

Vivado Design Suite ユーザー ガイド

デザイン解析および クロージャ テクニク

UG906 (v2012.2) 2012 年 8 月 20 日





Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.

©Copyright 2012 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

本資料は英語版 (v2012.2) 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。

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 までお知らせください。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。

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	改訂内容
2012 年 8 月 20 日	2012.2	初版

目次

第 1 章 : デザイン解析	5
レポートおよびメッセージ	5
レポート	7
タイミング サマリ レポート	21
クロック ネットワーク レポート	33
クロックの相互関係レポート	34
パルス幅レポート	35
タイミング レポート	36
第 2 章 : ロジック解析	41
RTL 解析	41
[Netlist] ビュー	41
[Hierarchy] ビュー	43
回路図の表示	44
第 3 章 : インプリメンテーション結果の解析	57
[Design Runs] ビュー	57
その他の解析	59
配線の表示	67
表示オプション	69
[Device] ビューのナビゲーション	69
第 4 章 : タイミング解析	71
タイミングの最終確認	71
タイミング パス レポートの解釈	73
第 5 章 : クロージャ テクニック	80
入力制約およびソースのチェック	80
ツールのエフォートレベルの増加	81
手動フロアプラン	82
配線の変更	92
ロジックの変更	93
変更の使用	94

付録 A : その他のリソース	95
ザイリンクス リソース	95
ソリューション センター	95
リファレンス.....	95

デザイン解析

『Vivado Design Suite ユーザー ガイド：デザイン解析およびクロージャ テクニック』(UG906) では、次の内容を説明します。

- Vivado™ 統合設計環境 (IDE) の理解 (次の情報を含む)
 - メッセージおよびレポート
 - デザイン ルール チェック (DRC) 構造
- チップ設計の理解
- タイミング解析
- 配置配線結果の解析
- タイミング要件が満たされない場合に使用する手法

Vivado IDE の使用法およびビューの管理については、『Vivado Design Suite ユーザー ガイド：Vivado IDE の使用』(UG893) を参照してください。

レポートおよびメッセージ

Vivado IDE では、さまざまな処理の実行中に発生したイベントに関する情報がレポートおよびメッセージに表示されます。ユーザーが実行したほとんどの操作に対して、メッセージが表示されます。メッセージを確認し、デザインに問題が発生している箇所がないかどうかを判断します。

メッセージおよびレポートは、Vivado IDE の下部の [Messages] ビューおよび [Reports] ビューに表示されます。

[Messages] ビューでは、メッセージが生成された操作ごとにメッセージがグループ化されます。このビューのツールバーを使用して、メッセージ ID またはファイルでメッセージを分類できます。

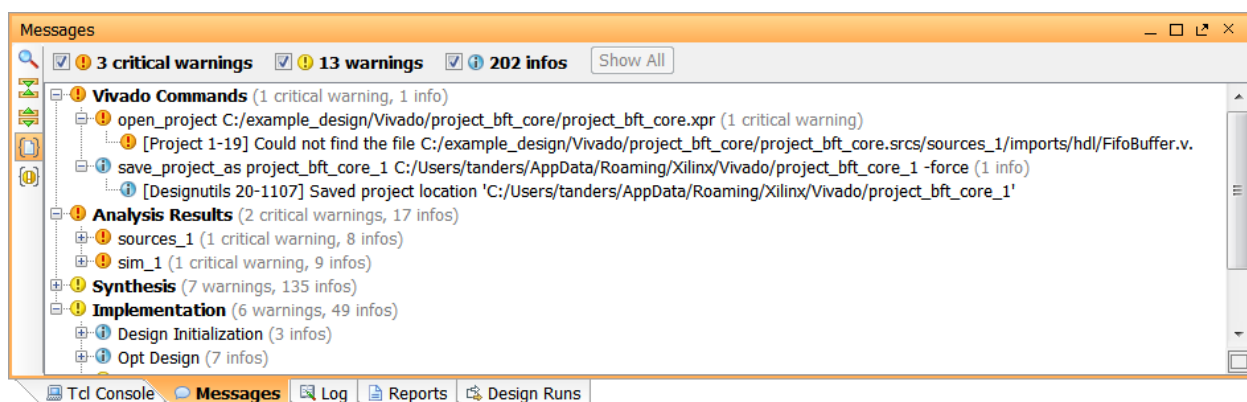






図 1 : [Messages] ビュー

一部のメッセージにはファイルまたはデザイン エレメントへのハイパーリンクが含まれており、デバッグに役立ちます。ファイル名または行番号をクリックし、ソースを表示します。ポップアップ メニューから [Copy Message] をクリックすると、メッセージ全体をコピーできます。

メッセージの重要度

表 1：メッセージの重要度

アイコン	重要度	メッセージ
	情報	デザイン プロセスに関する一般的なステータスおよびフィードバックを示します。
	警告	制約または仕様が意図どおりに適用されていないために、最適なデザイン結果にならない可能性を示します。
	クリティカル 警告	ユーザー入力または制約に適用されなかったり、適切な方法に従っていないものがあり、フローの後方でエラーが発生する可能性が高いことを示します。ソースおよび制約を確認して、変更を加えることを強くお勧めします。
	エラー	デザイン結果が使用できない状態になり、ユーザーが対処しないと回避されない問題を示します。デザインフローは停止します。

メッセージのフィルター機能

メッセージのフィルター機能を使用して、各操作で重要度の高いメッセージのみを表示できます。メッセージを重要度でフィルターするには、メッセージの重要度の横にあるチェック ボックスをクリックします。デザインの読み込み中にクリティカル メッセージまたはエラー メッセージがある場合、[Messages] ビューがアクティブになり、ほかのメッセージは非表示になります。

メッセージ タイプ

次の 2 種類のメッセージがあります。

- ディスクに保存されるメッセージ
- メモリに保存されるメッセージ

新しいプロセス

次のいずれかのコマンドを実行すると、新しいプロセスが開始します。

- 合成
- インプリメンテーション
- launch_runs (Tcl)

プロセスによりメッセージおよびレポートが生成され、run をリセットするまでディスクに保持されます。プロジェクトが開いている場合、run に関連するメッセージが表示されます。[Messages] ビューには、アクティブ run に関するメッセージのみが表示されます。

デザインの読み込み中に表示されるメッセージ

次のデザインを読み込むときに表示されるメッセージは、[Messages] ビューの該当するデザイン フォルダーの下に表示されます。

- エラボレート済みデザイン
- 合成済みデザイン
- インプリメント済みデザイン

フォルダーは、デザインがメモリに読み込まれたときに表示されます。

プロジェクトを開いたときに表示されるメッセージ

Vivado IDE GUI または Tcl コンソールでプロジェクトを開いたときに、メッセージが表示されることがあります。これらのメッセージは、デザインが開いている間のみ保持されます。

メッセージ ID 番号

メッセージには、固有の ID 番号が付けられています。この ID を使用して、メッセージに関する情報を検索できます。

メッセージの重要度の変更

ツールのデフォルトのメッセージ優先度が、そのデザインでのメッセージの優先度と異なる場合があります。メッセージの重要度を変更するには、`set_msg_severity` コマンドを使用します。次に例を示します。

```
set_msg_severity "Common 17-81" "CRITICAL WARNING"
```

レポート

次のレポートがあります。

- [生成されるレポート](#)
- [Vivado 合成レポート](#)
- [制御セット レポート](#)
- [DRC レポート](#)
- [配線ステータス レポート](#)
- [タイミング サマリ レポート](#)
- [WebTalk レポート](#)

レポートが生成される操作

レポートは、Vivado IDE でのさまざまな操作により生成されます。

- デザインを読み込むと、[Tools] メニューのコマンドを使用して異なるレポートを生成できるようになります。
- 合成またはインプリメンテーションを実行すると、合成レポートまたはインプリメンテーション レポートが生成されます。
- その他のレポートは、デザインをメモリに読み込んでから生成します。

[Reports] ビューには、アクティブな合成またはインプリメンテーション run に関するレポートが表示されます。レポートをダブルクリックすると、テキスト エディターで開きます。[Run Properties] ビューの [Reports] タブから、[Design Runs] ビューで選択している run のレポートを表示できます。

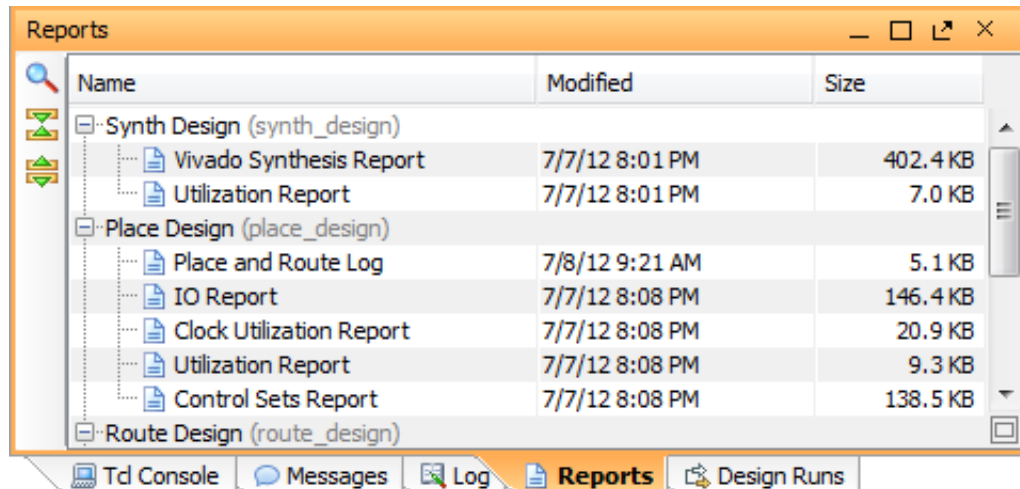


図 2 : [Reports] ビュー

生成されるレポート

次のレポートが生成されます。

- 合成中に生成されるレポート
- 配置中に生成されるレポート
- 配線中に生成されるレポート
- ビットストリーム生成中に生成されるレポート

合成中に生成されるレポート

合成中には、次のレポートが生成されます。

- Vivado 合成レポート
- リソース使用率レポート (合成後)

配置中に生成されるレポート

配置中には、次のレポートが生成されます。

- 配置配線ログ
- I/O レポート
- クロック使用量レポート
- リソース使用率レポート (配置後)
- 制御セット レポート

配線中に生成されるレポート

配線中には、次のレポートが生成されます。

- [WebTalk レポート](#)
- [DRC レポート](#)
- [配線ステータス レポート](#)
- [タイミング サマリ レポート](#)

ビットストリーム生成中に生成されるレポート

ビットストリーム生成の出力は、ログ ファイルに記述されます。ビットストリーム生成中に生成されるレポートは、[WebTalk レポート](#)のみです。

Vivado 合成レポート

Vivado 合成レポートは Vivado 合成の主な出力で、次の情報を含みます。

- 処理されたファイル
 - VHDL
 - Verilog
 - SystemVerilog
 - XDC
- セルごとのパラメーター設定
- ドライバーが複数あるネット
- 駆動されていない階層ピン
- 最適化情報
- ブラック ボックス
- 最終的なプリミティブ数
- 階層ごとのセルの使用量
- ランタイムおよびメモリ使用量

このレポートまたは [Messages] ビューでエラー、クリティカル警告、および警告を確認してください。合成では、フローの後の方で深刻となる可能性のある問題に対して、クリティカル警告または警告が生成されることがあります。

リソース使用率レポート

リソース使用率レポートは、フローのさまざまな段階で `report_utilization` コマンドを使用して生成できます。

このレポートには、`run` に使用されたデバイスと、次のエレメントの使用率が表示されます。

- スライス ロジック
 - LUT
 - MuxFx
 - レジスタ

- メモリ
 - ブロック RAM
 - FIFO
- DSP48
- I/O リソース
- 次のようなクロック リソース
 - BUFGCTRL
 - BUFR
 - BUFHCE
 - MMCME2_ADV
 - PLLE2_ADV
- 次のような特定のデバイス リソース
 - STARTUPE2
 - XADC
- 使用されるタイプごとのプリミティブ数
- ブラック ボックス

Tcl コンソールから実行する場合、`-cells` オプションを使用して特定の階層セルの使用率をレポートできます。Vivado IDE GUI から実行した場合、この情報は表で示されます。

ロジック最適化コマンドでネットリストが変更されることがあるので、これらの値はフローの段階によって異なる場合があります。

配置配線ログ

配置配線ログには、次の情報が含まれます。

- 場所、ネットリスト、および使用された制約に関する情報
- ロジック最適化タスク。小型で高速なネットリストが生成するため、デフォルトでロジック最適化ルーチンが実行されます。
- 配置フェーズおよび配置後のタイミング予測 (WNS および TNS のみ)
- 配線フェーズおよび配線後のタイミング予測とタイミング サマリ (WNS、TNS、WHS、および THS のみ)
- 各インプリメンテーション コマンドおよびフェーズの経過時間とメモリ

このレポートまたは [Messages] ビューでエラー、クリティカル警告、および警告を確認してください。配置で生成された警告が、フローの後の方でエラーとなる場合があります。個々の段階をそれぞれ実行した場合は、最後に実行した段階の結果のみが含まれます。

重要：タイミング サマリ レポートでパルス幅タイミング サマリも参照し、タイミング違反や不足している制約に関する情報を確認してください。

I/O レポート

I/O レポートは、ISE の PAD レポートに置き換わるものです。I/O レポートには、次の情報がリストされます。

- **Pin Number**
デバイスに含まれるすべてのピンの番号
- **Signal Name**
ピンに割り当てられたユーザー I/O 名
- **Pin Usage**
ピンで使用されるパッドまたはバッファのタイプ
- **Pin Name**
ピンの名前
- **Direction**
ピンの方向 (入力、出力、入出力、または未使用)
- **IO Standard**
ユーザー I/O の I/O 規格
アスタリスク (*) はデフォルトを示します。これは、Vivado IDE の [I/O Ports] ビューとは異なります。
- **IO Bank Number**
ピンが配置される I/O バンク
- **Drive (mA)**
駆動電流 (mA)
- **Slew Rate**
バッファのスルー レート (FAST または SLOW)
- **Termination**
オフチップ終端設定
- **IOB Delay**
ピンの遅延値
- **Voltage**
VCCO、VCCAUX、および関連のピンの値
- **Constraint**
ピンがユーザーにより制約されている場合は「FIXED」と表示
- **IOB Sequential Element**
ポートの隣の I/O バンクにフリップフロップがパックされているか
- **Signal Integrity**
オフチップ終端設定

クロック使用量レポート

クロック使用量レポートは、デバイス内のクロック リソースの使用状況を解析するのに役立ちます。クロックの配置問題をデバッグする際に有益です。

クロック使用量レポートには、次の情報が含まれます。

- 使用可能、使用済み、および制約されているクロック プリミティブ数
- BUFG ごとのロードおよびスキュー
最大遅延およびスキューの大きいネット
- MMCM ごとのロードおよびスキュー
予測以上のロード数、最大遅延、およびスキューのネット

リージョナル クロック：リージョナル クロック ネットワークは、グローバル クロック ネットワークからは独立しています。グローバル クロックとは異なり、リージョナル クロック信号 (BUFR) の範囲は 1 つのクロック領域内に制限されます。1 つの I/O クロック信号が 1 つのバンクを駆動します。ソース同期インターフェイス デザインで特に有益です。7 シリーズ FPGA の I/O バンクのサイズは、クロック領域のサイズと同じです。

ローカル クロック：汎用配線リソースに配線されるクロック ネットワークです。ローカル クロックの使用はできる限り避けてください。クロック スキューが大きくなり、PVT の変動に影響されやすくなります。インプリメンテーションを実行するたびに、クロックの配線が異なるものになる可能性があります。

クロック使用量レポートの [Locked] 列には、クロックの配置が固定されているか、クロック リソースの配置に制限がないかが示されます。

グローバル クロック数が多すぎる場合、ファンアウトの小さいクロックを BUFGH または BUFR などのクロック リソースに移動することを考慮してください。

制御セット レポート

制御セットとは、クロック信号、クロック イネーブル信号、およびセット/リセット信号の組み合わせです。各スライスでは 1 つの制御セットがサポートされ、スライス内のすべてのフリップフロップで使用できます。異なる制御セットのフリップフロップを同じスライスに配置することはできません。

制御セット レポートには、デザインに含まれる固有の制御セット数 (Number of unique control sets) と、デザインの配置に基づく制御セットの配置制限により使用できなくなったレジスタ サイトの最小数 (Minimum Number of register sites lost to control set restrictions) が表示されます。

- Clock Signal
論理クロック信号名
- Enable Signal
論理クロック イネーブル信号名
- Set/Reset Signal
論理セット/リセット信号名
- Slice Load Count
制御セットに接続されているセルに含まれるスライス数
- Bel Load Count
制御セットに接続されているセルの数

DRC レポート

DRC レポートは配線により生成されます。配線実行の前に、一般的なデザインの問題がチェックされます。レポートには、run で使用されたチェックがリストされます。クリティカル警告メッセージを確認してください。特定のチェックの重要度が、フローの後の方で上がる可能性があります。

配線ステータス レポート

配線ステータス レポートには、次の情報が含まれます。

- デザインに含まれるネットの数
- 次のようなタイル外の配線リソースを使用しないネットの数
 - CLB
 - ブロック RAM
 - I/O パッド
- 配線リソースが不要なネットの数

レポートに次のように示されていれば、デザインは完全に配線されています。

of unrouted nets.....: 0 :

タイミング サマリ レポート

タイミング サマリ レポートには、デザインのタイミングが満たされたかどうかを示されます。これは、ISE の TWR レポートに相当します。このレポートには、次のセクションが含まれています。

サマリ

次の情報が含まれます。

- デザイン名
- パーツ名
- スピード ファイル
- Vivado IDE のバージョン
- 生成にかかった時間
- コマンド ライン

Timer Settings

レポートの生成に使用されたタイミング エンジンのコンフィギュレーションを示します。

Check Timing

不足しているタイミング制約、問題があると思われるタイミング制約を示します。

Design Timing Summary

デザインで制約が満たされたかどうかを示します。TNS、THS、または TPWS のいずれかが 0ns でない場合、タイミングは満たされていません。

Clock Summary

ユーザーまたはタイミング エンジンにより定義された各クロックについて、波形、周期、周波数を示します。

Intra Clock Table

クロックドメイン内のパスのタイミング サマリを示します。

Inter Clock Table

クロックドメインをまたがるパスのタイミング サマリを示します。

Path Group Table

デフォルト クロック パス グループ以外のパス グループのタイミング サマリを示します。非同期タイミング チェック レポート (リカバリリムーバブル)、ユーザー定義グループ (group_path コマンドを参照) が含まれます。

Timing Details

「Intra Clock Table」、「Inter Clock Table」、および「Path Group Table」セクションの各クロック ペアに対し、N 個のワースト タイミング パスの詳細を示します。

インプリメンテーション が終了した後、レポートを手動で開く必要があります。ツールで自動的に開きません。
[Report Timing Summary] コマンドを使用すると、インタラクティブ レポートを作成できます。

WebTalk レポート

WebTalk レポートは、ビットストリーム生成中に生成されます。このレポートは、ユーザーのザイリンクス デバイスの使用に関する情報を収集します。この情報は、ザイリンクスでよりよいツールを提供するために使用させていただきます。機密情報は収集されません。

詳細は、<http://japan.xilinx.com/ise/webtalk/> を参照してください。

その他のレポートの生成

その他のレポートを生成するには、デザインをメモリに読み込む必要があります。よく使用されるレポートは、Flow Navigator から生成できます。

エラボレート済みデザイン

- [Report DRC]
- [Report Noise]
- [Report Power]

合成済みデザインおよびインプリメント済みデザイン

- [Report Timing Summary]
- [Report Clock Networks]
- [Report Clock Interaction]
- [Report DRC]
- [Report Noise]
- [Report Utilization]
- [Report Power]

[Tools] → [Timing] をクリックすると、タイミング関連のレポートを生成できます。たとえば、合成済みデザインを開いている場合、次のコマンドを使用できます。

- [Report Timing]
- [Report Timing Summary]
- [Report Pulse Width]
- [Report Clock Interaction]

この後のセクションで、各レポートに関する次の情報を説明します。

- レポートの実行方法
- レポートに含まれる情報

ヒント：フローのすべての段階ですべてのレポートを使用できるわけではありません。RTL 解析では、一部のレポートのみがサポートされます。

DRC レポート

[Report DRC] を使用すると、一般的なデザインの問題およびエラーを確認するデザイン ルール チェックが実行されます。

エラボレート済みデザイン

I/O およびクロック配置に関する DRC が実行されます。RTL ネットリストには、I/O バッファ、クロック バッファ、および合成済みデザインには含まれるその他のプリミティブがすべて含まれているわけではありません。エラボレート済みデザインの DRC では、その後の DRC ほど多くのエラーはチェックされません。

合成済みデザインおよびインプリメント済みデザイン

- 合成済みネットリストに関する DRC が実行されます。
- I/O、BUFG、および配置がチェックされます。
- MGT、IODELAY などのプリミティブの配線に関する属性の基本的なチェックが実行されます。
- 同じ DRC が使用可能な配置配線を考慮して実行されます。
- DRC レポートでは、メッセージと同じ 4 つの重要度が使用されます。この時点では、クリティカル警告およびエラーでフローが停止することはありません。

