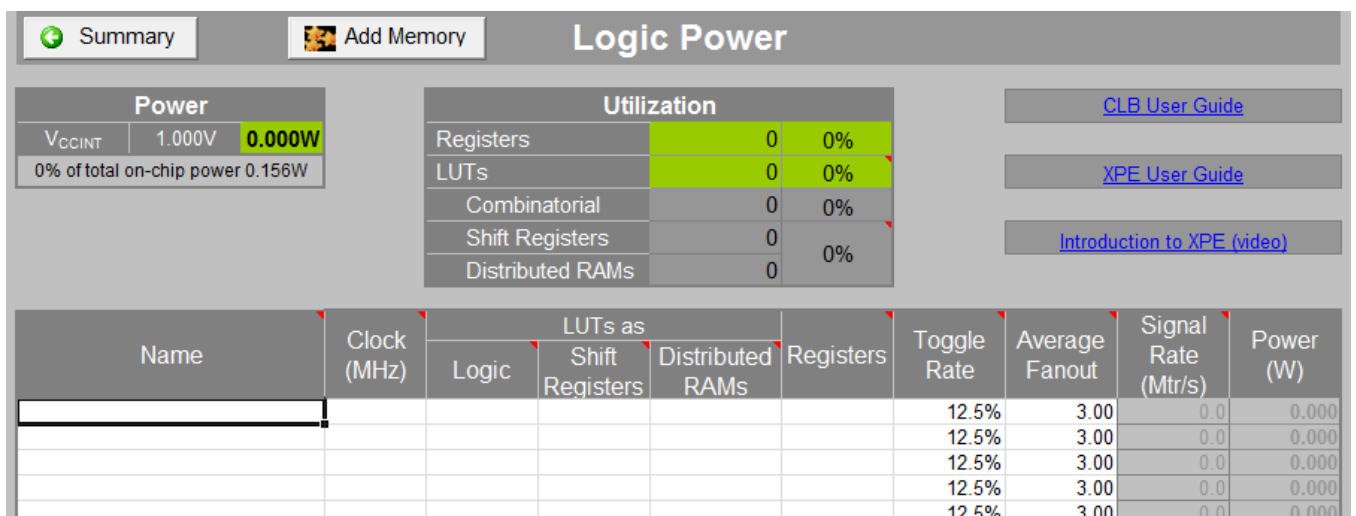
- クロック ツリーの消費電力 ([Clock Tree Power]): [Clock] シートでは、各クロック、予測される周波数 ([Frequency])、使用されるクロック リソースを入力します。使用されるクロック リソースが不明な場合は、[Type] をデフォルトの [Global] のままにします。この時点では、ファンアウト ([Fanout]) を気にする必要はありません。ファンアウトについては、手順 6 で説明します。[Clock Buffer Enable] は [100%]、[Slice Clock Enable] は [50%] のデフォルト値のままにします。

図 10: 7 シリーズ デバイスの [Clock] シート



- ロジックの消費電力 ([Logic Power]): [Logic] シートでは、使用されるスライス リソースの見積もり (数) を入力します。[LUTs as] 列には、演算またはロジックに使用される LUT 数 ([Logic])、SRL としてコンフィギュレーションされる LUT 数 ([Shift Registers])、メモリとしてコンフィギュレーションされる LUT 数 (Distributed RAMs) を入力します。[Registers] 列には、デザインにコンフィギュレーションされるレジスタまたはラッチの数を入力します。異なるロジック ファンクションまたは特性 (クロック スピード、トグル レートなど) に個別の行を使用してください。

図 11: 7 シリーズ デバイスの [Logic] シート



ザイリンクスでは、デバイス デザインの初期段階でリソースの正確な数を把握することは困難なので、多めに見積もった値を使用し、デザイン プロセスを進めながら、値を更新してより正確な見積もり値が得られるようにします。



**ヒント:** クロック周波数情報を入力する際は、Excel のセルを関連付ける機能を使用し、[Clock] シートに入力されているセルと関連付けます。これには、[Logic] シートで [Clock (MHz)] セルを選択し、「=」と入力して、そのロジックのクロックソースに関連するセルを [Clock] シートで選択します。これで、そのセルに [Clock] シートの値が自動的に入力されるようになります。こうしておく、仕様の変更や消費電力と周波数のトレードオフを検討するときなど、クロック周波数を変更する必要がある場合に、値を 1 箇所変更するだけですみます。また、データの入力ミスも避けることができます。

- I/O の消費電力 ([I/O Power]): チップのすべてのレールの全体的な見積もり値を正確なものにするには、[IO] シートを入力することが重要です。選択した I/O 規格および I/O 回路によっては、VCCO レールだけでなく、VCCINT および VCCAUX レールでも多くの電力が消費される可能性があります。たいていの場合は、各デバイス インターフェイスを個別に入力し、またインターフェイス信号をデータ、制御、クロック信号に分けておくのが一番簡単です。こうしておく、異なる I/O 規格はもちろん、ロードやトグル レートなどの I/O 特性を指定しやすくなります。



**推奨:** XPE で [Add Memory Interface] をクリックし、[XPE Memory Interface Configuration] ダイアログ ボックスを使用すると、複雑なメモリ インターフェイスに関連する I/O を簡単に追加できます。

図 12: 7 シリーズ デバイスの [IO] シート

Summary

Add Memory Interface

Active Current

Source	On-Chip	Off-Chip	Active
VCCINT	1.000V	0.000A	0.000A
VCCAUX	1.800V	0.000A	0.000A
VCCAUX_IO	2.000V		
VCC033	3.300V	0.000A	0.000A
VCC025	2.500V	0.000A	0.000A
VCC018	1.800V	0.000A	0.000A
VCC015	1.500V	0.000A	0.000A
VCC0135	1.350V	0.000A	0.000A
VCC012	1.200V	0.000A	0.000A

Summary

Power (on-chip)

0.000W

→ Logic

0.000W

→ Buffer

0.000W

0% of total on-chip power 0.156W

Power (off-chip)

0.000W

I/O Count

0

I/O Utilization

0%

→ High Performance

0%

→ High Range

0%

Select I/O User Guide

XPE User Guide

Introduction to XPE (video)

Banks With Internal Vref

0

I/O Delay Controllers

0

Show External Board Termination Settings

Name	Bank	I/O Type	I/O Settings							Activity							Output Load (pF)	Signal Rate (Mtr/s)	On Chip Power (W)				Off Chip
			I/O Standard	Input Pins	Output Pins	Bidir Pins	I/O LOGIC SERDES	I/O DELAY	IBUF	Input Term	Clock (MHz)	Toggle Rate	Data Rate	Output Enable	Term Disable	IBUF Disable			VCCINT	VCCAUX	VCCAUX_IO	VCCO all rails	VCCO all rails
	HP		LVCMOS 1.8V 12mA (Slow)				No	Off	Low Power			12.5%	SDR				0.0	0.000	0.000	0.000	0.000	0.000	
	HP		LVCMOS 1.8V 12mA (Slow)				No	Off	Low Power			12.5%	SDR				0.0	0.000	0.000	0.000	0.000	0.000	
	HP		LVCMOS 1.8V 12mA (Slow)				No	Off	Low Power			12.5%	SDR				0.0	0.000	0.000	0.000	0.000	0.000	
	HP		LVCMOS 1.8V 12mA (Slow)				No	Off	Low Power			12.5%	SDR				0.0	0.000	0.000	0.000	0.000	0.000	
	HP		LVCMOS 1.8V 12mA (Slow)				No	Off	Low Power			12.5%	SDR				0.0	0.000	0.000	0.000	0.000	0.000	
	HP		LVCMOS 1.8V 12mA (Slow)				No	Off	Low Power			12.5%	SDR				0.0	0.000	0.000	0.000	0.000	0.000	

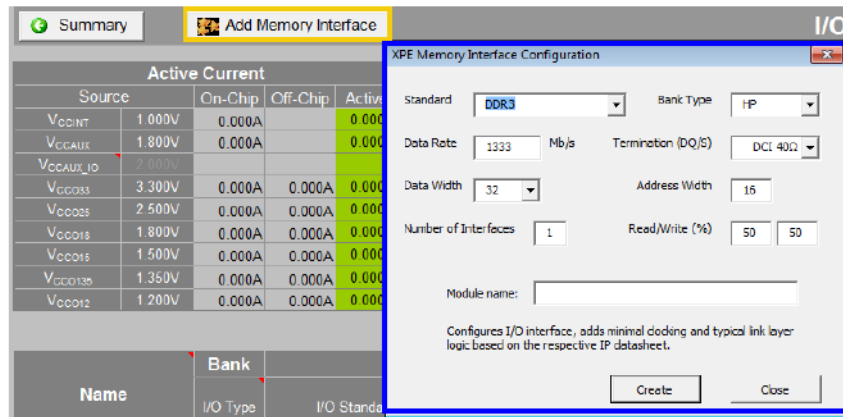
I/O 電流計算では、標準ボード トレースおよび終端が適用されていると想定した上で消費電力が見積もられます。



**ヒント:** 差動 I/O を使用している場合、各入力および出力をペアで指定する必要があります。スプレッドシートで 1 つの差動入力に対して 2 つの入力を指定しないでください。

DDR 規格などの複雑な規格のデータを入力するには、[XPE Memory Interface Configuration] ダイアログ ボックスを使用すると簡単です。このダイアログ ボックスで関連オプションを入力すると、[IO] シートの該当行に情報が自動入力されます。

図 13: [IO] シートの [XPE Memory Interface Configuration] ダイアログ ボックス



- ブロック RAM の消費電力 ([Block RAM Power]): [BRAM] シートでは、デザインで使用するブロック RAM の数と設定を入力します。[Enable Rate] を ENA または ENB ポートがイネーブルになる時間の割合に合わせて変更してください。RAM がイネーブルの時間はダイナミック消費電力に直接比例するので、ブロック RAM の消費電力を正しく見積もるには、このパラメーターに適切な値を入力することが重要です。BRAM の [Mode] の設定が消費電力見積もりにどのように影響するかについては、『Xilinx Power Estimator ユーザー ガイド』 (UG440: [英語版](#)、[日本語版](#)) の「BRAM モードの設定による精度の向上」を参照してください。



**推奨:** XPE で [Add Memory] をクリックし、[XPE Memory Configuration] ダイアログ ボックスを使用すると、デザインのブロック RAM を簡単に追加できます。

図 14: [BRAM] シート

Summary

Add Memory

Block RAM Power

Power

V<sub>CCINT</sub>1.000V0.000W

V<sub>CCBRAM</sub>1.000V0.000W

0% of total on-chip power 0.156W

Utilization

RAMB1800%

RAMB3600%

Memory Resources User Guide

XPE User Guide

Introduction to XPE (video)

Name	Block RAMs	Mode	Toggle Rate	Clock (MHz)	Enable Rate	Bit Width	Port A			Port B			Signal Rate (Mtr/s)	Power (W)		
							Write Mode	Write Rate	Clock (MHz)	Enable Rate	Bit Width	Write Mode		Write Rate	V <sub>CCINT</sub> 1.000V	V <sub>CCBRAM</sub> 1.000V
		RAMB18	50.0%		25.0%	1	NO_CHANGE	50.0%		25.0%	1	NO_CHANGE	50.0%	0.000	0.000	0.000
		RAMB18	50.0%		25.0%	1	NO_CHANGE	50.0%		25.0%	1	NO_CHANGE	50.0%	0.000	0.000	0.000
		RAMB18	50.0%		25.0%	1	NO_CHANGE	50.0%		25.0%	1	NO_CHANGE	50.0%	0.000	0.000	0.000
		RAMB18	50.0%		25.0%	1	NO_CHANGE	50.0%		25.0%	1	NO_CHANGE	50.0%	0.000	0.000	0.000
		RAMB18	50.0%		25.0%	1	NO_CHANGE	50.0%		25.0%	1	NO_CHANGE	50.0%	0.000	0.000	0.000
		RAMB18	50.0%		25.0%	1	NO_CHANGE	50.0%		25.0%	1	NO_CHANGE	50.0%	0.000	0.000	0.000

- UltraRAM の消費電力 ([UltraRAM Power]): [URAM] シートでは、デザインで使用する UltraRAM の数と設定を入力します。ダイナミック消費電力に最も影響する可能性のある設定 ([Cascade Group Size]、[Input Toggle Rate]、[Output Toggle Rate]、[Enable Rates]、[Write Enable] など) には現実的な値を使用してください。UltraRAM の消費電力見積もりの詳細は、『Xilinx Power Estimator ユーザー ガイド』 (UG440: [英語版](#)、[日本語版](#)) を参照してください。

図 15: [URAM] シート

Summary

Power

V<sub>CCINT</sub>0.720V0.085W

V<sub>CCBRAM</sub>0.850V0.000W

0% of total on-chip power 1.200W

Utilization

URAM2882827%

Memory Resources User Guide

XPE User Guide

Introduction to XPE (video)

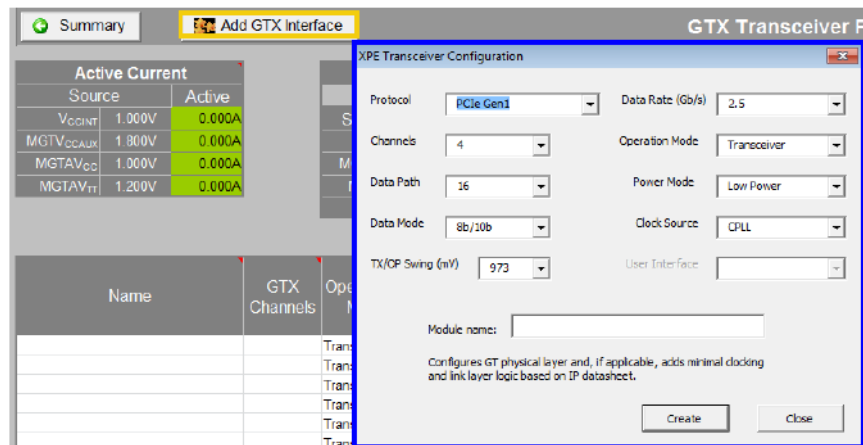
Name	URAMs	Cascade Group Size	Latency	Mode	Sleep Rate	Avg Inactive Cycles	Input Toggle Rate	Output Toggle Rate	Clock (MHz)	Port A			Port B			Power (W)	
										Data Width	Enable Rate	Write Enable	Data Width	Enable Rate	Write Enable	V <sub>CCINT</sub> 0.720V	V <sub>CCBRAM</sub> 0.850V
URAM0	16	4	1	URAM288	0.0%	10	12.5%	12.5%	300.0	72	25.0%	12.5%	72	25.0%	12.5%	0.048	0.000
URAM1	12	4	1	URAM288	0.0%	10	12.5%	12.5%	300.0	72	25.0%	12.5%	72	25.0%	12.5%	0.036	0.000
		1	0	URAM288	0.0%	10	12.5%	12.5%	72	25.0%	12.5%	72	25.0%	12.5%	0.000	0.000	
		1	0	URAM288	0.0%	10	12.5%	12.5%	72	25.0%	12.5%	72	25.0%	12.5%	0.000	0.000	

- DSP の消費電力 ([DSP Power]): [DSP] シートを入力します。DSP ブロックは、SP ブロックは、カウンター、パレルシフター、MUX といった一般的なファンクションなど、乗算器以外の目的でも使用できます。
- クロック マネージャーの消費電力 ([Clock Manager Power]): デザインで MMCM または PLL が使用されている場合は、[CLKMGR] シートでその使用法と設定を指定します。
- [GT]: デザインで GT (シリアル トランシーバー) が使用されている場合は、[GT] シートでその使用法と設定を指定します。



**推奨:** [Add GTX Interface] ボタンをクリックし、[XPE Transceiver Configuration] ダイアログ ボックスを使用すると、データを簡単に正確に入力できます。

図 16: [XPE Transceiver Configuration] ダイアログ ボックスを使用した GT のコンフィギュレーション



## 手順 6: トグルおよび接続パラメーターの設定

[Toggle Rate]、[Average Fanout]、または [Enable Rate] を含むタブで、設定値を確認します。トグル レートおよびイネーブル レートがわかっていない場合は、サイリンクスではデフォルト値をそのまま使用することをお勧めしますが、デフォルト値がデザインの特性を表していない場合は、適宜変更してください。たとえば、メモリ インターフェイスにインターフェイス上のトグル レートを高く保つトレーニング パターン ルーチンがあることがわかっている場合は、これを反映してトグル レートを高くします。また、回路のクロック イネーブルが回路全体のアクティビティを削減するように指定されている場合は、トグル レートを低くします。トグル レートの決定方法の詳細は、『Xilinx Power Estimator ユーザー ガイド』 (UG440: [英語版](#)、[日本語版](#)) を参照してください。

クロック ファンアウトに対しては、特定のクロック ドメインにあるすべての同期エレメントを加算する式を作成するのが、XPE では最も簡単な方法です。たとえば、あるクロックの [Fanout] セルを選択し、「=SUM(」と入力して、そのクロックが供給される同期エレメント (BRAM、フリップフロップ、シフト レジスタ、SelectRAM など) の数を指定するセルをすべて選択します。選択し終わったら閉じかっこを入力します。これで、[Fanout] セルに適切な値が入力されるようになります。クロック ファンアウトをこのように入力するのは簡単で、また、リソース数が変更された場合に自動的にアップデートされるので便利です。最終的な式は、次のようになります。

```
=SUM(LOGIC!I12:I15, BRAM!E10:E12, DSP!E8, CLKMGR!E10:E12)
```

ロジックのファンアウトでは、データパスおよび制御パスの特性を考慮する必要があります。DSP デザインなど、順次データパスが適切に構築されているデザインでは、ファンアウトは通常デフォルト値よりも小さくなります。エンベデッド デザインなど、データ実行パスが多いデザインでは、ファンアウトはデフォルト値よりも大きくなる場合があります。トグル レートと同様、この情報がわからない場合は、デフォルト値のままにし、後で適宜調整します。

I/O の [Output Load] には、各デザイン出力の単純な容量性負荷を入力します。この値は、駆動される出力のダイナミック消費電力に影響します。[Output Load] の値は、主にその出力に接続されている各デバイスの入力容量の合計から求められます。入力容量は通常、デバイス I/O が接続されているデバイスのデータシートに記載されています。

## 手順 7: 結果の解析

結果を解析する前に、必要に応じて手順 1～6 を実行してください。これらの手順を完了したら、結果を解析します。ジャンクション温度が範囲を超えておらず、消費電力がプロジェクトの消費電力バジェットを超過していないことを確認してください。放熱または電力特性が目標値内にない場合は、適切な結果が得られるように環境特性 (エアフロー、ヒートシンクなど) を調整するか、デザインのリソースおよび電力特性を調整してください。厳しい消費電力要件を満たしながらも必要とする機能を得るには、トレードオフを考慮する必要がありますが、そのトレードオフを検討するのは、デザイン プロセスの初期段階がベストです。データの入力完了し、選択したグレードの制限温度範囲内でデバイスが動作している場合は、XPE でレポートされた消費電力を使用して、デザインの電源レールを決定できます。入力したデータの信頼性が高くない場合は、デバイスの消費電力システムを過小設計しないように、数値を大きくします。データの信頼性が十分高い場合は、ツールでレポートされた数値を増加する必要はありません。

デザイン プロセスを進めながら、スプレッドシートの情報を確認して更新し、最新の要件およびインプリメンテーションの詳細が反映されるようにします。こうすることでデザインで消費される電力の最新に見積もることができ、消費電力要件の調整が必要かどうかを早期に判断できます。

デザイン サイクルの後半でデザインの消費電力を評価する方法は第 3 章: 消費電力の見積もり: Vivado Design Suite でのデザイン フロー段階を、デザインの消費電力を削減する手法は第 6 章: 消費電力削減のためのヒントおよび手法を参照してください。



# 消費電力の見積もり: Vivado Design Suite でのデザイン フロー段階

## 概要

この章では、デザイン フロー段階での消費電力見積もりを自動的または簡単にする Vivado® Design Suite の機能を説明します。Vivado Design Suite で消費電力を見積もり、解析したら、[第 6 章: 消費電力削減のためのヒントおよび手法](#)に進み、デバイスの消費電力を最小限に抑えるため、システムを調べて変更する手法を学びます。

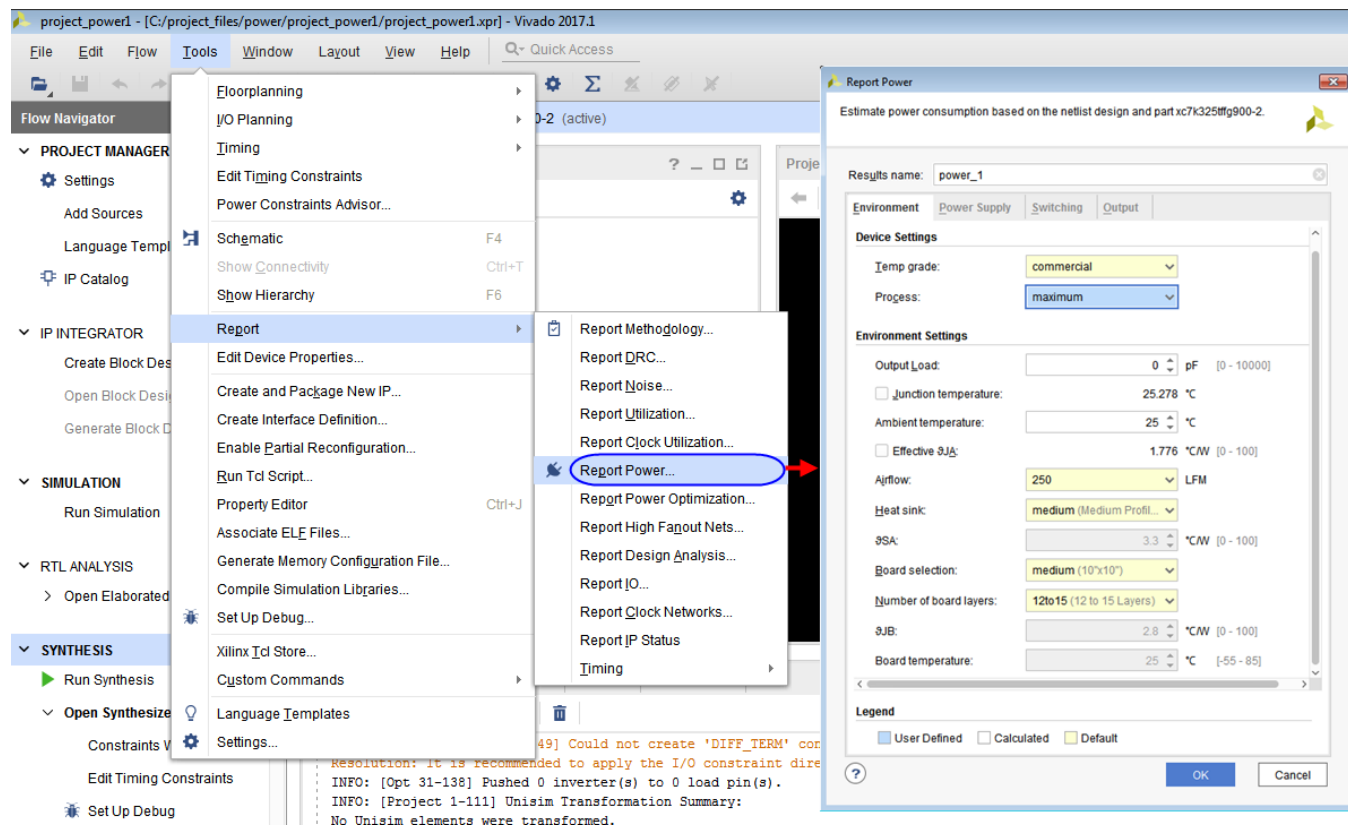
## 消費電力見積もり

合成およびインプリメンテーション段階に入ると、消費電力を定期的にモニターし、検証する必要があります。放熱量が要件のバジェット内に収まっていることを確認して、制限に近づいているエリアを早期に発見して対処できるようにする必要があります。消費電力見積もりの正確さは、デザインのどの時点で見積もられたかによって変わります。

## Vivado IDE での消費電力の見積もり

このセクションでは、Vivado® IDE の [Report Power] コマンドを使用した消費電力解析について説明します。合成後に初めて消費電力解析を設定すると想定しているので、まずツールにアクティビティ 情報を入力します。その後の run で、Vivado IDE の [Report Power] を使用して消費電力レポートを表示するか、または同等の Tcl コマンド (`report_power`) を使用して、Vivado IDE を使用せずにテキスト形式の消費電力レポートを表示するかを選択できます。次の図に、Vivado IDE での消費電力解析の実行方法を示します。

図 17: Vivado での消費電力解析: 解析用のデータを入力



## Vivado IDE からの消費電力解析の設定

[Power Analysis] ウィンドウで環境、アクティビティ、電源、およびツール デフォルトを指定するには、次の手順に従います。

1. [Flow] → [Open Synthesized Design or Flow] → [Open Implemented Design] をクリックします。  
Flow Navigator から同じ作業ができます。
2. [Reports] → [Report Power] をクリックします。  
または、Flow Navigator で [Report Power] をクリックします。
3. [Report Power] ダイアログ ボックスで、デバイスの環境およびツール設定を指定します。
  - [Report Power] ダイアログ ボックスの各タブで、デザインの環境に合わせて設定を変更します。
  - 環境および電圧設定は、デバイスのスタティック消費電力に大きく影響します。
  - アクティビティ レートおよび電圧設定は、ダイナミック消費電力に大きく影響します。
  - どの値に設定すべきか不明の場合は、デフォルトの値を使用してください。
  - シミュレーション結果からのアクティビティ ファイルがある場合、このダイアログ ボックスで指定できません。

設定の詳細は、[デバイス/デザイン設定を確認して既知の要素のアクティビティを調整](#)を参照してください。



4. レポート名を指定します。

## Vivado IDE からの消費電力解析の実行

[Report Power] ダイアログ ボックスで [OK] をクリックし、消費電力解析を開始します。ツールは次を実行します。

1. 環境、デバイス、およびツール オプションを考慮します。
2. ネットリストの接続および設定を読み込みます。
3. 定義されたノードのアクティビティを適用します。  
ノードとは、ネット、ピン、ポートなどのコンポーネントのことです。
4. 熱電力および電源を計算する前に、残りの未定義ノードのアクティビティを決定します。

消費電力解析では、アクティビティの定義に次の情報が使用されます。

- シミュレーション ファイル (SAIF)
- ベクターレス消費電力解析手法を使用した自動計算
- `set_switching_activity Tcl` コマンドを使用した手動定義

詳細は、[Tcl コンソールからの消費電力解析の実行](#)を参照してください。

## ベクターレス (確率) の見積もり

デザイン ノードのアクティビティが、ユーザーまたはシミュレーション結果から得られない場合は、ベクターレス消費電力見積もりアルゴリズムでこのアクティビティを予測できます。ベクターレス エン진은、未定義ノードすべてに初期シード (デフォルトの信号レートおよびスタティック確率) をまず割り当てます。それから、デザインのプライマリ入力から内部ノードの出力へとアクティビティを伝搬し、プライマリ出力に到達するまでこの操作を繰り返します。このアルゴリズムでは、デザインの接続性、リソースの機能、およびコンフィギュレーションが考慮されています。その経験則から、ネットリストに含まれるどのノードのグリッチ レートも見積もることができます。グリッチは、デザイン エレメントがアクティブなクロック エッジ間で数回状態が変わり、最終的な値に安定するまでに発生します。ベクターレス伝搬エンジンは、適当な長さの現実的なスティミュラスを使用する配線後のシミュレーションほど正確ではありませんが、精度と計算速度のバランスが取られている点で優れています。

**注記:** ベクターレスの消費電力見積もりでは、GT の出力ポートにアクティビティが伝搬されません。デザイン ロジックがこれらのアクティビティ レートに依存する場合は、正確な解析結果を得るため、`set_switching_activity -type <rx_data|tx_data>` コマンドで GT 出力のアクティビティ レートを明示的に指定する必要があります。



**ヒント:** ベクターレスの消費電力見積もりは、スイッチング レートおよびスタティック確率を上書きしない限り、デザインの平均的な消費電力見積もりになります。

### ベクターレス見積もりを改善するユーザー入力

特定のノードのアクティビティは、システム仕様またはデバイスが通信するインターフェイスによって決定されるので、どのデザインでもわかっているのが一般的です。特にデバイスの複数のセルを駆動するノード (セット、リセット、クロック イネーブル、またはクロック信号) に対してアクティビティをツールに入力すると、消費電力をより正確に見積もることができます。ノードには次のものがあります。

- クロック アクティビティ: すべてのデバイス クロック ドメインの正確な周波数、クロックが外部から供給されるのか (入力ポート)、内部で生成されるのか、または外部からプリント回路基板に供給されるのか (出力ポート) は、通常わかっています。デザインには、`create_clock` 制約を使用してクロックを少なくとも 1 つ定義しておく必要があります。クロックがまったく定義されていない場合、[Report Power] を実行すると警告メッセージが表示され、スイッチング アクティビティの計算に 10 GHz クロック周波数が使用されます。

- I/O データ ポート: デバイスに入出力するデータのプロトコルおよびフォーマットがわかっている場合、少なくとも一部の I/O に対して信号の遷移レートまたは信号のスタティック確率をツールで指定できます。たとえば、プロトコルに DC バランス要件 (信号のスタティック確率が 50%) がある場合や、メモリ インターフェイスに対するデータの書き込みおよび読み出し頻度がわかっている場合は、ストロブ信号およびデータ信号のデータ レートを設定できます。プライマリ入力にユーザー アクティビティ レートが指定されていない場合、[Report Power] を実行するとデフォルトのスタティック確率 0.5 とデフォルトのトグル レート 12.5% が割り当てられます。
- I/O および内部制御信号: システムおよびその機能から、セット、リセット、クロック イネーブルなどの制御信号のアクティビティを予測できる場合があります。これらの信号は、通常、デザイン ロジックの広い範囲をオン/オフできるので、このアクティビティ 情報を入力すると、より正確に消費電力を見積もることができるようになります。プライマリ入力のリセットである (順次エレメントの RESET ピンに直接接続されている) 場合は、スタティック確率 0、信号レート 0 のデフォルト値が割り当てられます。同様に、プライマリ入力クロック イネーブルである (順次エレメントの CE ピンに直接接続されている) 場合は、スタティック確率 0.99、信号レート 2 のデフォルト値が割り当てられます。



**推奨:** 特にデバイスの複数のセルを駆動するノード (セット、リセット、クロック イネーブル、またはクロック 信号) に対してアクティビティをツールに入力すると、消費電力をより正確に見積もることができます。

**注記:** ベクターレスの消費電力見積もりでは、GT の出力ポートにアクティビティが伝搬されません。デザイン ロジックがこれらのアクティビティ レートに依存する場合は、正確な解析結果を得るため、`set_switching_activity -type gt_txdata|gt_rxdata` コマンドで GT 出力のアクティビティ レートを明示的に指定する必要があります。

## ベクター (SAIF) ベースの消費電力見積もり

デザイン開発の各段階でシミュレーションを実行し、デザインが要件どおりに動作するかを検証します。検証方法には、デザインの開発段階、複雑さ、または企業の方針に応じて、さまざまなものがあります。次のセクションでは、収集可能な重要なデータについて、消費電力解析にこのデータを使用した場合によくあるミスについて説明します。消費電力を正確に見積もるには、デザインのアクティビティ レートを現実的なものにすることが重要です。アクティビティ レートが、シミュレーションされるブロックに入力されるデータの標準動作またはワースト ケースの動作を表すようにします。このような情報は、機能の検証中に提供されるとは限りません。場合によっては、無効なデータやコマンドが入力されたときにシステムがそれに対処し、安定した状態を保つことができることを検証するため、無効なデータが入力されることもあります。そのようなテスト ケースを使用して消費電力解析を実行すると、デザイン ロジックが標準システム動作でシミュレーションされないため、消費電力を正確に見積もることができない可能性があります。