第 1 章：デザイン解析

インプリメンテーションの段階でも DRC が実行されますが、クリティカルなポイントでフローが停止することがあります。配置配線で配置の妨げとなる問題がチェックされます。特定のメッセージの重要度がフローの段階によって低い場合があります。重要度が低い場合、`opt_design`、`place_design`、`route_design` が停止することはありませんが、ボード上で問題が発生する可能性があります。

たとえば、ユーザーによりすべてのデザイン ポートのパッケージ ピン ロケーションおよび I/O 規格が手動で指定されているかを確認する DRC があります。不足している制約がある場合、`place_design` および `route_design` ではクリティカル警告が表示されますが、これらの DRC 違反は `write_bitstream` ではエラーとなります。これらの制約がない状態ではデバイスはプログラムできません。

フローの初期段階で重要度を低くしているのは、最終的なピン配置が決定される前にデザインをインプリメンテーションまで実行できるようにするためです。最終的な検証となる包括的な DRC を実行するには、ビットストリームを生成する必要があります。

次の図に、DRC レポートを表示する [DRC] ビューを示します。

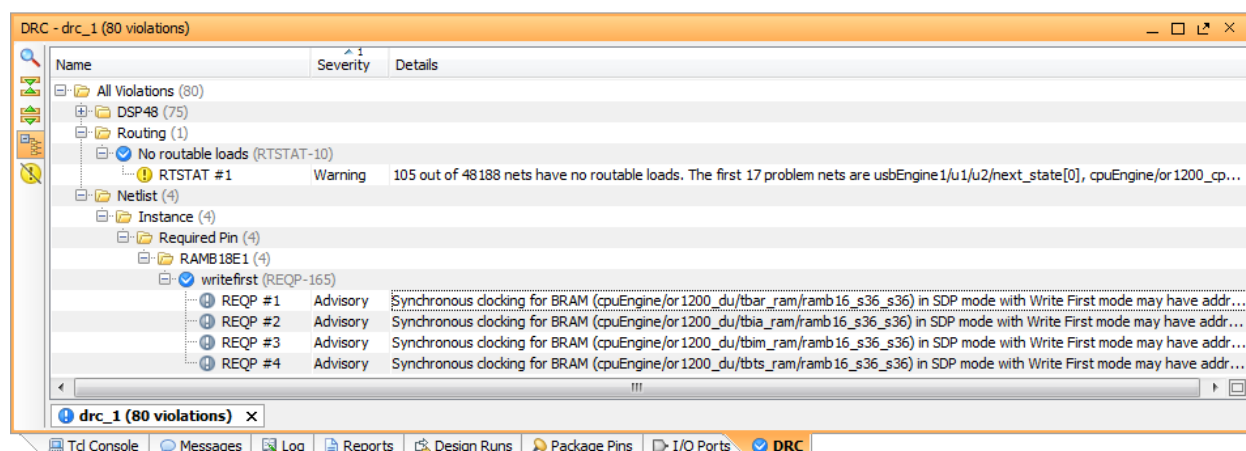


図 3：DRC レポート

DRC をクリックすると、[Violation Properties] ビューにメッセージの詳細が表示されます。[Detailed] タブをクリックして詳細を確認してください。多くのメッセージには、ネット、セル、およびポートへのハイパーリンクが含まれています。

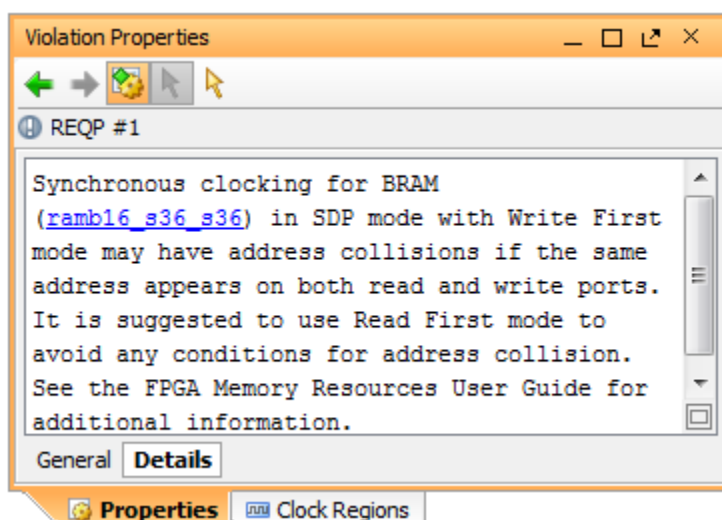


図 4：[Violation Properties] ビュー

DRC レポートは、デザインを変更したときにダイナミックに変化しません。デザインを変更したら、[Rerpot DRC] を再実行する必要があります。オブジェクトを削除または移動した場合などの操作によりリンクが無効になる場合、ツールでそれが判断されリンクが使用できなくなります。

リンクをクリックするとオブジェクトが選択されますが、[Properties] ビューは更新されません。オブジェクトのプロパティを表示するには、オブジェクトの選択を一度解除してから選択し直す必要があります。

Tcl で DRC レポートを作成するには、次のコマンドを使用します。

```
report_drc
```

次のコマンドを使用すると、結果をファイルに出力できます。

```
report_drc -file myDRCs.txt
```

ヒント: report_drc コマンドの詳細は、「report_drc -help」と入力してください。

SSN レポート

[Report Noise] コマンドは、7 シリーズ デバイスの同時スイッチ ノイズ (SSN) 解析を実行します。デフォルトでは、SSN レポートは Vivado IDE の下部の [Noise] ビューに表示されます。結果は CSV または HTML ファイルにエクスポートできます。

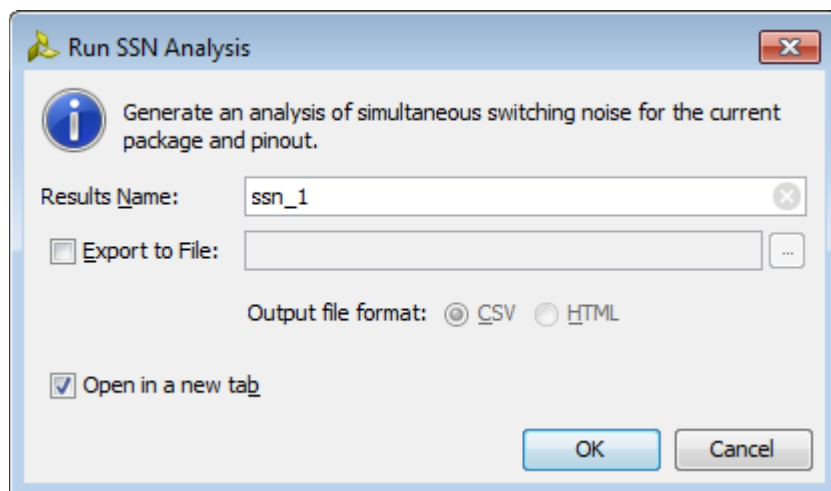


図 5 : [Run SSN Analysis] ダイアログ ボックス

SSN レポートには、次の 4 つのセクションがあります。

- Summary
- Messages
- I/O Bank Details
- Links

[Summary] セクション

次の情報を表示します。

- レポートが生成された日時
- 解析されたポート数とその割合
- ステータス (問題が検出されたかどうか)
- クリティカル警告、警告、および情報メッセージの数

[Messages] セクション

レポートの作成中に生成されたメッセージを示します。

[I/O Bank Details] セクション

ピン、I/O 規格、ノイズ マージン値の残りなどの情報を示します。

[Links] セクション

japan.xilinx.com/support 上の関連の資料へのリンクを示します。

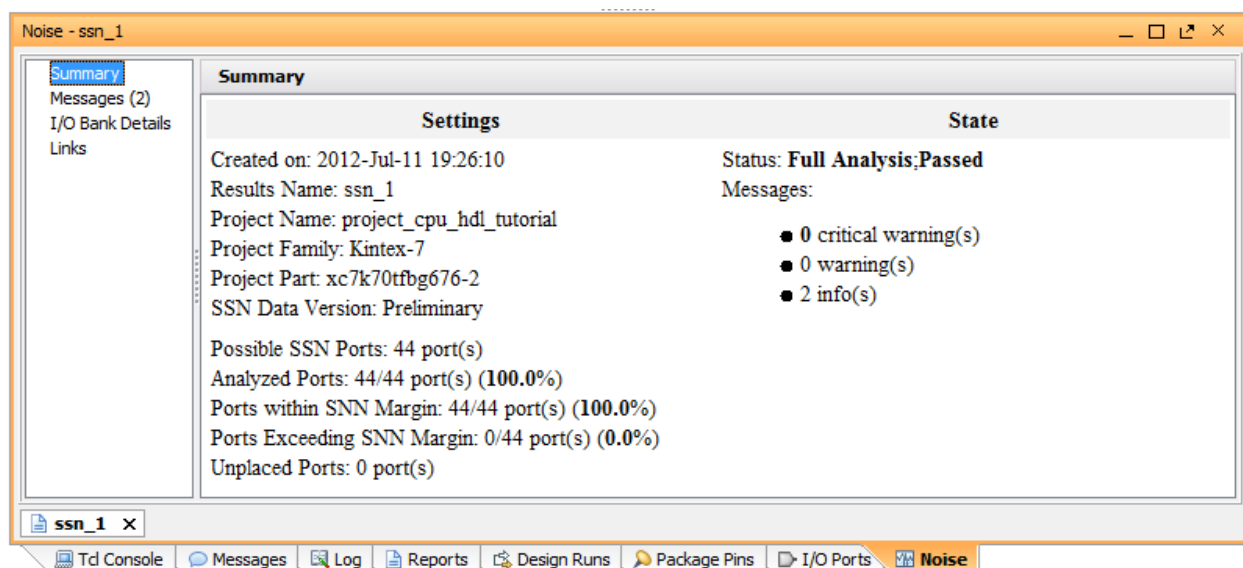


図 6 : SSN レポート

HTML 形式のレポートを生成するには、[Run SSN Analysis] ダイアログ ボックスでオプションを選択するか、次の Tcl コマンドを使用します。

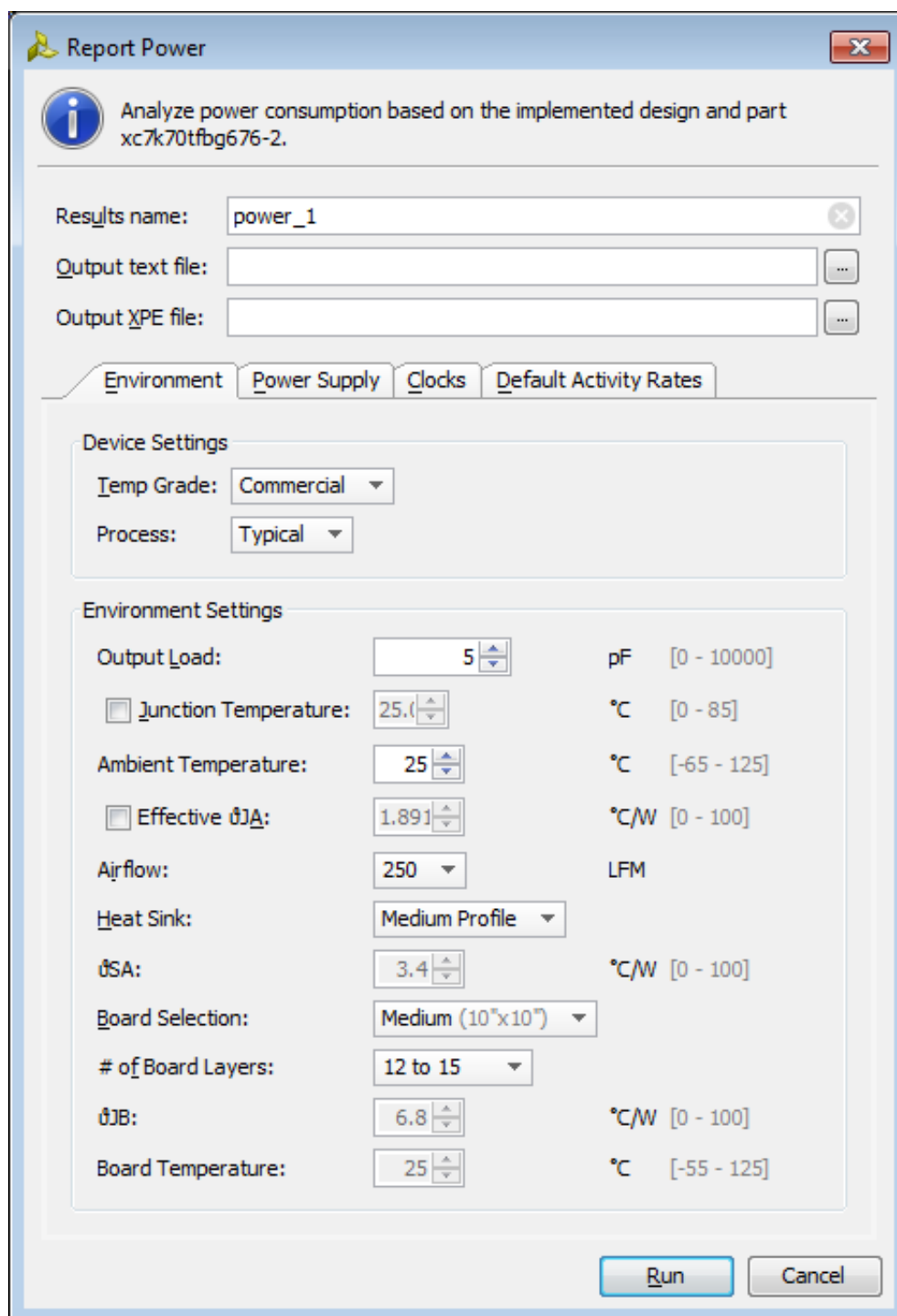
```
report_ssn -format html -file myImplementedDesignSSN.html
```

消費電力レポート

[Report Power] は、合成済みデザインまたはインプリメント済みデザインを開いているときに実行できます。

このレポートでは、次のようなデザイン入力に基づいて消費電力が予測されます。

- 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
- デザインで使用す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 FPGA 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、温度による消費電力の変動、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



The 'Report Power' dialog box is used to analyze power consumption. It features a title bar with a close button. Below the title bar is an information icon and a text box stating: 'Analyze power consumption based on the implemented design and part xc7k70tfbg676-2.' The dialog is divided into several sections. The top section contains three input fields: 'Results name:' with the value 'power_1', 'Output text file:', and 'Output XPE file:'. Below these are four tabs: 'Environment', 'Power Supply', 'Clocks', and 'Default Activity Rates'. The 'Environment' tab is selected, showing 'Device Settings' and 'Environment Settings'. 'Device Settings' includes 'Temp Grade:' set to 'Commercial' and 'Process:' set to 'Typical'. 'Environment Settings' includes a list of parameters with their values and units: 'Output Load:' (5 pF), 'Junction Temperature:' (25.0 °C), 'Ambient Temperature:' (25 °C), 'Effective θ_{JA} :' (1.891 °C/W), 'Airflow:' (250 LFM), 'Heat Sink:' (Medium Profile), ' θ_{SA} :' (3.4 °C/W), 'Board Selection:' (Medium (10"x10")), '# of Board Layers:' (12 to 15), ' θ_{JB} :' (6.8 °C/W), and 'Board Temperature:' (25 °C). At the bottom right are 'Run' and 'Cancel' buttons.

Report Power

Analyze power consumption based on the implemented design and part xc7k70tfbg676-2.

Results name: power_1

Output text file:

Output XPE file:

Environment Power Supply Clocks Default Activity Rates

Device Settings

Temp Grade: Commercial

Process: Typical

Environment Settings

Output Load: 5 pF [0 - 10000]

☐ Junction Temperature: 25.0 °C [0 - 85]

Ambient Temperature: 25 °C [-65 - 125]

☐ Effective θ_{JA} : 1.891 °C/W [0 - 100]

Airflow: 250 LFM

Heat Sink: Medium Profile

θ_{SA} : 3.4 °C/W [0 - 100]

Board Selection: Medium (10"x10")

of Board Layers: 12 to 15

θ_{JB} : 6.8 °C/W [0 - 100]

Board Temperature: 25 °C [-55 - 125]

Run Cancel

図 7 : [Report Power] ダイアログ ボックス

消費電力レポートの解析

消費電力レポートでは、次のものに基づいて解析できます。

- 設定
- 消費電力の合計
- 階層
- 電圧レール
- ブロックタイプ

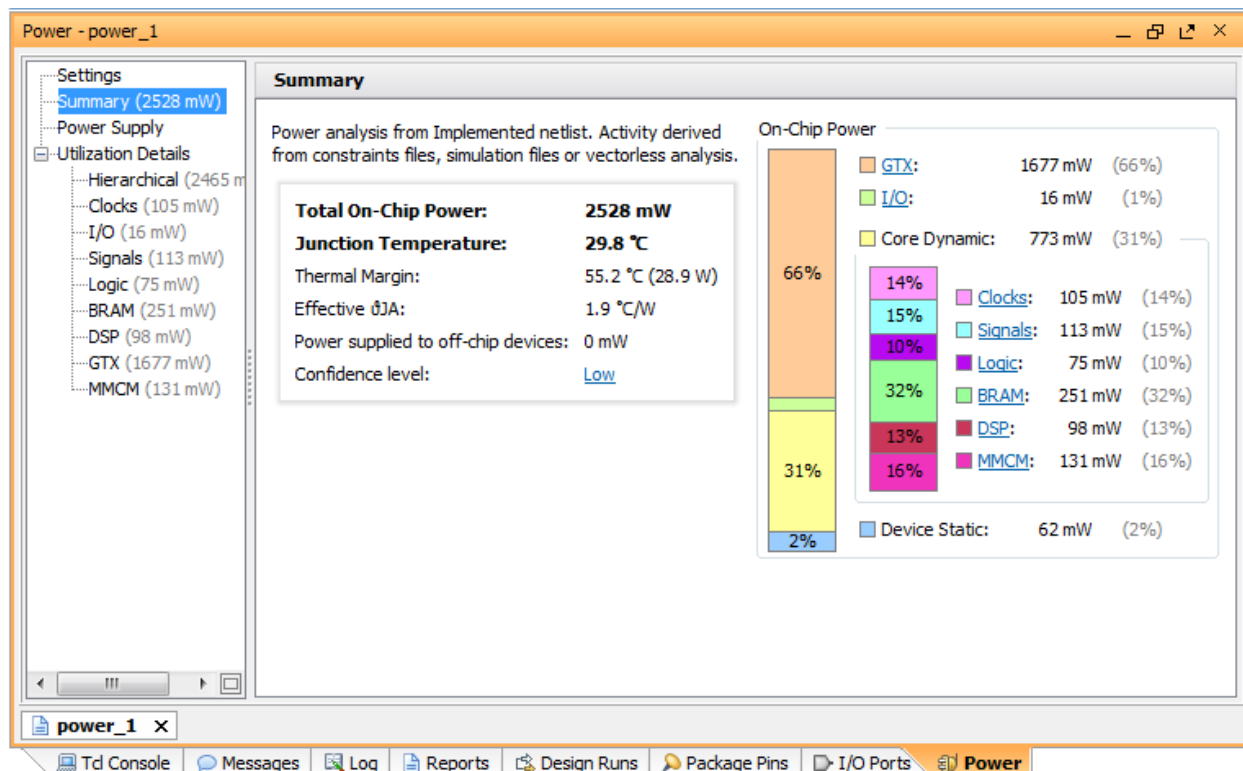


図 8：消費電力レポート

消費電力レポートと結果の解析の詳細は、『Vivado Design Suite ユーザー ガイド：消費電力解析および最適化』(UG907)を参照してください。

非プロジェクト モードでの消費電力レポートの生成

非プロジェクト モードでは、link_design または synth_design コマンドを実行した後に report_power コマンドを使用して消費電力レポートを生成できます。使用可能な配置配線情報を使用して、より正確な消費電力予測値が示されます。Tcl コンソールまたはスクリプトでこのレポートを生成するには、次のコマンドを使用します。

```
report_power
```

タイミング サマリ レポート

タイミング解析は、合成後であればどの地点でも実行できます。合成およびインプリメンテーション run で自動的に生成されるタイミング サマリ レポートを確認できます。合成済みデザインまたはインプリメント済みデザインをメモリに読み込むと、Flow Navigator の [Synthesis] または [Implementation] の下から [Report Timing Summary] をクリック、または [Tools] → [Timing] → [Report Timing Summary] をクリックして、インタラクティブなタイミング サマリ レポートを生成できます。これに相当する Tcl コマンドは `report_timing_summary` です。このコマンドの詳細は、『Vivado Design Suite Tcl コマンドリファレンス ガイド』(UG835) を参照してください。

合成済みデザインでは、接続およびファンアウトに基づいてネット遅延が予測されます。セルが制約により既に配置されていると、その間のネット遅延はより正確なものとなります。

インプリメント済みデザインでは、ネット遅延は実際の配線情報に基づいています。最終的なタイミングの確認には、完全に配線されたデザインを使用してください。デザインが完全に配線されているかどうかは、配線ステータス レポートで確認できます。

[Report Timing Summary] ダイアログ ボックスのオプション

[Report Timing Summary] ダイアログ ボックスには、次のタブがあります。

- [Options]
- [Advanced]
- [Timer Settings]

[Results name] で名前を指定すると、結果が [Timing] ビューに表示されます。このフィールドが空白の場合、レポートは [Tcl Console] ビューに表示されます。[Open in a new tab] をオフにすると、[Timing] ビューに表示されていた結果が削除されます。

次のセクションで、ダイアログ ボックスの各オプションについて説明し、同等の `report_timing_summary` コマンドのオプションを示します。

[Options] タブ

次の図に、[Report Timing Summary] ダイアログ ボックスの [Options] タブを示します。

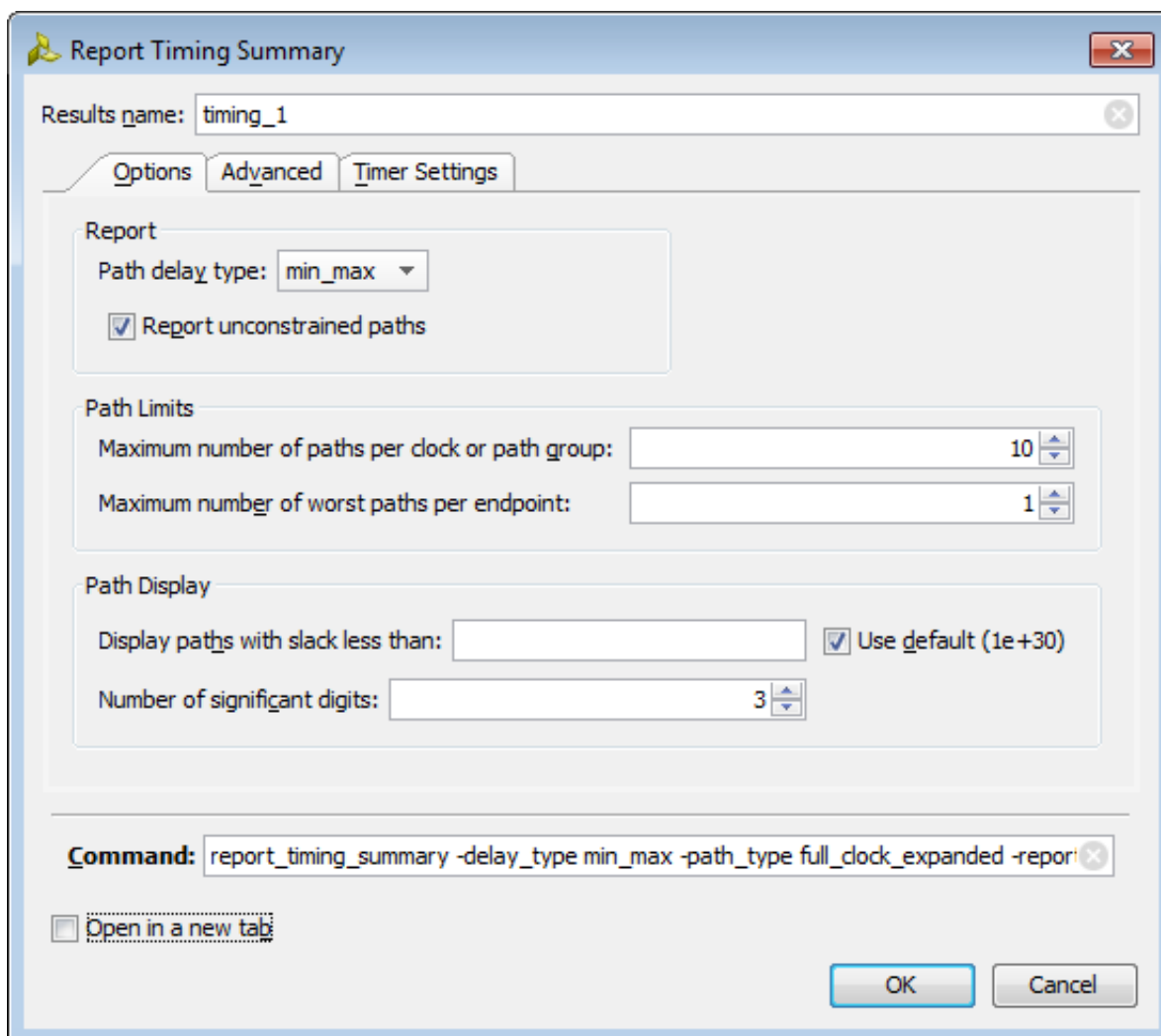


図 9 : [Report Timing Summary] ダイアログ ボックスの [Options] タブ

次のコマンドが含まれています。

- [Report]
 - [Path delay type] : 実行する解析のタイプを設定します。合成済みデザインの場合、デフォルトでは最大遅延解析 (セットアップ/リカバリ) のみが実行されます。インプリメント済みデザインの場合、デフォルトでは最小/最大遅延解析 (セットアップ/ホールド、リカバリ/リムーバル) の両方が実行されます。[min] を選択すると、最小遅延解析 (ホールドおよびリムーバル) のみが実行されます。
同等 Tcl オプション : -delay_type
 - [Report unconstrained paths] : タイミング要件のないパスに関する情報を生成します。このオプションは、Vivado IDE ではデフォルトでオンになっていますが、同等の Tcl コマンド report_timing_summary ではデフォルトでは使用されません。
同等 Tcl オプション : -report_unconstrained

- [Path Limits]
 - [Maximum number of paths per clock or path group] : クロック ペアまたはパス グループごとにレポートするパスの最大数を指定します。
同等 Tcl オプション : `-max_paths`
 - [Maximum number of worst paths per endpoint] : パスのエンドポイントごとにレポートするワースト パスの最大数を指定します。これは、クロック ペアまたはパス グループごとのパスの最大数によって制限されます。
同等 Tcl オプション : `-nworst`
- [Path Display]
 - [Display paths with slack less than] : スラックが指定した値未満のパスをレポートします。このオプションは、サマリ表には影響しません。
同等 Tcl オプション : `-slack_lesser_than`
 - [Number of significant digits] : レポートに表示される値の精度を指定します。
同等 Tcl オプション : `-significant_digits`

[Advanced] タブ

次の図に、[Report Timing Summary] ダイアログ ボックスの [Advanced] タブを示します。

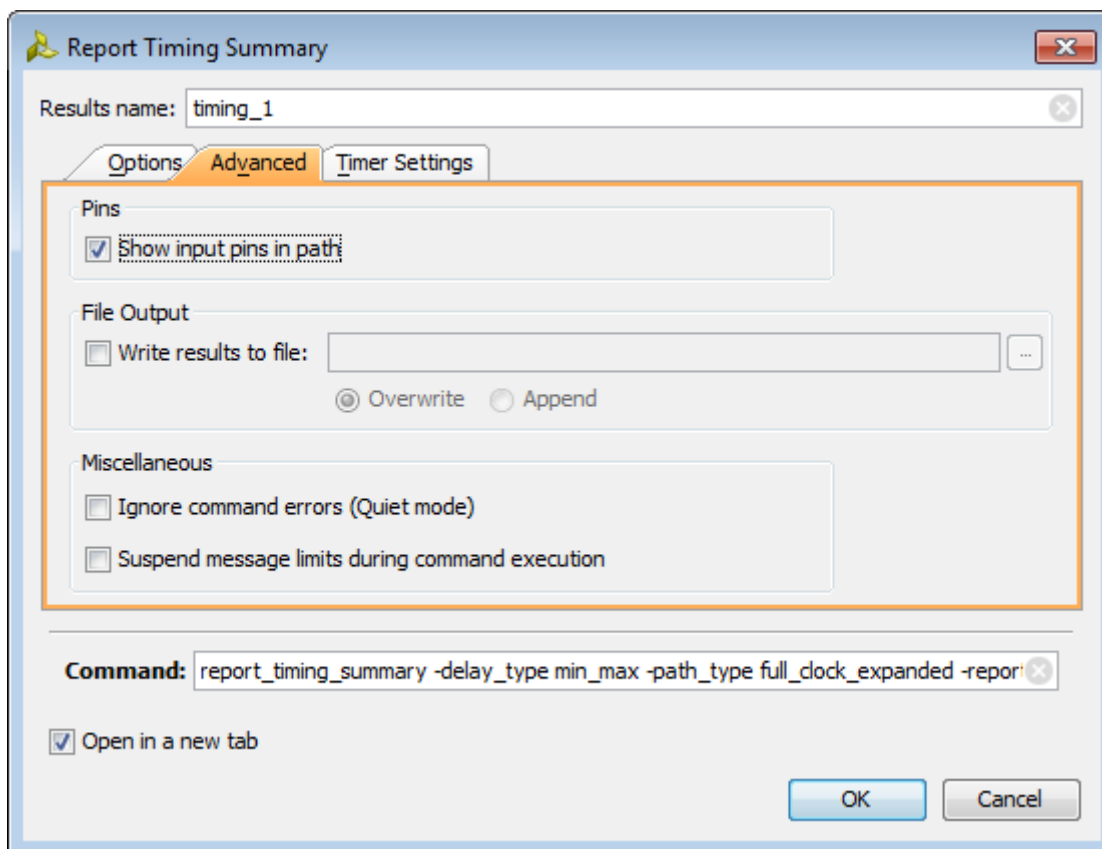


図 10 : [Report Timing Summary] ダイアログ ボックスの [Advanced] タブ

次のオプションがあります。

- [Show input pins in path] : パスの詳細に入力ピンを表示します。このオプションをオンにしておくことをお勧めします。

同等 Tcl オプション : `-input_pins`

- [File Output] : コマンドの結果をファイルに出力します。既存のレポート ファイルに追加するオプションもあります。

同等 Tcl オプション : `-file` および `-append`

- [Miscellaneous] :

- [Ignore command errors] : エラーを無視します。

同等 Tcl オプション : `-quiet`

- [Suspend message limits during command execution] : プログラム実行中メッセージの制限を解除し、すべてのメッセージを表示します。

同等 Tcl オプション : `-verbose`

[Timer Settings] タブ

タイマー設定を指定するには、ダイアログ ボックスまたは Tcl コンソールを使用します。これらの設定は、同じ Vivado IDE セッションで実行する合成およびインプリメンテーション以外のタイミング関連のコマンドに影響します。タイマー設定はツール プリファレンスとしては保存されないため、新しいセッションごとにデフォルト値に戻ります。

推奨 : デフォルト値を変更しないでください。デフォルト値では、最も正確な遅延値が使用され、タイミング解析の適用範囲が最大限になります。

次の図に、[Report Timing Summary] ダイアログ ボックスの [Timer Settings] タブを示します。

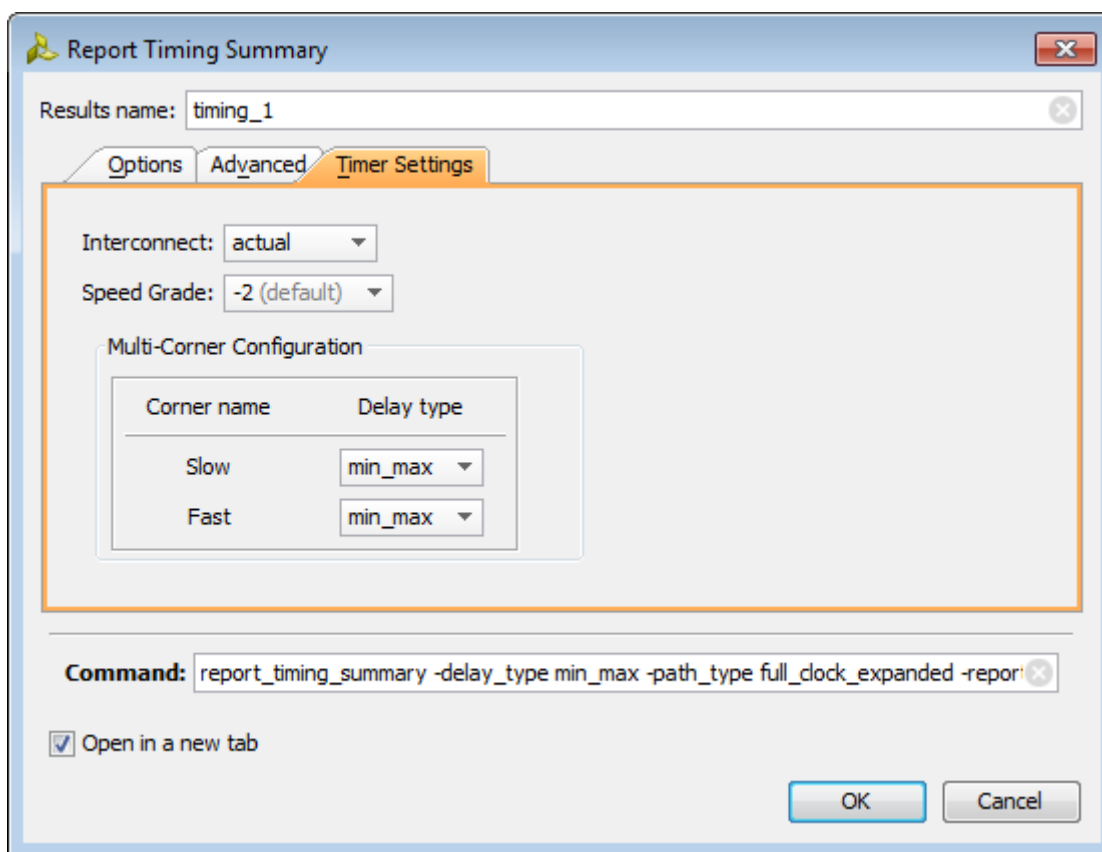


図 11 : [Report Timing Summary] ダイアログ ボックスの [Timer Settings] タブ

次のコマンドが含まれています。

- [Interconnect] : ネット遅延の計算方法を指定します。デフォルトでは、合成済みデザインの解析では [estimated] に、インプリメント済みデザインの解析では [actual] に設定されます。
 - [estimated]

未配置のセルの場合、ネット遅延値はドライバーおよびロード、ファンアウトの特性に基づく最適な配置での遅延値になります。このネットは、タイミング パス レポートでは「unplaced」と示されます。

配置済みセルの場合、ネット遅延はドライバーとロード間の距離およびファンアウトにより決まります。このネットは、タイミング パス レポートでは「estimated」と示されます。
 - [actual]

配線済みネットの場合、ネット遅延は実際のハードウェア遅延と同じになります。このネットは、タイミング パス レポートでは「routed」と示されます。

配置済みセル間の未配線のネットの場合、ネット遅延はドライバーとロード間の距離およびファンアウトにより決まります。このネットは、タイミング パス レポートでは「estimated」と示されます。
 - [none]

ネット遅延は 0 になります。

同等 Tcl オプション : `set_delay_model`

- [Speed Grade] : スピード グレードを設定します。デフォルトでは、プロジェクトを作成する際やデザイン チェックポイントを開く際に選択したデバイスに基づいて設定されます。このオプションを使用すると、インプリメンテーションフローをすべて再実行しなくても、別のスピード グレードで同じデザイン データベースのタイミングをレポートできます。

同等 Tcl オプション : `set_speed_grade`

- [Multi-Corner ConfigurationMulti-Corner Configuration] : 各コーナーで実行する解析をセットアップ ([max])、ホールド ([min])、または両方 ([min_max]) に設定します。

推奨 : 両方のコーナーでセットアップ (最大) およびホールド (最小) 解析の両方を実行することをお勧めします。

同等 Tcl オプション : `config_timing_corners`

タイミング サマリレポートのセクション

タイミング サマリレポートには、次のセクションが含まれます。

- General Information
- Timer Settings
- Design Timing Summary
- Check Timing
- Clock Summary
- Intra-Clock Paths
- Inter-Clock Paths
- Path Groups
- Unconstrained Paths

タイミング サマリレポートには、Vivado IDE で生成可能な複数のレポート ([Report Clock Interaction]、[Report Pulse Width]、[Report Timing]、[Check Timing]) の情報、および `report_clocks` などの Tcl のみで生成可能なレポートと同様の情報が含まれますが、制約の適用されていないパス (Unconstrained Paths) など、このレポートにしか含まれない情報もあります。

次に、各セクションについて説明します。

General Information

一般情報 (General Information) セクションには、次の情報が含まれます。

- デザイン名
- 選択されたデバイス、パッケージ、スピード グレード (スピード ファイル バージョン)
- Vivado Design Suite のバージョン
- 日付
- レポートを生成するために実行された同等の Tcl コマンド

Timer Settings

レポートのタイミング情報を生成するのに使用された Vivado タイミング解析エンジン設定の詳細を示します。次の図に、[Timer Settings] セクションの例を示します。次の情報が含まれます。

- [Enable Multi-Corner Analysis] : マルチコーナー解析がイネーブルかどうかを示します。各コーナーの解析での解析が実行されたかは、[Multi-Corner Configuration] に示されます。
- [Enable Pessimism Removal] および [Pessimism Removal Resolution] : これらの設定は、常にイネーブルにする必要があります。この設定により、各パスのソース クロックおよびデスティネーション クロックで共通ノードでのスキューはレポートされなくなります。
- [Enable Input Delay Default Clock] : ユーザー制約のない入力ポートにデフォルトのヌル入力遅延制約を作成します。このオプションは、デフォルトではオフになっています。
- [Enable Preset / Clear Arcs] : 非同期ピンを介してタイミング パスを伝搬します。リカバリリムーバル チェックには影響しません。デフォルトではオフになっています。

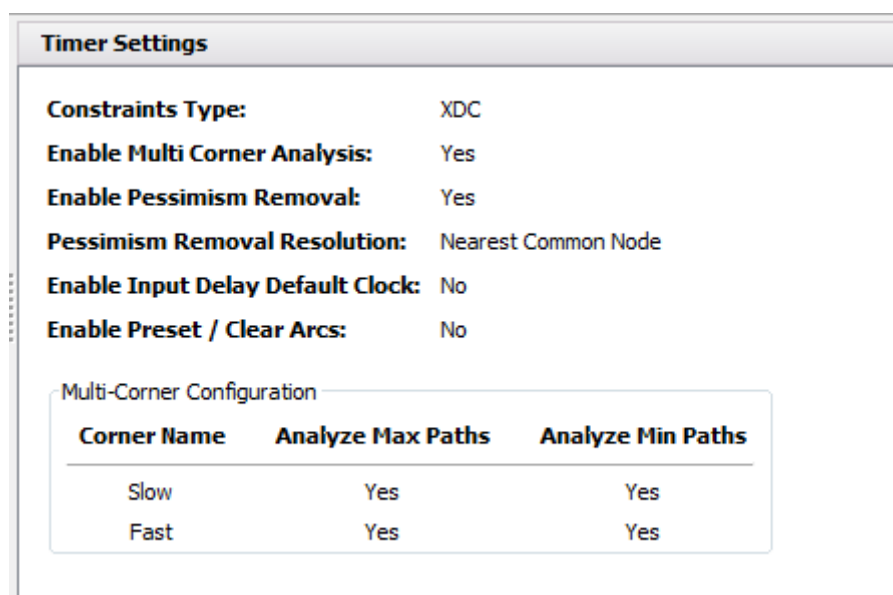


図 12 : タイミング サマリ レポートの [Timer Settings] セクション

Design Timing Summary

デザインのタイミングのサマリを示します。ほかのセクションの結果が統合されています。

推奨 : このサマリを使用すると、配線後にタイミング制約が満たされたか、デザインの現在のステータスなどを簡単に確認できます。

- [Setup]

最大遅延解析 (セットアップ、リカバリ、およびデータ チェック) の結果を示します。

 - [Worst Negative Slack (WNS)] : 最大遅延解析におけるすべてのタイミング パスの最悪のスラックを示します。この値は、正の場合と負の場合があります。
 - [Total Negative Slack (TNS)] : 各タイミング パス エンドポイントにおける最悪の違反を考慮した場合の、WNS 違反の合計を示します。最大遅延解析ですべてのタイミング制約が満たされている場合は 0ns になり、違反がある場合は負の値になります。
 - [Number of Failing Endpoints] : 違反が発生している (WNS < 0ns) エンドポイントの総数を示します。

- [Hold]

最小遅延解析 (ホールド、リムーバル、およびデータ チェック) の結果を示します。

- [Worst Hold Slack (WHS)] : 最小遅延解析におけるすべてのタイミング パスの最悪のスラックを示します。この値は、正の場合と負の場合があります。
- [Total Hold Slack (THS)] : 各タイミング パス エンドポイントにおける最悪の違反を考慮した場合の、WHS 違反の合計を示します。最小遅延解析ですべてのタイミング制約が満たされている場合、この値は 0ns になり、違反がある場合は負の値になります。
- [Number of Failing Endpoints] : 違反が発生している (WHS < 0ns) エンドポイントの総数を示します。

- [Pulse Width]

ピンのスイッチ制限に関する次のチェックの結果を示します。

- 最小 Low パルス幅
- 最小 High パルス幅
- 最小周期
- 最大周期
- 最大スキュー (同じ下位セルの 2 つのクロック ピン間)