- システム トランザクション レベル: デザイン サイクルの初期段階で、PCB 上のデバイス間のトランザクション、またはデバイス アプリケーションの異なるファンクション間のトランザクションを記述している場合があります。この記述から、一部の I/O ポートとほとんどのクロック ドメインのアクティビティをファンクション ブロックごとに抽出できます。この情報は、Xilinx® Power Estimator スプレッドシートに入力するときに役立ちます。
- デバイス記述レベル: アプリケーションの RTL を定義しながら、ビヘイビアー シミュレーションを実行して機能を検証する必要がある場合があります。これは、データフローおよびクロック サイクルに対する計算の有効性を検証するのに役立ちます。この段階では、使用されるデバイス リソース、その数、およびコンフィギュレーション データはありません。手動でリソース使用量を推定して、I/O ポートまたは内部制御信号 (セット、リセット、クロック イネーブル) のアクティビティを抽出できます。この情報は、Xilinx Power Estimator スプレッドシートの情報を調整するのに使用できます。シミュレータでノード アクティビティを抽出し、SAIF ファイル フォーマットでエクスポートできます。このファイルを保存しておくと、後の Vivado® デザイン フローでより正確な消費電力解析をするときに使用できます。たとえばインプリメンテーション後のシミュレーションを実行しない場合は、配置配線後にこのファイルを使用できます。
- デバイス インプリメンテーション レベル: インプリメンテーション プロセスのさまざまな段階でシミュレーションを実行すると、消費電力に関連して、それぞれに異なる結果が得られるので、それを抽出しておくことができます。この追加情報を使用すると、Xilinx Power Estimator スプレッドシートおよび Vivado 消費電力解析をより正確に見積もることができます。また、I/O ポートおよび特定のモジュールのアクティビティを保存することも可能で、合成後、配置後、配線後の Vivado 消費電力解析で再利用できます。
- 合成後: ネットリストがターゲット デバイスで使用可能な実際のリソースにマップされます。

- 配置後: ネットリスト コンポーネントが実際のデバイス リソースに配置されます。このパッキング情報から、最終的なロジック リソース数および設定がわかるので、Xilinx Power Estimator スプレッドシートを更新できます。
- 配線後: 配線が完了すると、使用される配線リソースに関するすべての詳細およびデザインに含まれる各パスの正確なタイミング情報が定義されます。インプリメントされた回路の機能をベスト ケースおよびワースト ケースのゲートおよび配線遅延で検証できるだけでなく、シミュレータは内部ノードの正確なアクティビティをレポートし、グリッチを含めることができます。このレベルの消費電力解析を基に、プロトタイプボードで消費電力を実際に計測する前に、最も正確な消費電力を見積もることができます。

## 解析用スイッチング アクティビティの指定

- シミュレーション結果 (SAIF ファイル): Vivado® の消費電力レポートを生成する際、デザイン データベース内のネット でシミュレーション結果のネットリスト内の名前と一致するものにはシミュレーションの値が適用されます。シミュレーション結果のネットリストは SAIF (Switching Activity Interchange Format) ファイルです。Vivado で一致したネットにスイッチング アクティビティとスタティック 確率が適用され、デザインの消費電力が算出されます。シミュレーション結果は、合成前または配置配線前のデザイン フロー早期に生成されている場合があります。この場合は、シミュレーション結果からモジュールの I/O ポートのアクティビティのみを取り込んで、ベクターレス エンジンで内部ノードのアクティビティが見積もられるようにする方が適切です。論理シミュレーションでは、グリッチ アクティビティはキャプチャされません。また、インプリメンテーション中のロジック変換 (最適化、複製、ゲーティング、リタイミングなど) のため、デザインとシミュレーション ネットリスト間で一部のノードが一致しないことがあります。ほとんどのプライマリ ポートおよび制御信号は一致するので、一致したノードに対しては現実的なアクティビティが提供されます。デザインの一致していない部分にはベクターレス エンジンによりアクティビティが伝搬されるので、消費電力見積もりの精度が上がります。

次のタイプのシミュレーション結果を使用してください。

- シミュレーションへのテスト ベクターおよび入力が、デザインの典型的な動作または意図した動作を表していることを確認します。エラー処理およびコーナー ケース (まれにしか発生しない状況) のシミュレーションでは、ロジックは標準動作条件と同じようにはシミュレーションされません。
- ビヘイビア シミュレーション結果よりもインプリメンテーション後のシミュレーション結果を使用します。フル タイミング シミュレーションの方が、SAIF 結果にタイミング グリッチ情報が含まれるので、より正確です。



**重要:** 指定した SAIF ファイルと一致しないデザイン ネットのアクティビティは、ベクターレス アルゴリズム およびデフォルトのスイッチング レートを使用して計算されます。これにより消費電力レポートのトグル レートが異なり、最終的に XPE にも反映されます。タイミング シミュレーションがサポートされるのは Verilog だけなので、VHDL で生成された .saif ファイルを使用することはお勧めしません。

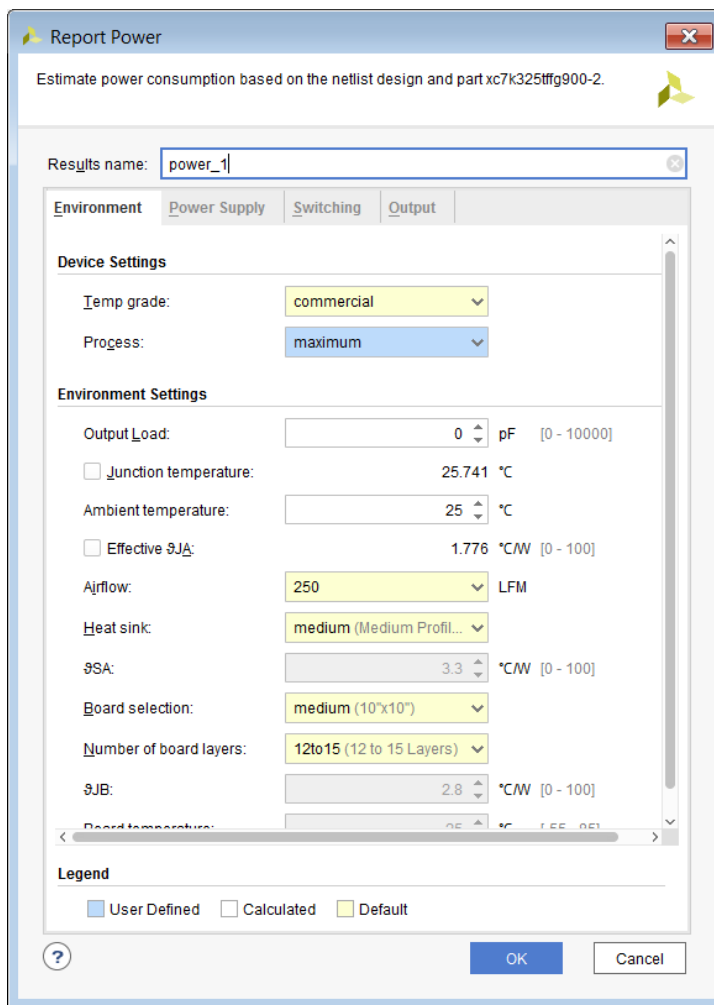


**重要:** 消費電力解析用の SAIF ファイルを Vivado シミュレータで生成するには、『Vivado Design Suite ユーザー ガイド: ロジック シミュレーション』(UG900: [英語版](#)、[日本語版](#)) を参照してください。Vivado® Design Suite 内で Mentor Graphics 社の ModelSim シミュレータを使用して SAIF ファイルを生成して消費電力を解析する場合は、[ザイリンクス アンサー 53544](#) を参照してください。フル タイミング シミュレーションには、`write_sdf` コマンドを使用してデザイン タイミング シミュレーション情報 (SDF) ファイルを生成し、シミュレーション実行中にアノテートします。

## デバイス/デザイン設定を確認して既知の要素のアクティビティを調整

[Report Power] ダイアログ ボックスは、Vivado® IDE の Flow Navigator で [Report Power] をクリックすると開きます。このダイアログ ボックスで消費電力設定を確認し、既知の要素のアクティビティを調整します。

図 18: [Report Power] ダイアログ ボックス



各シートを確認して、システムが正しく表現されているか確認します。[Report Power] ダイアログ ボックスには、次のタブが含まれます。

- [Environment] タブ
- [Power Supply] タブ
- [Switching] タブ
- [Output] タブ

## [Environment] タブ

[Environment] タブで編集可能なオプションを確認します。プロセス、電圧、環境データが意図した環境に近いものであることを確認します。これらの設定は、見積もられる総消費電力に大きく影響します。[Environment] タブでは、次のオプションを設定できます。

- [Device Settings]:

- [Temp Grade]: デバイスに適した温度グレードを選択します。通常は [Commercial] または [Industrial] です。設定によっては、スタティック消費電力仕様が異なるデバイスもあります。これを正しく設定しておく、選択したデバイスに対して正しいジャンクション温度範囲が表示されます。
- [Process]: ワースト ケース解析では、[Maximum] に設定することをお勧めします。デフォルト設定は [Typical] で、統計的な計測に基づくより正確な結果が得られますが、[Maximum] に変更すると、消費電力仕様がワースト ケースの値に変更されます。
- [Environment Settings]:
  - [Output Load] (pF): I/O ポートの出力で駆動されるボードおよびその他の外部容量を指定します。
  - [Junction Temperature] (°C): デバイスのジャンクション温度を指定します。ワースト ケース解析では、[User Override] オプションを使用して、この値をデバイスの温度グレードに基づく TJ (最大) に設定します。
  - [Ambient temperature] (°C): デバイス デザインを含むエンクロージャ内で達する可能性のある最大温度を指定します。この設定と、エアフローやその他の放熱経路 (ヒートシンクなど) により、ジャンクション温度が正確に計算され、デバイスのスタティック消費電力がより正確に計算されます。
  - [Effective  $\Theta_{JA}$ ] (°C/W): カスタム  $\Theta_{JA}$  (通常熱モデルから求める) の値を指定します。周辺温度および有効  $\Theta_{JA}$  に熱シミュレーションからの値を設定すると、より正確な見積もりが得られます。
  - [Airflow] (LFM): チップのエアフローは、LFM (リニア フィート/分) で計測されます。LFM は、CFM (立方フィート/分) で表されたファンの出力を通気経路の断面で除算して求めることができます。デバイスまたはファン (あるいはその両方) の配置が、デバイス上の通気、さらには放熱に影響する可能性があります。このパラメーターのデフォルト値は 250 LFM です。エアフローなし (空気が静止した状態) でデバイスを動作させる場合、デフォルトの 250 LFM を 0 LFM に変更する必要があります。
  - [Heat sink] (使用する場合): ヒートシンクを使用しており、詳細な放熱情報がない場合は、使用しているヒートシンクのタイプに合ったプロファイルを選択します。このパラメーターをほかのパラメーターと一緒に使用すると、有効  $\Theta_{JB}$  を計算しやすくなり、より正確なジャンクション温度および静止消費電力が計算されます。ソケットの設計および構造によっては、ヒートシンクとして機能するものもあります。
  - [Board selection] および [Number of board layers] (既知の場合): ボードのおおよそのサイズとスタックを選択すると、ボード自体の熱伝導性が考慮されるので、有効な  $\Theta_{JB}$  を計算できます。
  - [ $\Theta_{JB}$ ]: ボードおよびシステムのより正確な熱モデルがある場合は、デバイスからの放熱量を指定するために  $\Theta_{JB}$  (プリント回路基板の熱抵抗) を使用する必要があります。

[ $\Theta_{JB}$ ] に正確なカスタム値を指定すると、ジャンクション温度が正確に見積もられるため、デバイスのスタティック消費電力が正確に算出されます。



**重要:** [ $\Theta_{JB}$ ] にカスタム値を指定するには、[Board selection] を [Custom] に設定する必要があります。カスタム [ $\Theta_{JB}$ ] を指定する場合は、正確な消費電力を算出するため、[Board temperature] も指定する必要があります。

## [Power Supply] タブ

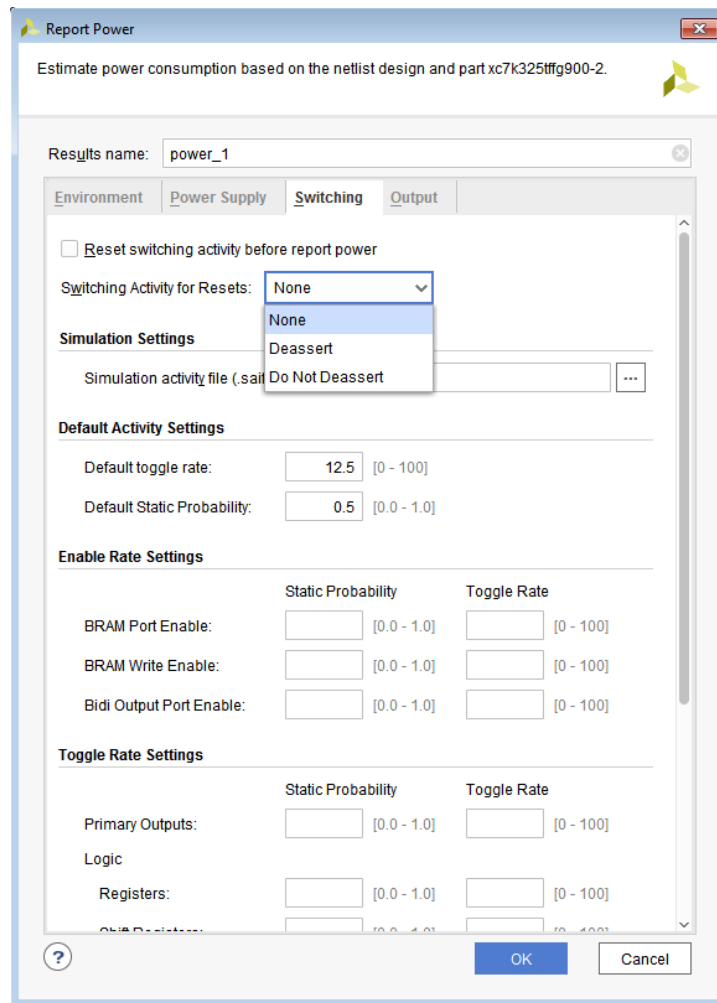
既知の場合は、[Power Supply] タブで各電源に正しい電圧値を設定します。電圧は、スタティック消費電力およびダイナミック消費電力の両方に大きく影響します。

## [Switching] タブ

[Switching] タブでデザインのシミュレーションおよびデフォルト アクティビティ 設定を確認してください。デザイン内の制約付きクロックもこのタブに表示されます。



図 19: 消費電力レポートのスイッチング設定



- [Reset switching activity before report power]: report\_power を実行する前に、適用されているスイッチング アクティビティをすべてクリア /リセットします。
- [Switching Activity for Resets]: 制御セットのスイッチング アクティビティを設定します。詳細は、[リセットのデassert](#)を参照してください。
- [Simulation Settings]:
  - [Simulation activity file (.saif)]: Vivado® の消費電力レポートを生成する際に、デザインに生成された SAIF シミュレーション データを入力として使用します。デザイン データベース内のネット内でシミュレーション結果のネットリスト内の名前と一致するものには、シミュレーションの値が適用されます。シミュレーション結果 (SAIF ファイル) からの情報を入力してさらに精度の高い消費電力解析を取得する方法については、[解析用スイッチング アクティビティの指定](#)を参照してください。
- [Default Activity Settings]:
  - [Default toggle rate]: 消費電力解析でデザインのプライマリ入力に使用するデフォルトのトグル レートを設定します。デフォルトのトグル レートは、スイッチ アクティビティがユーザーにより 設定されないプライマリ入力ネット、シミュレーション データ、またはデザインの制約に設定されます。非同期入力の場合、トグル レートはデザインの中でキャプチャ クロックに対して設定されます。有効な値は  $0 \leq \text{value} < 100$  で、デフォルト値は 12.5 です。

- [Default Static Probability]: デザインの消費電力解析で使用するデフォルトのスタティック確率を指定します。デフォルトのスタティック確率は、ユーザー、デザインのシミュレーション データまたは制約でスイッチング アクティビティが指定されていないプライマリ入力に使用されます。有効な値は  $0 \leq \text{value} \leq 1$  で、デフォルト値は 0.5 です。
- [Enable Rate Settings]:
  - [BRAM Port Enable]: デザインに含まれるブロック RAM のイネーブル信号すべてのアクティビティ レートを設定します。
  - [BRAM Write Enable]: デザインに含まれるブロック RAM のライト イネーブル信号すべてのアクティビティ レートを設定します。
  - [Bidi Output Port Enable]: デザインに含まれる双方向 I/O イネーブル信号 (IOBUF の T ピンなど) すべてのアクティビティ レートを設定します。

**注記:** [Static Probability] と [Toggle Rate] は一緒に指定する必要があります。

- [Toggle Rate Settings]:
  - [Primary Outputs]: デザインに含まれるプライマリ出力のイネーブル信号すべて (OBUFT の T ピンなど) のスイッチング アクティビティ レートを設定します。
- [Logic]:
  - [Registers]: デザインに含まれるレジスタすべての出力ピンのスイッチング アクティビティ レートを設定します。
  - [Shift Registers]: デザインに含まれるシフト レジスタすべての出力ピンのスイッチング アクティビティ レートを設定します。
  - [Distributed RAMs]: デザインに含まれる分散 RAM すべてのデータ出力ピンのスイッチング アクティビティ レートを設定します。
  - [LUTs]: デザインに含まれる LUT すべての出力ピンのスイッチング アクティビティ レートを設定します。
  - [DSPs]: デザインに含まれる DSP すべてのデータ出力ピンのスイッチング アクティビティ レートを設定します。
  - [Block RAMs]: デザインに含まれるブロック RAM すべてのデータ出力ピンのスイッチング アクティビティ レートを設定します。
- [GTs (Serial Transceivers)]:
  - [RX Data]: デザインに含まれる GT すべての RX データ出力ピンのスイッチング アクティビティ レートを設定します。
  - [TX Data]: デザインに含まれる GT すべての TX データ出力ピンのスイッチング アクティビティ レートを設定します。

**注記:** [Static Probability] と [Toggle Rate] は一緒に指定する必要があります。詳細およびガイドラインは、[ネットリスト エLEMENTのアクティビティ](#) の `set_switching_activity` コマンドの説明を参照してください。

- [Constrained Clocks]: [Constrained Clocks] を展開すると、デザインで制約されているクロックがすべてリストされます。クロック周波数が正しく設定されていることを確認します。



**ヒント:** すべてのプライマリ クロックを指定するようにします。デザイン クロックは、`create_clock` or `create_generated_clock` 制約に基づいてのみ識別されます。



**推奨:** 消費電力がより正確に算出されるようにするため、ザイリンクスではデザインの正しいクロック周波数を使用することをお勧めします。

## [Output] タブ

[Output] タブには、さまざまな消費電力結果ファイルが表示され、次の設定が含まれます。

- [Output text file]: プロジェクトの記録として消費電力見積もり結果を保存する必要がある場合があります。また、パフォーマンスまたはエリア制約を満たすために別のマップ、配置、および配線オプションを試す場合、それぞれの消費電力結果を保存しておく、複数の条件で要件が満たされたときに消費電力が最も低いソリューションの選択に役立ちます。
- [Output XPE file] (Xilinx® Power Estimator): 環境情報、デバイス使用量、デザイン アクティビティすべてを 1 つのファイル (.xpe) に保存します。このファイルは、Xilinx Power Estimator スプレッドシートにインポートできます。消費電力要件を超えており、ツールの最適化機能だけでは要件を満たすことができないような場合に便利です。この場合、現在のインプリメンテーション結果を Xilinx Power Estimator にインポートし、異なるマップ、ゲーティング、たたみ込み、およびその他の手法を試してそれらの消費電力への影響を見積もり、RTL コードを変更してインプリメンテーションを再実行します。また、Xilinx Power Estimator スプレッドシートでの見積もりと合成結果を比較して、XPE を必要に応じて調整します。
- [Output RPX file]: 消費電力レポートを RPX 形式で保存します。このレポートは、後から open\_report コマンドを使用して Vivado® IDE で開くことができます。

## 解析の実行

[Report Power] ダイアログ ボックスに必要なデータを入力したら、解析を実行します。ファイルおよびユーザー入力により供給されたアクティビティがネットリストにアノテートされ、残りの未定義ノードにデフォルト値が適用されます。その後、この初期アクティビティがプライマリ 入力からデザインのプライマリ出力に伝搬され、未定義ノードのアクティビティの見積もりが正確なものになります。最後に、使用される各リソースのダイナミック消費電力が算出され、これらのリソースでのスイッチング アクティビティにより生成される追加のスタティック消費電力が推論され、デバイスで予測されるジャンクション温度および総消費電力要件が算出されます。

## スイッチング アクティビティ制約の保持

[Report Power] ダイアログ ボックスへの入力はすべてプロジェクトの XDC 制約に保存され、[Report Power] をフローで再実行するときに使用されます。これは、シナリオを想定した what-if 解析に便利です。最後に使用したスイッチング アクティビティ 制約が保持され、ツールに表示されます。Vivado® IDE の [Tcl Console] ウィンドウで XDC ベースのコマンドを入力するか、[Net Properties] ウィンドウ ([Power] ビューの [Edit Properties] ボタン) で入力した場合でも、それらの入力値は [Report Power] ダイアログ ボックスに反映されます。スイッチング アクティビティの XDC 制約は、[Report Power] ダイアログ ボックスと一致します。ダイアログ ボックスで変更を加えるとそれが XDC 制約に反映され、XDC 制約を変更するとダイアログ ボックスに反映されます。

これは、[Report Power] ダイアログ ボックスのデフォルトのスイッチング アクティビティを上書きする場合にも便利です。この場合、適切なデフォルト値を使用して XDC 制約を作成し、report\_power を実行できます。

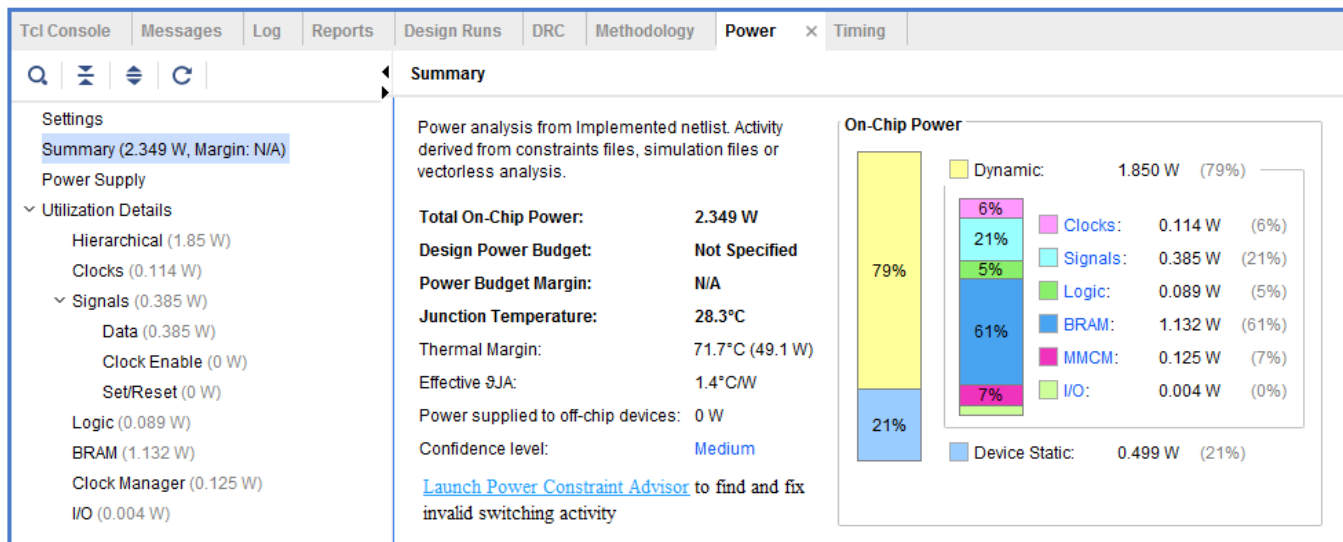
## デザインでの電力配分の確認

消費電力解析が完了したら、[Power] ウィンドウの [Summary] ページで [Total On-Chip Power] (総オンチップ消費電力) および温度プロパティを確認します。[On-Chip Power] グラフでは、デバイス リソース タイプごとの消費電力が示されます。このグラフから、デザインで消費電力が最大の箇所を特定できます。



[Summary] ページには、消費電力解析の [Confidence Level] も表示されます。[Confidence Level] は、入力データの正確さと完全性を示すもので、消費電力解析の実行時に使用されます。[Confidence Level] の値 ([Low]、[Medium]、[High]) をクリックすると詳細が表示され、これらの詳細から消費電力解析の精度を上げる方法がわかります。たとえば、より多くのクロックのアクティビティ レートやデザインの I/O 入力を指定すると、消費電力解析の精度が上がる可能性があります。

図 20: Vivado IDE での Vivado 消費電力解析: 消費電力レポート



[Power Supply] ページには、各電源の電流と、そのスタティック消費電力およびダイナミック消費電力の内訳が表示されます。

[Utilization Details] の下からリソース タイプを選択すると、そのリソースの消費電力の詳細が表示されます。各リソースのページは、ツリー形式の表になっています。列ヘッダーをドラッグすると、列の順序を変更できます。また、列ヘッダーをクリックすると並べ替え順を変更できます。

図 21: Vivado 消費電力解析: 消費電力の詳細

Power									
Settings	Utilization	Name	I/O Type	I/O Standard	Drive ...	Input Pins	Output Pins	Bidir Pins	IO LOGIC SERI
Summary (1.379 W)	0.004 W (<1% of...)	dut_fpga							
Power Supply	0.004 W (<1% of...)	sys_clk_...	HP	DIFF_SSTL15	N/A	1	0	0	No
Utilization Details	<0.001 W (<1% of...)	fmc_out	HR	LVC MOS33	12.000	0	10	0	No
Hierarchical (0.91i)	0 W	gpio_out...	HR	LVC MOS33	12.000	0	1	0	No
Clocks (0.011 W)	0 W	led	HP	LVC MOS15	12.000	0	1	0	No
Signals (0.061 W)									
Data (0.061 W)									
Clock Enable									
Set/Reset (0 V)									
Logic (0.011 W)									
BRAM (0.714 W)									
Clock Manager (0.000 W)									
I/O (0.004 W)									

レポートされている消費電力が要件を超えている場合は、[第 6 章: 消費電力削減のためのヒントおよび手法](#)のデバイスの消費電力を削減する手法を参照してください。使用できる手法は、デザインの完成度や開発プロセスの変更許容度によって異なります。



**重要:** [Device] 表の [Process] で [Maximum] が選択されているときに、パワーオン時の電源電流値のいずれかが動作時の電流要件の見積もりを超えている場合、[Power Supply] パネルで最小パワーオン時電源要件が青色で表示されます。電流値のいずれかが青色で表示された場合、[Power Supply] パネルに表示される総消費電力と Vivado 消費電力レポートの [Summary] セクションに表示される [Total On-Chip Power] の値は一致しません。

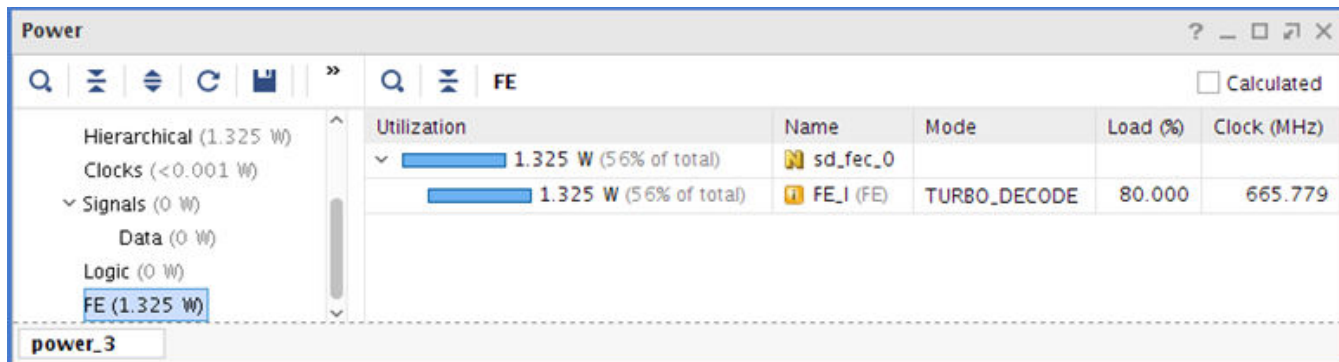
## 最大パッケージ電流に対するアラート

電流の見積もり値が、選択したパッケージの最大仕様値を超えた場合、[Power Supply] セクションの Iccint 電流の合計値フィールドが赤色になります。この機能は、UltraScale+™ デバイスにのみ適用されます。

## SD-FEC コアの消費電力見積もり

消費電力レポートでは、Zynq® UltraScale+™ RFSoc に含まれる Soft-Decision FEC コアの消費電力の見積もりがサポートされています。この消費電力見積もりは、次の図に示すように表示されます。

図 22: SD-FEC の消費電力見積もりを示す消費電力レポート



インプリメンテーション後、SD-FEC オブジェクトの消費電力レポートを生成する前に、次のプロパティを変更できます。

- LD\_PERCENT\_LOAD: LDPC Decoder コアの使用率 (%)
- LE\_PERCENT\_LOAD: LDPC Encoder コアの使用率 (%)
- TD\_PERCENT\_LOAD: Turbo Decoder コアの使用率 (%)

これらの 3 つのプロパティは、SD-FEC IP のカスタマイズ中に指定するか、またはインプリメント済みデザインで `set_property` コマンドを使用して指定できます。また、[Report Power] コマンドで生成された `.xpe` ファイルは、what-if 解析のため XPE スプレッドシートにインポートできます。

## RF コンバーターの消費電力見積もり

Zynq® UltraScale+™ RFSoc デバイス ファミリーには、RF データ コンバーター サブシステムが含まれます。消費電力 レポートでは、これらのコアの消費電力見積もりがサポートされます。コアは、Vivado® のザイリンクス IP カタログ に含まれる RF Data Converter IP で生成でき、さまざまな設定を指定できます。これらの IP を含むインプリメント済みデザインで [Report Power] を実行すると、次の図に示すような消費電力レポートが生成されます。

図 23: RFADC の消費電力レポート

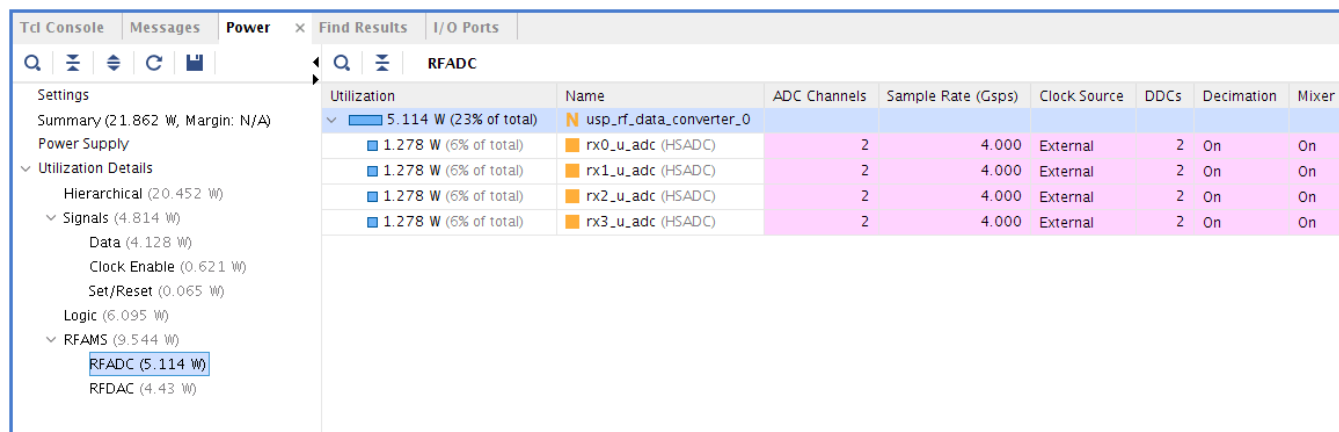


図 24: RFDAC の消費電力レポート

The screenshot shows the Power Analyzer interface with the 'RFDAC' component selected in the tree view. The main panel displays a table of power utilization for RFDAC components.