次の 3 つの値が表示されます。

- [Worst Pulse Width Slack (WPWS)] : 最小遅延および最大遅延を使用した上記のすべてのチェックにおける最悪のパルス幅スラックを示します。
- [Total Pulse Width Slack (TPWS)] : 各タイミング パス エンドポイントにおける最悪の違反を考慮した場合の、WPWS 違反の合計を示します。すべてのタイミング制約が満たされている場合は 0ns になり、違反がある場合は負の値になります。
- [Number of Failing Endpoints] : 違反が発生している (WPWS < 0ns) エンドポイントの総数を示します。

次の図に、[Design Timing Summary] セクションの例を示します。

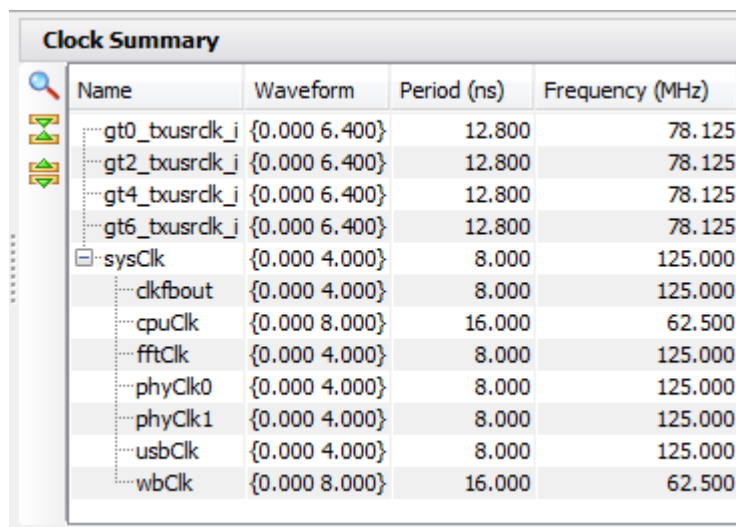
Design Timing Summary		
Setup	Hold	Pulse Width
Worst Negative Slack (WNS): -0.450	Worst Hold Slack (WHS): 0.109	Worst Pulse Width Slack (WPWS): 2.000
Total Negative Slack (TNS): -1.161	Total Hold Slack (THS): 0.000	Total Pulse Width Negative Slack (TPWS): 0.000
Number of Failing Endpoints: 3	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0

図 13 : タイミング サマリ レポートの [Design Timing Summary] セクション

Clock Summary

report_clocks コマンドを実行した場合と同様の情報が含まれ、create_clock コマンド、create_generated_clock コマンド、またはツールで自動的に生成されたクロックすべてが表示されます。名前、周期、波形、タイプ、周波数要件などの各クロックのプロパティも示されます。名前のインデントは、マスター クロックと生成クロックの関係を表します。

次の図に、[Clock Summary] セクションの例を示します。



Name	Waveform	Period (ns)	Frequency (MHz)
gt0_txusrclk_j	{0.000 6.400}	12.800	78.125
gt2_txusrclk_j	{0.000 6.400}	12.800	78.125
gt4_txusrclk_j	{0.000 6.400}	12.800	78.125
gt6_txusrclk_j	{0.000 6.400}	12.800	78.125
sysClk	{0.000 4.000}	8.000	125.000
clkfbout	{0.000 4.000}	8.000	125.000
cpuClk	{0.000 8.000}	16.000	62.500
fftClk	{0.000 4.000}	8.000	125.000
phyClk0	{0.000 4.000}	8.000	125.000
phyClk1	{0.000 4.000}	8.000	125.000
usbClk	{0.000 4.000}	8.000	125.000
wbClk	{0.000 8.000}	16.000	62.500

図 14 : タイミング サマリ レポートの [Clock Summary] セクション

Check Timing

不足しているタイミング制約や、制約に問題のあるパスに関する有益な情報を示します。すべてのパス エンドポイントに制約が設定されている必要があります。このレポートを個別に生成するには、[Tools] → [Timing] → [Check Timing] をクリックするか、check_timing コマンドを使用します。

デフォルトで実行されるチェックは、次のとおりです。

- **no_clock** : 定義されたタイミング クロックが供給されていないクロック ピンをレポートします。定数クロック ピンもレポートされます。
- **unconstrained_endpoints** : タイミング 要件がないパスのエンドポイントをレポートします。これは、no_clock チェックでもレポートされる不足しているクロック定義にも直接関係しています。
- **no_input_delay** : 入力遅延制約が設定されていない入力ポートをレポートします。
- **no_output_delay** : 出力遅延制約が設定されていない出力ポートをレポートします。
- **multiple_clock** : 複数のタイミング クロックが到達するクロック ピンをレポートします。この状況は、クロック ツリーにクロック マルチプレクサーがある場合に発生します。
- **loops** : デザインで検出された組み合わせループをレポートします。ループは、タイミングをレポートするため Vivado タイミング エンジンにより自動的に分離されます。
- **generated_clocks** : 同じクロック ツリーにないマスター クロック ソースを基準とする生成クロックをレポートします。
- **partial_input_delay** : 最小入力遅延制約または最大入力遅延制約のどちらかしか設定されていない入力ポートをレポートします。これらのポートは、セットアップ解析とホールド解析の両方にはレポートされません。
- **partial_output_delay** : 最小出力遅延制約または最大出力遅延制約のどちらかしか設定されていない出力ポートをレポートします。これらのポートは、セットアップ解析とホールド解析の両方にはレポートされません。
- **unexpandable_clocks** : Vivado タイミング エンジンで 1000 クロック サイクル以上、共通乗数を見つけることができなかったクロック ペアをレポートします。これらのクロック ペア間のパスのタイミングは確実に調整できず、クロック ペアを非同期として処理する必要があります。

制約の詳細は、『Vivado Design Suite ユーザー ガイド : 制約の使用』(UG903) を参照してください。

Intra-Clock Paths

ソース クロックとデスティネーション クロックが同じタイミング パスのワースト スラックおよび合計違反を示します。次の図に、[Intra-Clock Paths] セクションの例を示します。

Intra-Clock Paths											
Clock	Clock Edges...	WNS (ns)	TNS (ns)	# Fail...	Clock Edges...	WHS (ns)	THS (ns)	# Fail...	WPWS (ns)	TPWS (ns)	# Fail...
gt0_bxusrclk_i	rise - rise	9.605	0.000	0	rise - rise	0.302	0.000	0	6.050	0.000	0
gt2_bxusrclk_i	rise - rise	9.605	0.000	0	rise - rise	0.302	0.000	0	6.050	0.000	0
gt4_bxusrclk_i	rise - rise	9.605	0.000	0	rise - rise	0.302	0.000	0	6.050	0.000	0
gt6_bxusrclk_i	rise - rise	9.605	0.000	0	rise - rise	0.302	0.000	0	6.050	0.000	0
sysClk				0				0	2.000	0.000	0
clkfbout				0				0	6.592	0.000	0
cpuClk	rise - rise	5.604	0.000	0	rise - rise	0.234	0.000	0	7.650	0.000	0
fftClk	rise - rise	4.723	0.000	0	rise - rise	0.302	0.000	0	3.650	0.000	0
phyClk0	rise - rise	1.930	0.000	0	rise - rise	0.292	0.000	0	3.650	0.000	0
phyClk1	rise - rise	1.930	0.000	0	rise - rise	0.292	0.000	0	3.650	0.000	0
usbClk	rise - rise	2.260	0.000	0	rise - rise	0.302	0.000	0	3.650	0.000	0
wbClk	rise - rise	8.428	0.000	0	rise - rise	0.302	0.000	0	7.650	0.000	0

図 15 : タイミング サマリ レポートの [Intra-Clock Paths] セクション

左側のインデックス ペインで [Intra-Clock Paths] の下のクロック名をクリックすると、詳細な情報が表示されます。各クロックのスラックおよび違反のサマリ、および [SETUP]、[HOLD]、[Pulse Width] をクリックしてセットアップ、ホールド、パルス幅チェックの N 個のワースト パスに関する詳細を表示できます。

各解析タイプのラベルの横に、ワースト スラック値とレポートされるパス数が表示されます。次の図を参照してください。

Timing - Timing Summary - timing_1											
Intra-Clock Paths - phyClk0 - SETUP											
			Id	Name	Slack	From	To	Total Delay	Lc		
			1	Path 1	1.930	usbEngine0/u4/csr_reg[26]/C	usbEngine0/u1/u2/sizd_c_reg[12]/D	6.007			
			2	Path 2	1.983	usbEngine0/u4/csr_reg[26]/C	usbEngine0/u1/u2/sizd_c_reg[8]/D	5.954			
			3	Path 3	1.984	usbEngine0/u4/csr_reg[26]/C	usbEngine0/u1/u2/sizd_c_reg[13]/D	5.953			
			4	Path 4	1.986	usbEngine0/u4/csr_reg[26]/C	usbEngine0/u1/u2/sizd_c_reg[11]/D	5.951			
			5	Path 5	2.001	usbEngine0/u4/csr_reg[26]/C	usbEngine0/u1/u2/sizd_c_reg[10]/D	5.936			
			6	Path 6	2.036	usbEngine0/u4/csr_reg[26]/C	usbEngine0/u1/u2/sizd_c_reg[4]/D	5.901			
			7	Path 7	2.037	usbEngine0/u4/csr_reg[26]/C	usbEngine0/u1/u2/sizd_c_reg[9]/D	5.900			
			8	Path 8	2.039	usbEngine0/u4/csr_reg[26]/C	usbEngine0/u1/u2/sizd_c_reg[7]/D	5.898			
			9	Path 9	2.054	usbEngine0/u4/csr_reg[26]/C	usbEngine0/u1/u2/sizd_c_reg[6]/D	5.883			
			10	Path 10	2.090	usbEngine0/u4/csr_reg[26]/C	usbEngine0/u1/u2/sizd_c_reg[5]/D	5.847			

図 16 : タイミング サマリ レポートの [Intra-Clock Paths] セクション

ソース クロックとデスティネーション クロックが異なるタイミング パスのワースト スラックおよび合計違反を示します。次の図に、[Inter-Clock Paths] セクションの例を示します。

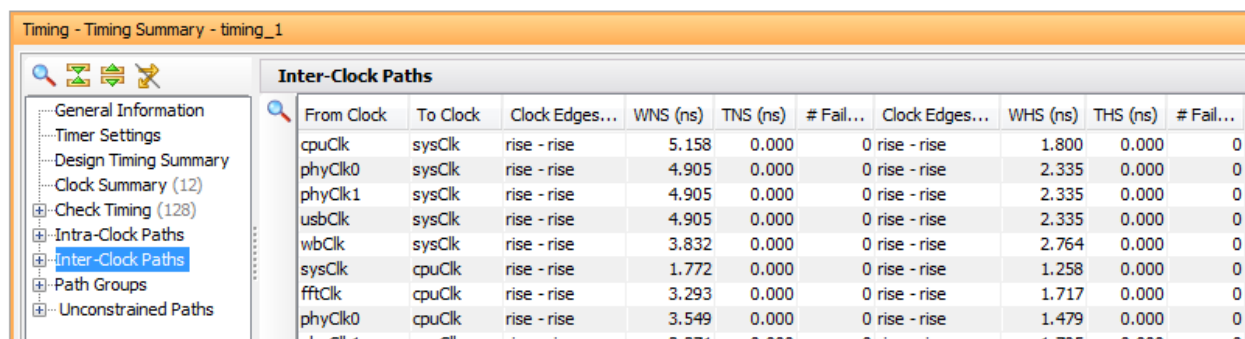


図 17: タイミング サマリ レポートの [Inter-Clock Paths] セクション

左側のインデックス ペインで [Inter-Clock Paths] の下のクロック名をクリックすると、詳細な情報が表示されます。各クロックのスラックおよび違反のサマリ、および [SETUP]、[HOLD]、[Pulse Width] をクリックしてセットアップ、ホールド、パルス幅チェックの N 個のワースト パスに関する詳細を表示できます。

各解析タイプのラベルの横に、ワーストスラック値とレポートされるパス数が表示されます。次の図を参照してください。

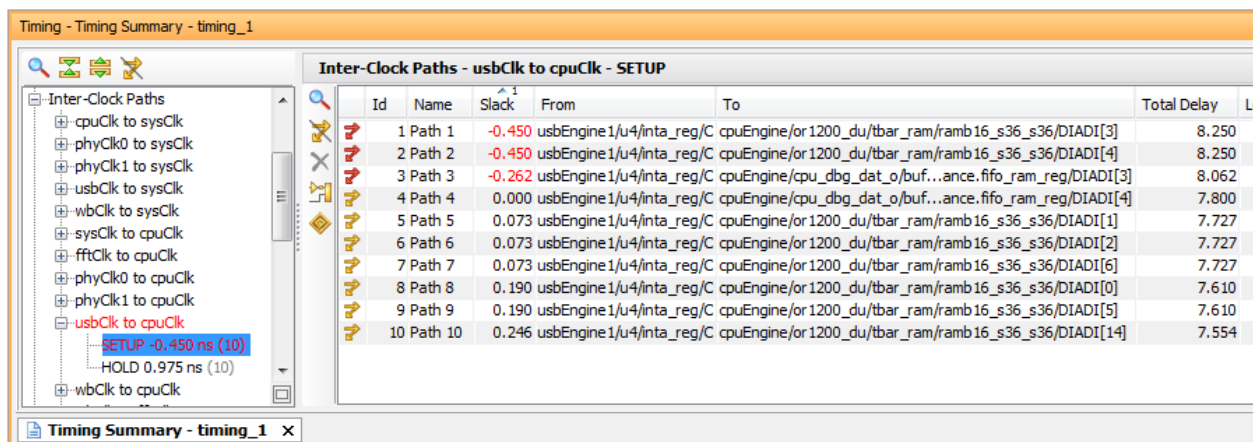


図 18: タイミング サマリ レポートの [Inter-Clock Paths] セクションの詳細

Path Groups

デフォルトのパス グループおよびユーザー定義のパス グループを表示します。次の図に、[Path Groups] セクションの例を示します。

Path Group	From Clock	To Clock	Clock Edges...	WNS (ns)	TNS (ns)	# Fail...	Clock Edges...	WNS (ns)	THS (ns)	# Fail...
async_default	wbClk	cpuClk	rise - rise	12.622	0.000	0	rise - rise	0.219	0.000	0
async_default	wbClk	fftClk	rise - rise	7.070	0.000	0	rise - rise	0.219	0.000	0
async_default	gt0_txusrclk_j	gt0_txusrclk_j	rise - rise	9.829	0.000	0	rise - rise	2.374	0.000	0
async_default	gt2_txusrclk_j	gt2_txusrclk_j	rise - rise	9.829	0.000	0	rise - rise	2.374	0.000	0
async_default	gt4_txusrclk_j	gt4_txusrclk_j	rise - rise	9.829	0.000	0	rise - rise	2.374	0.000	0
async_default	gt6_txusrclk_j	gt6_txusrclk_j	rise - rise	9.829	0.000	0	rise - rise	2.374	0.000	0
async_default	wbClk	usbClk	rise - rise	7.070	0.000	0	rise - rise	0.219	0.000	0
async_default	sysClk	wbClk	rise - rise	4.472	0.000	0	rise - rise	1.128	0.000	0
async_default	wbClk	wbClk	rise - rise	15.200	0.000	0	rise - rise	0.412	0.000	0

図 19 : タイミング サマリ レポートの [Path Groups] セクション

注記 : **async_default** は、Vivado タイミング エンジンで自動的に作成されたパス グループで、リカバリ/リムーバルのような非同期タイミング チェックで終わるすべてのパスが含まれます。これらのチェックは、[SETUP] (最大遅延解析) および [HOLD] (最小遅延解析) に個別にレポートされます。group_paths コマンドで作成したグループもすべてこのセクションに表示されます。ソース クロックおよびデスティネーション クロックはどの組み合わせでも、パス グループを構成できます。

Unconstrained Paths

制約がないか、パスを無視するタイミング例外が追加されているために、タイミングが適用されていない論理パスを示します。これらのパスは、ソースおよびデスティネーション クロック ペアごとに分類されます。パスの開始点または終点に関連付けられているクロックがない場合、クロック名情報は空 (NONE) になります。

次の図に、[Unconstrained Paths] セクションの例を示します。

Path Group	From Clock	To Clock
(none)		
(none)	clkfbout	
(none)	cpuClk	
(none)	gt0_txusrclk_j	
(none)	gt2_txusrclk_j	
(none)	gt4_txusrclk_j	
(none)	gt6_txusrclk_j	
(none)	phyClk0	
(none)	phyClk1	
(none)	sysClk	

図 20 : タイミング サマリ レポートの [Unconstrained Paths] セクション


タイミング パスの詳細の確認

ほとんどのセクションは、展開してクロック ペアごとのパスの情報を表示できます。[SETUP]、[HOLD]、[Pulse Width] サブセクションには、N 個のワースト パスが表示されます。これらのパスのいずれかを選択すると、その詳細が [Path Properties] ビューの [Report] タブに表示されます。パスをダブルクリックすると、同じ詳細情報が新しいビューに表示されます。タイミング パスの詳細については、第 4 章「[タイミング解析](#)」を参照してください。

パスを右クリックして次のコマンドをクリックすると、パスの解析に有益です。

- [Schematic] : パスの回路図を表示します。
- [Report Timing on Source to Destination] : パスのタイミング解析を再実行します。
- [Highlight] : [Device] および [Schematic] ビューでパスをハイライトします。

違反のあるパスのみを表示

タイミングが満たされていないパスのスラック値は、レポートで赤で表示されます。ツールバーの [Show only failing checks]  をクリックしてオンにすると、これらの違反のみを表示できます。次の図に、[Timing] ビューで違反のあるパスのみを表示した例を示します。

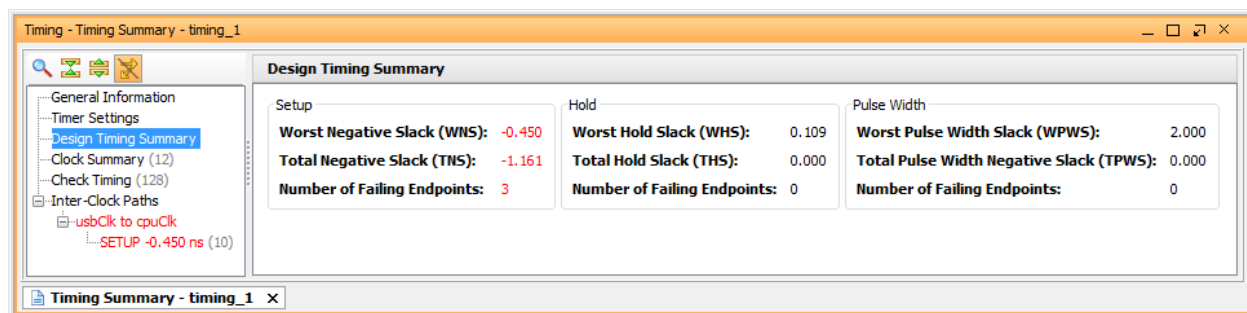


図 21: タイミング サマリレポートで違反のあるパスのみを表示

クロック ネットワーク レポート

[Report Clock Networks] を実行すると、デザインのクロックがツリー表示で示されます。

- I/O ポートからロードまで
- ユーザー定義または自動的に定義されたクロック名を表示
- 完全なツリーの詳細は GUI のみで表示。このレポートのテキスト版には、クロック ルートの名前のみが表示されます。
- BUFG を駆動する BUFG の検索に使用
- クロック ロードでないものを駆動するクロックを検索

各プライマリ クロックおよび生成クロック (デザインで定義されている場合) を含むフォルダーが表示されます。フォルダーが別の場合、制約が適用されていないクロック ルートを示します。

各ツリーにはソースからエンドポイントまでのクロック ネットワークが示され、エンドポイントのタイプごと並べられています。

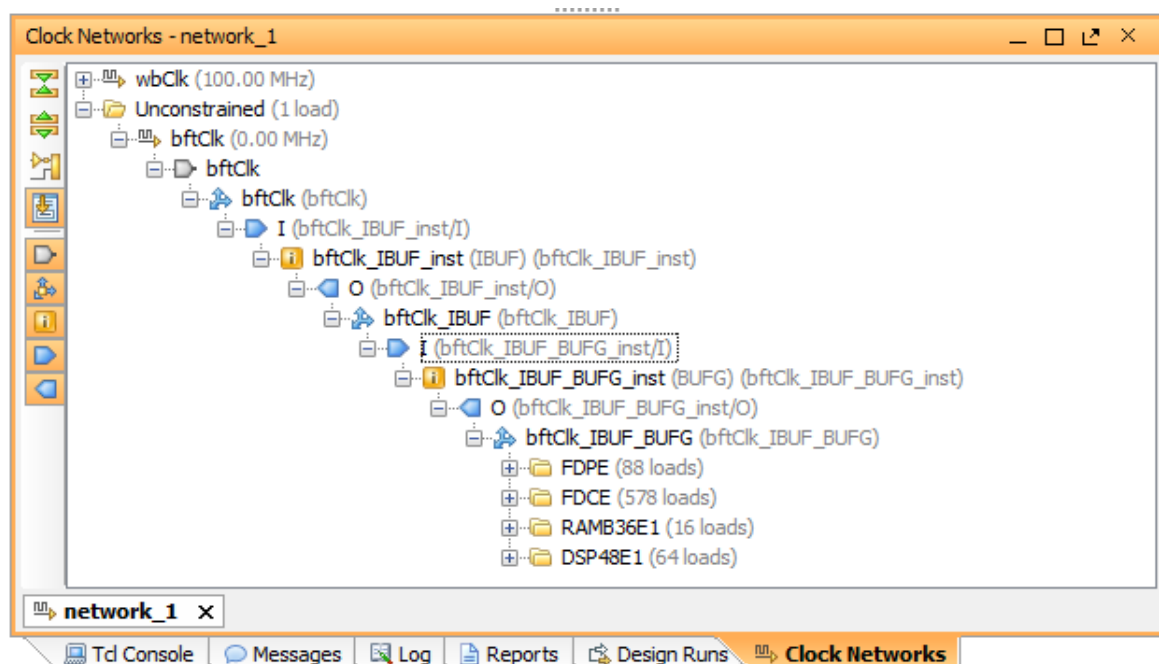


図 22：クロック ネットワーク レポート

ツールバーの [Filter Ports]、[Filter Nets]、[Filter Instance]、[Filter Input Pins]、[Filter Output Pins] ボタンを使用して、クロック ツリーに表示される情報を削減できます。ツリーでオブジェクトを右クリックし、[Trace to Source] をクリックすると、クロック パスの回路図が生成されます。