Utilization	Name	DAC Channels	Sample Rate (Gsp/s)	Clock Source	DUCs	Interpolation	Mixer
4.43 W (20% of total)	usp_rf_data_converter_0						
2.215 W (10% of total)	tx0_u_dac (HSDAC)	4	6.400	External	4	On	On
2.215 W (10% of total)	tx1_u_dac (HSDAC)	4	6.400	External	4	On	On

RF データ コンバーター IP のカスタマイズで、ADC/DAC チャネル カウント、サンプル レート、クロック ソース、間引き、ミキサーなどのすべてのユーザー 設定値を設定します。また、消費電力データは XPE シートにインポートし戻して、見積もられた消費電力をさらに解析することもできます。

## 消費電力レポート用の HBM 設定

HBM は IP インテグレーター ブロック デザインまたは IP カタログの HBM IP ウィザードを使用して設定できます。HBM インスタンスのセル プロパティを設定すると、report\_power の HBM 設定をさらに詳細に設定できます。セル プロパティは、詳細な消費電力解析に使用できます。HBM インスタンスに使用可能なセル タイプは、次のとおりです。

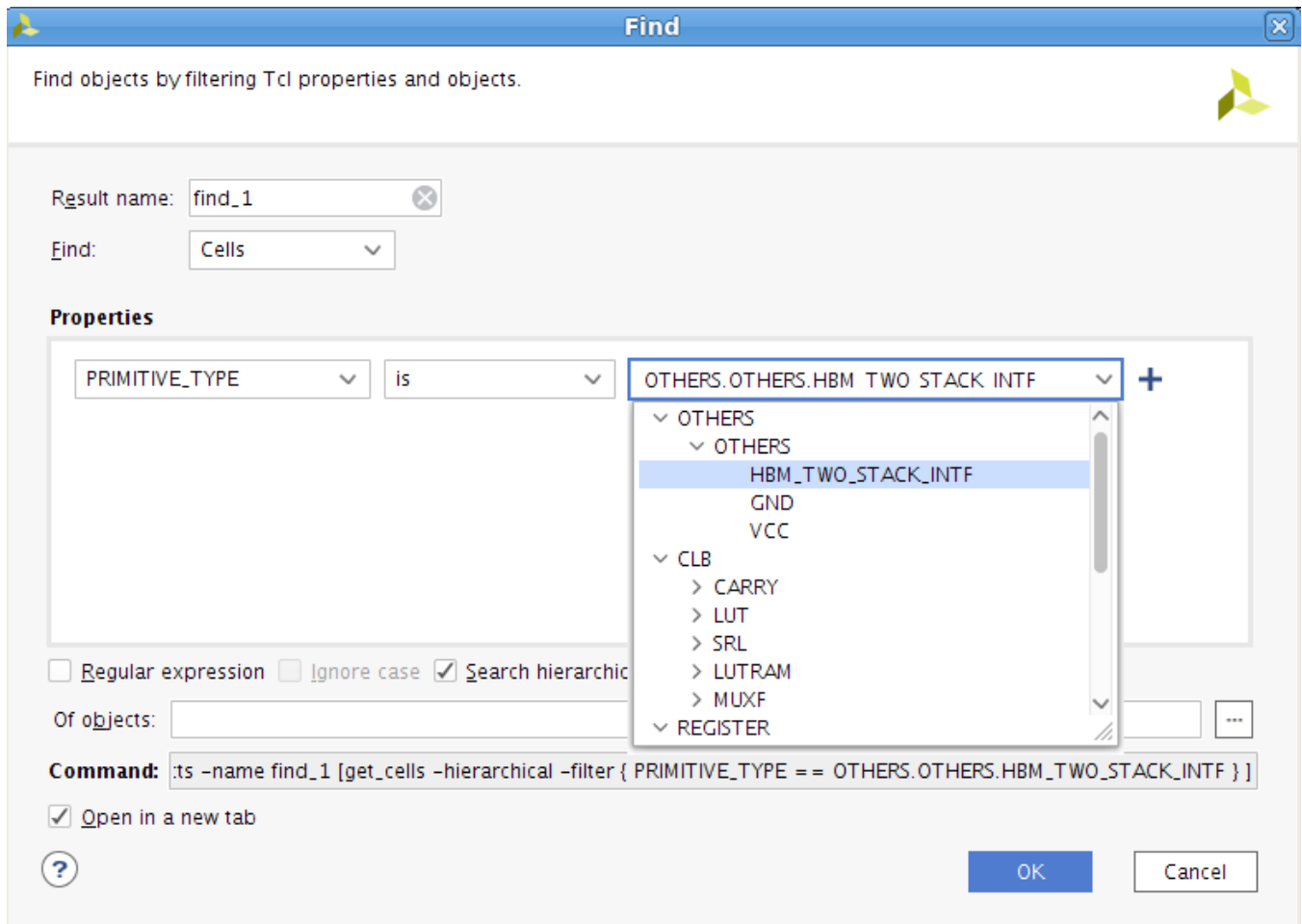
- HBM\_ONE\_STACK\_INTF: 単一の HBM スタックをターゲットにする場合
- HBM\_TWO\_STACK\_INTF: 2 つの HBM スタックをターゲットにする場合

get\_cells コマンドを使用して HBM インスタンスを見つけます。

```
set hbm_inst [get_cells -hier -filter {REF_NAME == HBM_TWO_STACK_INTF}]
```

HBM インスタンスは、Vivado® IDE で [Find] ダイアログ ボックスを使用しても検索できます。

図 25: Vivado で [Find] ダイアログ ボックスを使用した HBM インスタンスの検索



プロパティの値は report\_power を実行する前に変更できます。消費電力解析には、次のプロパティが使用されます。

- PAGEHIT\_PERCENT\_00、PAGEHIT\_PERCENT\_01: オープン ページにアクセスする HBM トランザクションのサイクル率で、最高速のアクセスになります。たとえば、消費電力を削減してより効率的なオープン ページ内の場合、順次メモリ アクセスになる可能性が高くなります。
- READ\_PERCENT\_00 ~ READ\_PERCENT\_15 (Stack 0)、READ\_PERCENT\_16 ~ READ\_PERCENT\_31 (Stack 1): 擬似チャネルが HBM から読み込まれるサイクル率 (%)。
- WRITE\_PERCENT\_00 ~ WRITE\_PERCENT\_15 (Stack 0)、WRITE\_PERCENT\_16 ~ WRITE\_PERCENT\_31 (Stack 1): 擬似チャネルが HBM に書き込まれるサイクル率 (%)。

PAGEHIT\_PERCENT に基づいて READ\_PERCENT および WRITE\_PERCENT が合理的な値になります。次のガイドラインを使用してください。

- PAGEHIT\_PERCENT < 75%: READ\_PERCENT + WRITE\_PERCENT は 50% 以下になる必要があります。
- PAGEHIT\_PERCENT >= 75%: READ\_PERCENT + WRITE\_PERCENT は 90% 以下になる必要があります。

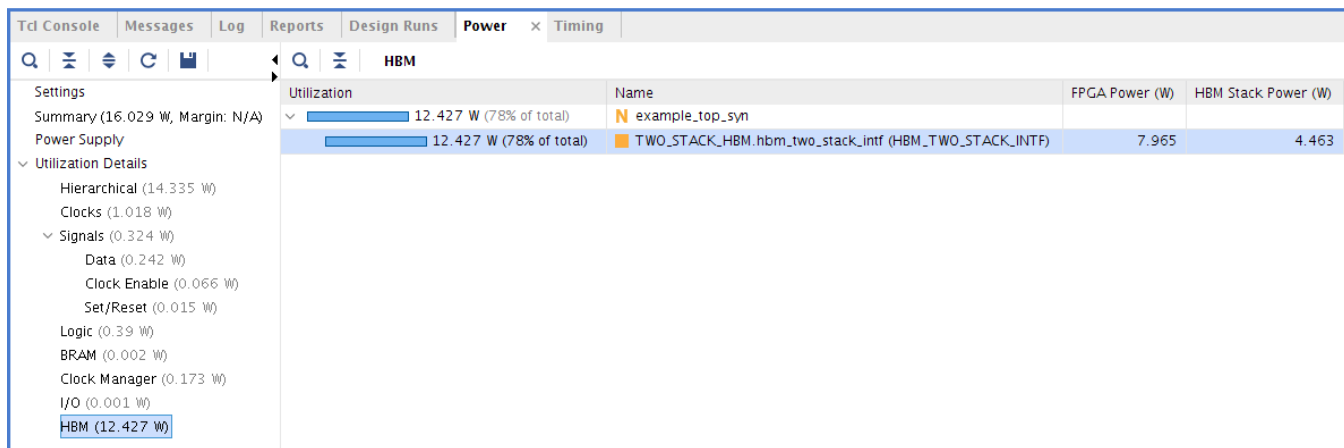
**注記:** 現在のリリースでは、PAGEHIT\_PERCENT\_00 および PAGEHIT\_PERCENT\_01 のデフォルト値は 50 です。デフォルト値は今後のリリースで 75 に修正される予定です。

次のプロパティは、HBM IP 設定で割り当てられるので、変更できません。

- DATARATE\_00 ~ DATARATE\_15: 各メモリ コントローラーのデータ レート (Gbps)。プロパティ 00 ~ 07 は Stack 0 に、08 ~ 15 は Stack 1 に適用されます。
- SWITCH\_ENABLE\_00、SWITCH\_ENABLE\_01: 専用 AXI スイッチがスタックに対してイネーブルまたはディスエーブルかを反映します。

次の図に、消費電力レポートのデバイスおよび HBM スタック間の消費電力の内訳を示す [HBM] ビューの例を示します。

図 26: HBM の消費電力レポート



Tcl Console   Messages   Log   Reports   Design Runs   Power x Timing			
HBM			
Utilization	Name	FPGA Power (W)	HBM Stack Power (W)
12.427 W (78% of total)	example_top_syn		
12.427 W (78% of total)	TWO_STACK_HBM.hbm_two_stack_intf (HBM_TWO_STACK_INTF)	7.965	4.463

Left sidebar details:

- Settings
  - Summary (16.029 W, Margin: N/A)
  - Power Supply
  - Utilization Details
    - Hierarchical (14.335 W)
    - Clocks (1.018 W)
    - Signals (0.324 W)
      - Data (0.242 W)
      - Clock Enable (0.066 W)
      - Set/Reset (0.015 W)
    - Logic (0.39 W)
    - BRAM (0.002 W)
    - Clock Manager (0.173 W)
    - I/O (0.001 W)
    - HBM (12.427 W)**

## 消費電力レポート用の GTM 設定

GTM は、IP インテグレーター ブロック デザイン内で IP カタログの GTM IP ウィザードを使用して設定します。

図 27: 消費電力レポート用の GTM 設定

Component Name
gtm\_wizard\_ultrascale\_0

Basic
Physical\_Resources
Optional\_features
FEC\_Options
AM\_50G
AM\_100G

System

GT Type
GTME4

Transceiver configuration preset
Start from scratch

Transmitter

TX Line rate (Gb/s)
53.125

Transmitter PAM mode selection
PAM4

TX User data width
128

TX Internal data width
128

TXOUTCLK source
TXPROGDIVCLK

Differential swing and emphasis mode
Custom

Receiver

RX Line rate (Gb/s)
53.125

Receiver PAM mode selection
PAM4

RX User data width
128

RX Internal data width
128

RXOUTCLK source
RXPROGDIVCLK

Reference clock Frequency

Requested reference clock(MHz)
156.25

Actual Reference clock(MHz)
156.25

☐ Resulting Fractional divider Enable

消費電力レポートに必要な主なパラメーターはすべて、UNISIM プロパティを使用して設定できます。UNISIM プロパティは、次のとおりです。

- MODULATION\_MODE: GTM 信号変調モード。NRZ および PAM4 信号伝送に使用されます。
- DATARATE: 指定した変調モードの GTM チャンネル ライン。PAM4 では 19.6 Gb/s ~ 58 Gb/s、NRZ GTM では 9.8 Gb/s ~ 29 Gb/s です。
- FEC\_MODE: ハード化された RS-FEC の使用。このパラメーターを BYPASS に設定すると、GTM はハード化された FEC ブロックをバイパスします。FEC を使用するには、このプロパティを KP4 に設定します。
- INTERFACE\_WIDTH: GTM インターフェイス幅。このプロパティは、今後追加されます。現時点では、インターフェイス幅は MODULATION\_MODE から求められます。
- INS\_LOSS\_NYQ: イコライゼーション モード。低消費電力モードでは 10 dB 以下、高パフォーマンス モードでは 10 dB より大きくする必要があります。
- TX\_AMPLITUDE\_SWING: TX ドライバーの差動振幅。有効な値は、250、275、300、...、1000、および 1025 です。

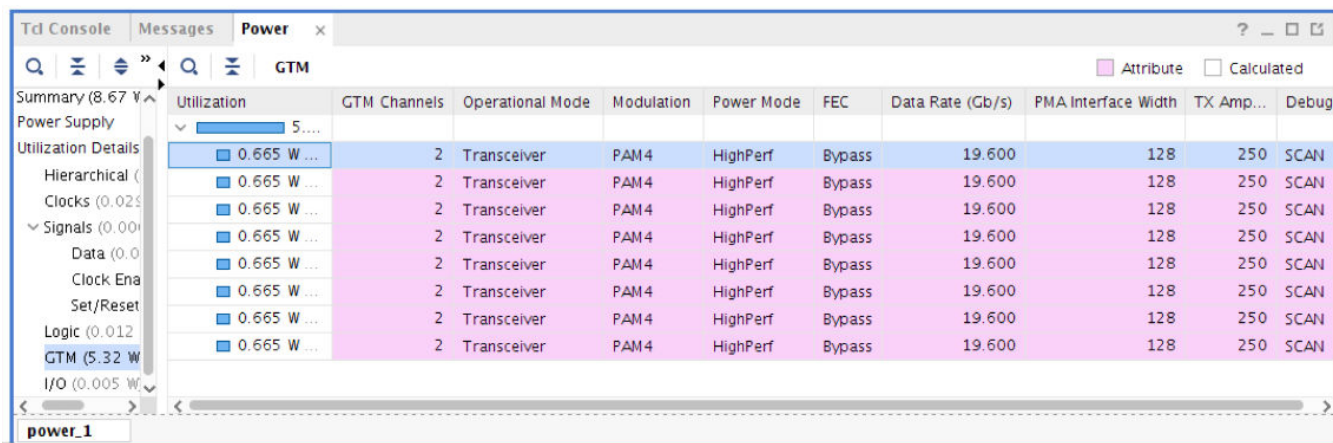
GTM デバッグ モードは CH\*\_RX\_PAD\_CFG1[10] 属性により決定されます。CH\*\_RX\_PAD\_CFG1[10] は各チャンネルの ACJTAG\_EN ビットで、ACJTAG がアクティブかどうかを判断するために使用されます。

**注記:** FEC\_MODE パラメーターを KP4 に設定すると、PAM4 信号伝送を使用したときに GTM でハード化された FEC ブロックをバイパスできません。PAM4 を使用する場合は、FEC\_MODE パラメーターを KP4 に設定してください。



次の図に、GTM の消費電力レポートの例を示します。

図 28: GTM の消費電力レポート



Summary (8.67 W)	Utilization	GTM Channels	Operational Mode	Modulation	Power Mode	FEC	Data Rate (Gb/s)	PMA Interface Width	TX Amp...	Debug
Power Supply	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN
Utilization Details	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN
Hierarchical (0.025 W)	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN
Clocks (0.001 W)	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN
Signals (0.001 W)	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN
Data (0.001 W)	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN
Clock Ena	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN
Set/Reset	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN
Logic (0.012 W)	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN
GTM (5.32 W)	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN
I/O (0.005 W)	0.665 W	2	Transceiver	PAM4	HighPerf	Bypass	19.600	128	250	SCAN



# Vivado Design Suite での消費電力解析および最適化

## 概要

この章では、Vivado<sup>®</sup> Design Suite で実行可能な消費電力に関連する機能およびフローを説明し、消費電力見積もり、解析、および最適化を実行できるようにします。消費電力解析は、合成後、最適化後、配置後、または配線後に実行できます。RTL エラボレーション後には実行できません。消費電力最適化は、配置前後にのみ実行できます。消費電力の解析および最適化は、Vivado 統合設計環境または Tcl コンソールのいずれかを使用して実行可能で、動的に what-if シナリオを試すことができます。

## Vivado IDE での消費電力解析

Vivado<sup>®</sup> IDE の消費電力に関連した機能を使用すると、デザインのインプリメンテーションで次の消費電力見積もりおよび解析を実行できます。

- デザインのスタティック消費電力に影響する次の温度特性をレポート
  - 。 ジャンクション温度、周囲温度などの温度値
  - 。 ボード層の数、ボードの温度など、選択したボードに関するデータ
  - 。 デザインで使用されるエアフローおよびヒートシンク プロファイルのデータ
- 各種電源のデバイス電流要件のレポート
- 消費電力の分配を詳細に解析することにより、ダイナミック消費電力、熱消費電力、またはオフチップ消費電力を削減する消費電力節約ストラテジを特定

次の図に、典型的な消費電力見積もりおよび解析フローを示します。最も正確な結果を得られるよう、見積もりまたは解析を実行する前に必要なツールへの入力および設定をする手順も含まれています。消費電力見積もりおよび解析コマンドは、Vivado IDE または Tcl プロンプトから実行できます。

図 29: 消費電力見積もりおよび解析フロー



X12401-010320

## サポートされるデバイス アーキテクチャ

Vivado® Design Suite のアーキテクチャ サポートは、『Vivado Design Suite ユーザー ガイド: リリース ノート、インストール、およびライセンス』 (UG973: [英語版](#)、[日本語版](#)) を参照してください。

## サポートされる入力

- タイミング制約を指定する XDC 制約ファイル。
- ビヘイビアまたはタイミング シミュレーションからのシミュレーション出力アクティビティ ファイル (SAIF ファイル)。
- 環境、動作条件、ツール デフォルト、および個々のネットリスト ノード アクティビティ を指定する XDC/Tcl ファイル コマンド。UltraScale+™ デバイスの場合、XPE で Vivado® IDE から読み込まれた XDC ファイルがダンプされます。
- Vivado 消費電力解析ツールには、デフォルトの値およびノード アクティビティ レートを入力するメカニズムが複数あります。次に、それらのメカニズムを優先順にリストします。
  1. スタティック (定数は GND または VCC に接続)。
  2. [Power] ウィンドウの [Utilization Details] セクションでユーザーが入力した値。
  3. インポートされたシミュレーション アクティビティ ファイル (SAIF)。
  4. インポートされた制約ファイル: 制約ファイル (XDC) またはデザイン ネットリストからインポートされたクロック制約。
  5. ベクターレス見積もり: ノードが上記の入力いずれでも定義されていない場合、デフォルト値と、ノードへの入力アクティビティに基づいて、アクティビティが見積もられます。
  6. デフォルト値: ノードがベクターレスで見積もることができなかった場合、デザインのプライマリ入力やブラックボックス出力の場合と同じように、デフォルト値が割り当てられます。

注記: デフォルト値は、[Report Power] ダイアログ ボックスで変更できます。詳細は、[デバイス/デザイン設定を確認して既知のエレメントのアクティビティを調整](#)を参照してください。

## サポートされる出力

- GUI I/O バス、ネット、およびセル消費電力プロパティ
- GUI およびテキスト形式の消費電力レポート
- Xilinx® Power Estimator スプレッドシートにインポート可能な XML ベースの消費電力レポート
- Tcl コマンドを使用したアクティビティ レートと動作条件レポート

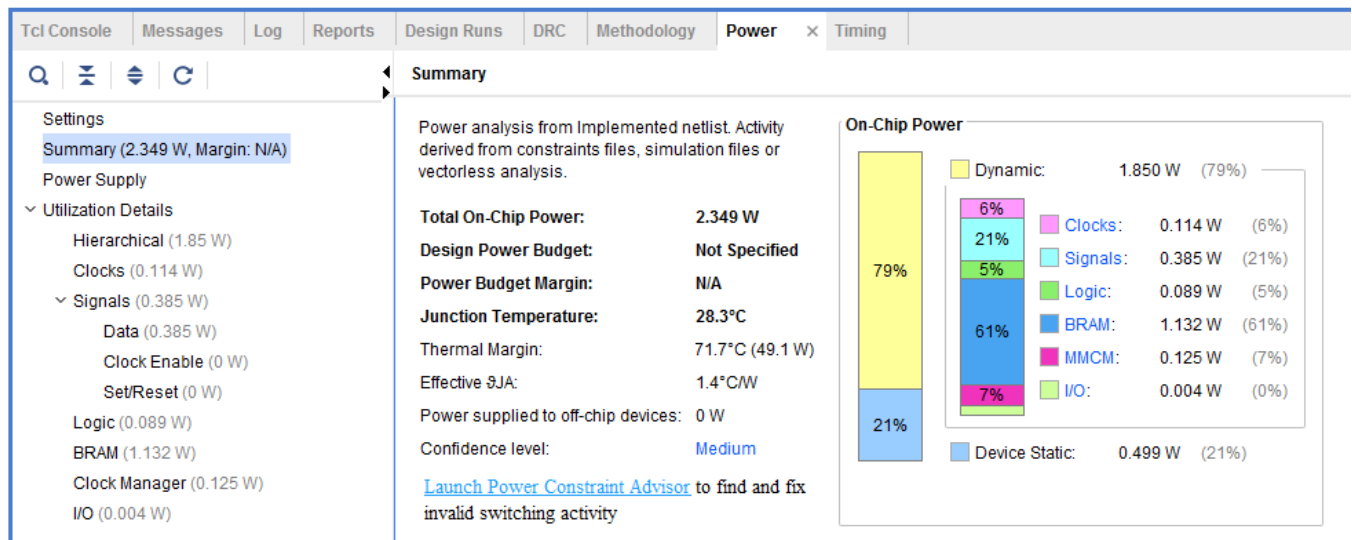
## 制御信号のアクティビティの調整

正確な消費電力解析用に SAIF ベースのアノテーションを使用していない場合は、初期段階の解析を実行した後に消費電力解析を調整できます。[Report Power] コマンドを実行すると、すべての制御信号が [Power] ウィンドウの [Signals] にリストされます。アプリケーションの予期動作から、標準デザイン動作では、セット/リセット信号の一部がアクティブでないことがわかります。その場合は、これらの信号のアクティビティを調整するとよいかもしれません。同様に、ブロックが使用されないときに、アプリケーションに含まれる一部の信号がブロック全体をディスエーブルにすることがあります。その場合は、機能に合わせてアクティビティを調整してください。合成ツールや配置配線アルゴリズムが、RTL 記述を最適化するために制御信号を推論またはマップし直すことがあるため、これらのビューに不明な信号が含まれていることがあります。その場合は、ツールにこれらのアクティビティを自動設定させてください。

## Vivado IDE での消費電力レポートの解析

消費電力レポートおよび解析のウィンドウは、Vivado® IDE のワークスペースに統合されています。消費電力に関するさまざまなビューを表示できるだけでなく、既存のビューへのクロスプローブも可能です。

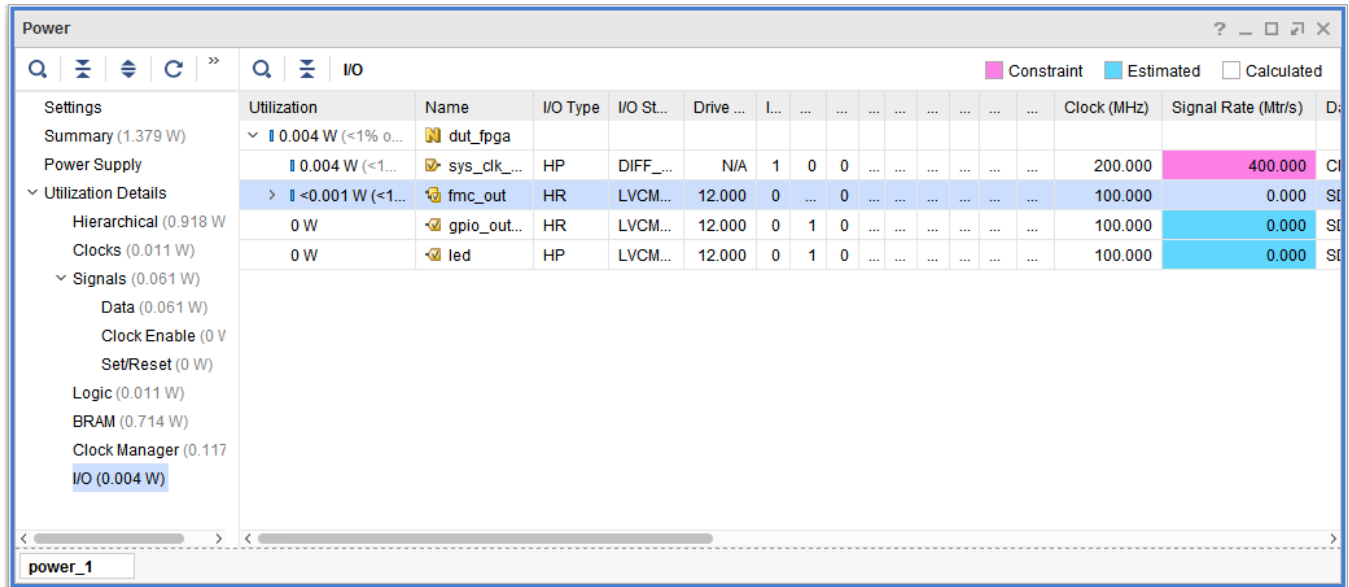
図 30: Vivado IDE での消費電力解析



- [Power] ウィンドウの [Settings] ページには、消費電力の計算に使用されたデバイス、ツール、および環境設定がすべて表示されます。
- [Summary] ページには、重要な温度および電源電力の結果のまとめが表示されます。

[Utilization Details] セクションの各リソース タイプのページ、または [Netlist] ウィンドウでリソース タイプを選択すると、[Properties] ウィンドウの [Statistics] ビューに選択したエレメントの設定、使用数、アクティビティの詳細が表示されます。異なる動作条件やアクティビティ パターンで複数のレポートを生成し、消費電力を見積もることができます。次の図に示すように、[Utilization Details] セクションの値 ([Clocks] ページの [Frequency]、[I/O] ページの [Signal Rate] など) の一部は色分けされ、消費電力解析のために使用された値のソースが表示されます。色分けの説明は、ウィンドウの下部に表示されます。たとえば、値がシミュレーション アクティビティ ファイルからのものか、ユーザー定義か、ベクターレス伝搬エンジンにより割り当てられたデフォルト値なのかわかります。

図 31: 消費電力レポートの色分け



Settings	Utilization	Name	I/O Type	I/O St...	Drive ...	I...	...	...	...	...	...	...	Clock (MHz)	Signal Rate (Mtr/s)	D...
Summary (1.379 W)	0.004 W (<1% o...	dut_fpga													
Power Supply	0.004 W (<1...	sys_clk_...	HP	DIFF_...	N/A	1	0	0					200.000	400.000	Cl
Utilization Details	> 0.001 W (<1...	fmc_out	HR	LVC...	12.000	0		0					100.000	0.000	St
Hierarchical (0.918 W)	0 W	gpio_out...	HR	LVC...	12.000	0	1	0					100.000	0.000	St
Clocks (0.011 W)	0 W	led	HP	LVC...	12.000	0	1	0					100.000	0.000	St
Signals (0.061 W)															
Data (0.061 W)															
Clock Enable (0 V)															
Set/Reset (0 W)															
Logic (0.011 W)															
BRAM (0.714 W)															
Clock Manager (0.117 W)															
I/O (0.004 W)															

★ **重要:** 消費電力レポートでは、IP インテグレーターで設定された Zynq-7000/Zynq® UltraScale+™ MPSoC ブロックでの Zynq®-7000 SoC および Zynq® UltraScale+™ MPSoC の消費電力解析がサポートされます。PS の使用および機能は、IP インテグレーターで設定します。消費電力レポートでは、これらの設定に基づいて消費電力が見積もられます。Vivado® の消費電力見積もりは読み出し専用なので、PS 特定のプロセッサ、インターフェイス、またはメモリの [Signal Rate] または [Static Probability] をこの段階で変更することはできません。Xilinx® Power Estimator の [PS] タブの各フィールドの詳細は、『Xilinx Power Estimator ユーザー ガイド』(UG440: [英語版](#)、[日本語版](#)) の [PS] シートのセクションを参照してください。

★ **重要:** 消費電力レポートでは、Zynq UltraScale+ EV デバイスの VCU (ビデオ コーデック ユニット) の消費電力見積もりがサポートされます。VCU の解像度、色フォーマットなどのプロパティは、IP インテグレーターで設定します。消費電力レポートでは、これらの設定に基づいて消費電力が見積もられます。詳細は、『Xilinx Power Estimator ユーザー ガイド』(UG440: [英語版](#)、[日本語版](#)) の [Other] シートのセクションを参照してください。

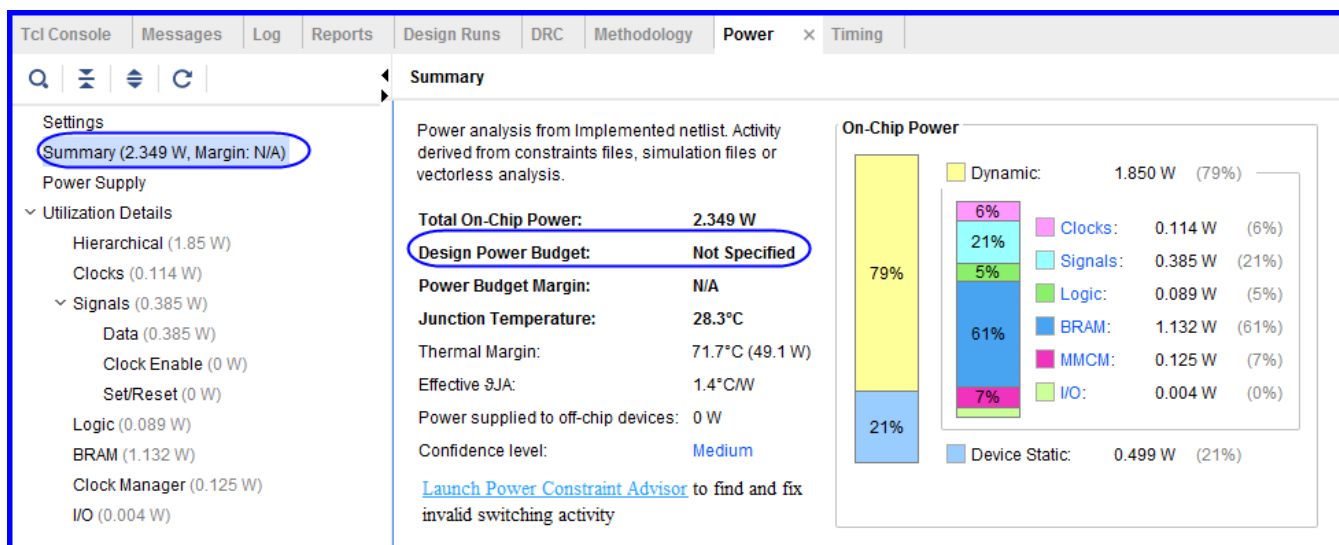
## UltraScale+ デバイスの消費電力および電源電流バジェットの設定

消費電力レポートを生成する前に、デザインの消費電力および電源電流バジェットを指定します。消費電力バジェットは、デザイン全体に対して指定します。電源電流バジェットは、電源ごとに指定します。消費電力および電源電流バジェットを指定するには、次のコマンドを使用します。

```
set_operating_condition -design_power_budget <Power in Watts>
set_operating_conditions -supply_current_budget {<supply rail name> <current budget in Amp>}
```

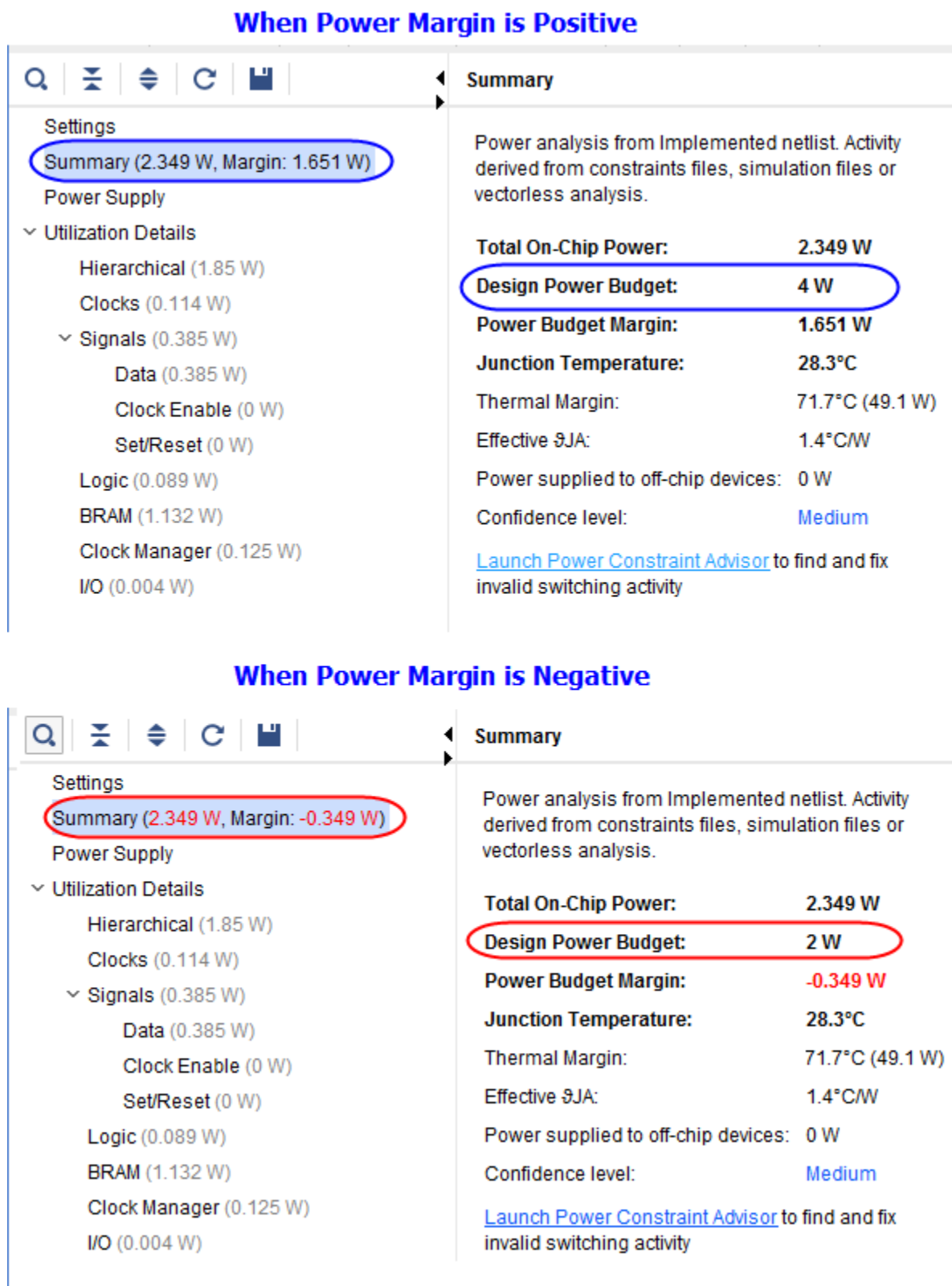
[Report Power] を実行すると、デザインの消費電力と個別の電源電流が指定したバジェットと比較されます。[Report Power] (GUI/テキスト) は、消費電力バジェットのマージンを示します。マージンが正の値の場合は、デザインの消費電力が消費電力バジェットよりも低いことを意味し、負の値 (赤色で表示) の場合は消費電力バジェットがデザインの消費電力を超えてしまったことを示します。消費電力バジェットを指定しなかった場合は、次のようにマージンが N/A と表示されます。次の図に、デザインの消費電力バジェットを指定しなかった場合の消費電力レポートの例を示します。

図 32: デザインの消費電力バジェットを指定しなかった場合の消費電力レポート



次の図に、デザインの消費電力バジェットを 4 ワットに指定し、消費電力マージンが正の場合の消費電力レポートを示します。2 つ目の図は、デザインの消費電力バジェットを 2 ワットに指定し、消費電力マージンが負の消費電力レポートを示します。

図 33: 正および負のマージンの消費電力バジェットを示す消費電力レポート



各電源の電源電流バジェットは、[Power Supply] セクションに示されます。各電源に対し、電源電流が指定したバジェットより小さい場合は正のマージンが示され、電源電流が指定したバジェットを超える場合は負のマージンが赤で示されます。電源に対して電源電流バジェットが指定されていない場合は、その電源レールの [Budget] 列に「Unspecified」（指定なし）、[Margin] 列に「N/A」と示されます。

図 34: 電源電流バジェット

Supply Source	Voltage (V)	Total (A)	Dynamic (A)	Static (A)	Budget (A)	Margin (A)
Vccint	0.850	37.647	32.270	5.376	42.000	4.353
Vccint_io	0.850	0.751	0.001	0.750	0.500	-0.251
Vccbram	0.850	0.143	0.000	0.143	1.620	1.477
Vccaux	1.800	0.635	0.160	0.475	1.190	0.555
Vccaux_io	1.800	0.027	0.003	0.024	0.056	0.029
Vcco33	3.300	0.000	0.000	0.000	Unspecified	NA
Vcco25	2.500	0.000	0.000	0.000	Unspecified	NA
Vcco18	1.800	0.004	0.004	0.000	0.014	0.010
Vcco15	1.500	0.000	0.000	0.000	Unspecified	NA
Vcco135	1.350	0.000	0.000	0.000	Unspecified	NA
Vcco12	1.200	0.000	0.000	0.000	Unspecified	NA
Vcco10	1.000	0.000	0.000	0.000	Unspecified	NA
Vccadc	1.800	0.016	0.000	0.016	0.020	0.004
VCC_IO_HBM	1.200	0.164	0.000	0.164	3.840	3.676
VCC_HBM	1.200	0.188	0.000	0.188	4.160	3.972
VCCAUX_HBM	2.500	0.025	0.000	0.025	0.200	0.175
MGTYAVcc	0.900	0.000	0.000	0.000	Unspecified	NA
MGTYAVtt	1.200	0.000	0.000	0.000	Unspecified	NA

## 消費電力レポートの保存と復元

レポートの保存と復元は、2016.1 リリースで Vivado® に導入された新機能です。この機能を使用すると、Vivado IDE からの消費電力レポートを保存し、後で必要なときに開くことができます。レポートは RPX 形式で保存され、次の Vivado Tcl コマンドでいつでも開くことができます。

```
open_report
```

プロジェクト モードでインプリメント済みデザインを開くと、消費電力レポート impl\_1 がタイミング レポートのようにデフォルトで開きます。チェックポイント フローでは、report\_power Tcl コマンドで -rpx オプションを使用するとレポートを保存できます。

```
report_power -rpx design_1_power.rpx
```

保存したレポートは、次の Tcl コマンドを使用して Vivado IDE で開くことができます。

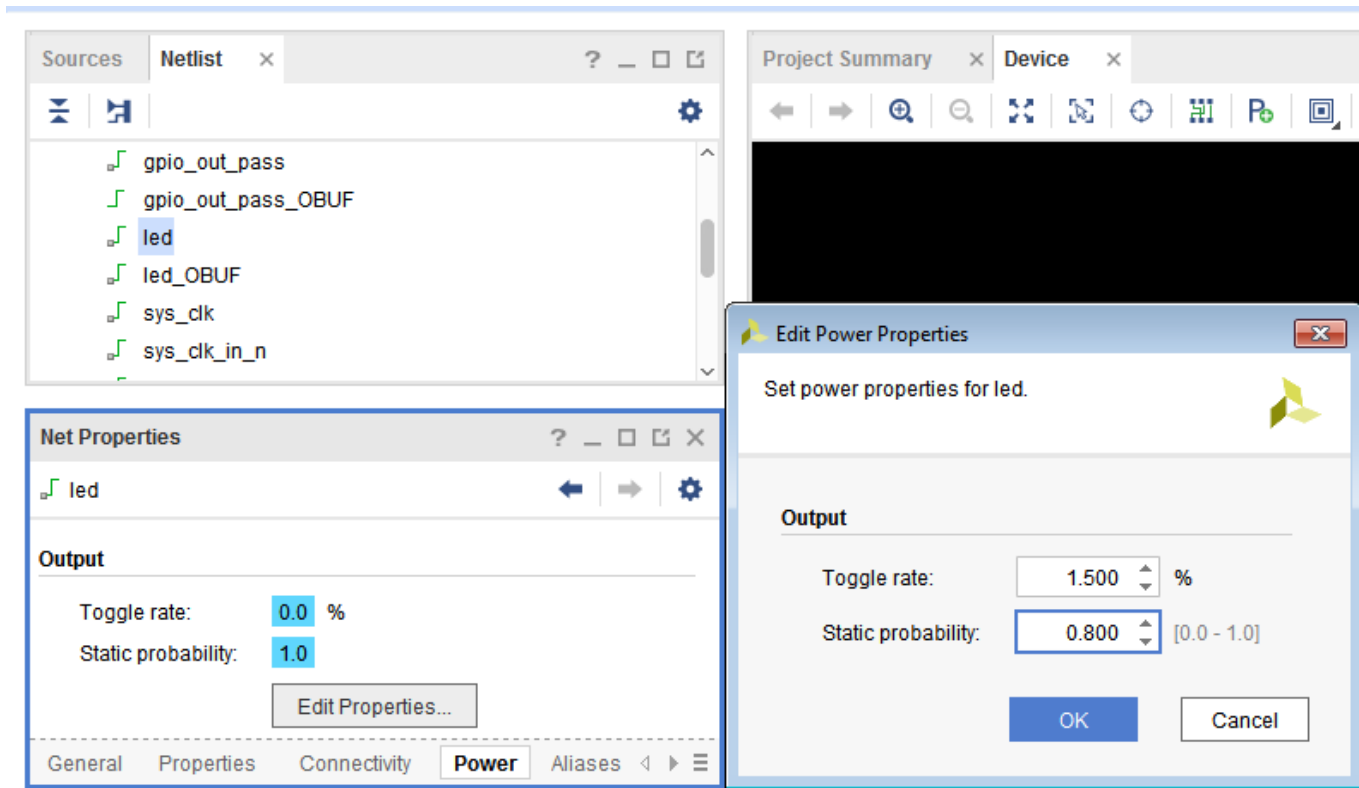
```
open_report -name rpx1 ./design_1_power.rpx
```



## Vivado IDE での what-if 解析の実行

what-if 解析を実行するには、デザインのネットおよびセルにトグル レートおよびスタティック確率を設定します。これらを設定するには、[Netlist] ウィンドウ、回路図、消費電力レポートでネットまたはセルを右クリックして [Properties] をクリックし、[Properties] ウィンドウの [Power] ビューを開き、[Load Properties] ボタンをクリックします。[Edit Properties] ボタンをクリックし、[Edit Power Properties] ダイアログ ボックスで [Toggle rate] および [Static probability] を設定し、[OK] をクリックします。

図 35: [Net Properties] ウィンドウの [Power] ビュー



上記の例では、[Toggle rate] は 1.5% に、[Static probability] は 0.8 に設定されています。[OK] をクリックすると、Vivado IDE で設定が確定され、[Tcl Console] ウィンドウに次の XDC 制約が表示されます。

```
set_switching_activity -toggle_rate 1.500000 -static_probability 0.800000
[get_nets led]
```



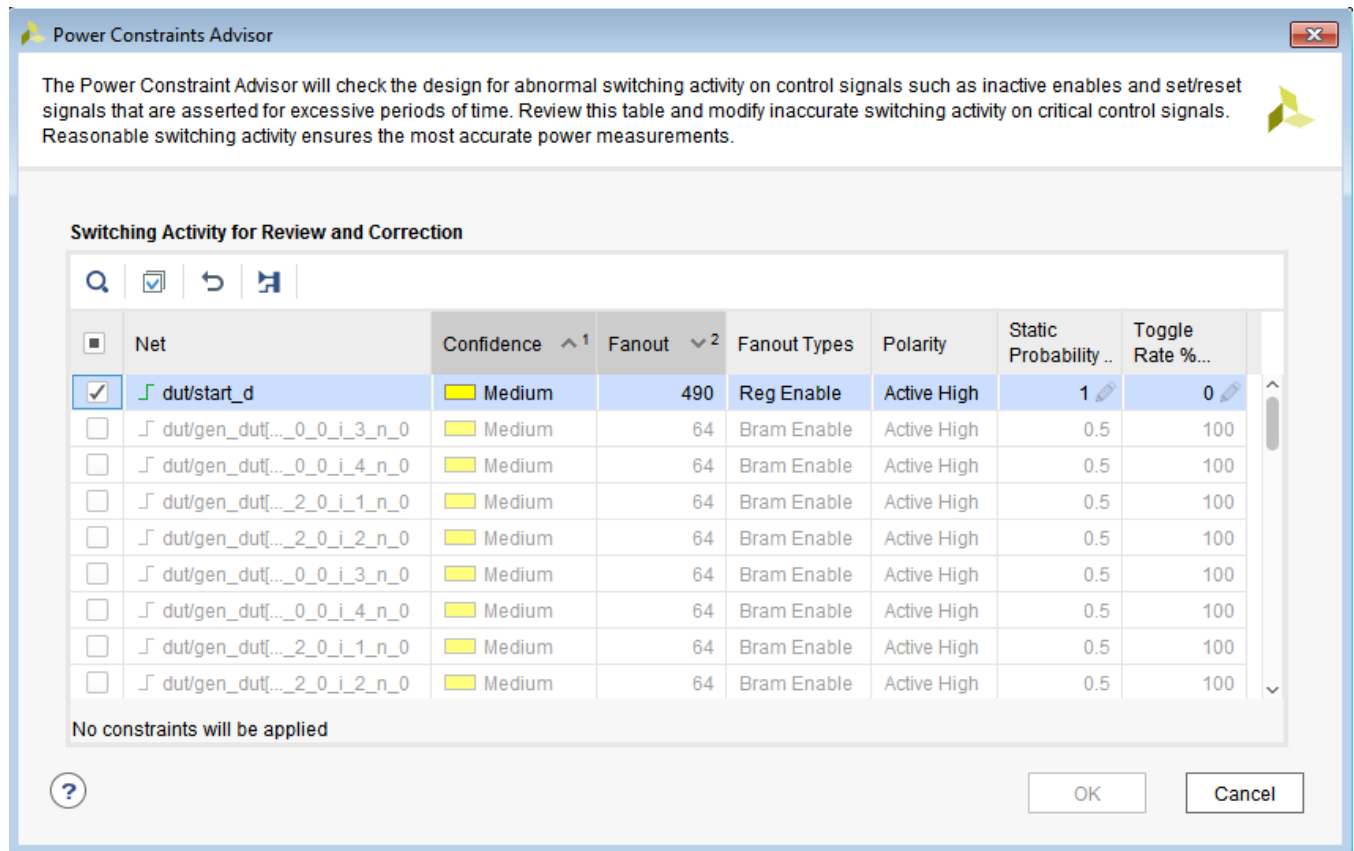
**重要:** この XDC 制約によりデザインが最新の状態でなくなるので、[Force-up-to-date] を使用してデザインのステータスを最新の状態にします。

## 消費電力制約アドバイザー

消費電力制約アドバイザーは、デザイン内の制御信号すべてを対象に、ツールで計算したスイッチング アクティビティをレポートします。制御信号には、リセットおよびイネーブル (リセット、セット、クリア、プリセットなど) が含まれます。妥当なスイッチング アクティビティを 供給することにより、消費電力をより正確に見積もることができます。

消費電力制約アドバイザーを実行するには、Vivado IDE で [Tools] → [Power Constraints Advisor] をクリックします。

図 36: 消費電力制約アドバイザー



レポートの表を確認し、非アクティブなイネーブルや必要以上の時間アサートされるリセット信号など、クリティカルな制御信号のスイッチング アクティビティが正しくない場合は修正します。消費電力制約アドバイザー レポートには、次の制約が表示されます。

- [Net]: ネットは制御セット、ブロック RAM イネーブル、レジスタ イネーブルのいずれかです。
- [Confidence]: ネットのスイッチング アクティビティがどの程度正確か (信頼性レベル) を示します。ネットの信頼性レベルを計算するときに消費電力ツールで使用するしきい値は、次のとおりです。
- セット/リセット/プリセット/クリア:

表 1: 消費電力ツールのしきい値

信頼性レベル	スタティック確率
小	> 8%
中	5% - 8%
大	< 5%

- ブロック RAM イネーブル:

表 2: ブロック RAM イネーブル

信頼性レベル	スタティック確率
小	< 1%

信頼性レベルが Low の場合、ブロック RAM はデザインでアクティブではないので、削除できるかどうかを再検討してください。

- レジスタ イネーブル:

表 3: レジスタ イネーブル

信頼性レベル	スタティック確率
小	< 3%
中	> 3%

信頼性レベルが Low の場合、デザインのレジスタがアクティブではないので、再検討する必要があります。信頼性レベルが Medium の場合、ユーザーの定義した、またはツールで設定された妥当な時間でレジスタがイネーブルになることを示します。

- [Fanout]: 各制御信号のファンアウト (駆動される最下位プリミティブの数) を示します。ファンアウトが大きい信号は最も重要で、確認して修正すべき信号です。ファンアウトの大きい信号は、デザインを広い範囲に渡り、ダウンストリーム スイッチングをディスエーブルにしてしまうので、消費電力がかなり低くレポートされる可能性があるからです。ファンアウトの小さい信号だと、スイッチングが間違っている影響が小さいので、重要度は低くなります。
- [Fanout Types]: ネットが制御セット (セット、リセット、クリア、プリセット) かブロック RAM イネーブルかを示します。制御ネットに複数のエントリがあると、そのネットには複数のファンアウトがあり、ファンアウトセルにある複数のピンを駆動しているということになります。
- [Polarity]: 制御セットの極性を示します。ネットのスタティック確率を設定する際は、極性に注意してください。
- [Static Probability]: 編集可能なフィールドで、ネットのファンアウト タイプと極性に基づいた正しいアクティビティを入力する必要があります。
- [Toggle Rate]: ネットのトグル レートを示します。このフィールドは編集可能で、スタティック確率に基づいて入力する必要があります。

**注記:** デフォルトでは、[Confidence] が [Low] のものから、[Fanout] が [High] から [Low] の順にリストされますが、列に基づいてフィルターすることもできます。列フィルターを使用するには、表のヘッダーを右クリックし、[Enable Column Filtering] をクリックします。

消費電力制約アドバイザーは、次のように使用することをお勧めします。

- [Confidence] 列をクリックして Low が一番上になるように並べ替えます。
- Ctrl キーを押しながら [Fanout] 列を 2 回クリックし、ファンアウトの大きいものからリストされるようにします。
- [Confidence] が [Low] でファンアウト 200 を超える制御ネットすべてを確認し、新しい [Static Probability] と [Toggle Rate] を定義します。
- [OK] をクリックしてデザインに制約を適用し、[Report Power] コマンドを再実行します。

次に、制御セットおよびブロック RAM イネーブルに正確なスイッチング アクティビティを設定するのに役立つ例を示します。

### アクティブ High リセット、スタティック確率 0.9

これは、リセットが 90% の時間 High (アクティブ) であることを示します。ロードセルが 90% の時間リセットされるということなので、これは多すぎます。[Static Probability] を 0、[Toggle Rate] を 0 に設定して、リセットが非アクティブ、つまり、より現実的な状態になるようにスイッチング アクティビティを変更します。

### スタティック確率 0、トグル レート 0 のブロック RAM イネーブル

ブロック RAM がイネーブルにならないことを示していますが、これは非現実的です。[Static Probability] を 0.25、[Toggle Rate] を 50 に設定してイネーブル レートを 25% にするなど、ブロック RAM イネーブルにより現実的なスイッチング アクティビティを割り当ててください。次のコマンドを使用すると、消費電力制約アドバイザーのテキストレポートを生成できます。

```
report_power -advisory -file power_report.pwr
```

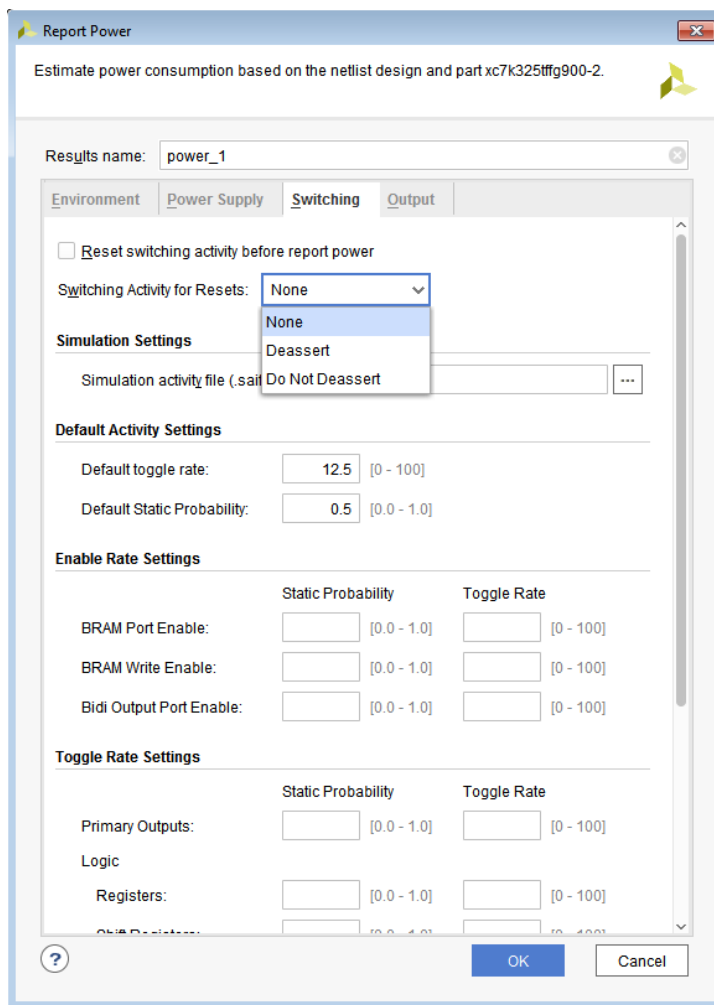
アドバイザーの表は、このレポート ファイルの最後に追加されます。

## リセットのディアサート

デザインのすべてのリセットをディアサートできます。このオプションを使用すると、消費電力見積もり値がハードウェア上の値に近いものになります。このオプションの設定は、ツールから、または Tcl モードで、イネーブルまたはディスエーブルにできます。[Switching Activity for Resets] の設定には、次の 3 つのオプションがあります。

- [None]: これがデフォルト モードです。ツールで値は設定されず、ベクターレス伝搬によるアクティビティをそのままにします。
- [Deassert]: デザインのリセットがすべてディアサートされます。
- [Do Not Deassert]: このモードでは、ディアサート オプションの変更が元の値に戻されます。

図 37: リセットのデアサート



set\_switching\_activity および reset\_switching\_activity コマンドに、次の Tcl オプションが追加されています。

```
set_switching_activity -deassert_resets
```

これは、[Switching Activity for Resets] の [Deassert] と同じです。

```
reset_switching_activity -no_deassert_resets
```

これは、[Switching Activity for Resets] の [Do Not Deassert] と同じです。[Deassert] オプションは、次の例外条件下では設定されません。

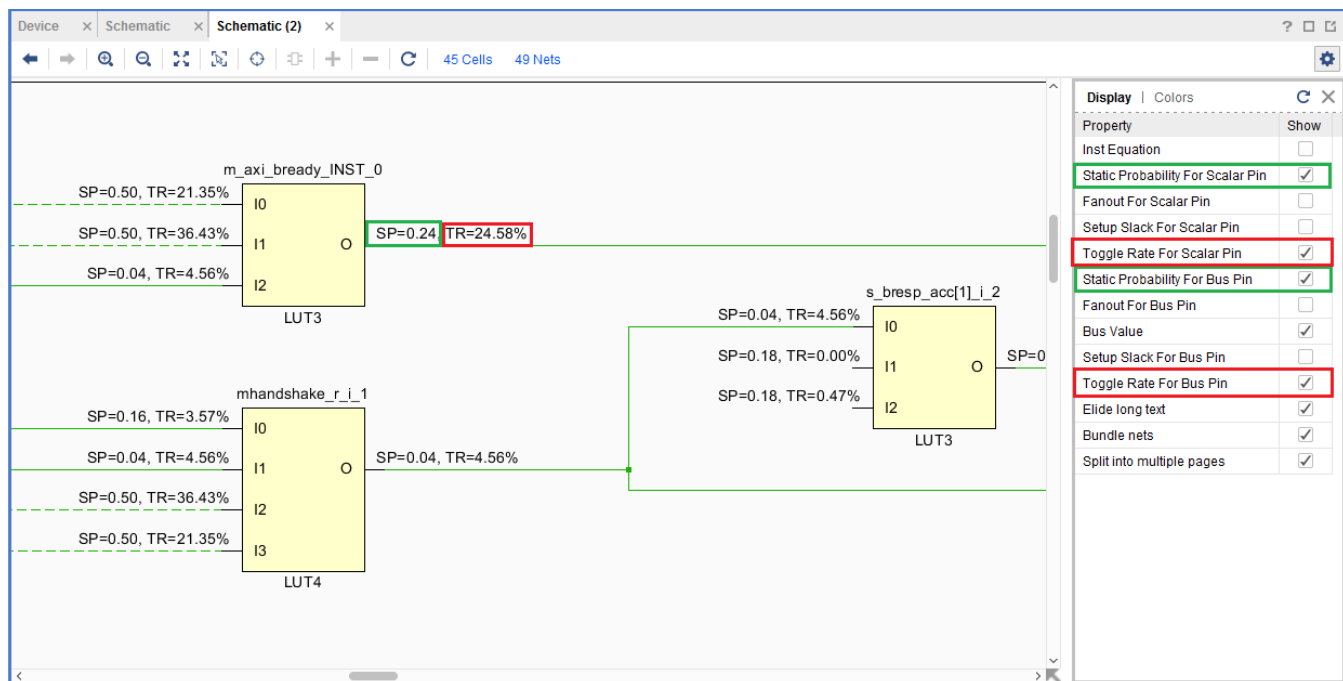
- リセット ネットが異なる極性のピンに接続されている場合。たとえば、リセット ネットがアクティブ High のリセット ピンとアクティブ Low のリセット ピンの両方に接続されている場合、コマンドでこのネットに値は設定されません。
- アクティブ High のリセット ピンに接続されているセットが、同時にアクティブ High のイネーブル ピンにも接続されている場合、このコマンドでは何も実行されません。

- 異なるクロック ドメインをまたぐ際のメタスタビリティを回避するために、非同期クリアと同期ディassertを実行する同期回路に接続されているネット。

## 回路図でのスイッチング アクティビティの表示

この機能を使用すると、回路図内の特定ネットのスタティック確率およびトグル レート情報が確認できます。

図 38: 回路図でのスイッチング アクティビティ



回路図でスイッチング アクティビティを表示するには、回路図ビューの右上の設定アイコンをクリックし、スカラーまたはバス ピンの SP/TR を選択します。

## Tcl インターフェイスを使用した消費電力解析

このセクションでは、Tcl インターフェイスを使用した典型的な消費電力解析フローを説明します。[消費電力解析の Tcl コマンド](#)に、消費電力解析に関連するコマンドをリストします。特定のコマンドのオプション、プロパティ、適用可能なエレメント、および戻り値は、次を参照してください。

- 「<command\_name> -help」と入力。
- 『Vivado Design Suite Tcl コマンド リファレンス ガイド』 (UG835: [英語版](#)、[日本語版](#)) を参照。
- 『Vivado Design Suite ユーザー ガイド: 制約の使用』 (UG903: [英語版](#)、[日本語版](#)) を参照。

## 消費電力解析の Tcl コマンド

- read\_saif
- set\_switching\_activity
- set\_operating\_conditions

- `report_switching_activity`
- `report_operating_conditions`
- `report_power`
- `reset_switching_activity`
- `reset_operating_conditions`
- `set_units`

## 消費電力解析に影響するタイミング制約

- `create_clock`
- `create_generated_clock`
- `set_input_delay`
- `set_case_analysis`

## Tcl プロンプトからの消費電力解析の設定

消費電力見積もりを実行する前に、デバイスの環境、デザイン ネットリストの既知のスイッチング アクティビティ レートを指定する必要があります。これにより、消費電力見積もりが正確になります。

- [デバイス環境](#)
- [ネットリスト エLEMENTのアクティビティ](#)
- `set_case_analysis`

## デバイス環境

次のようなデバイス動作条件をすべて指定します。

- 温度
  - 周辺温度
  - ヒートシンク
- 電圧
  - VCCINT
  - VCCAUX
  - VCCO
- デバイス
  - 温度グレード
  - プロセス コーナー

次のコマンドを使用します。

- `report_operating_conditions`



すべてまたは指定した動作条件設定をレポートします。次に例を示します。

```
report_operating_conditions    # Reports all
report_operating_conditions -voltage
```

- `set_operating_conditions`

指定されている動作条件パラメーターを変更します。次に例を示します。

```
set_operating_conditions -process maximum -junction_temperature 50
set_operating_conditions -voltage {vccint 0.97 vccaux 1.71}
```

- `reset_operating_conditions`

指定されている動作条件パラメーターまたはすべてのパラメーターを、選択したデバイスのデフォルト値に戻します。次に例を示します。

```
reset_operating_conditions    # Resets all
reset_operating_conditions -voltage
```

## ネットリスト エLEMENTのアクティビティ

次のコマンドを使用して、信号またはトグル レートおよびスタティック確率などのスイッチング アクティビティ、および既知のネットリスト エLEMENTのクロック波形情報を定義します。

- `set_switching_activity`

指定したELEMENTのアクティビティを設定します。スタティック確率と信号レート、またはスタティック レートとトグル レートのいずれかを設定できます。次に例を示します。

- プライマリ ポートとデザイン全体のブラック ボックス出力のデフォルトのスイッチング アクティビティを設定する場合

```
set_switching_activity -default_static_probability 0.5 -
default_toggle_rate 12.5
```

- ポート/ネット/ピンの信号レートを設定する場合

```
set_switching_activity -static_probability 0.5 -signal_rate 50
[get_ports din*]
```



**重要:** スタティック確率が 0 より大きく 1 未満の場合、信号レートは 0 より大きくする必要があります。同様に、信号レートが 0 の場合、スタティック確率は 0 または 1 にする必要があります。スタティック確率と信号レートは、一緒に指定する必要があります。

- ポート/ネット/ピンのトグル レートを設定する場合

```
set_switching_activity -static_probability 0.5 -toggle_rate 25 [get_nets
din_int*]
```

トグル レートはELEMENTに関連付けられたクロック専用のもので、有効な値の範囲は 0 ～ 100 です。

- ノード グループに対してスイッチング アクティビティを設定する方法は、次のとおりです。

`set_switching_activity` コマンドで `-type` オプションを使用すると、ノード グループ (タイプ) のアクティビティ レートを設定できます。次の表は、サポートされるタイプを示しています。