Tcl コンソールからこのレポートを生成して GUI に表示するには、次のコマンドを使用します。

```
report_clock_networks -name {network_1}
```

クロックの相互関係レポート

[Report Clock Interaction] コマンドは、次を実行します。

- クロック間のタイミング パスを解析します。
- 結果をグリッド形式で表示します。
 - 緑：完全に制約
 - 黒：パスなし
 - 赤：クロックドメイン間のすべてのパスにユーザー フォルス パス制約が設定されている
 - 黄色：クロックドメイン間の一部のパスにユーザー フォルス パス制約が設定されている
- データ損失やメタステーブル状態などの問題を特定するのに役立ちます。

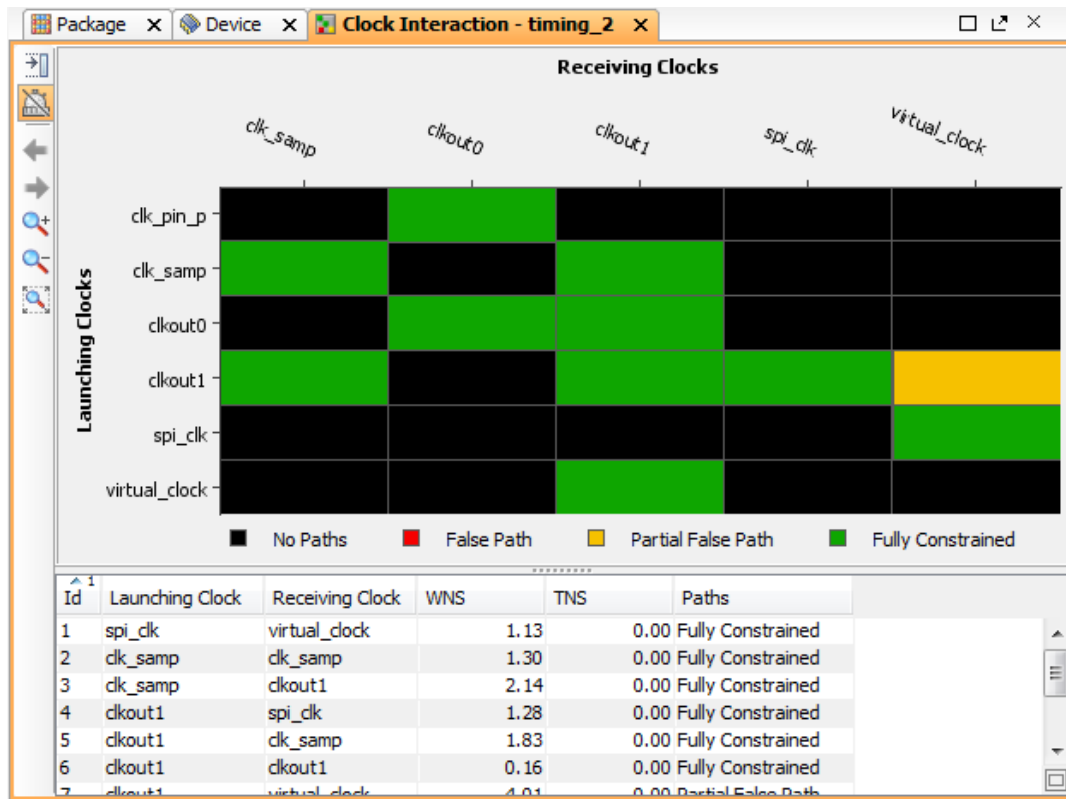


図 23: クロックの相互関係レポート

Tcl でクロック相互関係レポートを作成し、Vivado IDE で表示するには、次のコマンドを使用します。

```
report_clock_interaction -delay_type max -significant_digits 2 -name timing_1
```

パルス幅レポート

[Report Pulse Width] コマンドは、各インスタンス クロック ピンで最小周期、最大周期、High パルス時間、Low パルス時間の要件が満たされているかどうかをチェックします。また、PCIE クロックなど、インプリメント済みデザインの同じインスタンスの 2 つのクロック ピン間で最大スキュー要件が満たされているかもチェックします。

注記: ISE では、このチェックはコンポーネント スイッチ制限と呼ばれています。

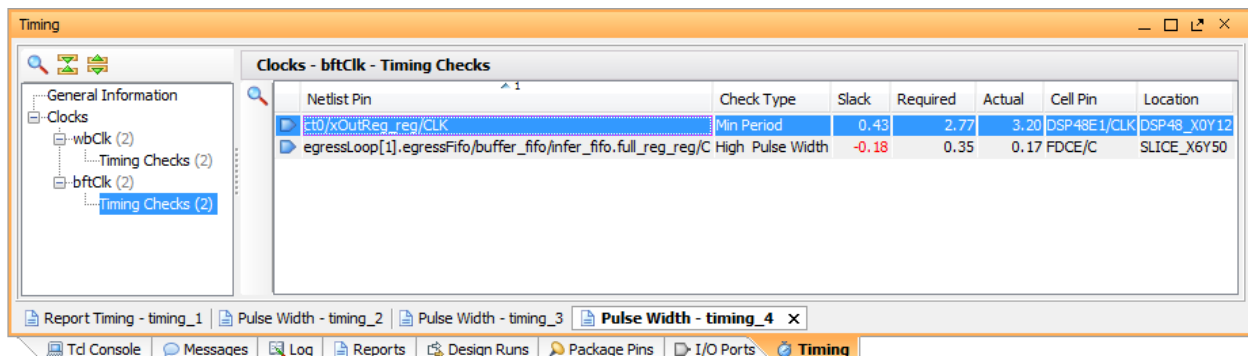


図 24: パルス幅レポート

Tcl コンソールでこのレポートを生成するには、次のコマンドを使用します。

```
report_pulse_width -name timing_1
```

セッション レポート

vivado.exe プロセスでは、プロセスを起動したディレクトリに vivado.log および vivado.jou ファイルが作成されます。vivado.exe プロセスが開始すると、Vivado IDE が起動する前に LOG および JOURNAL ファイルが初期化されます。LOG および JOURNAL ファイルは、[File] → [Open Log File] および [File] → [Open Journal File] をクリックするといつでも表示できます。

タイミング レポート

[Report Timing] を使用すると、合成後の任意の段階で、特定のタイミング パスを解析できます。タイミング サマリ レポートでレポートされたタイミング問題を調査する場合や、特定のタイミング制約の有効性および適用範囲を確認する場合に使用できます。[Report Timing] にはパルス幅レポートは含まれません。

デザインがメモリに読み込まれていれば、Vivado IDE のさまざまな場所から [Report Timing] を実行できます。

- メニュー：[Tools] → [Timing] → [Report Timing] をクリックします。
- クロック相互関係レポートから：クロック ペアを右クリックし、[Report Timing] をクリックして選択したクロック間のレポートを生成します。
- タイミング レポートおよびタイミング サマリ レポートのパスリストから：クロック ペアを右クリックし、[Report Timing] をクリックして選択したクロック間のレポートを生成します。

これに相当する Tcl コマンドは report_timing です。[Report Timing] ダイアログ ボックスでオプションを設定すると、[Command] フィールドに同等の report_timing コマンド構文が表示されます。このコマンド構文は、実行後に [Tcl Console] ビューにも表示されます。次のセクションで、ダイアログ ボックスの各オプションについて説明し、同等の report_timing コマンドのオプションを示します。

[Report Timing] ダイアログ ボックスのオプションは、[Report Timing Summary] ダイアログ ボックスのオプションとほぼ同じですが、追加のオプションがいくつかあります。

[Targets] タブ

特定のパスまたはパスのグループのレポートを生成するため、いくつかのフィルター オプションがあります。これらのフィルターは、タイミング パスの構造に基づきます。

- [Start Points] ([From])：シーケンシャル セル クロック ピン、シーケンシャル セル、入力ポート、双方向ポート、ソース クロックなど、開始点をリストします。複数の開始点をリストした場合、それらのネットリスト オブジェクトのいずれかから開始するすべてのパスがレポートされます。[Transition] では、特定のソース クロック エッジ (立ち上がり/立ち下がり) を選択します。

同等 Tcl オプション：-from、-rise_from、-fall_from

- [Through Point Groups] ([Through])：パスが通過するピン、ポート、組み合わせセル、またはネットのリストを指定します。複数のネットリスト オブジェクトをリストすると、そのいずれかを通過するパスがレポートされます。複数の通過点をすべて通過するパスをレポートするには、[More] ボタンをクリックして複数の [Through] オプションを指定します。[Transition] では、特定のデータ エッジ (立ち上がり/立ち下がり) を選択します。

推奨：[Transition] はデフォルトのままにします。

同等 Tcl オプション：-through、-rise_through、-fall_through

- **[End Points] ([To])**: シーケンシャル セルの入力データ ピン、シーケンシャル セル、出力ポート、双方向ポート、デスティネーション クロックなど、終点をリストします。複数の終点をリストした場合、それらのネットリスト オブジェクトのいずれかで終了するすべてのパスがレポートされます。**[Transition]** では、通常特定のデータ エッジ (立ち上がり/立ち下がり) を選択しますが、デスティネーション クロックを指定した場合は、特定のクロック エッジを選択します。

同等 Tcl オプション: `-to`、`-rise_to`、`-fall_to`

次の図に、**[Report Timing]** ダイアログ ボックスの **[Targets]** タブを示します。この例では、`cpuClk` の立ち上がりクロック エッジから、`u4/data[*]` ネットのいずれかを通過し、`fftClk` または `sysClk` の立ち下がりエッジで終了するパスを指定しています。

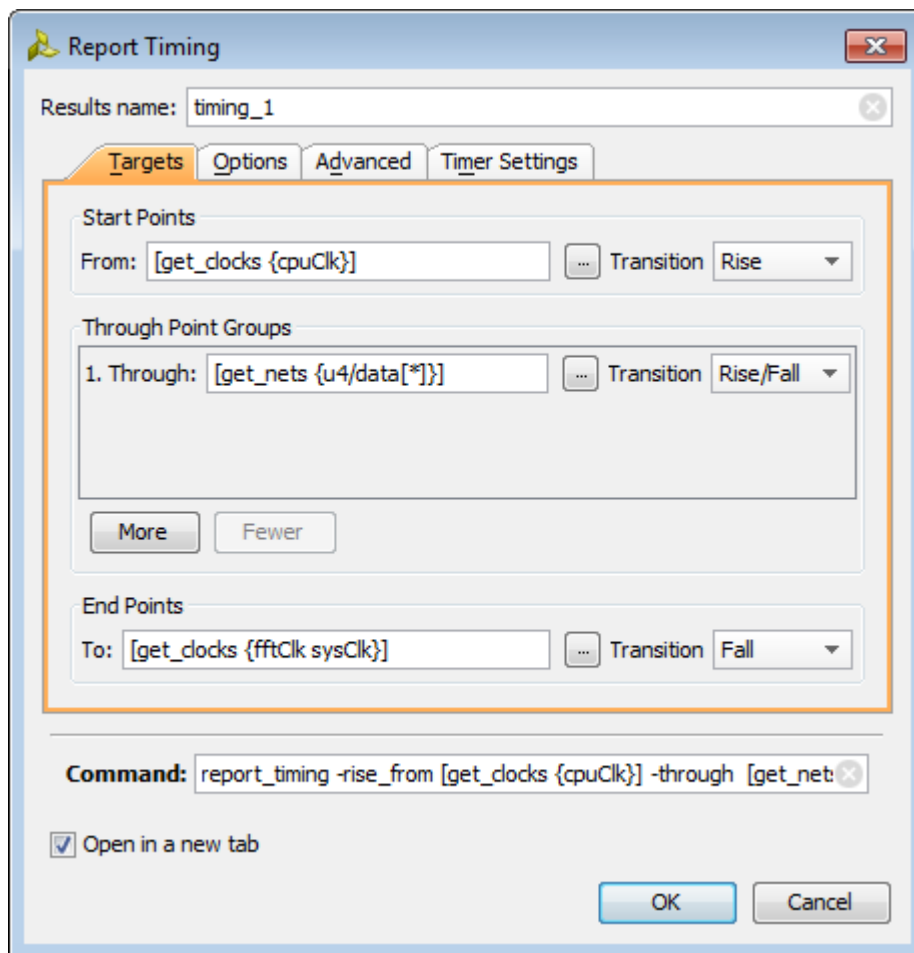


図 25 : **[Report Timing]** ダイアログ ボックスの **[Targets]** タブ

[Options] タブ

次のコマンドが含まれています。

- [Report]
 - [Path delay type] : 21 ページの「[タイミング サマリ レポート](#)」を参照してください。
 - [Do not report unconstrained paths:] : デフォルトでは、[Targets] タブで指定されたパスに制約されているものがない場合、制約されていないパスがレポートされます。制約されていないパスをレポートに含めないようにする場合は、このオプションをオンにします。
同等 Tcl オプション : `-no_report_unconstrained`
- [Path Limits]
 - [Number of paths per group] : 21 ページの「[タイミング サマリ レポート](#)」を参照してください。
 - [Number of paths per endpoint] : 21 ページの「[タイミング サマリ レポート](#)」を参照してください。
 - [Limit paths to group] : [Targets] タブで指定した条件に一致するパスで、指定したタイミング パス グループに属するもののみをレポートします。各クロックは、グループに関連付けられています。Vivado IDE タイミング エンジンでは、デフォルトでいくつかのグループが作成されます。たとえば、`**async_default**` はリカバリまたはリムーバブル タイミング チェックで終了するすべてのパスのグループです。
同等 Tcl オプション : `-group`
- [Path Display]
 - [Display paths with slack greater than] : スラックが指定した値より大きいパスをレポートします。
同等 Tcl オプション : `-slack_greater_than`
 - [Display paths with slack less than] : 23 ページの「[タイミング サマリ レポート](#)」を参照してください。
 - [Number of significant digits] : 23 ページの「[タイミング サマリ レポート](#)」を参照してください。
 - [Sort paths by] : レポートされたパスをグループごとに表示するか (デフォルト)、スラック順に表示するかを選択します。[group] を選択すると、各グループおよび各解析タイプ (`-delay_type min/max/min_max`) に対して N 個のワースト パスがレポートされます。グループはワースト パスを含むものから順に並べられるので、最悪の違反を含むグループがリストの一番上に表示されます。[slack] を選択すると、各解析タイプに対して N 個のワースト パスがレポートされ (すべてのグループを統合)、スラック順に並べられます。
同等 Tcl オプション : `-sort_by`

[Advanced] タブ

23 ページの「[タイミング サマリ レポート](#)」で説明されているオプションと同じです。

[Timer Settings] タブ

24 ページの「[タイミング サマリ レポート](#)」で説明されているオプションと同じです。

タイミング パスの詳細の確認

[OK] をクリックするとコマンドが実行され、[Timing] ビューの新しいタブに結果が表示されます。選択した各解析 (min/max/min_max) に対して N 個のワースト パスがレポートされます。次の図に、最大遅延解析および最小遅延解析の両方 (SETUP および HOLD) を選択し、N = 4 に設定した場合のタイミング レポートを示します。

Name	Slack	From	To	Total Delay	Logic Delay	Net %	Stages	Source Clock	Destination Clock
Constrained (12)									
cpuClk (4)									
Path 1	-0.450	usbEngine1/u4/inta_reg/C	cpuEngine/or1200_du/tbar_ram/ramb16_s36_s36/DIAD1[3]	8.250	1.388	83.2	12	usbClk	cpuClk
Path 2	-0.450	usbEngine1/u4/inta_reg/C	cpuEngine/or1200_du/tbar_ram/ramb16_s36_s36/DIAD1[4]	8.250	1.388	83.2	12	usbClk	cpuClk
Path 3	-0.262	usbEngine1/u4/inta_reg/C	cpuEngine/cpu_dbg_dat_o/buf...ance.fifo_ram_reg/DIAD1...	8.062	1.027	87.3	12	usbClk	cpuClk
Path 4	0.000	usbEngine1/u4/inta_reg/C	cpuEngine/cpu_dbg_dat_o/buf...ance.fifo_ram_reg/DIAD1...	7.800	1.027	86.8	12	usbClk	cpuClk
phyClk0 (4)									
Path 49	1.507	LineState_pad_0_i[0]	usbEngine0/LineState_r_reg[0]/D	1.964	0.867	55.9	1	sysClk	phyClk0
Path 50	1.507	LineState_pad_0_i[1]	usbEngine0/LineState_r_reg[1]/D	1.964	0.867	55.9	1	sysClk	phyClk0
Path 51	1.507	VStatus_pad_0_i[0]	usbEngine0/VStatus_r_reg[0]/D	1.964	0.867	55.9	1	sysClk	phyClk0
Path 52	1.507	VStatus_pad_0_i[1]	usbEngine0/VStatus_r_reg[1]/D	1.964	0.867	55.9	1	sysClk	phyClk0
Path 53	1.507	LineState_pad_1_i[0]	usbEngine1/LineState_r_reg[0]/D	1.964	0.867	55.9	1	sysClk	phyClk1

図 26 : タイミング レポート

これらのパスのいずれかを選択すると、その詳細が [Path Properties] ビューの [Report] タブに表示されます。次の図に、[Path Properties] ビューを示します。

Summary	
Name	Path 1
Slack	-0.450ns
Source	usbEngine1/u4/inta_reg/C (rising edge-triggered)
Destination	cpuEngine/or1200_du/tbar_ram/ramb16_s36_s36
Path Group	cpuClk
Path Type	Max at Slow Process Corner
Requirement	8.000ns
Data Path Delay	8.250ns (logic 1.388ns (16.824%) route 6.862ns (83.176%))
Logic Levels	12 (LUT2=1 LUT4=1 LUT5=2 LUT6=8)
Clock Path Skew	-0.010ns
Clock Uncertainty	-0.189ns
Source Clock Path	
Delay Type	Delay
(clock usbClk rise edge)	(r) 8.000
	(r) 0.000
net (fo=0)	0.000


図 27 : タイミング パスのプロパティ

パスをダブルクリックすると、同じ詳細情報が新しいビューに表示されます。タイミング パスの詳細については、第 4 章「タイミング解析」を参照してください。上図は、[Path Properties] ビューの [Report] タブを示しています。

パスを右クリックして次のコマンドをクリックすると、パスの解析に有益です。

- [Schematic] : タイミング パスを回路図で表示します。
- [Report Timing on Source to Destination] : パスのタイミング解析を再実行します。
- [Highlight] : [Device] および [Schematic] ビューでパスをハイライトします。

違反のあるパスのみを表示

タイミングが満たされていないパスのスラック値は、レポートで赤で表示されます。ツールバーの [Show only failing checks]  をクリックしてオンにすると、これらの違反のみを表示できます。

ロジック解析

この章では、次の情報を示します。

- ネットリストに含まれるエレメント
- ネットリストの接続性
- ゲートおよびプリミティブ タイプの検索
- チップでのデザインフロー
- 使用率
- DRC を使用した一般的なデザインの問題