表 4: スイッチング アクティビティの Tcl コマンドのタイプ (-type オプション)

タイプ名	スイッチング アクティビティ	ピン名	セル名
bram_enable	ブロック RAM のイネーブル ピン	ENARDEN/ENBWREN	RAMB36/18
bram	ブロック RAM のアクティブ データ出力すべて	DOADO/DOBDO	RAMB36/18
bram_wr_enable	ブロック RAM のライト イネーブル ピン	WEA/WEBWE	RAMB36/18
register	フリップフロップ/ラッチの出力ピン	Q	FD*
shift_register	シフトレジスタの出力ピン	Q	SRL*
lut_ram	RAM の出力データ ピン	O	RAM(32 64 128 256)*
lut	出力ピン	O	LUT*
dsp	DSP データ出力すべて	P/ACOUT/BCOUT/PCOUT	DSP48
gt_txdata	ポートの TX データ	TXDATA	GT*_CHANNEL
gt_rxdata	RX データ出力ポート	RXDATA	GT*_CHANNEL
io_output	プライマリ出力	get_ports -filter {DIRECTION = OUT} && 'I' pin of OBUF* & IOBUF*	OBUF*
io_bidir_enable	Bidir ポートのイネーブル ピン	T	OBUF*

次のセクションに、set\_switching\_activity コマンドの使用方法を説明します。デザインの最上位スコープの LUT に指定したスイッチング アクティビティを設定:

```
set_switching_activity -type lut -static_probability 0.5 -toggle_rate 25
[get_cells]
```

CPU/MEM の階層のすべてのレジスタに指定したトグル レートとスタティック確率を設定:

```
set_switching_activity -type register -toggle_rate 0.4 -static_probability
0.5 [get_cells CPU/MEM]
```

CPU の階層とその下の階層のすべてのレジスタに指定したトグル レートとスタティック確率を設定:

```
set_switching_activity -type register -toggle_rate 0.4 -static_probability
0.5 -hier [get_cells CPU]
```

すべてのプライマリ出力に指定したスイッチング アクティビティを設定:

```
set_switching_activity -type io_output -static_probability 0.5 -toggle_rate
0.4 -all
```



**重要:** トグル レートにグリッチ レートを含めると、 $(toggle\_rate/200) \leq static\_probability \leq 1 - (toggle\_rate/200)$  という条件を満たす必要がありますので、理想的には含めるべきではありません。グリッチ スイッチを考慮するには、信号レート設定と実際のアクティビティ レートを一緒に使用します。



**重要:** set\_switching\_activity を使用してもデザイン クロック ネットには影響しません。クロック ネットのアクティビティを変更するには、タイミング制約 (create\_clock, create\_generated\_clock, set\_case\_analysis など) を使用してください。

- report\_switching\_activity

指定したエレメントのアクティビティをレポートします。スタティック確率、信号レートおよびトグル レートを表示します。割り当てられたスイッチング アクティビティのソースも 表示します。次に、`report_switching_activity` コマンドの例を示します。

- 。 単一ネットのスタティック確率、信号レート、トグル レートをレポート:

```
Vivado% report_switching_activity -static-probability [get_ports clk_p]
clk_p: static probability = 0.5 (C)

Vivado% report_switching_activity [get_ports clk_p]
clk_p: static probability = 0.5 (C) signal rate = 400 (C) toggle rate
= 200 (C)
```

割り当てられたスイッチング アクティビティのソースは、(C) = XDC 制約、(D) = ツールのデフォルト、(S) = SAIF アノテート、(A) = ユーザー割り当て、となります。

- 。 グループ ノードをレポート:

階層 CPU の分散 RAM すべてのスイッチング アクティビティをレポート:

```
report_switching_activity -type lut_ram [get_cells CPU/*]
```

デザインの GT RXDATA すべてのスイッチング アクティビティをレポート:

```
report_switching_activity -type gt_rxdata -all
```

サポートされるタイプの詳細は、[表 4: スイッチング アクティビティの Tcl コマンドのタイプ \(-type オプション\)](#)を参照してください。

- `reset_switching_activity`

特定のネットリスト エレメントのアクティビティ レート (スタティック確率、信号レート、およびトグル レート) をツールのデフォルト値にリセットします。このコマンドは、ユーザー指定の値とシミュレーション アクティビティ レート設定の両方をリセットします。次に例を示します。

- 。 プライマリ ポートとデザイン全体のブラック ボックス出力のデフォルトのスイッチング アクティビティをリセットする場合

```
reset_switching_activity -default
```

- 。 デザイン全体のアクティビティ レートをリセットする場合

```
reset_switching_activity -all
```

- 。 特定のポート/ネット/ピンのアクティビティ レートをリセットする場合

```
reset_switching_activity [get_ports din*]
```

- 。 ノード グループのアクティビティ レートをリセットする場合

デザイン全体の BRAM イネーブル (ENARDEN/ENBWREN) すべてのスイッチング アクティビティをリセットする場合

```
reset_switching_activity -type bram_enable -all
```

階層 CPU/ とその下のレベルのすべての LUT のスイッチング アクティビティをリセットする場合

```
reset_switching_activity -type lut -hier [get_cells CPU/MEM]
```

サポートされるタイプの詳細は、[表 4: スイッチング アクティビティの Tcl コマンドのタイプ \(-type オプション\)](#)を参照してください。

- read\_saif

SAIF シミュレーション出力ファイルを読み込んで、一致するネットリスト エlement にこのファイルに記述されているスイッチング アクティビティをアノテートします。次に例を示します。

```
read_saif -out_file read_saif.rpt -strip_path tb/tb_core/core -file
routed.saif
```

read\_saif のオプションは、次のとおりです。

- out\_file: 不一致のシミュレーションおよびデザイン ネット リストをファイルにダンプします。
- strip\_path: デフォルトでは、デザインの最上位がテストベンチにインスタンス化されていると想定されます。このため、階層の最初の 2 レベルは SAIF データがデザインにアノテートされるときに削除されます。シミュレーション設定に複数の階層レベルが含まれる場合は、SAIF から削除される階層を指定して、実際のデザインに近い状態になるようにします。

read\_saif コマンドでは、SAIF アノテーション サマリも表示され、一致するデザイン ネットの数が表示されます。正確な解析には、デザイン ネットが 100% 一致しているのが理想的です。



**重要:** デザインに暗号化された IP/ブロックが含まれていると、シミュレータではこれらの IP/ブロックおよび暗号化された階層の内部ブロックに対して SAIF 情報が出力されません。この不完全な SAIF 情報が消費電力見積りの精度に影響する可能性があります。read\_saif コマンドでは、デザイン クロック ネットのアクティビティは変更されません。クロック ネット アクティビティは、タイミング制約で駆動されます。

read\_saif コマンドは、各 SAIF ファイルに対して複数回実行できます。これにより、異なるブロックの複数の SAIF ファイルを読み込むことができます。すべての SAIF ファイルのスイッチング アクティビティを考慮して消費電力が見積もられます。同じネットが複数の SAIF ファイルに含まれている場合は、read\_saif コマンドを使用して最後に読み込んだ SAIF ファイルからのスイッチング アクティビティが適用されます。

- create\_clock

クロック波形を指定する合成/インプリメンテーション制約です。次に例を示します。

```
create_clock -name clk -period 5 [get_ports clk]; # 200MHz
```

- create\_generated\_clock

生成クロック波形を指定する合成/インプリメンテーション制約です。次に例を示します。

```
create_generated_clock -name gen_clk -source clk1 -divide_by 2 [get_net -
hier sys_clk]
```

- set\_input\_delay

プライマリ入力を特定のクロックに関連付けます。これはマルチクロック デザインの場合、特にプライマリ ポートが異なるクロックで起動する場合に重要です。次に例を示します。

```
create_clock -name clk1 -period 5 [get_ports clk]
set_input_delay -clock clk1 1 [get_ports d]
```

**注記:** プライマリ ポートがどのクロックとも関連付けられていない場合、スイッチング レートはパスのデスティネーション クロックに基づいて計算されます。

デフォルトでは、create\_clock および create\_generated\_clock は XDC ファイルで定義されているので、再実行する必要はありません。ただし、消費電力レポートのクロック周波数を変更するなど what-if 解析を実行するには、変更を反映させるため create\_clock または create\_generated\_clock を使用する必要があります。

## set\_case\_analysis

グローバル クロック プリミティブ (BUFG、BUFGCE、BUFGCE\_DIV、BUFG\_GT、BUFGCTRL) のクロックのイネーブル/選択は、set\_case\_analysis コマンドで指定します。このコマンドにより、タイミング解析でクロック ロジックを通過するクロックが特定されます。たとえば、タイミング解析で正しいクロックが選択されるように、BUFGMUX のセレクト信号を set\_case\_analysis を使用して設定する必要があります。これにより、消費電力レポートでも正しいクロックを使用して消費電力が見積もられるようになります。BUGCE ブロックの場合は、CE 入力を set\_case\_analysis を使用して設定し、クロック出力をイネーブルまたはディスエーブルにする必要があります。

## 最小限の入力セット

消費電力見積もりの前に、次を実行します。

- アクティビティは、ネットリストのすべてのクロックに対して定義されるようにします。
- 可能であれば、Tcl コマンドを使用してデザインに含まれるプライマリ入力ポートのアクティビティを指定するか、またはシミュレーション出力ファイルを読み込みます。これらのポートのアクティビティ レートが内部ロジックのアクティビティ レートを決定するので、ツールのデフォルト設定がアプリケーションと一致していないと、内部ロジックのアクティビティが正しく見積もられない可能性があります。
- HDL コードで定義されているグローバル セット、リセット、クロック イネーブル信号など、ファンアウトが大きいネットのアクティビティがわかっている 場合は指定します。

シミュレーション結果ファイルを読み込む際には、アクティビティがワースト ケース デザインの論理アクティビティを記述したもの (シミュレーション結果が最大デザイン コード カバレッジが達成された時点のもの) であることを確認してください。ベーシックおよびコーナー ケース テストのシミュレーション結果を使用すると、消費電力を正確に見積もることができない可能性があります。

## Tcl コンソールからの消費電力解析の実行

環境およびアクティビティをすべて 定義したら、report\_power コマンドを使用して消費電力解析アルゴリズムを実行できます。次に例を示します。

```
Vivado% report_power -file routed.pwr -xpe design_top.xpe
```

ツールは次のことを実行します。

1. 環境設定およびデザインのネットリストを読み込みます。
2. 入力ファイルまたは Tcl コマンドを使用して指定したネットリスト エLEMENTのアクティビティをアノテートします。  
**注記:** ノードが未定義の場合は、ベクターレス伝搬エンジンが、既知のエLEMENTのアクティビティ、ロジックの構成および接続に基づいて、アクティビティを見積もります。
3. デザインの温度および電源電力を計算およびレポートします。

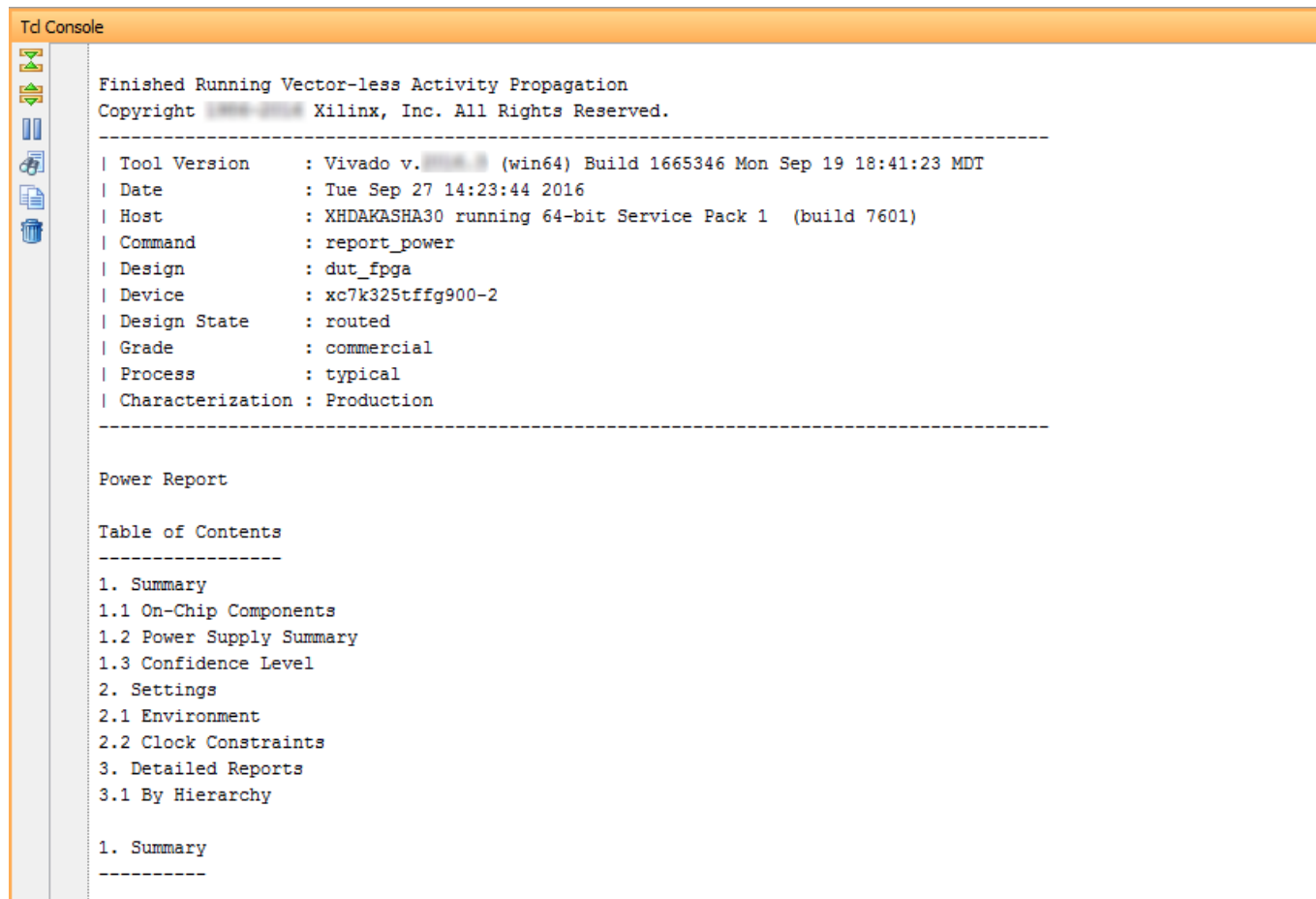
## Tcl コンソールからの消費電力レポートの解析

デザインの消費電力を解析するには、まず消費電力レポートに含まれている温度および電源電力情報を確認します。その後、要件に対するデザイン マージンによりませんが、リソースまたは階層セクションを確認します。これらのセクションには、デザインの消費電力の分配が詳細に示されています。解析結果しだいでは、Xilinx® Power Estimator に戻り、デザインのアーキテクチャ シナリオを実行することが必要な場合があります。また、想定したシナリオを実行し、次の設定を変更した場合の影響を評価することもできます。

- 環境

- デバイス
- インプリメンテーション
- 消費電力ツール

図 39: 消費電力および温度情報を含むテキスト形式レポート



```

Tcl Console

Finished Running Vector-less Activity Propagation
Copyright 2016-2017 Xilinx, Inc. All Rights Reserved.

-----
| Tool Version      : Vivado v.2016.2 (win64) Build 1665346 Mon Sep 19 18:41:23 MDT
| Date              : Tue Sep 27 14:23:44 2016
| Host              : XHDAKASHA30 running 64-bit Service Pack 1 (build 7601)
| Command           : report_power
| Design            : dut_fpga
| Device            : xc7k325tffg900-2
| Design State      : routed
| Grade             : commercial
| Process           : typical
| Characterization   : Production
-----

Power Report

Table of Contents
-----
1. Summary
1.1 On-Chip Components
1.2 Power Supply Summary
1.3 Confidence Level
2. Settings
2.1 Environment
2.2 Clock Constraints
3. Detailed Reports
3.1 By Hierarchy

1. Summary
-----

```

## 使用モデルの例

Vivado に含まれている `cpu_hdl` デザインを使用したスクリプト例を紹介します。上記のセクションで説明したコマンドのほとんどが、これら例に含まれています。消費電力レポートは Tcl コマンドを使用してダイナミックに実行できます。次に例を示します。

```
vivado -mode batch -source power_analysis.tcl
```

Tcl スクリプトを使用することもできます。次に、バッチ モードでスクリプトを実行する場合のスクリプト例を示します。

## 例 1: プロジェクト モードでの合成後およびインプリメンテーション後の消費電力見積もりおよび比較

```
#----- Setup estimation -----

# Open example project with HDL source files and timing constraints
create_project project_1 $work_dir/project_1 -part xc7k70tfbg676-2 -force
set_property target_language VHDL [current_project]
instantiate_example_design -template xilinx.com:design:cpu_hdl:1.0
#----- Run Synthesis then Power estimation -----

# Run Vivado Design Suite synthesis and automatically
launch_runs synth_1
wait_on_run synth_1

#open design
open_run synth_1

# Display tool default assumed operating conditions
report_operating_conditions -all

# Set specific device and environment operating conditions
set_operating_conditions -ambient 25
set_operating_conditions -voltage {vccint 1.0 vccaux 1.71}

# Generate verbose post-synthesis power report
report_power -verbose -file ex1_post-synthesis.pwr

#----- Run Implementation then Power estimation -----

launch_runs impl_1
wait_on_run impl_1

#open design
open_run impl_1

# Generate post-implementation verbose power report
report_power -file ex1_post-implementation.pwr

# Return operating conditions to default for device
reset_operating_conditions -ambient -voltage {vccint vccaux}
```

## 例 2: 非プロジェクト モードでの合成後およびインプリメンテーション後の消費電力見積もりおよび比較

```
#----- Setup estimation -----

# Open netlist in projectless mode
read_edif -name top.edf

# AND link the design
link_design

# OR open Vivado checkpoint
open_checkpoint -file post_synth.dcp

# read design constraints, if it is not part of a design checkpoint (DCP)
read_xdc -name top_full.xdc
```



```
# Display tool default assumed operating conditions
report_operating_conditions -all

# Set specific device and environment operating conditions
set_operating_conditions -ambient 25
set_operating_conditions -voltage {vccint 0.95 vccaux 1.71}

#----- Power estimation at post synthesis -----

# Generate verbose post-synthesis power report
report_power -verbose -file ex1_post-synthesis.pwr

#----Run various Implementation steps then run Power estimation after every
step ----
opt_design
report_power -verbose -file ex1_post-opt_design.pwr
power_opt_design ;# Optional
report_power -verbose -file ex1_post_pwr_opt_design.pwr
place_design
report_power -verbose -file ex1_post_place_design.pwr
phys_opt_design ;# Optional
report_power -verbose -file ex1_post_phys_opt_design.pwr
route_design

# Generate post-route verbose power report
report_power -verbose -file ex1_post_route_design.pwr

# Return operating conditions to default for device
reset_operating_conditions -ambient -voltage {vccint vccaux}
```

### 例 3: リセット時のスタティック確率の値が正確であることを確認

```
# Query the Static Probability value of the reset
report_switching_activity -static_probability [get_ports reset]
# Output is - reset: static probability = 0.5 (D)

# Set Static Probability value and signal rate of reset to 0
set_switching_activity -static_probability 0.0 -toggle_rate 0 [get_ports
reset]

# Generate post-route verbose power report
report_power -verbose -file ex1_post_route_design.pwr
```

### 例 4: what-if デザイン解析/レポート、デザイン アクティビティの変更/リセット

消費電力解析は動的な作業であり、想定したシナリオを即座に試す what-if 解析を実行できます。インプリメント済みデザインを開き、次のコマンドを入力します。これによりサブモジュール `fftEngine` の制御信号(クロック イネーブルおよびリセット)のアクティビティが変更され、この階層レベルおよびデザイン全体の消費電力への影響を評価できます。

```
#----- Report power and activity with default settings
-----

# Report power
report_power -file ex3_power_before.pwr

# Get activity of signals of interest
report_switching_activity [get_nets {fftEngine/reset fftEngine/wb_we_i_reg}]
```

```
#----- scenario with no reset and higher CE activity -----

# disable reset and enable clock enables in module fftEngine most of the
time
set_switching_activity -static_probability 0 -signal_rate 0 [get_nets
fftEngine/reset_reg]
set_switching_activity -static_probability 1 -toggle_rate 0 [get_nets
fftEngine/wb_we_i_reg]
report_power -file ex3_power_no_reset_activ.pwr
report_switching_activity [get_nets fftEngine/reset_reg fftEngine/
wb_we_i_reg]

#----- scenario with active reset and low CE activity -----

# enable reset and disable clock enable in module fftEngine most of the time
set_switching_activity -static_probability 1 -toggle_rate 0 [get_nets
fftEngine/reset_reg]
set_switching_activity -static_probability 0 -signal_rate 0 [get_nets
fftEngine/wb_we_i_reg]
report_power -file ex3_power_reset_activ.pwr
report_switching_activity [get_nets fftEngine/reset_reg fftEngine/
wb_we_i_reg]
```

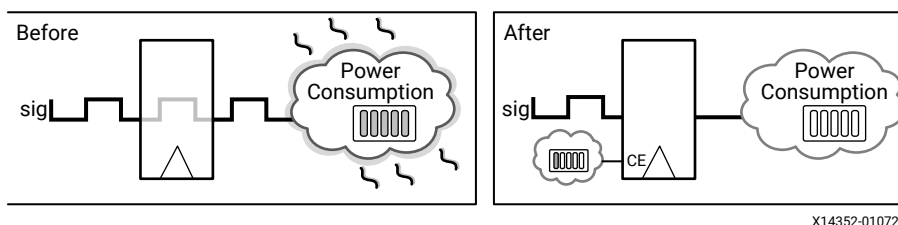
## 消費電力最適化の機能

Vivado® デザイン ツールでは、さまざまな消費電力最適化が提供されており、デザインのダイナミック消費電力を最大 30% 削減できます。これらの最適化では、ASIC で使用されるクロック ゲーティング手法が使用され、デザインの出力に影響しない部分や、そのクロック サイクルでステートのアップデートが不要な部分のアクティビティを最小限に抑えることができます。これらの最適化は、デザイン全体に適用するか、選択した部分のみに適用できます。

デバイスのダイナミック消費電力は、デザインのさまざまなノードにおけるクロック周波数 (f)、ノード容量 (C)、デバイスの動作電圧 (V)、およびアクティビティ (α) から算出されます。ほとんどのデザインでは、上記のパラメーターの一部はデバイス テクノロジ (電圧など) またはデザイン要件 (動作周波数など) により決定されますが、デザイン内のノードにはデバイスの出力には影響しないのにトグルし続けるものもあり、無駄なダイナミック消費電力が発生する原因となっています。このようなノードは、デバイスのクロック イネーブル (CE) を使用してゲーティングできます。これをコード記述で達成することも可能ですが、デザインにはほかのソースからの IP が含まれていたり、このような詳細なゲーティングを設定するには多大な労力を要するので、この手法が使用されることはほとんどありません。Vivado では、このような消費電力の最適化を自動的に実行でき、最小限の労力で最大限の消費電力削減を達成できます。

Vivado は、レガシ IP ブロックおよびサードパーティ IP ブロックを含め、デザイン全体を解析します。各クロック サイクルの結果に影響しないソース レジスタの出力ロジックを特定し、細粒度クロック ゲーティングまたはロジック ゲーティング信号を作成して、不要なスイッチング アクティビティを除去します。

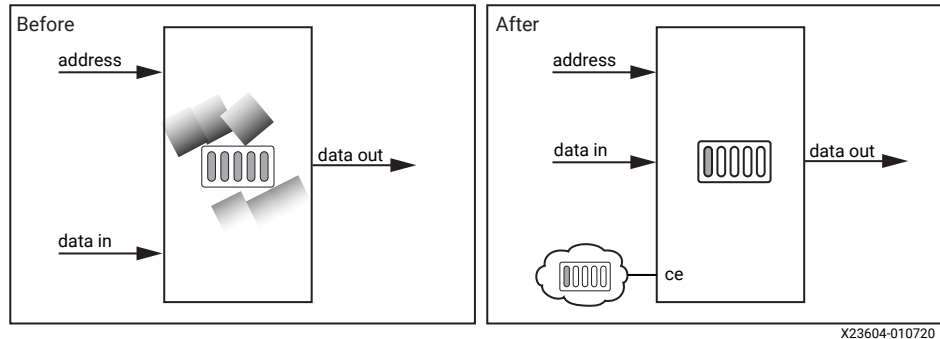
図 40: クロック ゲーティング



X14352-010720

クロックゲーティングによる最適化では、シンプルデュアルポートモードまたは完全なデュアルポートモード両方の専用ブロック RAM の消費電力も削減されます。これらのブロックには、アレイイネーブル、ライトイネーブル、および出力レジスタのクロックイネーブルなどのイネーブル信号があります。節約される消費電力のほとんどはアレイイネーブルの使用によるもので、データが書き込まれず、出力が使用されないときに、消費電力を削減する機能がインプリメントされます。

図 41: クロックゲーティングによる最適化でブロック RAM イネーブルを活用



ザイリンクスのクロックゲーティング最適化では、追加のゲーティングロジックが作成され、ユーザーロジックが変更されることはありませんので、デザインの機能は常に保持されます。ただし、この最適化によって、特にこの最適化がクリティカルパスに使用された場合などは、タイミングが影響が出る場合があります。

## ブロック RAM の WRITE\_MODE の消費電力最適化

ザイリンクス 7 シリーズ デバイスでは、完全なデュアルポート (TDP) のブロック RAM の場合、そのポートの出力が接続されていないか、書き込み中に不要であれば、WRITE\_MODE は WRITE\_FIRST から NO\_CHANGE に変更できます。同様に、完全なデュアルポート (TDP) のブロック RAM では、そのポートの出力が接続されていなければ、WRITE\_MODE は WRITE\_FIRST から NO\_CHANGE に変更できます。

UltraScale™ デバイスでは、上記の最適化に加え、シンプルデュアルポート (SDP) モードの場合、読み出しポートと書き込みポートの WRITE\_MODE は、読み出しおよび書き込みポートのクロックが非同期であれば、安全に NO\_CHANGE に変更できます。これらの変更により、ブロック RAM の出力ポートがアップデートされなくなるので、書き込みサイクルの消費電力を削減できます。この最適化は、ユーザー定義の機能およびパフォーマンスに影響のない場合にのみ実行されます。

## ブロック RAM のカスケード最適化

ザイリンクス 7 シリーズ デバイスでブロック RAM がカスケード接続される場合、一度にアクティブにできるブロック RAM は 1 つのみなので、残りのブロック RAM はアドレスおよび既存のイネーブル条件に基づいてディスエーブルにできます。そうすると消費電力を大幅に削減できます。これらの最適化は、Vivado® Design Suite の opt\_design 段階でデフォルトで実行されます。

## Vivado IDE での消費電力最適化

Vivado での消費電力最適化は、次の 2 つの段階で実行します。

- opt\_design
- power\_opt\_design

opt\_design 中に実行される最適化は、ユーザーの操作なしで自動的に実行されます。これらの最適化は、主にブロック RAM の消費電力削減を目的としています。



**重要:** 消費電力最適化は、opt\_design、power\_opt\_design のいずれかまたは両方で、デザインのタイミング パフォーマンスに影響することがあります。

UltraScale™ デバイスでは、タイミングに悪影響を及ぼす可能性のあるより積極的なブロック RAM 消費電力最適化は、power\_opt\_design にのみ含まれます。つまり、消費電力を抑えるよりもパフォーマンスを優先させることができます。UltraScale+™ デバイスでは、power\_opt\_design で XPM-URAM 消費電力最適化が実行されます。

デフォルトでは、opt\_design コマンドでブロック RAM の消費電力最適化が実行されます。ブロック RAM の消費電力最適化は、-bram-power-opt オプションを使用して明示的にスタンドアロンで実行することもできます。

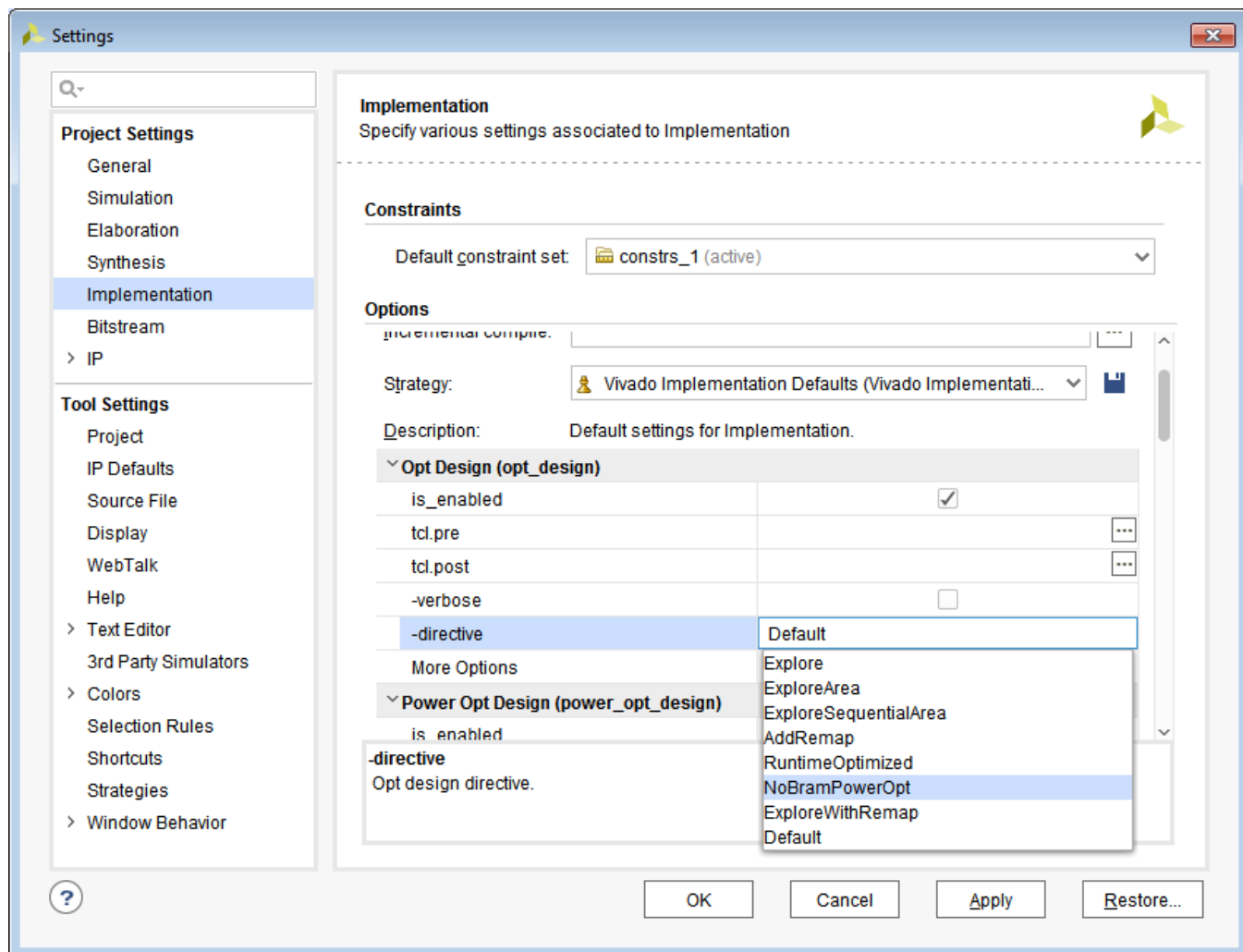
```
opt_design -bram-power-opt
```

ブロック RAM の消費電力最適化をデフォルトの opt\_design フローからディスエーブルにするには、opt\_design コマンドに NoBramPowerOpt 指示子を設定します。

```
opt_design -directive NoBramPowerOpt
```

この指示子は、[Settings] ダイアログ ボックスの [Implementation] ページからも設定できます。

図 42: 最適化中のブロック RAM の消費電力最適化をオフ

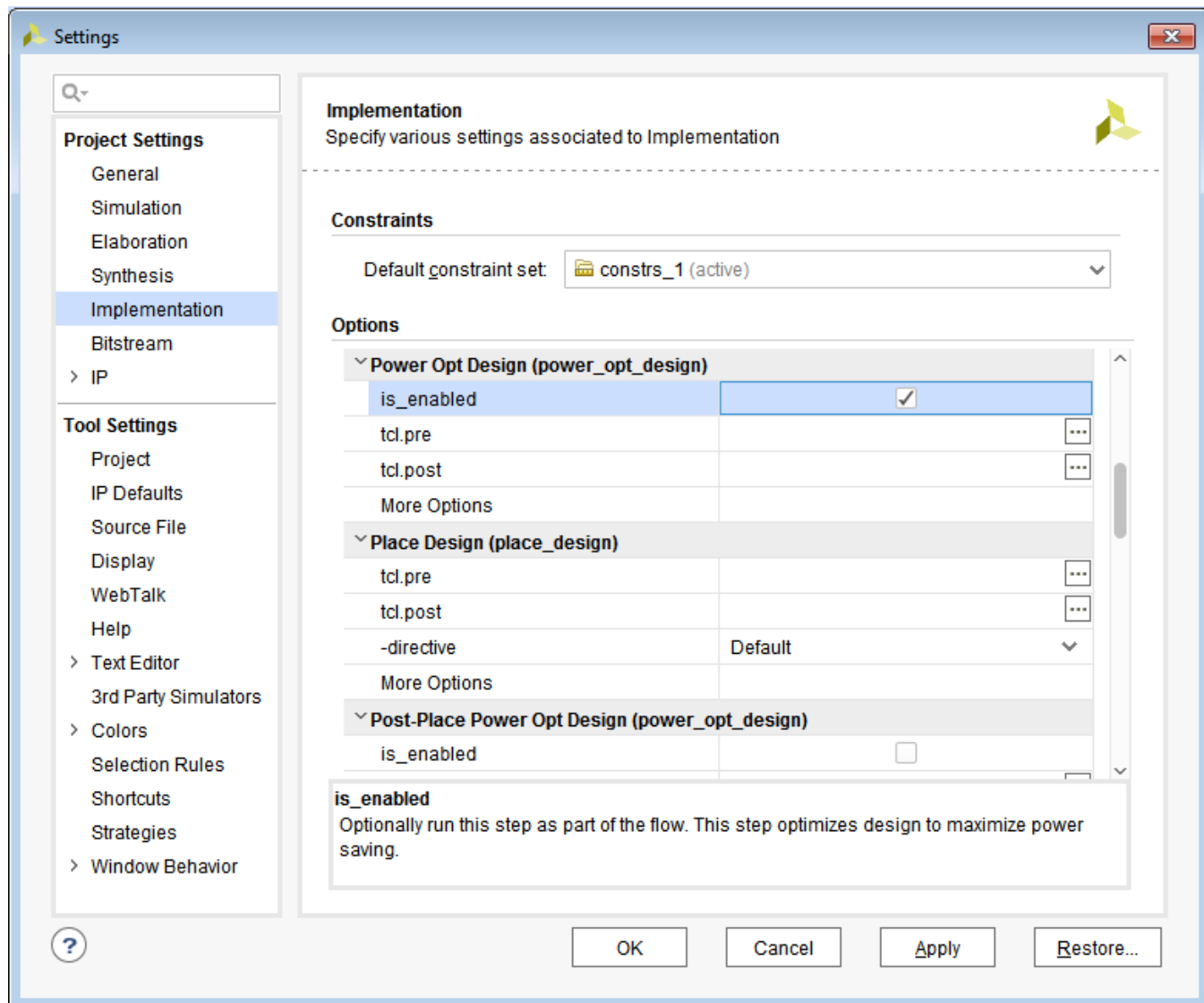


Vivado® IDE で `power_opt_design` を使用した消費電力最適化をイネーブルにするには、[Tools]→[Project Settings]→[Implementation]→[Power Opt Design] をクリックし、[is\_enabled] オプションをオンにします。このオプションをオンにすると、消費電力最適化が Vivado IDE のインプリメンテーションの一部として実行されます。最適化を詳細に設定し、その結果をレポートする場合は、[消費電力解析の Tcl コマンド](#)を参照してください。



**重要:** [Power Opt Design] は、デザイン フローの配置前または配置後のいずれかのタイミングでイネーブルにできますが、両方でイネーブルにすることはできません。詳細は、「消費電力最適化の実行」を参照してください。

図 43: 消費電力最適化オプション



## Vivado IDE での消費電力最適化レポートの表示

Vivado® IDE では、デザインで実行された消費電力最適化のレポートを表示できます。この消費電力最適化レポートは、合成またはインプリメンテーション後に表示できます。



**重要:** Vivado では、消費電力最適化は Vivado デザイン フローの `opt_design` および `power_opt_design` 段階で実行されます。どちらもデザイン合成後のインプリメンテーション中の段階です。合成済みデザインで消費電力最適化レポートを生成すると、レポートには、元のデザイン コードに記述された消費電力最適化機能に関する情報(クロック イネーブル (CE) を使用したブロック RAM のゲート制御など)のみが含まれます。インプリメンテーション中に自動的に実行される消費電力最適化情報は含まれません。

Vivado IDE での消費電力最適化レポートを表示するには、次の手順に従います。

1. Flow Navigator で [Open Synthesized Design] または [Open Implemented Design] をクリックします。

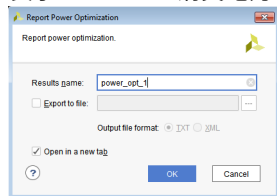
2. [Reports]→[Report Power Optimization] をオンにします。

次の Tcl コマンドでも同じことが実行できます。

```
report_power_opt -name <report_name>
```

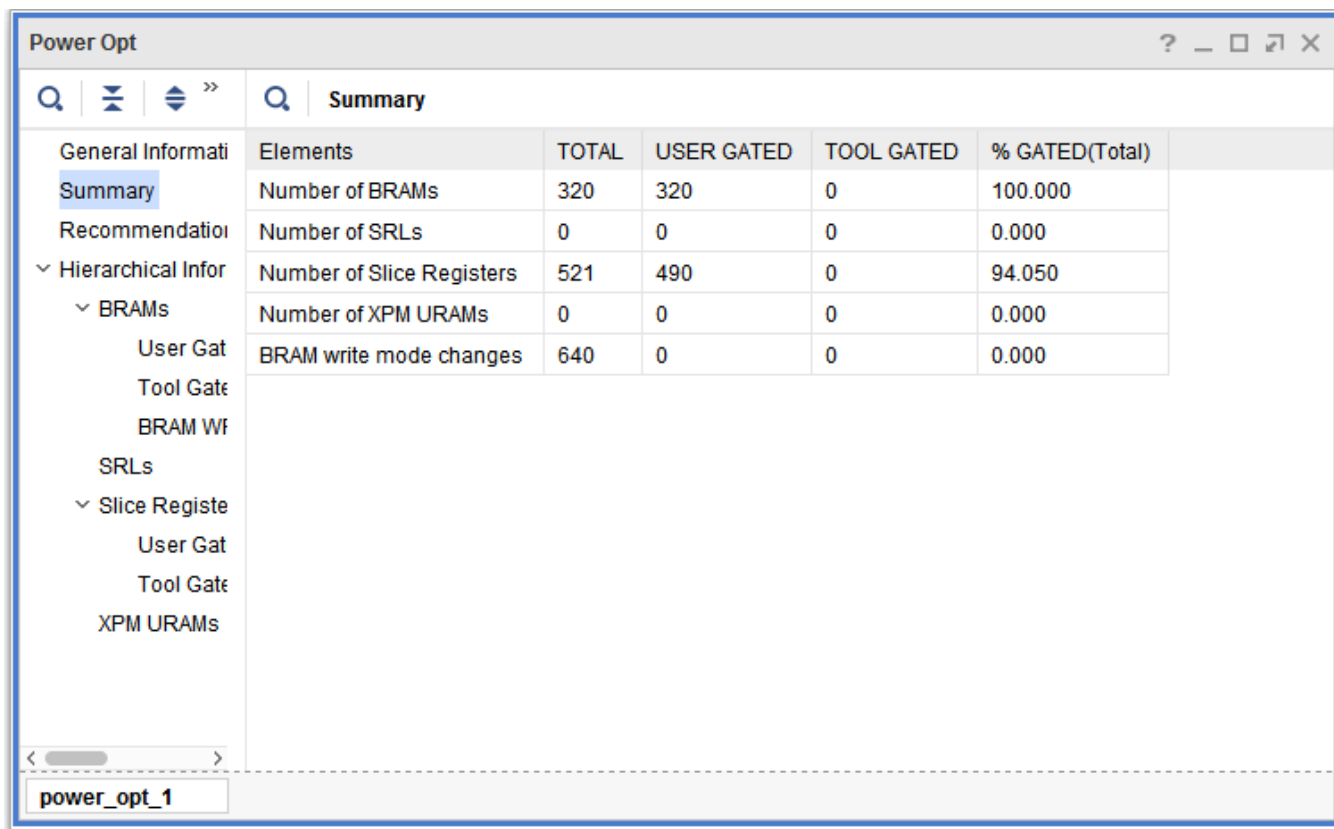
3. [Report Power Optimization] ダイアログ ボックスで次のオプションを指定します。

- [Results name]: Vivado IDE で表示される消費電力最適化レポートの名前を指定します。
- [Export to file]: Vivado IDE に消費電力最適化レポートを表示するのに加え、テキスト形式のレポートも生成します。テキスト ファイルのファイル名とディレクトリを指定し、[TXT] または [XML] 形式を選択します。
- [Open in a new tab]: Vivado IDE に表示されている消費電力最適化レポートはそのままにし、新しく生成された消費電力最適化レポートを新しいタブに開きます。このチェック ボックスをオフにすると、Vivado に現在表示されている消費電力最適化レポートが新しい消費電力最適化レポートで上書きされます。



4. [OK] をクリックします。

消費電力最適化レポートが Vivado IDE の下部に表示されます。



Elements	TOTAL	USER GATED	TOOL GATED	% GATED(Total)
Number of BRAMs	320	320	0	100.000
Number of SRLs	0	0	0	0.000
Number of Slice Registers	521	490	0	94.050
Number of XPM URAMs	0	0	0	0.000
BRAM write mode changes	640	0	0	0.000

消費電力最適化レポートは、次のカテゴリ別に表示されます。



- [General Information]: デザインに関する情報、デザインがインプリメントされるサイリンクス デバイス、この消費電力最適化レポートを生成した Tcl コマンドなどの情報を示します。
- [Summary]: ユーザーまたは消費電力最適化ツールにより最適化されたブロック RAM、SRL、スライス レジスタの数を示します。
- [Recommendations]: 消費電力をさらに最適化するために推奨される項目を示します。
- [Hierarchical Information]: Vivado で消費電力最適化が実行されたブロック RAM、SRL、およびシフト レジスタの詳細を示します。

Vivado IDE で実行される消費電力最適化の詳細は、[消費電力最適化の機能](#) および [ブロック RAM の WRITE\\_MODE の消費電力最適化](#) を参照してください。



ヒント: 階層モジュールまたはインスタンスに DONT\_TOUCH 属性が設定されている場合は、そのロジックが消費電力最適化により削除されることはありません。

## Tcl インターフェイスを使用した消費電力最適化の実行

Vivado® には、消費電力最適化用に 4 つの Tcl コマンドがあります。

- `set_power_opt`
- `opt_design -bram_power_opt`
- `power_opt_design`
- `report_power_opt`

これらのコマンドは、消費電力最適化を有効にし、デザインの最適化を適用する部分を指定し、実行された最適化の効果をレポートします。特定のコマンドのオプション、プロパティ、適用可能なエレメント、および戻り値は、次を参照してください。

- 「<command\_name> -help」と入力
- 『Vivado Design Suite Tcl コマンド リファレンス ガイド』 (UG835: [英語版](#)、[日本語版](#)) および 『Vivado Design Suite ユーザー ガイド: 制約の使用』 (UG903: [英語版](#)、[日本語版](#)) を参照

## 消費電力最適化制約の設定

消費電力最適化を実行する前に、オプションで消費電力最適化制約を設定し、消費電力最適化を実行する箇所を特定できます。`set_power_opt` コマンドを使用すると、消費電力最適化に含める/除外するセル タイプ、階層レベル、クロック ドメインなどを指定できます。



ヒント: この場合も、`power_opt_design` コマンドを使用して消費電力最適化を有効にする必要があります。`set_power_opt` コマンドは、最適化を実行する部分を指定するためにのみ使用します。

`set_power_opt` コマンドの構文は、次のとおりです。

```
set_power_opt [-include_cells <args>] [-exclude_cells <args>] [-clocks
<args>] [-cell_types <args>] [-quiet] [-verbose]
```

表 5: set\_power\_opt のオプション

オプション名	オプション	デフォルト	説明
-include_cells	○	すべて	クロック ゲーティングに含めるセルを指定します。
-exclude_cells	○	なし	クロック ゲーティングから除外するセルを指定します。
-clocks	○	すべてのクロック	指定したクロックが供給されるセルにクロック ゲーティングを適用します。
-cell_types	○	すべて	次のセル タイプのみのクロック ゲート: [all bram uram reg srl none]
-quiet	○	N/A	コマンド エラーを表示しません。
-verbose	○	N/A	メッセージの非表示設定を解除し、すべてのメッセージを表示します。

## 例

次の例では、ブロック RAM および REG セルに対して消費電力最適化を設定し、その後 SRL を追加しています。

```
set_power_opt -cell_types {bram reg}
set_power_opt -cell_types {srl}
```

次の例では、BRAM セルにのみ消費電力最適化を設定し、その後 cpuEngine ブロックを最適化から除外して、cpuEngine/cpu\_dbg\_dat\_i ブロックを追加しています。

```
set_power_opt -cell_types bram
set_power_opt -exclude_cells cpuEngine
set_power_opt -include_cells cpuEngine/cpu_dbg_dat_i
```

## 消費電力最適化の実行

消費電力最適化は、消費電力を最小にするため、デザイン全体またはデザインの一部 (set\_power\_opt を使用する場合) に実行できます。消費電力最適化は、デザイン フローの配置前または配置後のいずれかで実行できますが、両方では実行できません。配置前の消費電力最適化では、消費電力の削減量を最大にすることに焦点が置かれ、まれにタイミングが悪化することがあります。タイミングを保持することが主な目標である場合は、消費電力最適化を配置後に実行することをお勧めします。配置後の消費電力最適化では、タイミングを保持する消費電力最適化のみが実行されます。また、配置後に phys\_opt\_design -bram\_enable\_opt を実行すると、タイミングに影響するブロック RAM イネーブルの最適化の一部を元に戻すこともできます。典型的な配置前の消費電力最適化のスクリプトは、次のようになります。

```
synth_design
opt_design
power_opt_design
place_design
route_design
report_power
```

このコマンドの構文は、次のとおりです。

```
power_opt_design [-quiet] [-verbose]
```

表 6: power\_opt\_design のオプション

オプション名	オプション	デフォルト	説明
-quiet	<input type="radio"/>	なし	コマンド エラーを表示しません。
-verbose	<input type="radio"/>	なし	メッセージの非表示設定を解除し、すべてのメッセージを表示します。

## 消費電力最適化テキスト レポートの生成

report\_power\_opt コマンドを実行すると、ブロック RAM、SRL、レジスタなど、消費電力最適化が実行されたすべてのセルがテキスト レポートに階層別に示されます。各セルに使用されたイネーブルに関する情報と、イネーブルが Vivado® で作成されたのかユーザーにより作成されたのかが示されます。このコマンドの構文は、次のとおりです。

```
report_power_opt [-cell <arg>] [-file <arg>] [-quiet] [-verbose]
```

表 7: report\_power\_opt のオプション

-オプション名	オプション	デフォルト	説明
-cell	<input type="radio"/>	最上位	指定したセルの消費電力最適化をレポートします。
-file	<input type="radio"/>	なし	レポートを出力するファイルを指定します。指定したファイルが既に存在する場合は、上書きされます。
-quiet	<input type="radio"/>	なし	コマンド エラーを表示しません。
-verbose	<input type="radio"/>	なし	メッセージの非表示設定を解除し、すべてのメッセージを表示します。

### 例

次の例では、myopt.rep というファイルを作成し、デザイン全体の消費電力最適化をレポートしています。

```
report_power_opt -file myopt.rep
```

次の例では、myopt.rep というファイルを作成し、デザインの mctrl0 サブ階層の消費電力最適化をレポートしています。

```
report_power_opt -file myopt2.rep -cell mcore0/mctrl0
```

## 消費電力最適化を使用して消費電力を最大限に削減するためのガイドライン

Vivado® ツールで消費電力最適化を実行したときに消費電力が最大限に削減されるようにするには、消費電力最適化をデザイン全体に実行し、デザインの一部を除外しないようにします。消費電力最適化を有効にしても消費電力が削減されない場合は、デザインが正しく制約されているかどうかを確認してください。デザインのすべてのレジスタに制約が設定されているかどうかを確認するには、check\_timing コマンドを使用します。

デザインが正しく制約されている場合は、消費電力最適化に影響しているかもしれないコーディング スタイルがないかどうか確認します。デバッグが必要になる可能性があるのは、グローバル セット/リセット信号、ブロック RAM イネーブル生成、およびレジスタ クロック ゲーティングの 3 つです。消費電力最適化で生成されたイネーブルの数が少なければ、コード記述方法、または合成およびインプリメンテーションのオプション/プロパティを見直す必要があります。

- グローバル セット/リセット信号

特にデータパス、パイプライン フリップフロップ、ブロック RAM に対しては、非同期セット/リセット信号をできるだけ使用しないようにします。

また、`power_opt_design` でグローバル セット/リセット信号に `dont_touch` プロパティを設定し、イネーブルとして使用されないようにすることも考慮します。HDL で `dont_touch` プロパティを設定すると、フローのすべての段階でこの属性が適用されます。このオプションは、消費電力最適化用の XDC 制約として設定することをお勧めします。次にその例を示します。

```
set_property DONT-TOUCH true [get_cells u1]
```

最後に、消費電力最適化およびベクターレス消費電力見積もりの前に、グローバル セット/リセット信号の信号レートおよび確率が正しく設定されていることを確認してください。

- スライス レジスタおよび SRL

`power_opt_design` でデザインのスライス レジスタまたは SRL に対してクロック イネーブルが生成されない原因は、多数あります。次に、その例を示します。

- 。 デザインに組み合わセループが存在する
- 。 デザインへのプライマリ入力ソースとなるフリップフロップおよび SRL にセット/リセット信号が使用されている
- 。 データパス フリップフロップに非同期セット/リセット信号が使用されている
- 。 デザインに多数のクロック ドメインがあり、クロック ドメインをまたがっているためにイネーブルが生成されない
- 。 SRL のサイズ: SRL のシフト レジスタの段数が多いほど、すべての段に対して 1 つのクロック イネーブルを生成するのが困難になります。

- ブロック RAM

ブロック RAM (BRAM) が多数使用されるデザインでは、消費電力を大きく削減できる可能性があります。Vivado では、さまざまな最適化手法を使用してイネーブルを生成し、消費電力を削減します。`power_opt_design` を使用してもブロック RAM ゲーティングがそれほど適用されない場合は、次のような原因が考えられます。

- 。 ブロック RAM が主に FIFO18/FIFO36 セルである。これらのインスタンスはツールで最適化できません。
- 。 推論またはインスタンス化されたメモリが、A ポートと B ポートに非同期クロックを使用する完全なデュアル ポート (TDP) モードであり、`power_opt_design` で最適化できない。
- 。 ブロック RAM 自体、またはブロック RAM に供給されるアドレス/ライト イネーブル フリップフロップに非同期リセットが使用されている。

## 消費電力最適化後のタイミングの保持

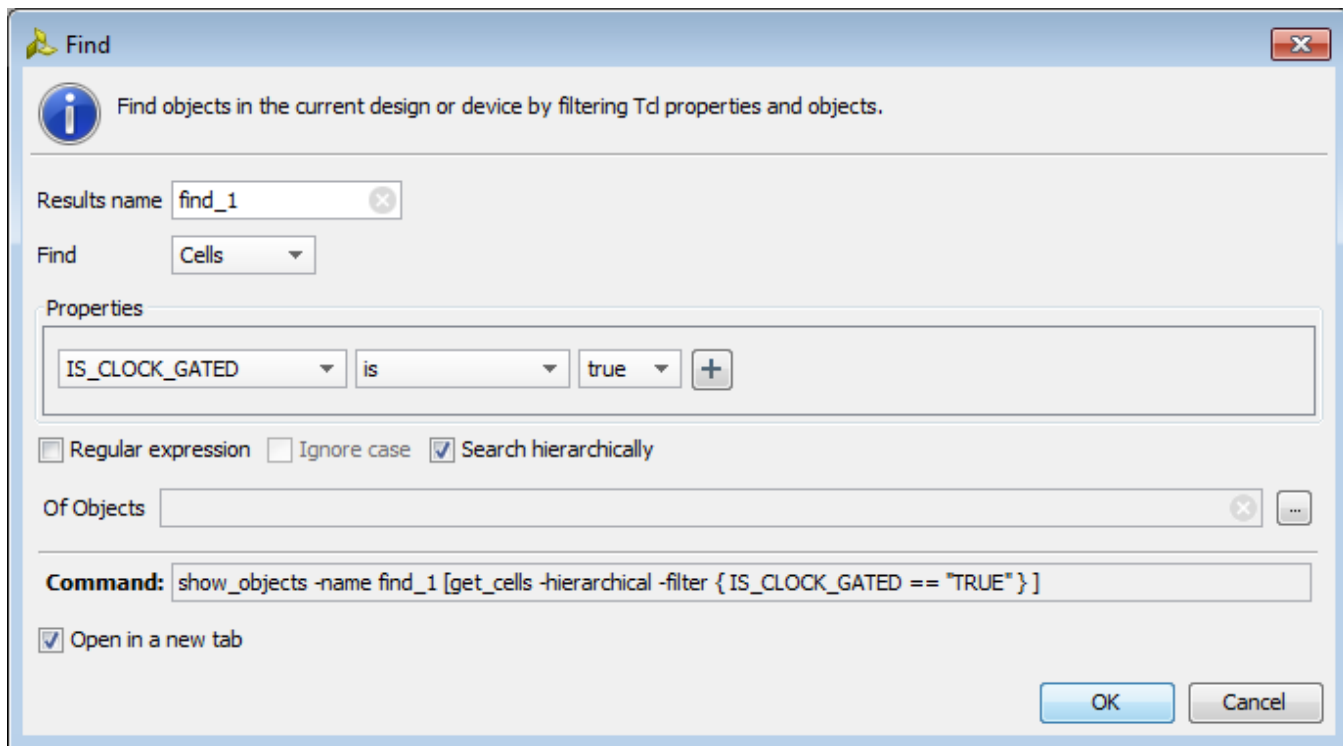
消費電力最適化では、消費電力を最大限に削減しながら、タイミングへの影響は最小限に抑えます。消費電力最適化によりタイミングが悪化した場合は、この影響を補正するための手法を使用できます。

可能な場合、タイミング クリティカルでないクロック ドメインまたはモジュールを特定し、`set_power_opt` XDC コマンドを使用してそれらだけに消費電力最適化を適用します。最もクリティカルなクロック ドメインがデザインの大部分を占めている場合や電力の大部分を消費している場合は、クリティカル パス上のセルが消費電力最適化で削除されていないかを確認します。消費電力最適化で最適化されたオブジェクトには、`IS_CLOCK_GATED` プロパティが設定されます。これらのセルを消費電力最適化から除外します。クロック ゲーティングが適用されたセルを検索するには、次の `Tcl` コマンドを使用します。

```
get_cells -hier -filter {IS_CLOCK_GATED==1}
```

[Find] ダイアログ ボックスを使用して、これらのセルを検索できます。

図 44: 消費電力最適化されたセルの検索



より簡単な方法は、ブロック RAM への消費電力最適化を制限することです。これによりタイミングへの影響は最小限に抑えられますが、その効果はデザインで使用されているブロック RAM の数や、ブロック RAM がどのようにゲーティングされているかによります。ブロック RAM への消費電力最適化を制限するには、`opt_design` または `power_opt_design` コマンドを実行する前に `set_power_opt -cell-types {bram}` コマンドを実行します。

# Vivado 消費電力レポートを使用した正確な消費電力解析

## 概要

消費電力解析では、ツールでさまざまな要因を想定する必要があるため、正しい見積もりを得るのは簡単なことではありません。ユーザーができるだけ情報を指定してこれらの想定を最小限に抑えることにより、より正確な消費電力を得ることができます。正確な消費電力解析を実行するには、次の点を考慮する必要があります。

- [温度設定](#)
- [電源設定](#)
- [クロック仕様](#)
- [制御信号](#)
- [プライマリ入力](#)
- [コンポーネント レベル](#)

## 温度設定

スタティク消費電力は、理想的にはトランジスタのソースからドレインへのリーク電力とゲート リーク電力の合計です。スタティク消費電力は、温度条件に完全に左右されます。正確に消費電力を見積もるには、より正確な温度情報を提供することが基本です。

## プロセス コーナー

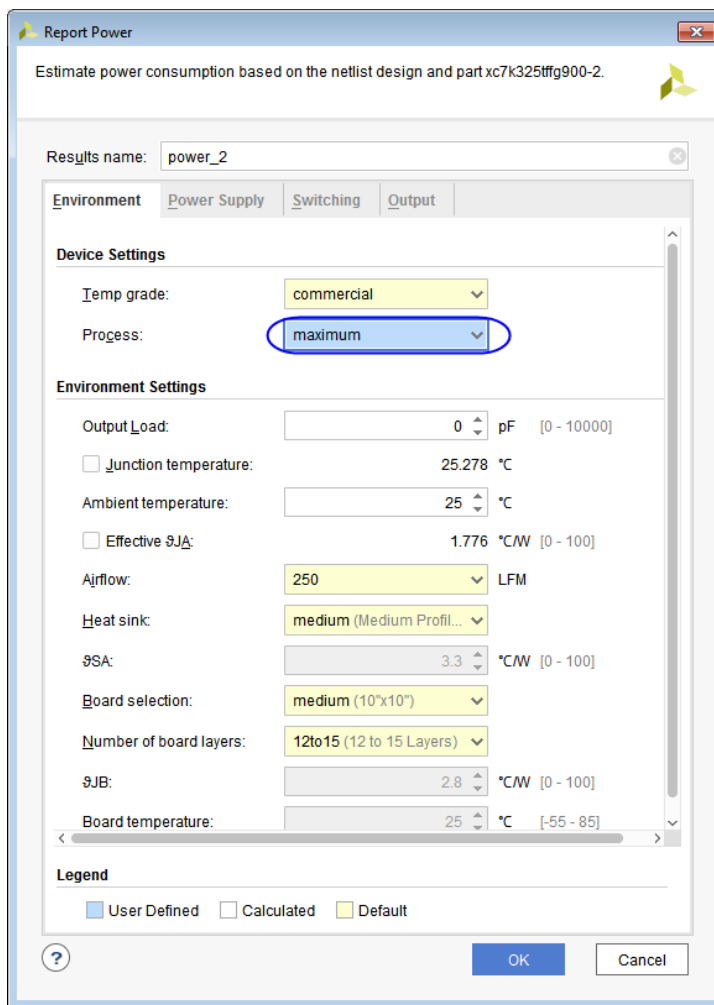
デバイスが製造されるとき、その製造プロセスによって、デバイスごとにパフォーマンスおよび消費電力は変わります。[Report Power] コマンドでは、標準 ([typical]) と最大 ([maximum]) の 2 つのプロセス コーナーのスタティク消費電力を見積もることができます。すべてのデバイスが [typical] の見積もり値をみたとするのが理想的ですが、プロセスの違いによってデバイスにばらつきがあるので、[typical] の値が中心となるように、特定のデバイスのプロセス変動に基づいて手動で調整する必要があります。[maximum] 設定を使用すると、レポートされる数値が動作範囲内に収まり、ハードウェアの測定値に近いものになります。ジャンクション温度が固定されている場合、[typical] と [maximum] のスタティク消費電力に予測される変動は、コマーシャル デバイスで約 2.5 倍です。



**重要:** 正確なワースト ケースのスタティク消費電力を取得するには、[Process] を [maximum] に設定します。

Vivado® では、[Report Power] ダイアログ ボックスの [Process] のデフォルト設定は [typical] です。これは、[Report Power] ダイアログ ボックスの [Environment] タブで [maximum] に変更できます。

図 45: [Report Power] ダイアログ ボックスの [Process] 設定



同等 Tcl コマンド:

```
set_operating_conditions -process maximum
```

## ジャンクション温度

リーク電流はジャンクション温度の増加に応じて指数関数的に増加するので、スタティック消費電力が高くなる要因となります。ジャンクション温度は、デバイスの全消費電力、冷却システム、ボード選択、周囲条件などのさまざまな要因に依存します。デフォルトでは、ジャンクション温度は周囲温度、ヒートシンク、ボード選択などのその他の温度設定入力に基づいて算出されます。ジャンクション温度は全消費電力に直接比例するため、ダイナミック消費電力が増加すると上昇します。正確なスタティック消費電力を見積もるには、正しいジャンクション温度を指定することが非常に重要です。

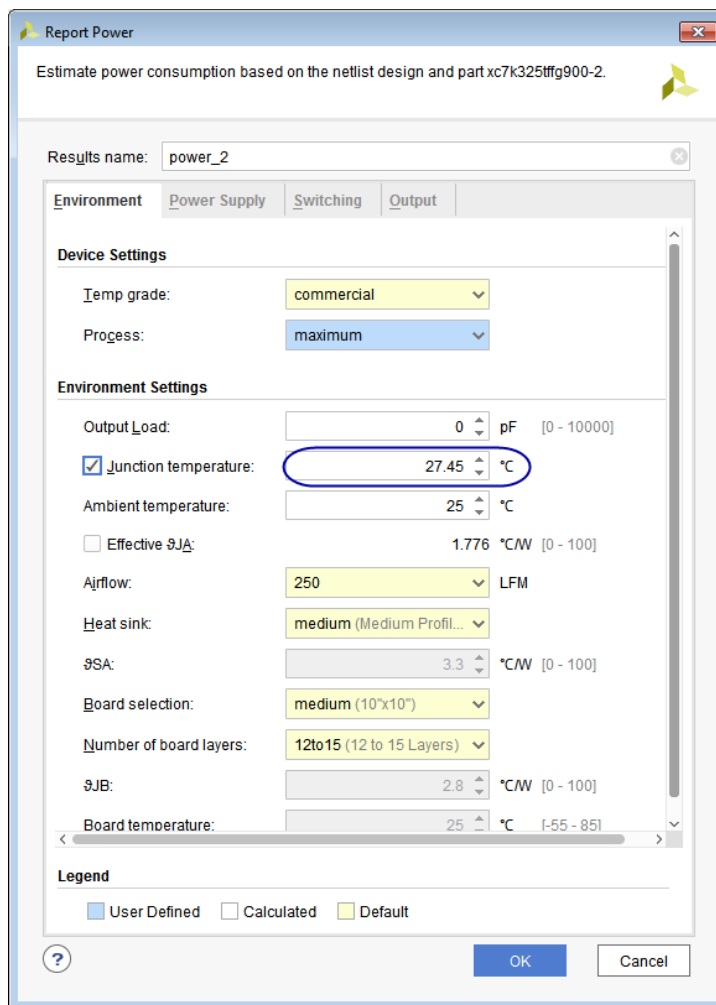


**重要:** ジャンクション温度はハードウェア上の電力を測定する際に読み出し、[Report Power] ダイアログ ボックスの既存の設定を上書きしてください。

Vivado® IDE でジャンクション温度を設定するには、[Report Power] ダイアログ ボックスの [Environment] タブで [Junction temperature] をオンにし、値を入力します。