RTL 解析

RTL 解析の詳細は、『Vivado Design Suite ユーザー ガイド：システム レベル デザイン入力』(UG895) を参照してください。

[Netlist] ビュー

[Netlist] ビューには、合成ツールで処理されたネットリストでのデザイン階層が表示されます。

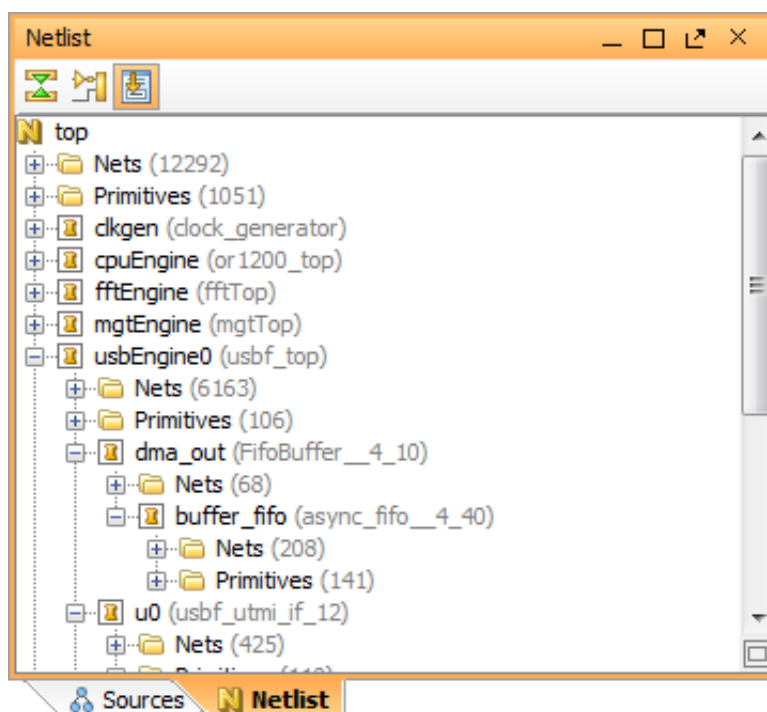


図 28 : [Netlist] ビュー

合成設定によって、ネットリストの階層が元の RTL と 100% 同じであったり、階層がなかったりします。一般的には、合成ツールはデフォルトではほとんどのユーザー階層を保持しながらエッジを最適化し、小型で高速のネットリストを作成します。合成ツールのデフォルトを使用すると、ネットリストの階層は認識できますが、階層へのインターフェイスが変更されている場合があります。一部のポートや階層レベルがない場合もあります。

階層の各レベルは、その階層ツリーを示します。各レベルに、次のものが表示されます。

- そのレベルのすべてのネットを含む [Nets] フォルダー
- そのレベルのすべてのプリミティブを含む [Primitives] フォルダー
- そのレベルにインスタンス化されている階層

ツリーをたどると、すべてのブランチを表示できます。セルおよびプリミティブの横に表示されているアイコンは、デザインの状態を示します。

詳細は、『Vivado Design Suite ユーザー ガイド : Vivado IDE の使用』(UG893) の「[Netlist] ビュー」を参照してください。

階層の各レベルの [Properties] ビューの [Statistics] タブには、次のようなリソース使用率統計が表示されます。

- その階層ブランチ全体のプリミティブの使用状況
- 階層の境界を横切るネットの数
- 階層で使用するクロック

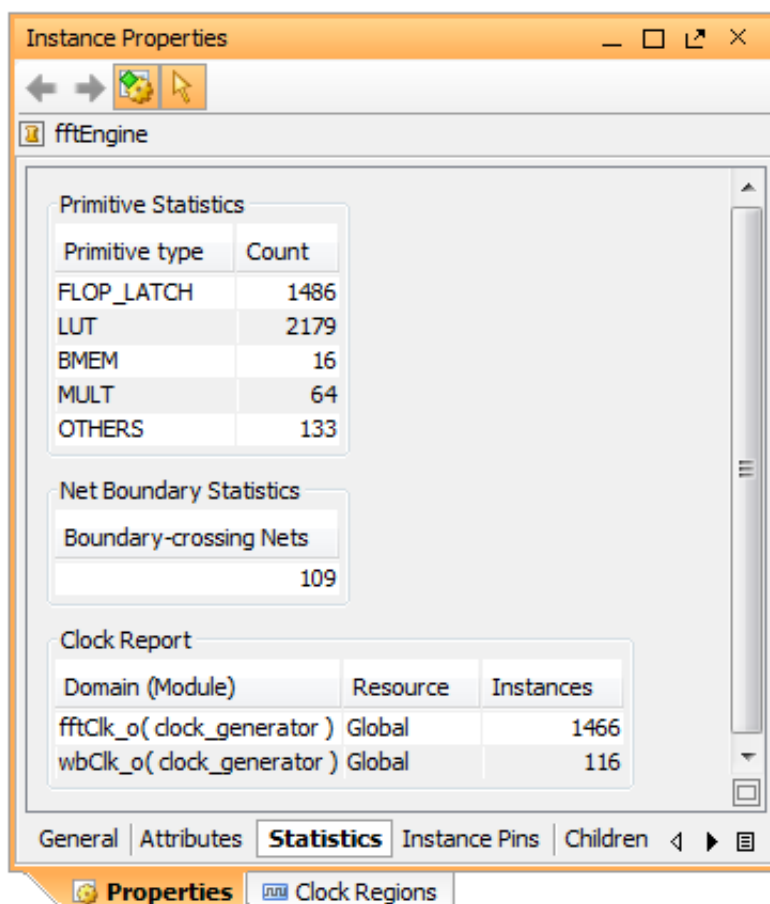


図 29 : [Instance Properties] ビュー

デザインをフロアプランする場合、Pblock に対して同様のプロパティが表示されます。

[Hierarchy] ビュー

このビューでは、階層のリソースの使用量を理解できます。[Hierarchy] ビューを開くには、[Tools] → [Show Hierarchy] をクリックします。

[Hierarchy] ビューには、ネットリストの階層ツリーが表示されます。各行は、ネットリスト内の階層レベルを示します。下に行くほど、ネットリストの下位階層になります。各階層レベルは、そのレベルのほかの階層に相対したサイズで表示されます。

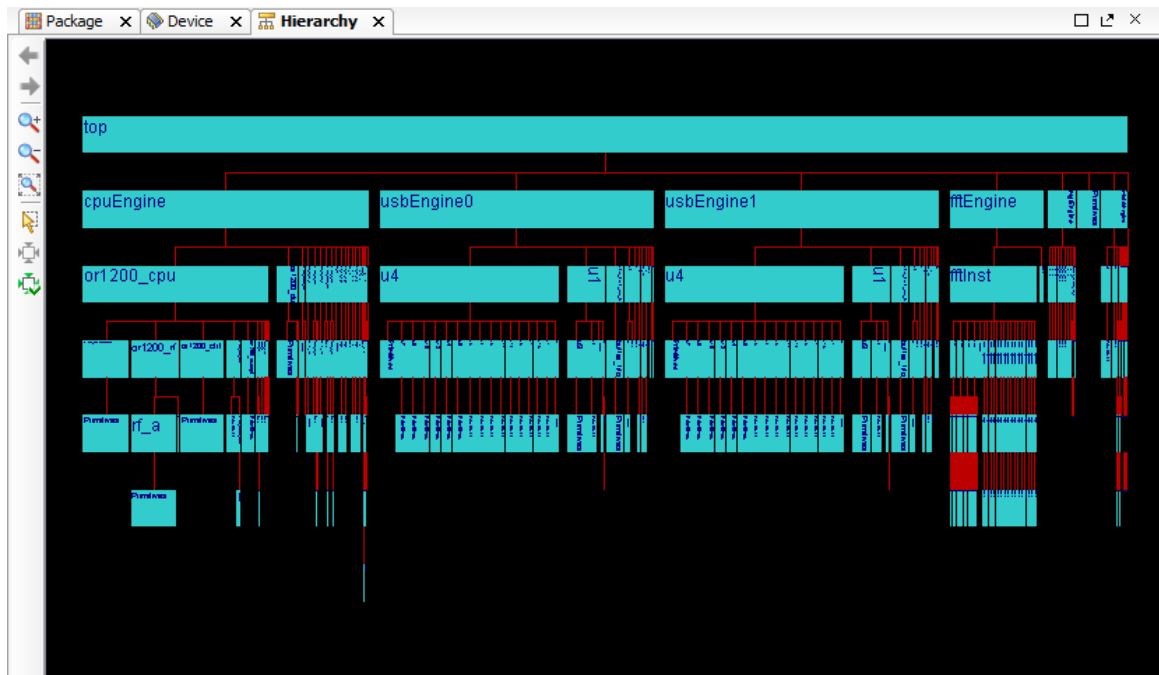


図 30 : [Hierarchy] ビュー

上図の `cpuEngine`、`usbEngine0`、および `usbEngine1` は、次のようになっています。

- デザインのほとんどのロジックを占有します。
- どれもおよそ同じ数のリソースを使用します。

リソース使用率レポートには、次のような特徴があります。

- デザインをリソース タイプごとに表示します。
- 各リソース タイプ別に階層レベルごとの使用率を表示します。

リソース使用率レポートを表示するには、[Tools] → [Report Utilization] をクリックします。

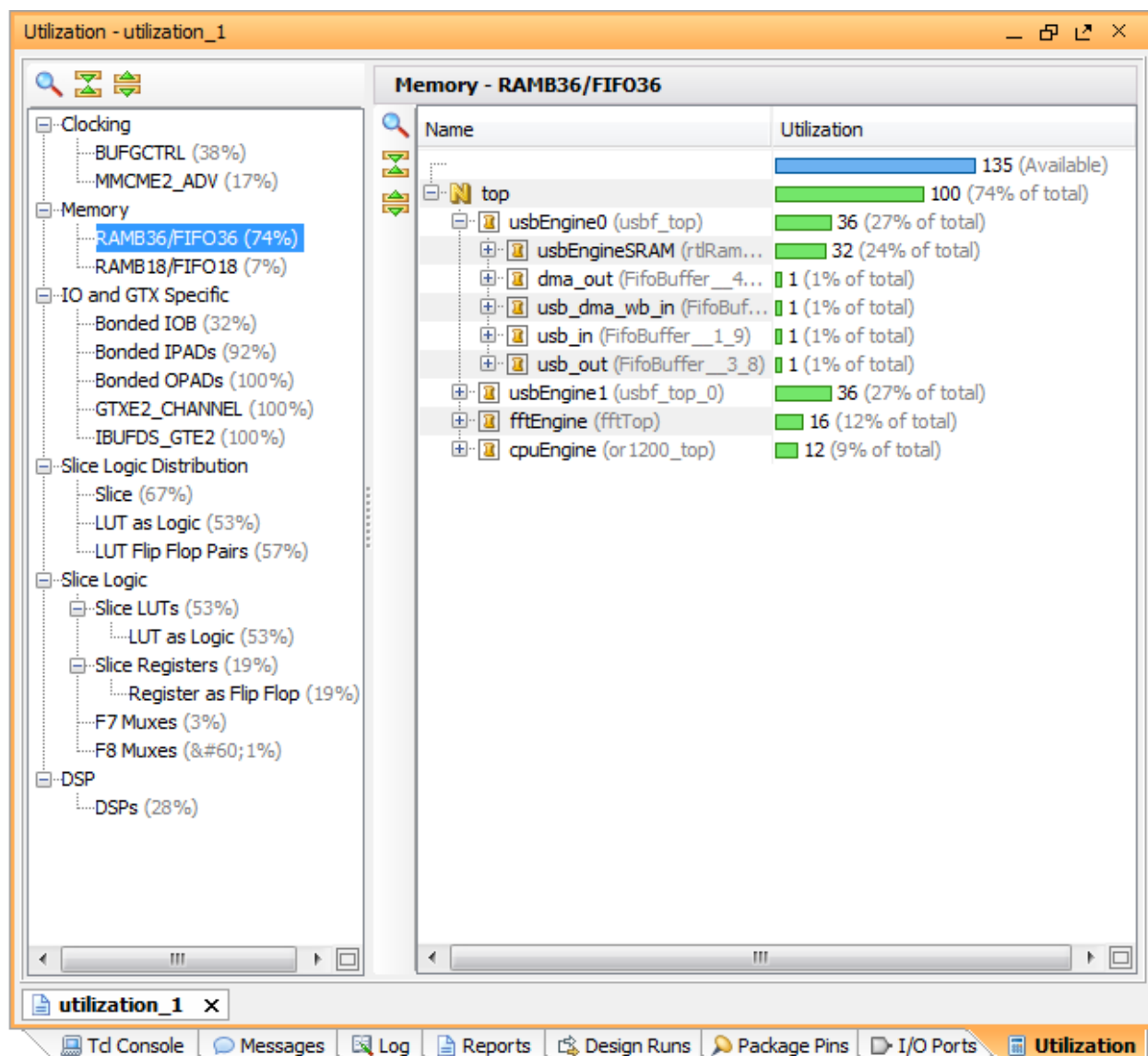


図 31 : リソース使用率レポート

このデザインでは、2 つの usbEngine ブロックにより RAMB36 および FIFO36 ブロックの大半が消費されています。プラス記号 (+) をクリックすると、下位階層でのリソース使用率を表示できます。

回路図の表示

回路図は、ネットリストのグラフィカル表示です。回路図を表示すると、次を実行できます。

- ネットリストをグラフィカルに表示します。
- ゲート、階層、および接続を確認します。
- ロジック コーンをたどったり展開したりします。
- デザインを解析します。
- デザインの内部をより理解しやすくなります。

エラボレート済みデザインの RTL レベルでは、ツールでコードがどのように解釈されたかを確認できます。合成済みデザインおよびインプリメント済みデザインでは、合成ツールで生成されたゲートを確認できます。

回路図を表示するには、[Tools] → [Schematic] をクリックします。オブジェクトを選択していない場合は、デザインの最上位のゲート、階層、および接続が表示されます。

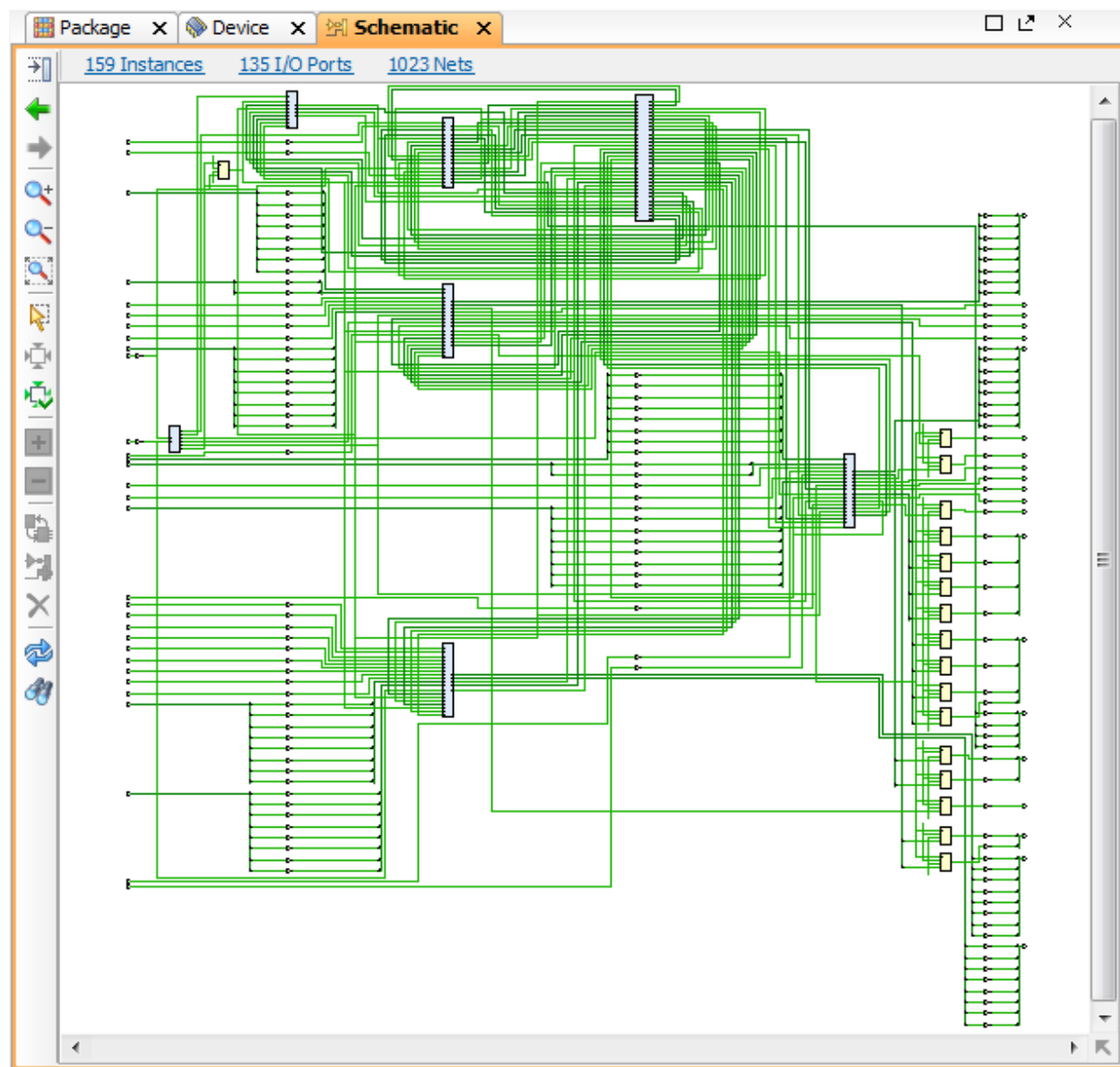


図 32 : 最上位回路図

回路図の拡大/縮小および移動方法の詳細は、『Vivado Design Suite ユーザー ガイド : Vivado IDE の使用』(UG893) を参照してください。

ヒント : 1 つの階層レベルを選択すると、回路図がシンプルになります。選択したエレメントが青色でハイライトされ、1 つの階層のポートが表示されます。

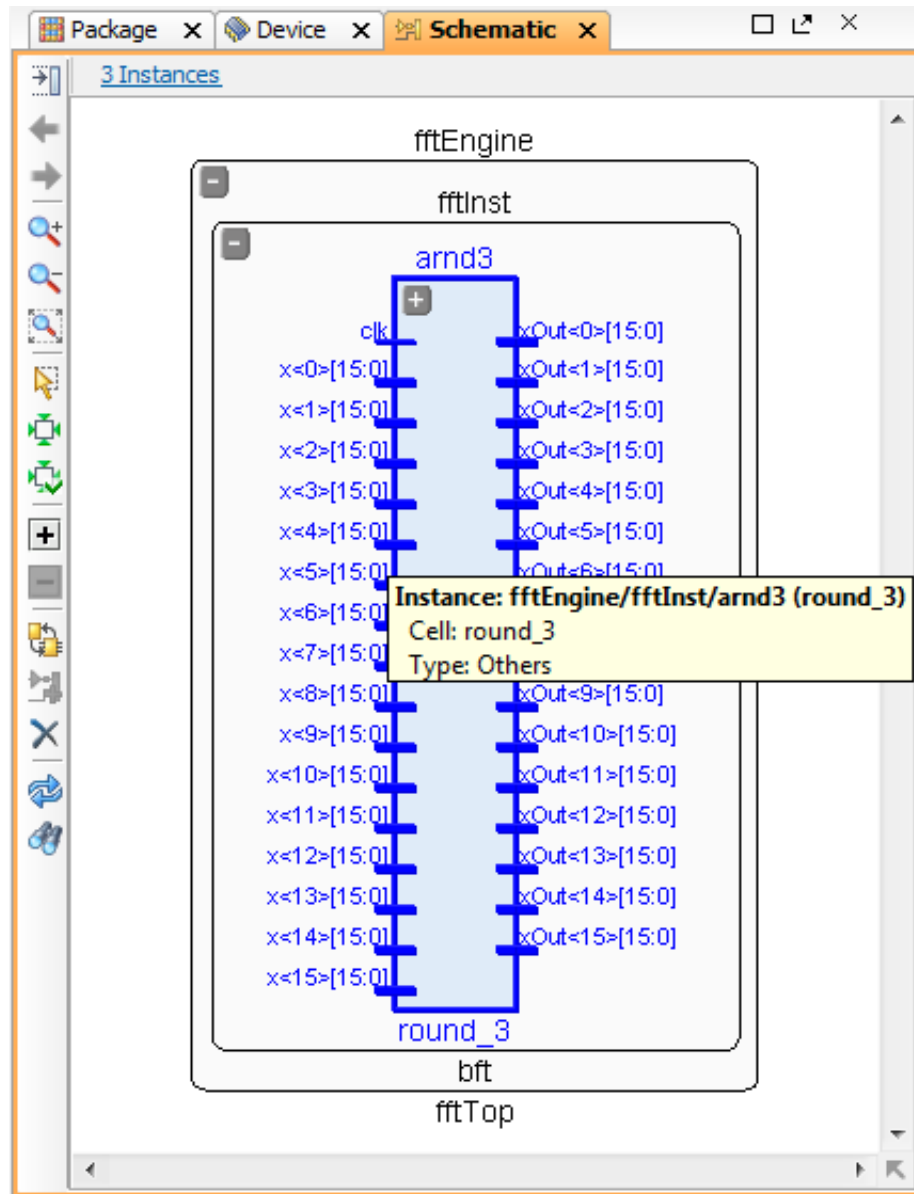


図 33 : 1 つの階層を選択した場合の回路図

回路図では、次の操作を実行できます。

- 階層の左上にある [+] をクリックしてゲートを表示します。
- ポートまたはエレメントをダブルクリックして展開します。
- ポップアップメニューを使用します。

詳細は、『Vivado Design Suite ユーザー ガイド : Vivado IDE の使用』(UG893) の「[Schematic] ビュー」を参照してください。

- 前の表示に戻ったり次の表示に進んだりするには、ツールバーの [Previous schematic] および [Next schematic] ボタンをクリックします。
- ツールバーの [Expand all logic inside selected instance] ボタンをクリックして、ロジックおよび接続をより詳細に表示します。
- ツールバーの [Collapse all logic inside selected instance] をクリックして、回路図を簡略化します。

インプリメンテーション後にタイミングパスのゲートを視覚的に確認するには、回路図が最も簡単な方法です。パスを選択して回路図を開くと、そのパスのゲートとネットが表示されます。