図 46: [Report Power] ダイアログ ボックスでのジャンクション温度の設定



同等 Tcl コマンド:

```
set_operating_conditions -junction_temp 45
```

ザイリンクス デバイスに単純なサーミスターなどの携帯型温度測定デバイスを配置すると、大まかなジャンクション温度を測定できます。ザイリンクス ハードウェア プログラム ツールのいずれかを使用してデバイスをプログラムすると、ダイ温度値を読み出すことができます。たとえば、ISE の iMPACT では [Debug] → [Read Status Register] をクリックするとダイ温度値を読み出すことができます。Vivado ハードウェア マネージャーでは、システム モニター ウィンドウにダイ温度のプロット図が描画されます。

## 電源設定

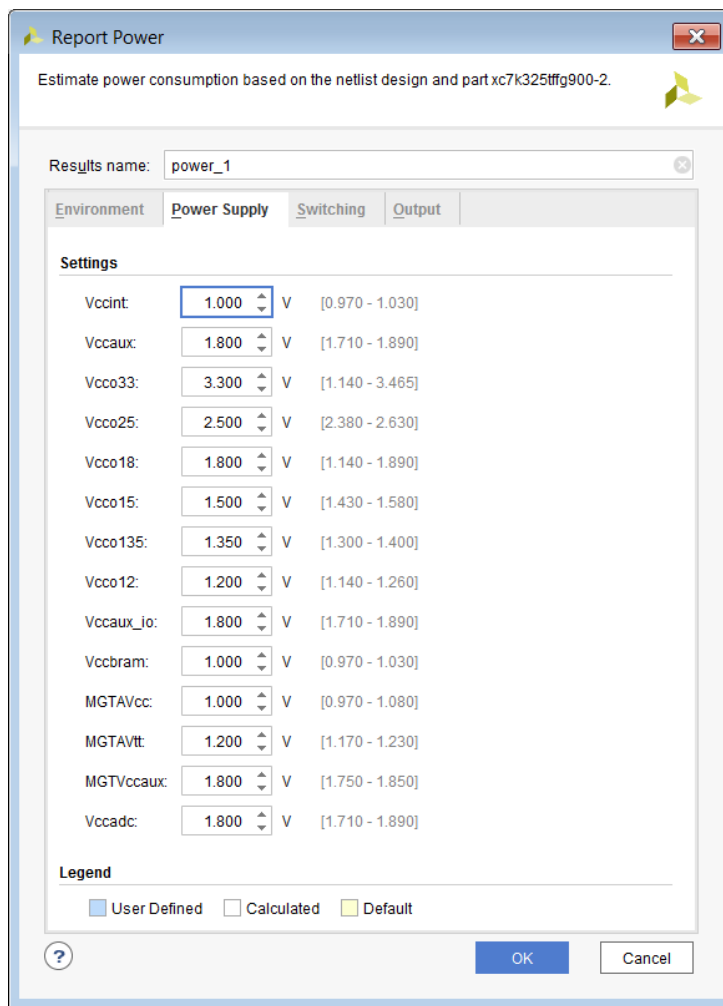
電圧スケールリングは、ザイリンクス デバイスにおける効果的な消費電力削減方法の 1 つです。デバイスには、ロジックに特定の電圧を提供するさまざまな電圧レールがあります。各デバイスのデータシートには、これらの電圧レールの推奨動作条件がリストされています。たとえば、Kintex®-7 デバイスは 0.97V ~ 1.03V の電圧の VCCINT レールで動作します。消費電力バジェットを満たすために電圧スケールリングを利用できます。ハードウェア設定では、これらのレールに詳細な制御が可能な外部電源レギュレータにより電源が供給されます。電源電圧が増加すると消費電力も増加するため、正しい消費電力を見積もるには、正確な電源電圧を指定する必要があります。



**重要:** [Report Power] ダイアログ ボックスの [Power Supply] タブで正確な電源値を指定します。

Vivado® IDE で電源電圧を指定するには、[Report Power] ダイアログ ボックスの [Power Supply] タブに値を入力します。

図 47: [Report Power] ダイアログ ボックスでの電源値の設定



同等 Tcl コマンド:

```
set_operating_conditions -voltage {vccint 0.98 vccaux 1.8}
```

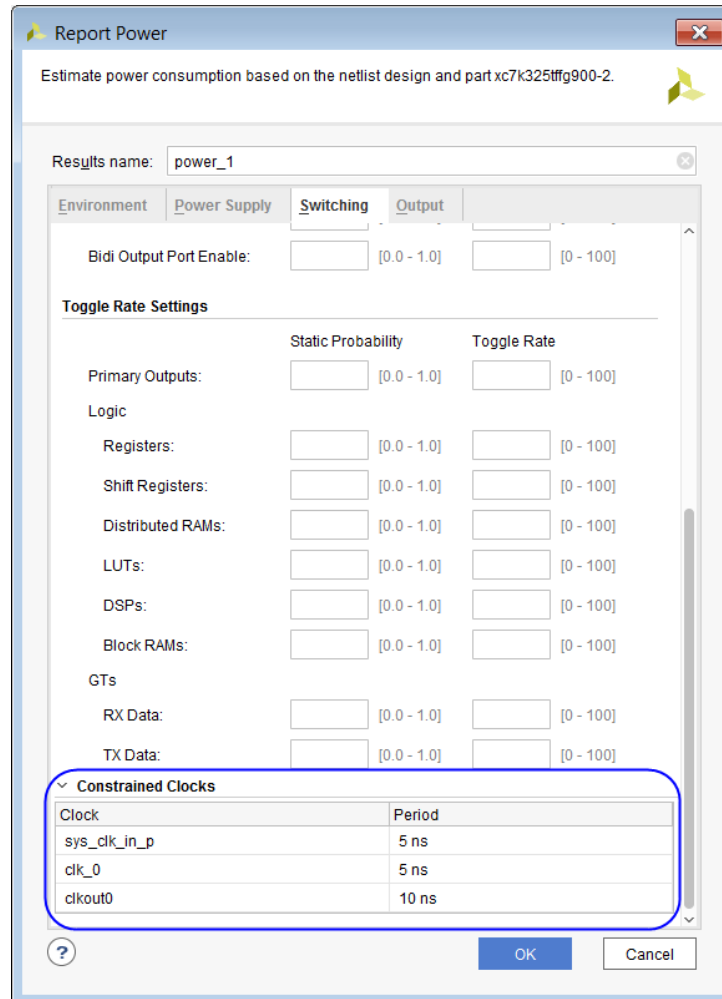
## クロック仕様

デザイン クロックは、ダイナミック消費電力計算における主なコンポーネントです。クロックが定義されていないと、スイッチング アクティビティの見積もりが不正確になり、正しい消費電力見積もりが得られません。クロック ノードは、XDC コマンドの `create_clock` または `create_generated_clock` を使用して定義されるタイミング制約から特定されます。

**注記:** デザインに必要なクロックはすべて `create_clock` または `create_generated_clock` コマンドを使用して定義する必要があります。

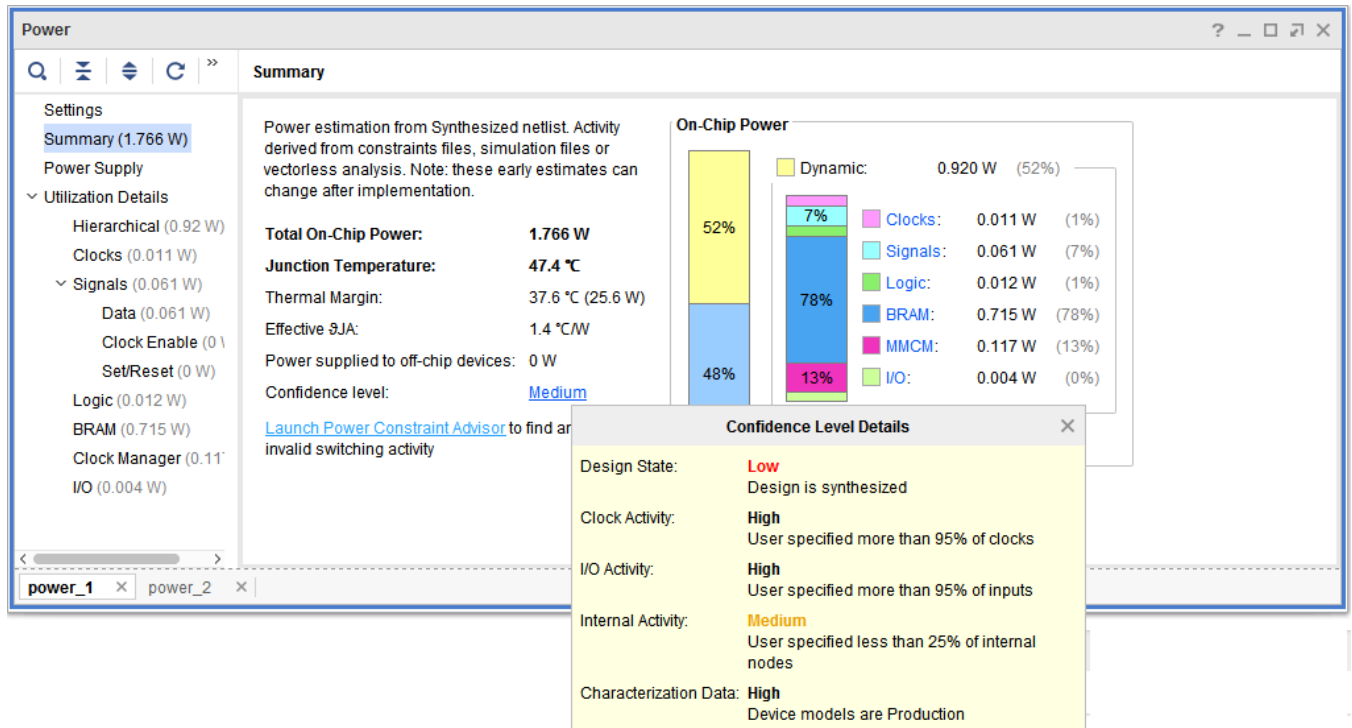
[Report Power] ダイアログ ボックスの [Switching] タブには、デザインで定義されているすべてのクロックが表示されます。

図 48: [Report Power] ダイアログ ボックスの制約が付いたクロック



デザインで定義されるクロックすべてが表示されていることを確認します。[Report Power] を実行し、[Summary] ページで [Confidence level] をクリックして詳細を表示すると、デザインで定義されているクロックの割合がわかります。この情報を使用して、クロック アクティビティの信頼性レベルが High になるようにします。

図 49: 信頼性レベル



Tcl モードでは、`get_clocks` および `report_clocks` コマンドを使用して定義されたクロックのリストを取得します。テキスト レポートには、クロック アクティビティの信頼性レベルが表示されます。

```
report_power -file power.rpt
```

図 50: テキスト レポート - クロック アクティビティの信頼性レベル

1.3 Confidence Level

User Input Data	Confidence	Details	Action
Design implementation state	Low	Design is synthesized	Accuracy of the tool is not optimal until design is fully placed and routed
Clock node activity	High	User specified more than 95% of clocks	
I/O node activity	High	User specified more than 95% of inputs	
Internal node activity	Medium	User specified less than 25% of internal nodes	Provide missing internal node activity with simulation results or by editing
Device models	High	Device models are Production	
Overall confidence level	Medium		

## 制御信号

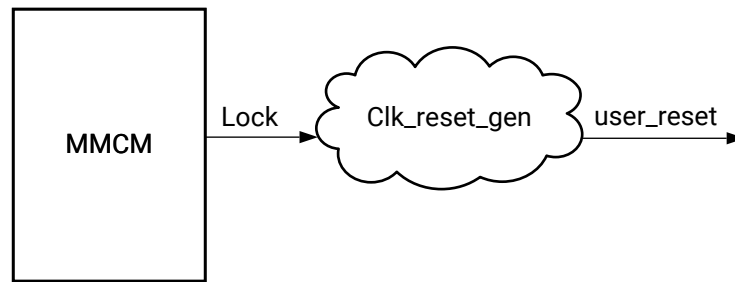
### グローバルおよびリージョナル リセット

グローバル リセットのアクティビティ レートによって、消費電力の見積もりが大幅に変わることがあります。アクティビティ レートは、デザインの各ロジック ブロックのステートと、ロジック出力が変わる確率を示します。このレートが正しいスイッチング情報で設定されていない場合、消費電力見積もりは非現実的な値になります。たとえば、リセットは run が開始してから数サイクル間だけアサート (アクティブ) され、その後は非アクティブになるのが理想的です。これは、スイッチング アクティビティを使用して次のように記述することもできます。

```
set_switching_activity -static_probability 0.01 -signal_rate 2 [get_ports
g1b_reset]
```

[Report Power] コマンドでグローバル リセットとなるプライマリ ポートが特定され、上記のスイッチング アクティビティが適用されます。グローバル リセットの特定には、保守的で安全な方法 (最下位プリミティブのリセット ピンに直接接続されたポート) が使用されます。ただしこれは、リセット ロジックが特別なロジック回路 (リセット ジェネレーター、デバウンサー、リセット ストレッチングなど) を使用して内部で生成されるような複雑なデザインではあまり役に立ちません。リセット生成に関係するロジックがある場合、[Report Power] コマンドで設計の意図が認識されず、デフォルトのスイッチング情報が適用されません。

図 51: リセット ロジック



X14354-012120

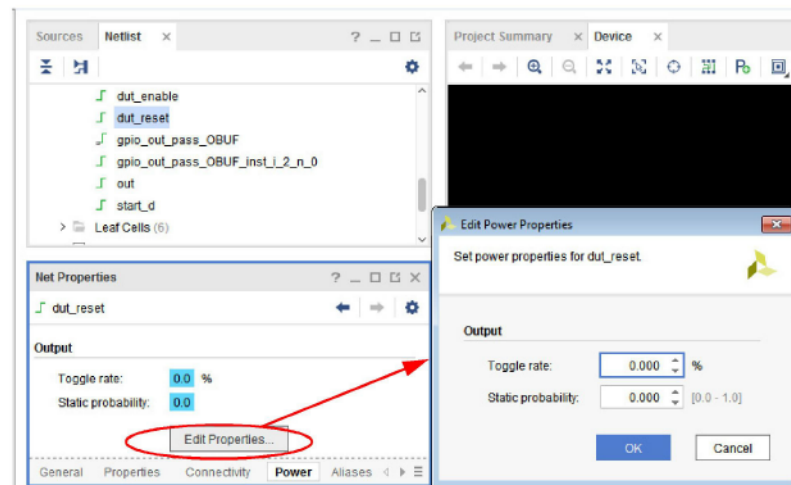
この場合、リセット アクティビティ 情報は確率計算と伝搬アルゴリズムを使用して生成されたロジックから算出されます。確率計算は、ロジックの最下位プリミティブ レベルで実行されます。確率アルゴリズムにより、特定のロジック ブロック (深くネスト化されたフィードバック ロジックなど) の処理にラグが発生することがあり、リセット ネットのスイッチング アクティビティが予期しない結果になることがあります。



**推奨:** グローバル/リージョナル リセット ネットに対して、正しいスイッチング情報を提供してください。

このようなグローバル リセット ネットに注意してください。これらのネットのアクティビティ レートは、[Net Properties] ウィンドウの [Power] ビューで直接設定します。

図 52: ネット アクティビティ レートの設定



同等 Tcl コマンド:

```
set_switching_activity -static_probability 0.01 -signal_rate 2 [get_nets u1/clkRst_gen/user_reset]
```

また、消費電力レポートを使用すると、デザインのリセット ネットを特定しやすくなるので、これらのネットのスイッチング情報を検証して、それを修正処置を取ることができます。[Report Power] コマンドの初回テスト run をデフォルト設定を使用して実行すると、リセット ネットのアクティビティを解析できます。

図 53: 消費電力レポートのリセット ネット

Utilization	Name	Signal Rate (Mtr/s)	% High	Fanout	Slice Fanout	Clock	Logic Type
0 W	dut/dut_reset	0.000	0.000	530	0	clkout0	FF LUT
0 W	led_OBUF	0.000	100.000	3	0	clkout0	FF I/O LUT

消費電力レポートには、このリセット ネットの影響を受けるロジック セルの数が [Fanout] 列に表示されます。最初のスイッチング アクティビティの見積もりが正しくない場合は、消費電力レポートでそのネットをクリックし、[Net Properties] ウィンドウの [Power] ビューでプロパティを変更します (上図参照)。

**注記:** レポートには、プリセット/セットおよびリセット ネットの両方がまとめて表示されます。上記のリセット ネットのガイドラインは、プリセット/セット ネットにも適用されます。

## グローバル クロック イネーブル

クロック イネーブルの扱いは通常リセットほど複雑ではなく、たいていのデザインではわかりやすく簡単です。ただし、リセット オン パワーが認識されるデザイン (クロック イネーブルが広範囲に使用され、特別ロジック回路を使用して制御される) のように複雑になることもあります。ロジック セルのダイナミック消費電力は、クロック イネーブルのスイッチング アクティビティによって変わります。アクティビティ レートが正しく設定されていないと、消費電力の正しい見積もり値は得られません。

たとえば、イネーブルは、実行中はアサートされ (アクティブ)、ロジック セルが使用されない場合にのみ非アクティブになるはずで (消費電力を削減するために明確に制御している場合)。これは、スイッチング アクティビティを使用して次のように記述できます。

```
set_switching_activity -static_probability 0.99 -signal_rate 2 [get_ports glb_enable]
```

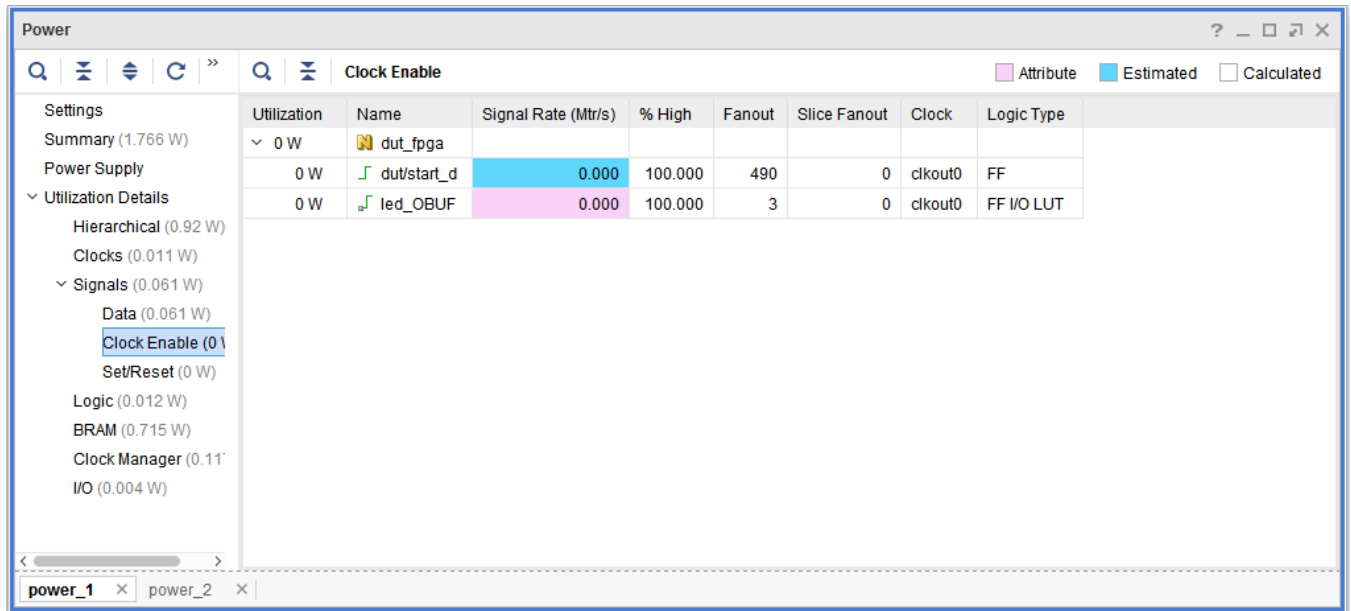
[Report Power] コマンドでグローバル イネーブルとなるプライマリ ポートが特定され、上記のスイッチング アクティビティが適用されます。グローバル イネーブルの特定には、保守的で安全な方法 (最下位プリミティブの CE ピンに直接接続されたポート) が使用されます。



**推奨:** グローバル/リージョナル イネーブル ネットに対して、正しいスイッチング情報を提供してください。

また、消費電力レポートを使用すると、デザインのイネーブル ネットを特定しやすくなるので、これらのネットのスイッチング情報をすばやく検証して、修正処置を取ることができます。[Report Power] コマンドの初回テスト run をデフォルト設定を使用して実行すると、イネーブル ネットのアクティビティを解析できます。

図 54: 消費電力レポートのクロック イネーブル ネット



Utilization	Name	Signal Rate (Mtr/s)	% High	Fanout	Slice Fanout	Clock	Logic Type
0 W	dut_fpga						
0 W	dut/start_d	0.000	100.000	490	0	clkout0	FF
0 W	led_OBUF	0.000	100.000	3	0	clkout0	FF I/O LUT

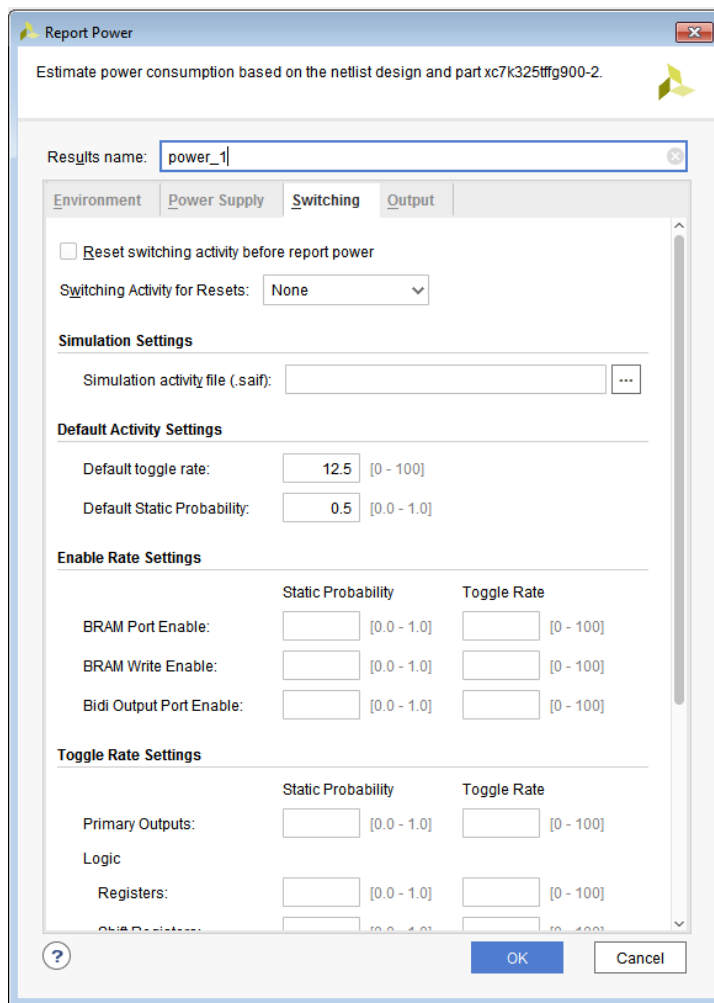
消費電力レポートには、このイネーブル ネットの影響を受けるロジックセルのタイプと数が [Logic Type] と [Fanout] 列に表示されます。最初のスイッチング アクティビティの見積もりが正しくない場合は、消費電力レポートでそのネットをクリックし、[Net Properties] ウィンドウの [Power] ビューでプロパティを変更します。

## プライマリ入力

共通ノードは、前述の推奨事項に従って処理されます。デザイン特有のハンドシェイク (プロトコル、メモリ インターフェイスなど) およびデータ ポートにも注意する必要があります。理想的には、プライマリ ポートのアクティビティ レートでデザインの全体的なアクティビティが決まります (このアクティビティ レートがダイナミック消費電力の精度に影響を与える)。消費電力レポートの生成では、プライマリ入力 (クロックおよび制御ポート以外) にデフォルトのスイッチング アクティビティ レート (Toggle\_rate=12.5 および Static\_Probability=0.5) が割り当てられます。つまり、ポートが 8 クロック サイクルで 1 回トグルし、50% の時間 High (ロジック 1) になるということです。この想定はデータ ポートではうまくいきますが、ハンドシェイク ノードに適用すると精度に大きく影響します。これで、プライマリ入力に正しいスイッチング情報を設定することがいかに重要かがわかります。デフォルトのアクティビティ設定は、[Report Power] ダイアログ ボックスの [Switching] タブに表示されます。



図 55: デフォルトのスイッチング アクティビティの設定



すべてのプライマリ入力(クロックでも制御信号でもない)に適用されるデフォルト値は変更できます。同等 Tcl コマンド:

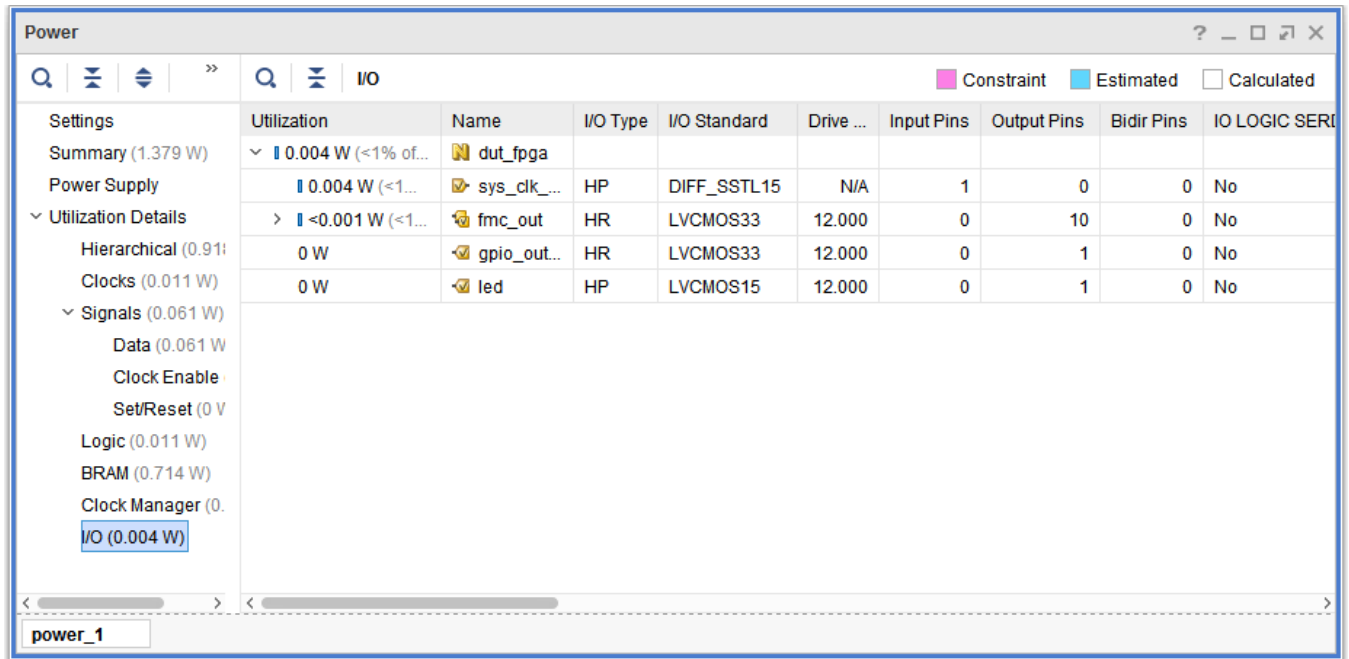
```
set_switching_activity -default_static_probability 0.5 -default_toggle_rate 25
```

すべてのプライマリ入力に同じアクティビティ レートが適用されます。データ ポートとハンドシェイク ポートは区別されないため、ハンドシェイク ポートに対してアクティビティ レートを手動で指定することが重要です。これは、Vivado® IDE または Tcl コマンドで実行できます。

**注記:** プライマリ I/O ポートには、正しいスイッチング値が設定されるようにしてください。

消費電力レポートの [I/O] ページには、すべてのポートとそのスイッチング アクティビティ 情報がリストされます。

図 56: 消費電力レポートの [I/O] ページ



Settings	Utilization	Name	I/O Type	I/O Standard	Drive ...	Input Pins	Output Pins	Bidir Pins	IO LOGIC SERI
Summary (1.379 W)	0.004 W (<1% of...)	dut_fpga							
Power Supply	0.004 W (<1% of...)	sys_clk_...	HP	DIFF_SSTL15	N/A	1	0	0	No
Utilization Details	<0.001 W (<1% of...)	fmc_out	HR	LVC MOS33	12.000	0	10	0	No
Hierarchical (0.91i)	0 W	gpio_out...	HR	LVC MOS33	12.000	0	1	0	No
Clocks (0.011 W)	0 W	led	HP	LVC MOS15	12.000	0	1	0	No
Signals (0.061 W)									
Data (0.061 W)									
Clock Enable									
Set/Reset (0 V)									
Logic (0.011 W)									
BRAM (0.714 W)									
Clock Manager (0.004 W)									

I/O ポートのアクティビティ レートを確認します。アクティビティ レートを変更するには、消費電力レポートで入力ポートを選択して、[I/O Port Properties] ウィンドウの [Power] ビューでプロパティを変更します。

同等の Tcl コマンド:

```
set_switching_activity -static_probability 0.25 -toggle_rate 10 [get_ports im_fcx_sync_in]

set_switching_activity -static_probability 0.5 -toggle_rate 50 [get_ports im_fcx_data_in]
```

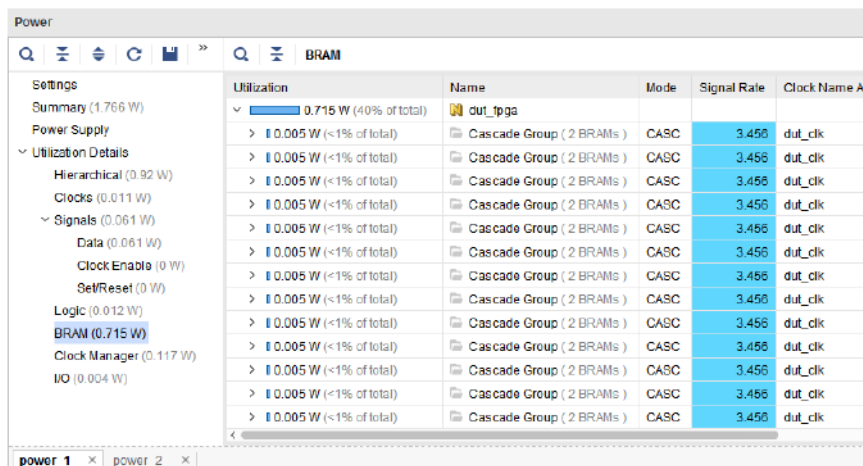
## コンポーネント レベル

最後に、消費電力が大きいプリミティブのアクティビティ レートを監視します。前述の注意点をすべて考慮して設定した場合、ブロック RAM、GT、DSP などのハード ブロックのアクティビティ レートは適切な値になっているはずですが、ザイリンクスではこれらの値をもう一度確認し、内部ロジック伝搬やツール内のモデリングに問題がないことを確認することをお勧めします。

たとえば、消費電力レポートではアクティビティ レートが GT に伝搬されないという制限があります。GT データ出力がロジックに接続されている場合は、アクティビティ レートを GT の TX/RX 出力に設定する必要があります。

[Report Power] ダイアログ ボックスを使用すると、レジスタ、シフト レジスタ、LUT、RAM、ブロック RAM、DSP、GT などのさまざまなタイプの出力アクティビティ レートを設定できます。これらは、set\_switching\_activity コマンドに -type オプションを使用しても設定できます。値を設定すると、その後の消費電力レポートの実行でその値が使用されます。グローバル設定は、デザインのハード プリミティブのインスタンスすべてに影響します。たとえば、ブロック RAM のトグル レート設定は、デザインのすべてのブロック RAM に適用されます。アクティビティ レートを変更するには、[Cell Properties] ウィンドウを使用することもできます。消費電力レポートで、ブロック RAM、DSP、および GT セクションを確認します。

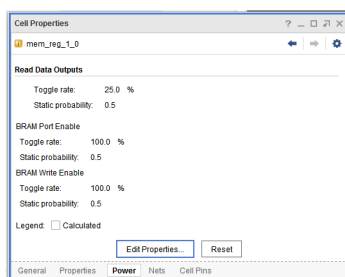
図 57: ブロック RAM のアクティビティ レート



Utilization	Name	Mode	Signal Rate	Clock Name A
0.715 W (40% of total)	out_tpga			
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk
> 0.005 W (<1% of total)	Cascade Group ( 2 BRAMs )	CASC	3.456	dut_clk

アクティビティ レートを変更するには、消費電力レポートでハード ブロック インスタンスを選択して、[Cell Properties] ウィンドウの [Power] ビューでプロパティを変更します。

図 58: ブロック RAM セルの [Cell Properties] ウィンドウの [Power] ビュー



タイプごとにアクティビティ レートを変更する Tcl コマンドは、次のとおりです。

- 特定のデザイン階層インスタンス (u1/transmit) の BRAM すべてにアクティビティ レートを設定するには、次のコマンドを使用します。

```
set_switching_activity -static_probability 0.25 -toggle_rate 10 -type
bram [get_cells u1/transmit]
```

- デザインのすべての GT にアクティビティ レートを設定するには、次のコマンドを使用します。

```
set_switching_activity -static_probability 0.5 -toggle_rate 50 -type gt -
all
```

# 消費電力削減のためのヒントおよび手法

## 概要

この章では、消費電力を削減する手法およびその手法による総消費電力への効果を説明します。この情報は、タイミング、消費電力要件、使用可能なリソース、およびデザインの変更許容度などに合わせて最適なオプションを評価する際に役立ちます。

## システム レベルの消費電力削減

### 冷却ストラテジ

冷却ストラテジは、デバイスで生成された熱が除去されて環境に吸収されるようにします。これらの冷却ストラテジは、デバイスのスタティック消費電力に大きく影響するのですが、通常、設計の初期段階で適用します。フロー後半になってからだと適用しづらくなります。

- エアフローを増やす。
- 周囲温度を下げる。
- ヒートシンク (またはさらに大きいヒートシンク) を使用するか、または別のレギュレータを選択する。

### 電源ストラテジ

電圧は、スタティック消費電力およびダイナミック消費電力の両方に大きく影響します。電圧レベルのアクティブ制御により、指定の電圧がデバイスに適用されます。

- スイッチング レギュレータを使用する。  
スイッチング レギュレータは、リニア レギュレータよりも電力効率に優れていますが、コンポーネント数は多くなります。
- 調整可能なレギュレータを使用する。  
同じ電源で複数のデバイスに電力を供給する場合、センス電圧を消費電力が最大のデバイスにできるだけ近くする。
- 許容誤差が小さいレギュレータを選択する。

## [Device Selection Dialog] ダイアログ ボックス

- 製品に適したデバイスを選択する。  
デバイスを選択する上で、消費電力が大きな決め手になってきています。集積度、機能、パフォーマンス要件が最適で、消費電力要件も満たすデバイスを選択してください。
- デバイス数を最小限に抑える。  
これにより、エリア、I/O インターコネクトの消費電力、全体的なリーク電流、などを抑えることができます。通常、プロセッサおよびデバイスなど、複数のコンポーネントをサイズの大きい 1 つのデバイスに置き換えると、スタティック消費電力を削減できます。
- 可能な限り小型のデバイスを選択する。  
これによりリーク電流を削減できます。通常、1 つのデバイスでは同じパッケージを異なるダイ サイズで使用できます。たとえば、プロトタイプ段階やリリース前には大きめのダイを使用し、量産段階では小さめのダイを使用することも可能です。
- 可能な限り大型のパッケージを選択する。  
これにより、放熱量が増加します。パッケージが大きいほど、ダイの熱を外に逃す面積が広がります。パッケージの上面に装着するヒートシンクを大きくすると、下面のボール グリッド アレイを介してプリント回路基板にさらに熱を逃すことができます。
- 低電圧デバイスを使用する。  
一部のデバイス ファミリには、低消費電力オプションがあります。電圧要件を下げると、スタティック消費電力およびダイナミック消費電力を大幅に削減できます。
- リーク電流が小さいデバイスを使用する。  
一部のデバイス ファミリでは、特定のスピード グレードまたは温度グレードを使用すると、低リーク電流または低スタティック消費電力オプションを利用できます。これらのデバイスの購入価格は多少高くなるかもしれませんが、電気代、冷却ハードウェア、およびシステム管理費も考慮すると、最終的なコストを抑えることができる可能性があります。

## 消費電力および温度の計測

このセクションでは、デバイス デバイスの消費電力と放熱量を計測するテクニックを説明します。これらのテクニックには、デバイスの内部リソースを使用するものや、ボード コンポーネントまたは外部コンポーネントを使用するものがあります。アプリケーションによっては、運用後に消費電力と温度を常に監視して調整する必要があるものがある場合、プロトタイプ作成中や検証段階でラボの測定値を使用するものもあります。

### 消費電力の計測方法

消費電力を計測するには、次の方法があります。

- [電流検出抵抗器を使用](#)
- [アドバンス レギュレータおよびデジタルパワー コントローラーを使用](#)
- [オンボード監視を実行](#)
- [個別の電源レールを使用](#)

## 電流検出抵抗器を使用

レギュレータ出力とデバイスの間に電流検出抵抗器を直列に挿入すると、小さな電圧降下が発生します。これは、オームの法則によると電流に比例します。この電圧を XADC/SYSMON で計測すると、デバイスに供給される電流がわかります。正確な計測値を得るために必要な接続については、該当するデバイス ファミリの XADC ユーザー ガイド (『UltraScale アークテクチャ システム モニター ユーザー ガイド』 (UG580: [英語版](#)、[日本語版](#))、『7 シリーズ FPGA および Zynq-7000 SoC XADC デュアル 12 ビット 1MSPS アナログ-デジタル コンバーター ユーザー ガイド』 (UG480: [英語版](#)、[日本語版](#))) を参照してください。電流検出抵抗器の使用の詳細は、『XADC (Xilinx Analog-to-Digital Converter) の駆動』 (XAPP795: [英語版](#)、[日本語版](#)) を参照してください。

## アドバンス レギュレータおよびデジタルパワー コントローラーを使用

最新の評価キットには、アドバンス レギュレータとデジタルパワー コントローラーが含まれており、これらを使用してレギュレータの出力電流と電圧をキャプチャし、その情報を USB インターフェイスを介して監視コンピューターに送信できます。電源レールを監視するには、これが最も単純で便利な方法です。ほとんどのザイリンクス開発ボードには、統合電源コントローラーが搭載されており、PC 上の GUI ソフトウェアで PMBus (I2C) to USB インターフェイス モジュールを使用してアクセスできます。

## オンボード監視を実行

ザイリンクス 7 シリーズ 以降のデバイス ファミリーには、電源電圧とデバイスの温度を計測するため、内部センサーと、少なくとも 1 つのアナログ/デジタル コンバーターが含まれています。Vivado® ハードウェア マネージャーを使用すると、リアルタイムで JTAG にアクセスでき、デバイスのコンフィギュレーションの実行前と実行後にさまざまな電源電圧やデバイスのジャンクション温度を計測できます。システム モニターまたは XADC コンポーネントをコードにインスタンス化して、デバイス アプリケーションからこれらを計測することもできます。

## 個別の電源レールを使用

可能であれば、各電源電圧に個別の電圧レールを使用してください。電圧レールがひとまとめに接続されている場合は、これらのレール間で電力を計測する際にそれを考慮します。

## 温度の計測方法

温度を計測するには、次の方法があります。

- [外部監視を実行](#)
- [オンボード監視を実行](#)

### 外部監視を実行

デバイス パッケージによりシリコンにアクセスできないので、ジャンクション温度を直接計測することはできません。ジャンクション温度は、パッケージ、ヒートシンク、およびその他の熱電対がある場所の温度を計測することにより見積もることができます。また、温度カメラを使用して、デバイスの温度および周辺のコンポーネントや環境との放熱関係を視覚化できます。

### オンボード監視を実行

温度は、消費電力計測と同じ方法を使用して計測できます。Vivado® ハードウェア マネージャーは、デバイス コンフィギュレーションの前後に使用できます。デバイス ジャンクション温度を読み出すには、デザインに System Monitor/XADC プリミティブを使用することもできます。

## 消費電力および温度の計測方法

デザインの総消費電力の 3 つの要素を評価するため、計測前にデバイスのジャンクション温度を制御して安定させる必要があります。これは、デバイスおよびデザインのスタティック消費電力がデバイスのジャンクション温度に大きく依存しているからです。デザインの総消費電力は、次の 3 つの要素で構成されます。

- デバイスのスタティック消費電力
- デザインのスタティック消費電力
- デザインのダイナミック消費電力

### デバイスのスタティック消費電力

空のデザインをダウンロードし、入力ノイズがキャプチャされず、すべての内部ロジックおよびコンフィギュレーション回路が既知のステートになっていることを確認します。

**注記:** 空のデザインには、トグルしないゲートまたはフリップフロップが 1 つ含まれており、すべての出力がトライステートになっています。

ジャンクション温度が安定したら、VCCINT、VCCAUX、およびその他の電源を計測します。特殊な装置、単純なヒートガン、または冷却スプレーを使用すると、温度を強制的に変更でき、デバイスのスタティック消費電力への環境による影響を評価できます。VCCADC は常に VCCAUX に接続しておく必要があります。

### デザインのスタティック消費電力

デバイスにデザインをダウンロードして、すべての入力および内部アクティビティ (入力データ、外部および内部クロック生成) を開始しないようにします。デバイスの温度が安定したら、電源レールの消費電力を計測します。これらの値からデバイスのスタティック消費電力を差し引くと、デザインで使用される特定のロジックリソースおよびコンフィギュレーションで消費されるスタティック消費電力 (デザインのスタティック消費電力) が求められます。

### デザインのダイナミック消費電力

デバイスにデザインをダウンロードして、デザインを表すクロックおよび入力スティミュラスを供給します。ジャンクション温度が安定したら、電源の消費電力を計測します。この消費電力は、デザインの瞬間的な総消費電力を表しており、各クロックサイクルのアクティビティによって変化します。

---

## デザイン レベルの消費電力削減

次のセクションでは、デザインの消費電力要件を満たすためにデザインに適用できる手法を示します。デザインサイクルには、主に次の 2 つの状況で消費電力クロージャが含まれます。

- 制約が満たされた後、さらにデザインを最適化する場合。  
または
- デザインが消費電力要件を超えている場合。



## 制約が満たされた後にさらにデザインを最適化

通常、開発プロセスのこの段階では、RTL、ボード電源、冷却パラメーターの変更は、検証に時間がかかったり、PCB リスピン コストがかかったりするため、最小限に抑える必要がありますが、この段階でも異なるツール オプション や制約を試して、ロジックおよび配線リソース数、コンフィギュレーション、およびアクティビティを最適化できます。この最適化により、ダイナミック消費電力が最小限に抑えられ、同時にスタティック消費電力も削減されます。デザイン マージンにもよりますが、ダイナミック消費電力を通常 15% ~ 20% 削減でき、一部のデザインではそれ以上削減できます。

## 消費電力要件を超えている場合

通常この段階では、システムを市場にリリースするプレッシャーが大きく、ボード環境および冷却オプションなどのシステムに含まれる多くのパラメーターが詳細に定義されています。そのためエンジニアリング作業のやり直しは制限されますが、次の手法を使用すると、消費電力を削減できる可能性が高いエリアを特定できます。

### 手順 1: 消費電力要件を超えている箇所の特定

GUI を使用している場合は Vivado® 消費電力解析レポートの [Summary] ページ、コマンド ラインを使用している場合はレポート ファイルの [Summary] セクションを確認します。[On-Chip Power] および [Power Supply] セクションで、消費電力分配の概要を確認できます。[Summary] ページで要件を超えている消費電力の種類および電力量を確認します。

### 手順 2: 焦点を置くエリアの特定

Vivado® 消費電力解析レポートまたは Xilinx® Power Estimator で詳細を確認します。環境パラメーターと、各リソース、デザイン階層、クロック ドメインで消費される電力を解析します。消費電力が高いエリアを見つけた場合は、次の情報を使用してその原因を特定できます。

### 手順 3: 試行

上記の手順で特定した消費電力を最適化するデザイン箇所の候補リストを確認し、簡単なものから順に並べ替えて、実行する最適化または試みを決定します。消費電力ツールを使用すると what-if 解析を実行できるので、コードや制約を実際に変更したりインプリメンテーションを再実行したりせずに、デザインの変更を入力して簡単に消費電力を見積もることができます。

### リソースをより効果的に使用

- ブロック RAM:
  - ブロック RAM の消費電力量は、イネーブルになっている時間に直接比例します。消費電力を削減するには、デザインでブロック RAM が使用されていないクロック サイクルで RAM イネーブル信号を Low にします。ブロック RAM のイネーブル レートとクロック レートは、消費電力を最適化する際に最も重要なパラメーターです。
  - TDP モードで書き込み中に出力ラッチが変更されない場合は、NO\_CHANGE モードを使用します。これが電力効率が高いモードです。SDP モードでは、NO\_CHANGE モードは WRITE\_FIRST モードと同じなので、NO\_CHANGE モードはありません。
- I/O: I/O インターフェイスは長距離を駆動する必要があり、寄生効果の影響が大きくなる可能性があるため、通常デバイスの消費電力要件の大部分を占めます。
- VCCAUX: できるだけ低い VCCAUX を使用します。これにより、この電源のスタティック消費電力およびダイナミック消費電力の両方が最小限に抑えられます。

- 入力: 内部で参照される入力規格の使用を制限します。
  - IODELAY: IDELAY2 の HIGH\_PERFORMANCE\_MODE プロパティを FALSE に設定します。FALSE に設定すると、出力ジッターは増加しますが、消費電力は小さくなります。
  - IBUF\_LOW\_PWR: 双方向 I/O および入力 I/O の IBUF\_LOW\_PWR プロパティを TRUE に設定します。デザインパフォーマンスでこの設定が許容されることを確認してください。
  - I/O コンフィギュレーション: パフォーマンス要件に対して I/O 規格、駆動電流、およびオンチップ終端設定を確認し、トライステートが可能な DCI I/O 規格 (T\_DCI) を使用して駆動電流を下げたり、終端を省いたり、外部終端を使用したりできないかを評価します。
  - 出力:
    - 受信チップでサポートされる最小のスルー/駆動電流/電圧を使用します。
    - 並列終端よりも終端なしまたは直列終端を選択します。この決定には、シグナル インテグリティ シミュレーション ツールを使用できます。
    - デバイスの温度要件、システム コスト、およびボードのスペース 要件を考慮して、オンチップ終端とオフチップ終端のどちらが適しているかを検討します。
    - 電圧幅の低い差動規格を使用できないかを検討します。
    - アプリケーションで大型パラレル バスの代わりにトランシーバーを使用できないかを検討します。
    - IBUF、IODELAY などの I/O 機能の要件を評価し、許容される場合はディスエーブルにします。
  - トランシーバー:
    - GTX/GTH/GTP トランシーバーでは、消費電力を削減できる可能性のあるパワーダウン モードがサポートされています。
    - GTX/GTH レシーバーには、システム レベルでの消費電力とパフォーマンスのトレードオフに応じて、2 種類の適応フィルタがあります。GTX/GTH/GTP レシーバーには、チャネル損失が低く消費電力で最適化された、電力効率の高い LPM (低消費電力モード) という適応モードがあります。
    - 各 GTX/GTH/GTP トランシーバーでは、SATA (Serial ATA) および SAS (Serial Attach SCSI) 仕様に記述されている OOB (Out Of Band) シーケンスの生成と、PCI™ Express 仕様に記述されているビーコンがサポートされています。OOB シーケンスを使用しない場合、さらに消費電力を削減できます。
    - トランシーバーを可能な数だけ 1 つのタイルにパックして、サポート回路の複製を最小限に抑えます。
  - XADC:
    - XADC は、ランタイム中に DRP ポートからそのコンフィギュレーション レジスタ #2 (アドレス 0x42) に書き込むことでパワーダウンできます。各チャネルのパワーダウンは、このレジスタのビット DI4 と DI5 で制御されます。Vivado® でパワーダウン動作をスタティックにエミュレートするには、Vivado の Tcl コンソールに次のコマンドを入力し、コンフィギュレーション レジスタを設定します。
- ```
set_property INIT_42 {16'h0430} [get_cells <inst>]
```
- <inst> は XADC インスタンスです。上記のコマンドは、XADC の両方のチャネルをパワーダウンします。
- ロジック:
 

次の方法でデザイン記述を最適化できます。

    - ロジックの最適化の妨げになり、配置配線リソースを多く使用する非同期の制御信号を最小限に抑えます。

- 制御セット数を最小限に抑えます。制御セットは、クロック、クロック イネーブル、セット、リセット、ライト イネーブル (LUT RAM の場合) 信号の固有グループです。1 つのスライス内での信号数の制限および信号の共有のため、制御セットの情報は重要です。これはデバイス アーキテクチャによって異なりますが、制限に達すると、近接した関連ロジックをパックできず、配線リソースが増加する場合があります。
- パイプライン段を追加して、組み合わせロジック コーンのサイズを最小限に抑えます。これにより、各クロック サイクルで信号が最終ステートに到達するまで、レジスタ間のグリッチの伝搬を最小限に抑えることができます。
- リソースのタイム シェアリングを使用します。この手法では、同じハードウェア リソースに異なるファンクションを時分割多重化することで、デバイスのリソース使用量を最小限に抑えます。これにより、小さいデバイスを使用できるようになったり、配置配線の密集が緩和され、スタティック消費電力およびダイナミック消費電力を削減できます。
- 低速で類似しているプロセスは、別のリソースを使用せずに同じリソースで実行できます。ただし、処理するデータのバッファ処理、マルチプレクサー処理、初期化、および制御方法を慎重に検討する必要があります。複数の入力センサーを処理するなど、並列処理の実行されるアプリケーションでこのような最適化を実行します。多数の処理ユニットを入力として使用する代わりに、1 つの処理ユニットを高速に動作させ、入力チャネルを順次処理しながら各出力の応答時間は同じになるようにできます。Xilinx® Power Estimator の what-if 見積もりを実行すると、消費電力の削減が作業努力に値するかを判断できます。
- DSP およびブロック RAM のオプションのレジスタを使用します。たとえば、DSP ブロックで乗算器または MREG レジスタをイネーブルにすると、クロック サイクル間の内部グリッチの伝搬が最小限に抑えられ、最も電力効率の高いインプリメンテーションになります。

## Vivado 消費電力最適化機能での試行

Vivado® ツールで消費電力最適化を実行したときに消費電力が最大限に削減されるようにするには、消費電力最適化をデザイン全体に実行し、デザインの一部を除外しないようにします。消費電力最適化を有効にしても消費電力が削減されない場合は、デバッグが必要である可能性がある部分はグローバル セット/リセット信号、ブロック RAM イネーブル生成、およびレジスタ クロック ゲーティングの 3 つです。これらの部分で消費電力最適化に生成されたイネーブルの数が少ない場合、コードまたは合成およびインプリメンテーションのオプション/プロパティを確認する必要があります。

**注記:** Vivado 消費電力最適化では、消費電力を最大限に削減しながらタイミングへの影響は最小限に抑えますが、場合によってはタイミングが悪化することがあります。この影響を補正する方法は、[消費電力最適化後のタイミングの保持](#)を参照してください。

- グローバル セット/リセット信号:

可能な限り (特にデータパス、パイプライン フリップフロップ、ブロック RAM に対して)、非同期セット/リセット信号の使用を最小限に抑えます。また、power\_opt\_design でグローバル セット/リセット信号に dont\_touch プロパティを設定し、イネーブルとして使用されないようにすることも考慮します。HDL で dont\_touch プロパティを設定すると、フローのすべての段階でこの属性が適用されます。このオプションは、消費電力最適化用の XDC 制約として設定することをお勧めします。次にその例を示します。

```
set_property DONT-TOUCH true [get_cells u1]
```

最後に、消費電力最適化およびベクターレス消費電力見積もりの前に、グローバル セット/リセット信号の信号レートおよび確率が正しく設定されていることを確認してください。

- スライス レジスタおよび SRL: power\_opt\_design でデザインのスライス レジスタまたは SRL に対してクロック イネーブルが生成されない原因は、多数あります。次に、その例を示します。
  - デザインに組み合わせループが存在する
  - デザインへのプライマリ入力ソースとなるフリップフロップおよび SRL にセット/リセット信号が使用されている

- データパス フリップフロップに非同期セット/リセット信号が使用されている
- デザインに多数のクロック ドメインがあり、クロック ドメインをまたがっているためにイネーブルが生成されない
- SRL のサイズ: SRL のシフト レジスタの段数が多いほど、すべての段に対して 1 つのクロック イネーブルを生成するのが困難になります。
- ブロック RAM: ブロック BRAM が多数使用されるデザインでは、消費電力を大きく削減できる可能性があります。Vivado では、さまざまな最適化手法を使用してイネーブルを生成し、消費電力を削減します。  
`power_opt_design` を使用してもブロック RAM ゲーティングがそれほど適用されない場合は、次のような原因が考えられます。
  - BRAM が主に FIFO18/FIFO36 セルである。これらのインスタンスはツールで最適化できません。
  - 推論またはインスタンス化されたメモリが、A ポートと B ポートに別のクロックを使用した完全なデュアル ポート (TDP) モードであり、`power_opt_design` で最適化できない。
  - ブロック RAM 自体、またはブロック RAM に供給されるアドレス/ライト イネーブル フリップフロップに非同期リセットが使用されている。

## Vivado 消費電力解析機能での試行

Vivado® の [Report Power] ダイアログ ボックスで設定を変更して解析を実行し、消費電力への影響を確認できます。

- 環境: 温度パラメーター、プロセス、または電圧を設定します。
- デザインのアクティビティ: デザインに含まれるネットまたはセルのアクティビティを 調整します。1 つまたは複数のアイテムを同時に変更します。次も変更できます。
  - クロック ドメイン: スイッチング周波数を調整します。
  - デバイス ロジック: ダイナミック アクティビティ レートを調整します。
  - I/O: スタティック アクティビティおよびダイナミック アクティビティの 確率を調整します。負荷容量や近端ボード終端など、デバイス出力に接続されている外部コンポーネントのパラメーターも調整できます。
  - 信号: データ信号のダイナミック アクティビティ レートを調整します。制御信号のスタティック確率を変更すると、クロック イネーブル、セット、またはリセットの異なる条件下での消費電力を評価できます。
  - 特定のブロック: ダイナミック アクティビティ 確率に加え、ブロック RAM のポート イネーブルまたはライト イネーブルなどの制御信号のアクティビティも 調整できます。

## Xilinx Power Estimator (XPE) での試行

XPE では、複数のソースで開発されたモジュールからの Vivado® 消費電力解析結果をインポートし、これらの IP ブロックをデバイスにインプリメントしたときの総消費電力を確認できます。また、ネットリストを変更しなくてもはいけない状況の評価し、実際にコードを変更せずに消費電力への影響を評価できます。XPE では各ロジック エLEMENT または信号を個別に変更することはできないので、デザインのコア ロジックでの精度は Vivado 消費電力解析よりも低くなります。XPE では、次を試すこともできます。

- リソース使用量: リソース数を減らしてみます。ロジックの一部をスライス ロジックからブロック RAM や DSP などの専用ブロックにマップし直したり、またその逆を実行してみます。
- リソース コンフィギュレーション: デザインの I/O、ブロック RAM、クロック ジェネレーター、およびその他のリソースに別のコンフィギュレーション設定を使用してみます。

## RTL コードの変更

消費電力を削減するために RTL コードを変更する必要がある場合は、パイプラインを追加したり、キャリー チェーンや XOR ファンクションなどのアクティビティの 高いロジックでリタイミングを実行してみたりできます。キャリー チェーンを含む長いパスは、低速のクロック ドメインにあることが多いですが、グリッチが増え、デザインの消費電力が増加する原因となります。多くの場合、これらのパスをリタイミングまたはパイプライン処理すると有益です。

## 手順 4: 変更を反映し、消費電力を確認

時間、パフォーマンス、およびリソース制約での最適な変更を決定したら、これらを反映させます。一度に試すオプションや変更が多すぎると、競合や相互作用が発生する可能性があるため、結果が最適にならない可能性があります。時間がある場合は一度に試すオプションを限定して、ほかの変更を加える前に消費電力およびその他の制約への影響を評価する方法が最適です。

# その他のリソースおよび法的通知

## ザイリンクス リソース

アンサー、資料、ダウンロード、フォーラムなどのサポート リソースは、[ザイリンクス サポート](#) サイトを参照してください。

## Documentation Navigator およびデザイン ハブ

ザイリンクス Documentation Navigator (DocNav) では、ザイリンクスの資料、ビデオ、サポート リソースにアクセスでき、特定の情報を取得するためにフィルター機能や検索機能を利用できます。DocNav を開くには、次のいずれかを実行します。

- Vivado<sup>®</sup> IDE で [Help] → [Documentation and Tutorials] をクリックします。
- Windows で [スタート] → [すべてのプログラム] → [Xilinx Design Tools] → [DocNav] をクリックします。
- Linux コマンド プロンプトに「docnav」と入力します。

ザイリンクス デザイン ハブには、資料やビデオへのリンクがデザイン タスクおよびトピックごとにまとめられており、これらを参照することでキー コンセプトを学び、よくある質問 (FAQ) を参考に問題を解決できます。デザイン ハブにアクセスするには、次のいずれかを実行します。

- DocNav で [Design Hub View] タブをクリックします。
- ザイリンクス ウェブサイトで[デザイン ハブ](#) ページを参照します。

**注記:** DocNav の詳細は、ザイリンクス ウェブサイトの [Documentation Navigator](#) ページを参照してください。DocNav からは、日本語版は参照できません。ウェブサイトのデザイン ハブ ページをご利用ください。

## 参考資料

このガイドの補足情報は、次の資料を参照してください。

1. 『Vivado Design Suite ユーザー ガイド: リリース ノート、インストール、およびライセンス』 (UG973: [英語版](#)、[日本語版](#))
2. 『Vivado Design Suite Tcl コマンド リファレンス ガイド』 (UG835: [英語版](#)、[日本語版](#))
3. 『Vivado Design Suite ユーザー ガイド: 制約の使用』 (UG903: [英語版](#)、[日本語版](#))



4. 『Xilinx Power Estimator ユーザー ガイド』 (UG440: [英語版](#)、[日本語版](#))
5. 『Vivado Design Suite チュートリアル: 消費電力解析および最適化』 ([UG997](#))
6. 『Vivado Design Suite ユーザー ガイド: ロジック シミュレーション』 (UG900: [英語版](#)、[日本語版](#))
7. 『7 シリーズ FPGA パッケージおよびピン配置: 製品仕様』 (UG475: [英語版](#)、[英語版](#))
8. 『UltraScale および UltraScale+ FPGA パッケージおよびピン配置ユーザー ガイド』 (UG575: [英語版](#)、[日本語版](#))
9. 『7 シリーズ FPGA および Zynq-7000 SoC XADC デュアル 12 ビット 1MSPS アナログ-デジタル コンバーター ユーザー ガイド』 (UG480: [英語版](#)、[日本語版](#))
10. 『XADC (Xilinx Analog-to-Digital Converter) の駆動』 (XAPP795: [英語版](#)、[日本語版](#))
11. [Vivado Design Suite の資料](#)

## トレーニング リソース

ザイリンクスでは、この資料に含まれるコンセプトを説明するさまざまなトレーニング コースおよび QuickTake ビデオを提供しています。次のリンクから関連するトレーニング リソースを参照してください。

- [Vivado Design Suite QuickTake ビデオ: Vivado での消費電力の見積もりと解析](#)
- [Vivado Design Suite QuickTake ビデオ: Vivado を使用した消費電力の最適化](#)
- [Vivado Design Suite QuickTake ビデオ チュートリアル](#)

## お読みください: 重要な法的通知

本通知に基づいて貴殿または貴社 (本通知の被通知者が個人の場合には「貴殿」、法人その他の団体の場合には「貴社」。以下同じ) に開示される情報 (以下「本情報」といいます) は、ザイリンクスの製品を選択および使用することのためにのみ提供されます。適用される法律が許容する最大限の範囲で、(1) 本情報は「現状有姿」、およびすべて受領者の責任で (with all faults) という状態で提供され、ザイリンクスは、本通知をもって、明示、黙示、法定を問わず (商品性、非侵害、特定目的適合性の保証を含みますがこれらに限られません)、すべての保証および条件を負わない (否認する) ものとし、また、(2) ザイリンクスは、本情報 (貴殿または貴社による本情報の使用を含む) に関係し、起因し、関連する、いかなる種類・性質の損失または損害についても、責任を負わない (契約上、不法行為上 (過失の場合を含む)、その他のいかなる責任の法理によるかを問わない) ものとし、当該損失または損害には、直接、間接、特別、付随的、結果的な損失または損害 (第三者が起こした行為の結果被った、データ、利益、業務上の信用の損失、その他あらゆる種類の損失や損害を含みます) が含まれるものとし、それは、たとえば当該損害や損失が合理的に予見可能であったり、ザイリンクスがそれらの可能性について助言を受けていた場合であったとしても同様です。ザイリンクスは、本情報に含まれるいかなる誤りも訂正する義務を負わず、本情報または製品仕様のアップデートを貴殿または貴社に知らせる義務も負いません。事前の書面による同意のない限り、貴殿または貴社は本情報を再生産、変更、頒布、または公に展示してはなりません。一定の製品は、ザイリンクスの限定的保証の諸条件に従うこととなるので、<https://japan.xilinx.com/legal.htm#tos> で見られるザイリンクスの販売条件を参照してください。IP コアは、ザイリンクスが貴殿または貴社に付与したライセンスに含まれる保証と補助的条件に従うことになります。ザイリンクスの製品は、フェイルセーフとして、または、フェイルセーフの動作を要求するアプリケーションに使用するために、設計されたり意図されたりしていません。そのような重大なアプリケーションにザイリンクスの製品を使用する場合のリスクと責任は、貴殿または貴社が単独で負うものです。<https://japan.xilinx.com/legal.htm#tos> で見られるザイリンクスの販売条件を参照してください。



### 自動車用のアプリケーションの免責条項

オートモーティブ製品 (製品番号に「XA」が含まれる) は、ISO 26262 自動車用機能安全規格に従った安全コンセプトまたは余剰性の機能 (「セーフティ 設計」) がない限り、エアバッグの展開における使用または車両の制御に影響するアプリケーション (「セーフティ アプリケーション」) における使用は保証されていません。顧客は、製品を組み込むすべてのシステムについて、その使用前または提供前に安全を目的として十分なテストを行うものとし、セーフティ設計なしにセーフティ アプリケーションで製品を使用するリスクはすべて顧客が負い、製品責任の制限を規定する適用法令および規則にのみ従うものとし、

### 商標

© Copyright 2012-2020 Xilinx, Inc. Xilinx、Xilinx のロゴ、Alveo、Artix、Kintex、Spartan、Versal、Virtex、Vivado、Zynq、およびこの文書に含まれるその他の指定されたブランドは、米国およびその他の国のザイリンクス社の商標です。すべてのその他の商標は、それぞれの所有者に帰属します。

この資料に関するフィードバックおよびリンクなどの問題につきましては、[jpn\\_trans\\_feedback@xilinx.com](mailto:jpn_trans_feedback@xilinx.com) まで、または各ページの右下にある [フィードバック送信] ボタンをクリックすると表示されるフォームからお知らせください。フィードバックは日本語で入力可能です。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。