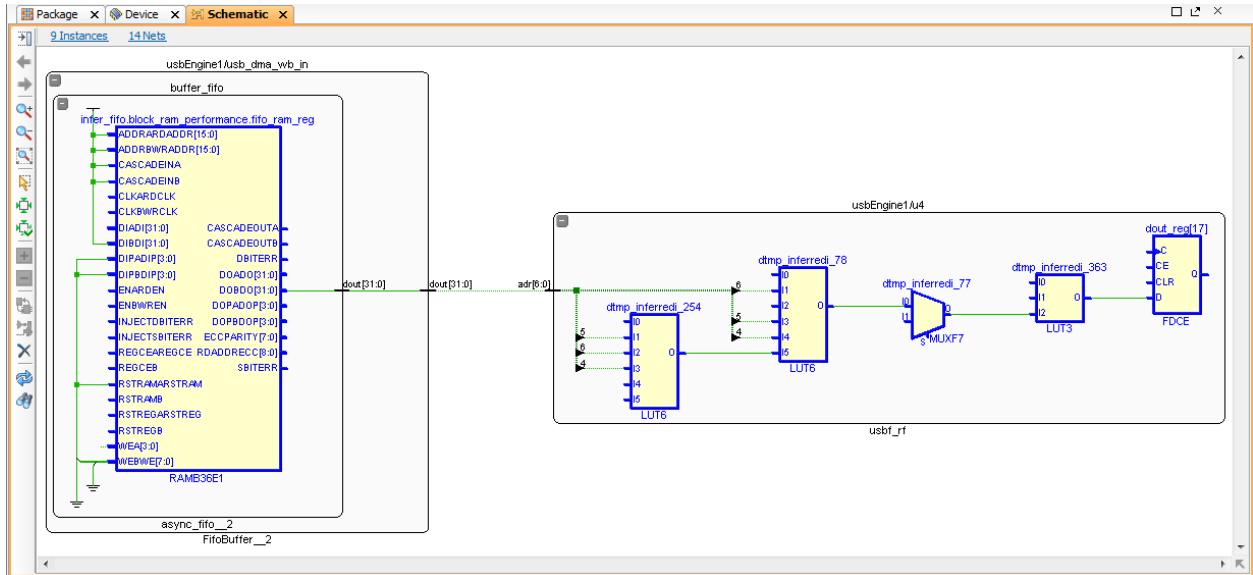


図 34 : タイミングパスの回路図

回路図で関連のある階層レベルを特定するには、ポップアップメニューから [Select Primitive Parents] をクリックします。

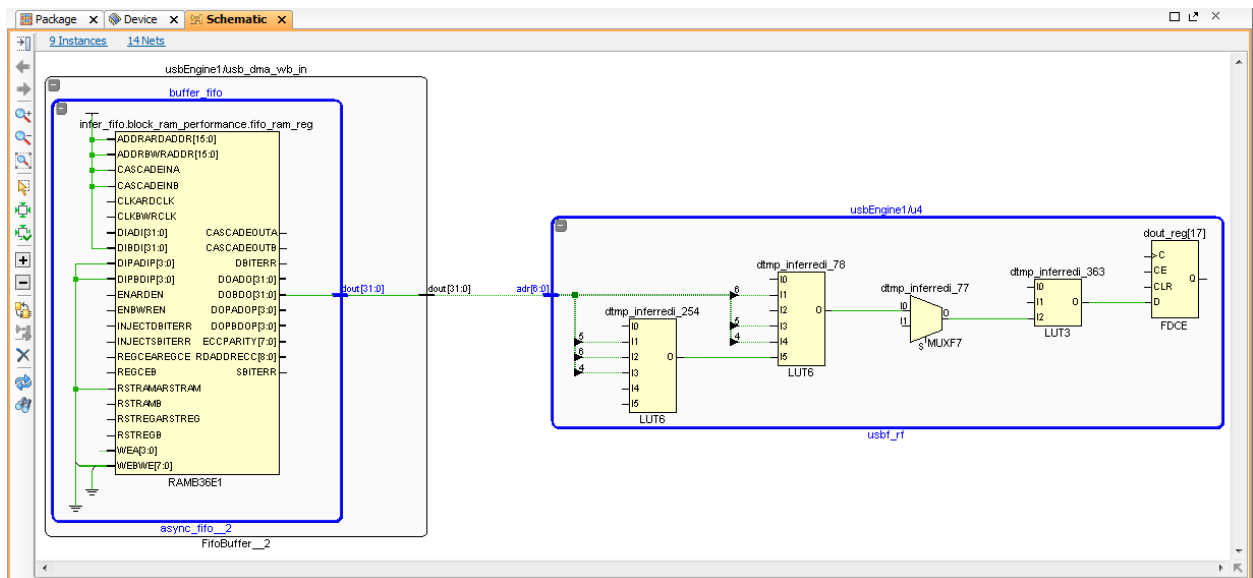


図 35 : タイミングパスの回路図でプリミティブの親を選択

[Highlight] および [Mark] コマンドを使用すると、関連のゲートが見やすくなります。[Highlight] および [Mark] コマンドを使用してプリミティブを色分け表示すると、どのロジックが元のパスにあるか、どのロジックが追加されたかなどがわかりやすくなります。

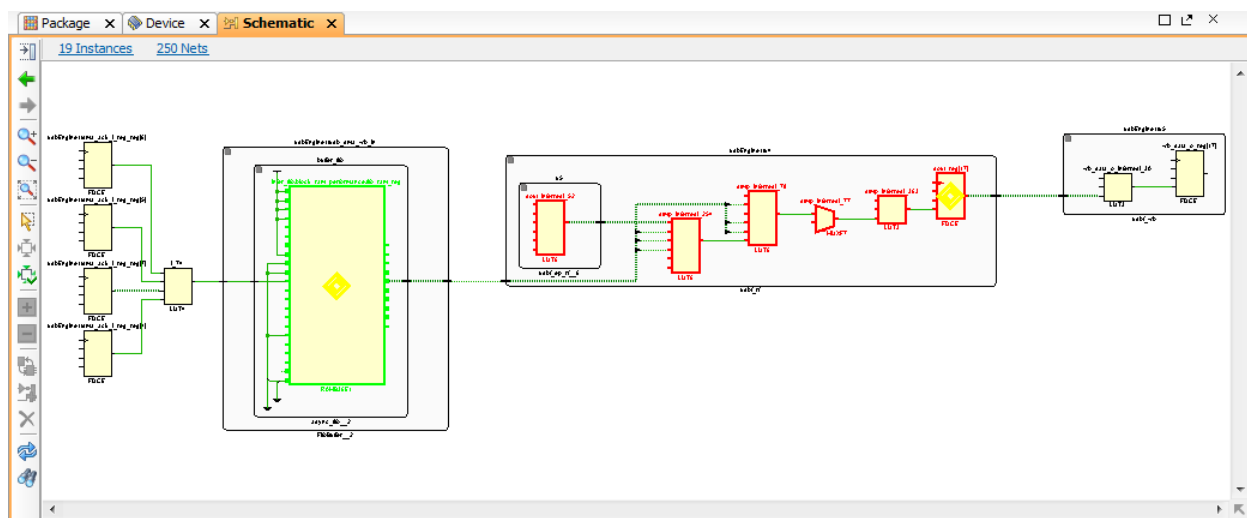


図 36 : 回路図でタイミング パスをマーク

[Find] コマンドを使用したオブジェクトの検索

Vivado™ IDE には、優れた検索機能が含まれています。検索機能を使用するには、[Edit] → [Find] をクリックします。

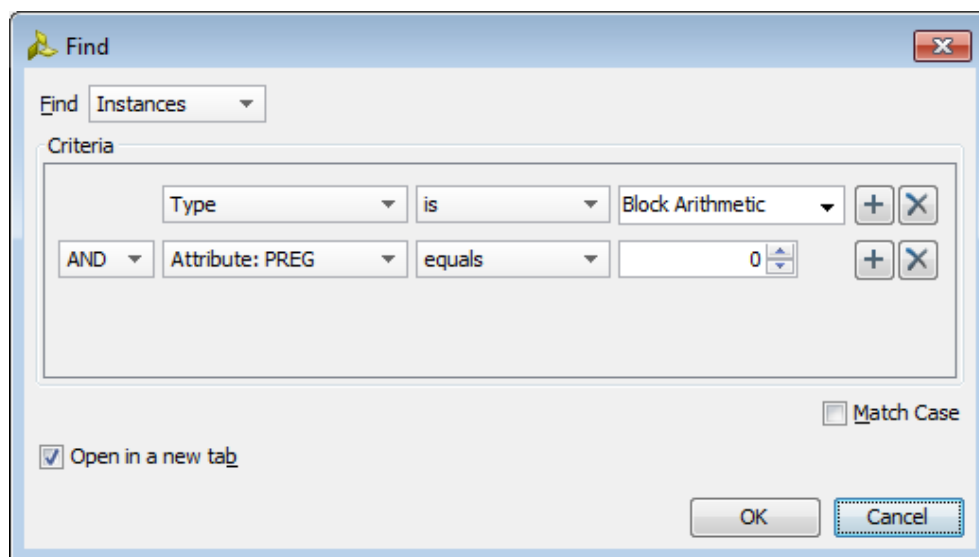


図 37 : [Find] ダイアログ ボックス

検索オブジェクト

[Edit] → [Find] を使用して、ネットリストで次のオブジェクトを検索できます。

- インスタンス
- ネット
- I/O ポート
- インスタンス ピン
- Pblock
- RPM

検索条件

各オブジェクトの検索では、複数の検索条件を指定できます。たとえば、インスタンスでは次の検索条件を指定できます。

- タイプ
- セル タイプ
- ブラック ボックス
- 次のようなプリミティブ
 - I/O バッファ
 - 演算ブロック
 - ブロック メモリ
 - LUT
- 名前
- ステータス
- 親 Pblock
- モジュール
- プリミティブ数
- 属性

ほかにもどんな検索条件があるか確認してみてください。

デバイス特定の検索条件

デバイス特定の検索条件は、次のとおりです。

- アーク
- ノード
- BEL ピン
- BEL
- サイト ピン
- サイト
- タイル
- I/O バンク
- クロック領域

検索例

[Edit] → [Find] を使用すると、たとえば次のものを検索できます。

- 未配置の I/O
- ツールで配置されたグローバル クロック
- ファンアウトが 10,000 以上のネット
- PREG エンベデッド レジスタを使用する DSP

複数の条件を使用した検索

複数の条件を指定して検索を実行するには、次の手順に従います。

1. 最初の検索条件を指定します。
2. [+] をクリックします。
3. 検索条件を追加します。
4. 検索条件の演算子 (AND または OR) を選択します。

Tcl での検索

Tcl コンソールまたはスクリプトで実行している場合、検索には `get_*` コマンドを使用します。

詳細は、『Vivado Design Suite ユーザー ガイド : Tcl スクリプト機能の使用』(UG894) を参照してください。

デザインのデータフロー (最上位フロアプラン)

RTL をデザインに統合する際、デバイス内のデザインをグラフィカル表示すると便利です。合成後のブロック間の接続や I/O ピン配置をグラフィカル表示で確認すると、デザインの理解に役立ちます。

インターコネクトを表示するには、階層の上位に Pblock を使用して最上位フロアプランを生成します。最上位 RTL を Pblock に分割するには、[Tools] → [Floorplanning] → [Auto Create Pblocks] をクリックします。

Pblock をデバイスに配置するには、[Tools] → [Floorplanning] → [Place Pblocks] をクリックします。スライス数および使用率に基づいて、Pblock のサイズが特定されます。

Pblock の使用率は、解析中は 100% を超えていてもかまいませんが、インプリメンテーションでは 100% 以下にする必要があります。Pblock の使用率を高くすると、デバイス上でのサイズが小さくなります。これは、デバイス全体を理解するために有効な手法です。

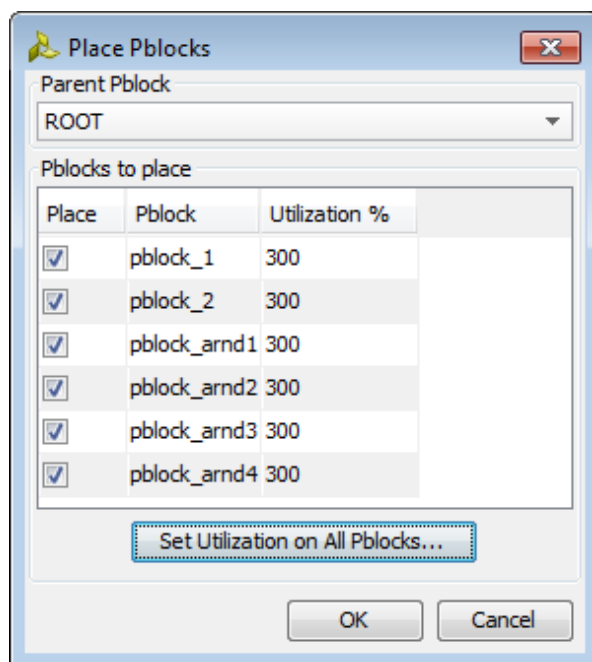


図 38 : [Place Pblocks] ダイアログ ボックス

最上位フロアプランには、ブロックと I/O の接続表示されます (緑色の線)。2 つの Pblock で共有されるネットは、まとめられます。線の太さおよび色は、共有されるネットの数により異なります。次の図に、最上位フロアプランの例を示します。

例 1: データパスを示す最上位フロアプラン

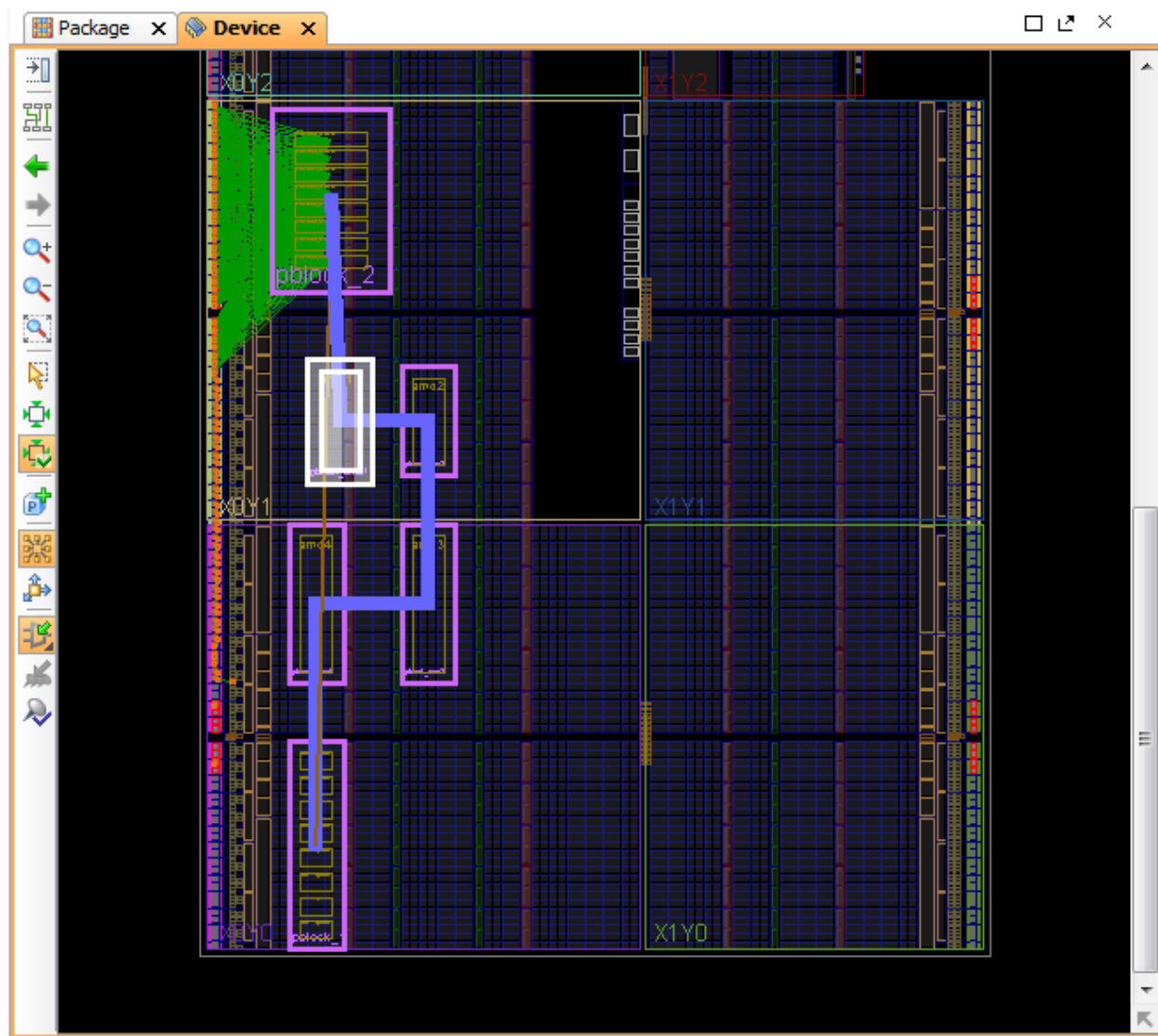


図 39: データパスを示す最上位フロアプラン

上図では、データパスが表示されています。各ブロックは、駆動ブロックとロードブロックのみに接続されています。緑色の線は、1 つのブロックのみが I/O に接続されていることを示しています。これは、フロアプランしやすいデザインです。

例 2 : 制御パスを含むフロアプラン

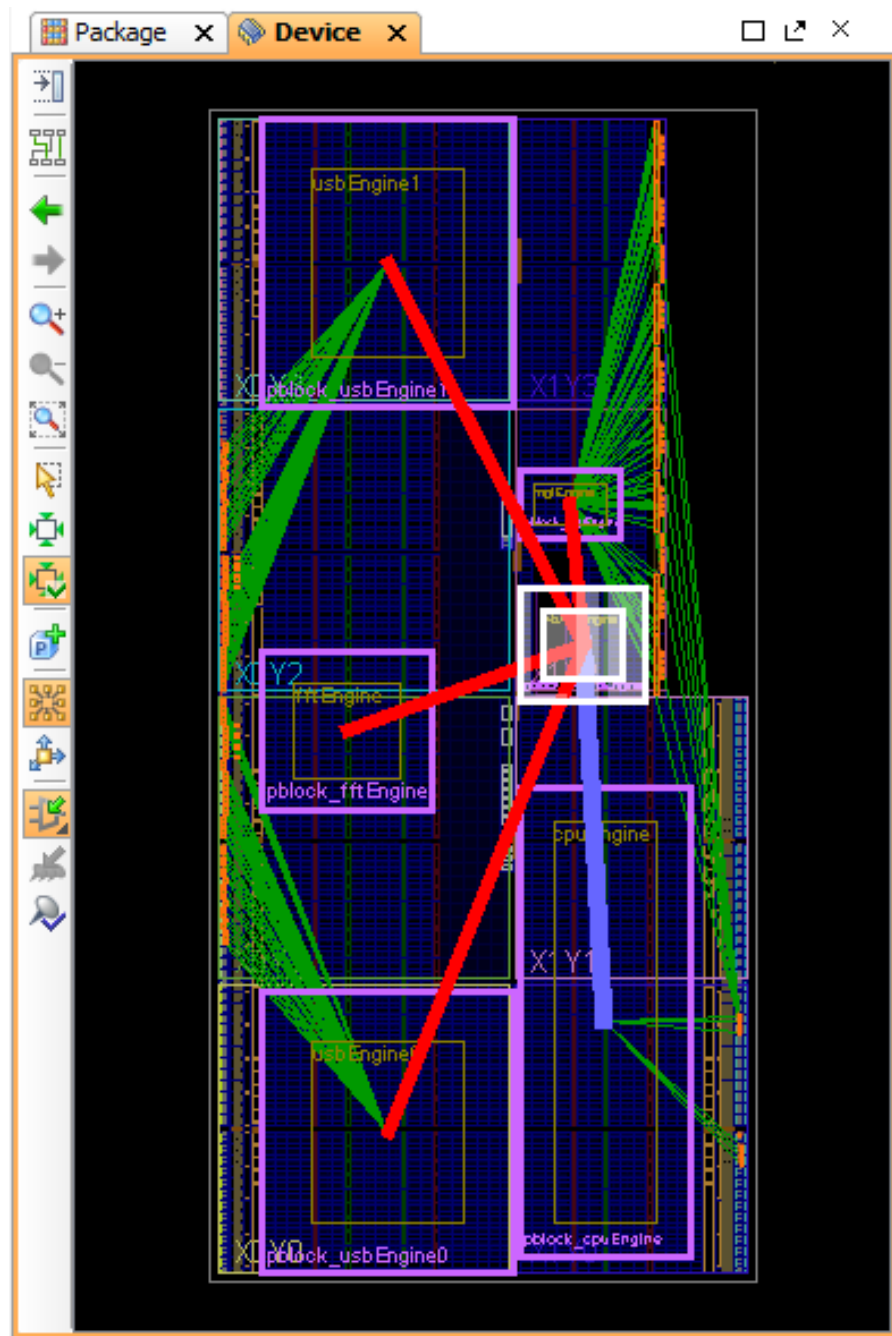


図 40 : 制御パスを含むフロアプラン

上図は、すべてのブロックが中央のブロックと接続されているデザインを示しています。接続は、中央のブロックと右下のブロックの間が最も多くなっています。この中央のブロックを分散させて、その他のロードと通信する必要がある場合があります。これは、フロアプランしにくいデザインです。

フロアプランの確認

フロアプランを確認する際は、デバイス リソースを考慮してください。Pblock のサイズ調整では、次のような特殊なデバイス リソースは考慮されません。

- ブロック RAM
- DSP48
- MGT
- クロック バッファ

フロアプランおよびリソース使用量を考慮してブロックを確認してください。

デバイス使用率統計の解析

インプリメンテーション問題のよくある原因は、ピン配置によるロジックおよびデバイスのレイアウトが考慮されていないことです。ほとんどのデバイスではスライス ロジックは均一ですが、次のような特殊リソースがロジックの配置に影響します。

- I/O
- HP (High Performance) バンク
- HR (High Range) バンク
- MGT
- DSP48
- ブロック RAM
- MMCM
- BUFG
- BUFR

特定の特殊リソースを多数消費するブロックは、デバイス全体に分散する必要がある場合があります。デザインのほかの部分とのインターフェイスを設計する際、このことを考慮してください。次を組み合わせることでブロック リソースを確認します。

- `report_utilization`
- ネットリスト プロパティ
- Pblock プロパティ

DRC

DRC はデザインをチェックし、一般的な問題をレポートします。DRC を実行するには、`report_drc` コマンドを使用します。インプリメンテーションでは、ツールで DRC が実行されます。配置配線後には、より完全で包括的な DRC が実行されます。

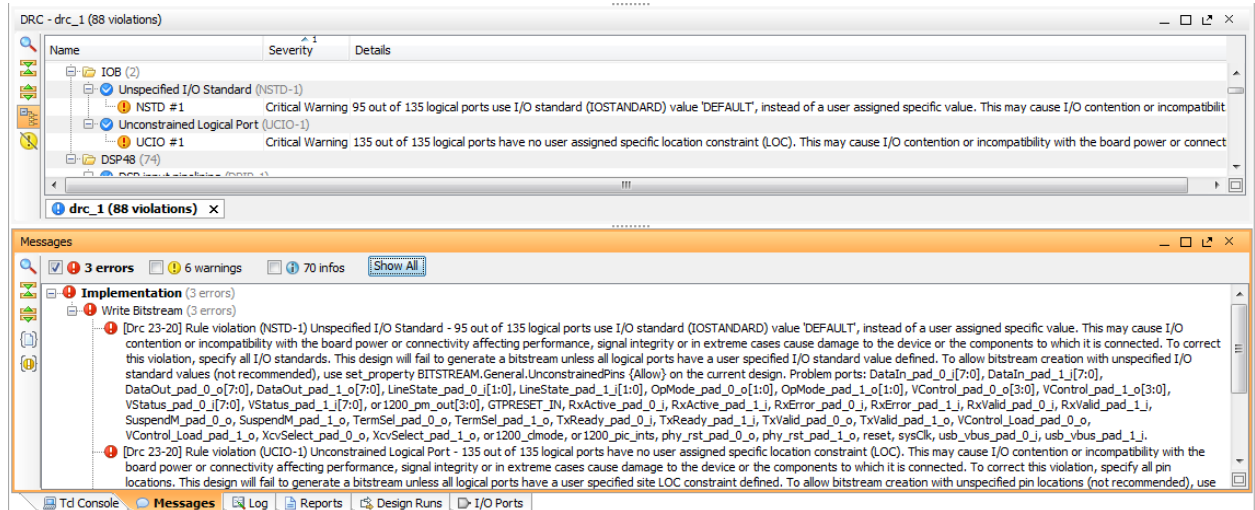


図 41 : DRC によるクリティカル警告およびエラー

推奨 : DRC のクリティカル警告およびエラーをフローの初期段階で確認し、フローの後の方で問題が発生しないようにしてください。

合成済みデザインで [Report DRC] を実行すると、制約の適用されていない I/O に対してクリティカル警告が表示されます。配線済みデザインでも、クリティカル警告がレポートされます。レポートを確認してください。ビットストリーム生成の段階では、同じ DRC でエラーがレポートされます。DRC レポートを早めに確認し、デザインで変更が必要な部分を特定するようにしてください。

複数のデザイン

デザインは、次のもので構成されます。

- ネットリスト
- 制約セット

制約セットとは、複数の XDC 制約ファイルのセットで、新しい制約を保存するファイルが指定されています。

- パーツ

複数のデザインを同時に開くことができます。コードを解析する際、エラボレート済みデザインとインプリメント済みデザインを同時に開いておくとう便な場合があります。フローの同じ段階のデザインを複数開くことも可能です。

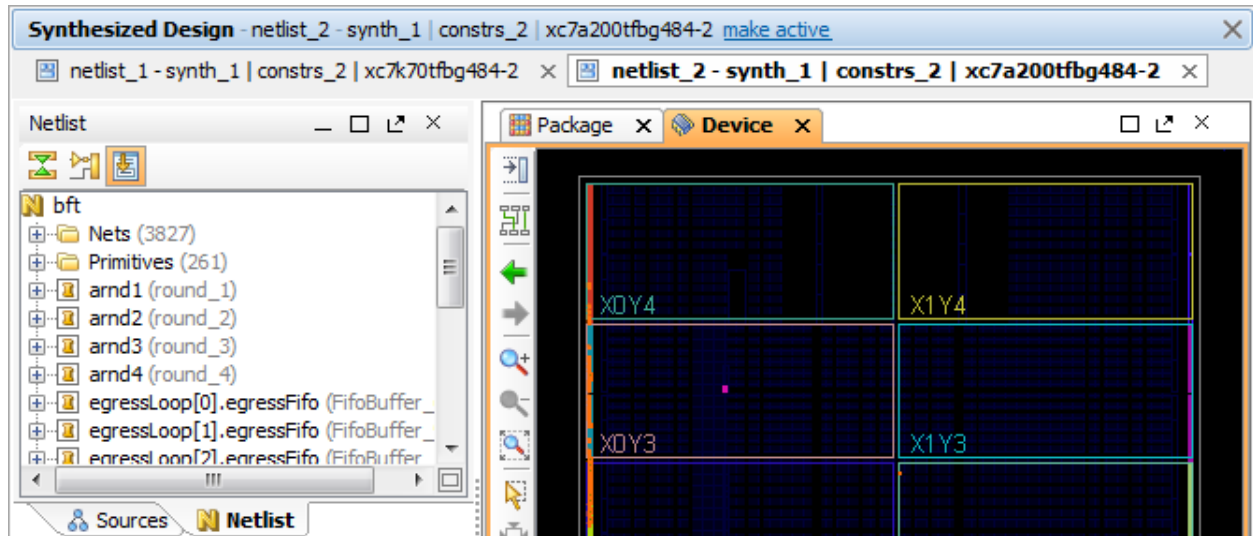


図 42 : 2 つの合成済みデザイン

アクティブでないデザインに対しては、[make active] リンクが表示されます。このリンクをクリックすると、そのデザインがアクティブになります。

この機能を使用して、次を比較できます。

- 異なる合成 run の結果
- 制約ファイルの異なるバージョン
- 異なるパーツを指定した同じネットリスト
- 異なる配置配線結果

この機能は、64 ビット OS で使用するのが適しています。同じ段階の 2 つの run を開くのが単純な使い道ですが、次のように 1 つのネットリストを複数のウィンドウに開くこともできます。

1. Flow Navigator で [Synthesized Design] を右クリックします。
2. ポップアップメニューから [New Synthesized Design] をクリックします。

エラボレート済みデザインでも、同じ機能がサポートされます。

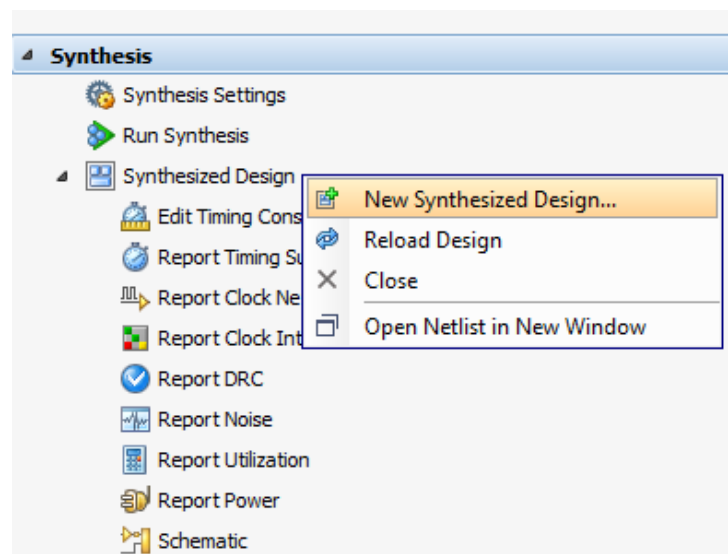


図 43 : [New Synthesized Design] コマンド

非プロジェクト モードでも、複数のデザインを開くことができます。これには、開いているチェックポイントを閉じずに 2 つ目のチェックポイントを開くのが簡単な方法です。2 つ目のチェックポイントを開く前に開いているチェックポイントを閉じるには、`close_design` コマンドを使用します。ただし、複数のデザインを管理するには、`start_gui` コマンドを実行して Vivado IDE GUI を使用するのが最も簡単です。

複数のビュー

複数のモニターを使用している場合、ビューまたはデザインをメイン ウィンドウから切り離してフロートさせると便利です。複数の合成済みデザインを開き、並べて比較する場合など、デザイン全体を別のウィンドウに開くには、デザインのバナーを右クリックし、[Open in New Window] をクリックします。

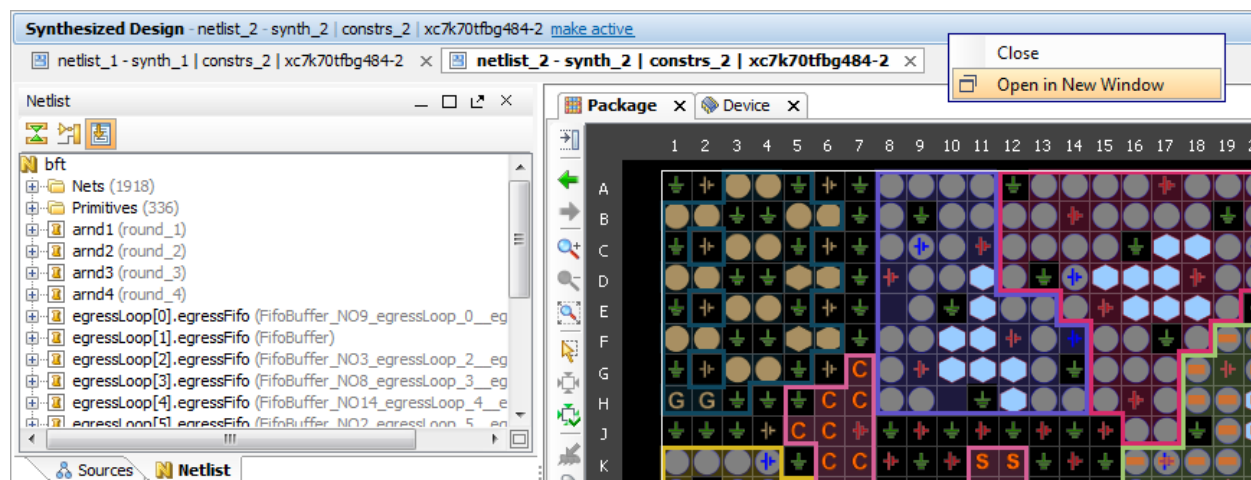


図 44：デザインを別のウィンドウで開く

ビューをフロートさせる方法は、『Vivado Design Suite ユーザー ガイド：Vivado IDE の使用』(UG893) の「ビューの使用」を参照してください。

インプリメンテーション結果の解析

この章では、インプリメンテーション後のデザインを解析してデバイス内での動作を理解するための次の手法を説明します。

- 階層ブロックの配置の確認
- I/O の確認
- 接続の表示
- ビュー間のクロスプローブ
- 詳細な配線の確認

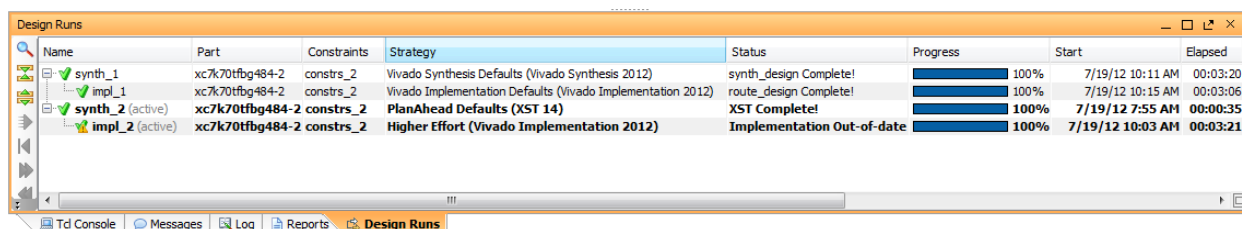
[Design Runs] ビュー

[Design Runs] ビューには、run の状態が表示されます。

詳細は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) の「[Design Runs] ビューの使用」を参照してください。

run が終了すると、[Design Runs] ビューが run が問題なく完了したか、エラーが発生したかが示されます。

ヒント：run が最新でない場合は、ポップアップ メニューから [Force up to date] をクリックして、ステータスを最新の状態にすることができます。



Name	Part	Constraints	Strategy	Status	Progress	Start	Elapsed
synth_1	xc7k70tbg484-2	constrs_2	Vivado Synthesis Defaults (Vivado Synthesis 2012)	synth_design Complete!	100%	7/19/12 10:11 AM	00:03:20
impl_1	xc7k70tbg484-2	constrs_2	Vivado Implementation Defaults (Vivado Implementation 2012)	route_design Complete!	100%	7/19/12 10:15 AM	00:03:06
synth_2 (active)	xc7k70tbg484-2	constrs_2	PlanAhead Defaults (XST 14)	XST Complete!	100%	7/19/12 7:55 AM	00:00:35
impl_2 (active)	xc7k70tbg484-2	constrs_2	Higher Effort (Vivado Implementation 2012)	Implementation Out-of-date	100%	7/19/12 10:03 AM	00:03:21

図 45 : [Design Runs] ビュー

[Design Runs] ビューでは、情報が次のように表示されます。

- run が問題なく完了すると、「route_design Complete!」と表示されます。
- 簡単なサマリが表示されます。
- デザイン エラーがレポートされます。
- デザインのタイミングが満たされたかどうかは示されません。
- クリティカル警告やその他のデザインの問題は表示されません。

Vivado™ IDE プロジェクト フローを使用している場合は、[Messages] ビューでアクティブなインプリメンテーションの結果を確認します。メッセージは、run フローの段階別にグループ化されています。インプリメンテーションでのクリティカルな情報は、このビューに表示されます。ログ ファイルでそのメッセージを見つけ、どういう状況で発生したかを確認します。

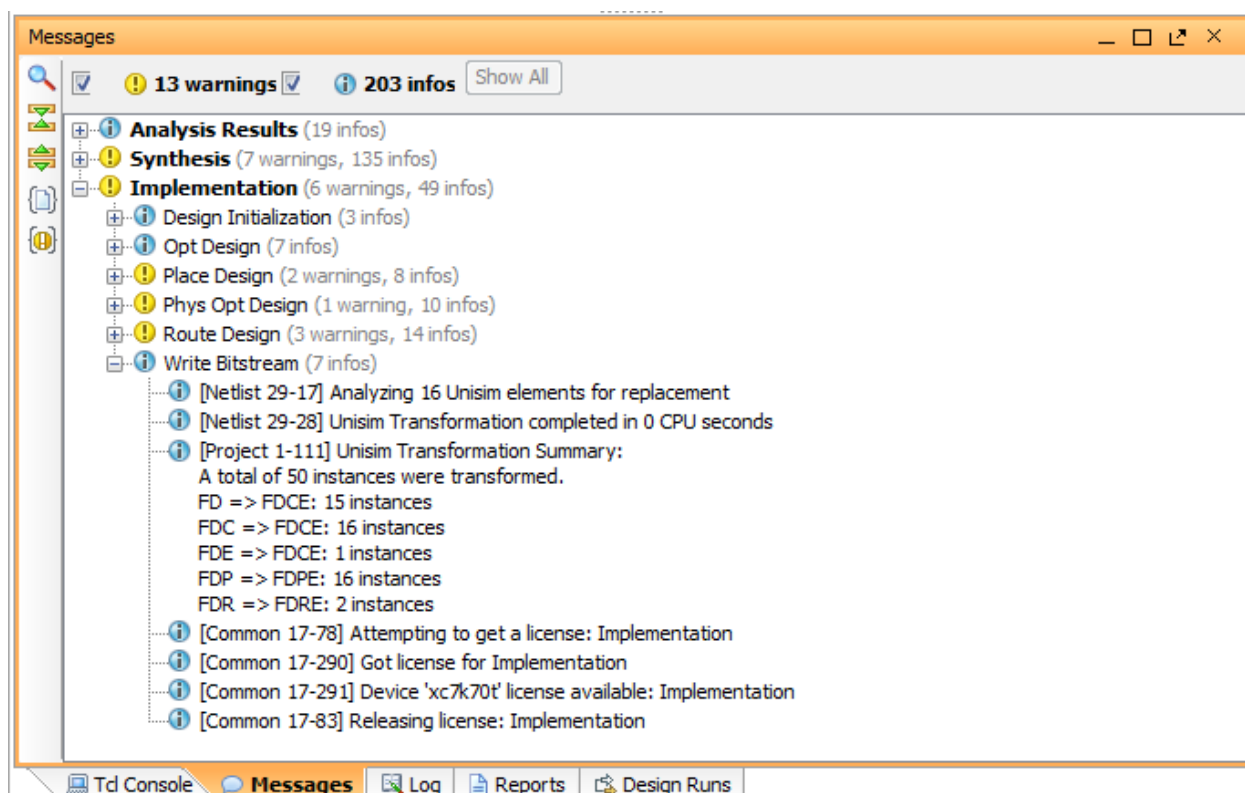


図 46：段階ごとにグループ化されたメッセージ

警告またはエラーがある場合は、いずれかのレポートを確認してください。一部のメッセージにはリンクが含まれており、関連するデザイン エLEMENT にクロスプローブできます。

デザインをメッセージの発生した状況で解析するため、次のいずれかデザインを開く必要がある場合があります。

- インプリメント済みデザイン (インプリメンテーション後のネットリストを使用)
- ネットリスト デザイン (インプリメンテーション前のネットリストを使用)

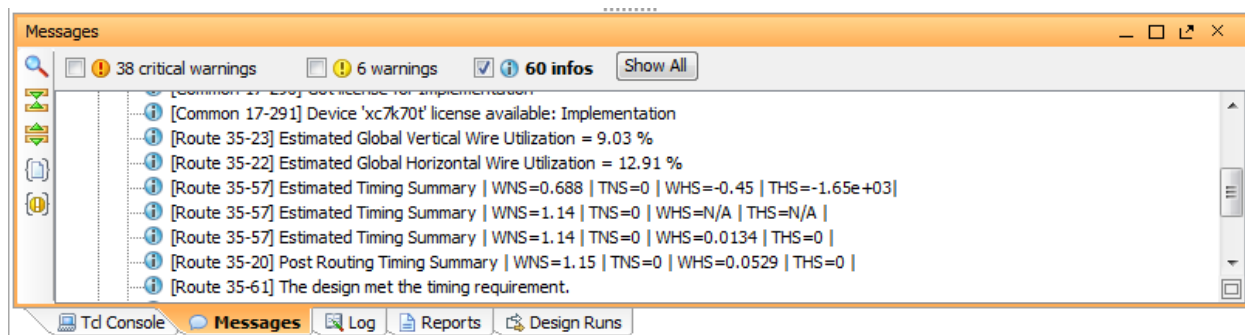


図 47：インプリメンテーションで表示されたメッセージ

配線では、デザインのタイミングが満たされたかどうかを示すメッセージが表示されます。

重要：このメッセージはタイミングの最終確認としては使用できません。タイミングの最終確認には、`report_timing_summary` コマンドを使用する必要があります。

デザインのタイミングが満たされたかどうかは、タイミング サマリ レポートを確認してください。タイミングが満たされていない場合は、第 5 章「クロージャ テクニック」を参照してください。

非プロジェクト フローでは、Vivado IDE GUI を開いた場合、[Messages] ビューにその実行セッションで生成されたメッセージが表示されます。次のものを確認してください。

- Vivado IDE ログ ファイル
- 情報、警告、およびエラー

Vivado IDE GUI を使用しない場合は、`report_timing_summary` コマンドを使用してデザインのタイミングが満たされたかどうかを確認してください。

その他の解析

インプリメンテーションが終了したら、デザインがデバイスでどのように動作するかを確認できます。

Vivado IDE では、デバイス内のロジックおよび配線リソースの使用状況を調べるためのメトリックが複数あります。メトリックは、[Device] ビューに指定した規則に従って色分けして表示されます。

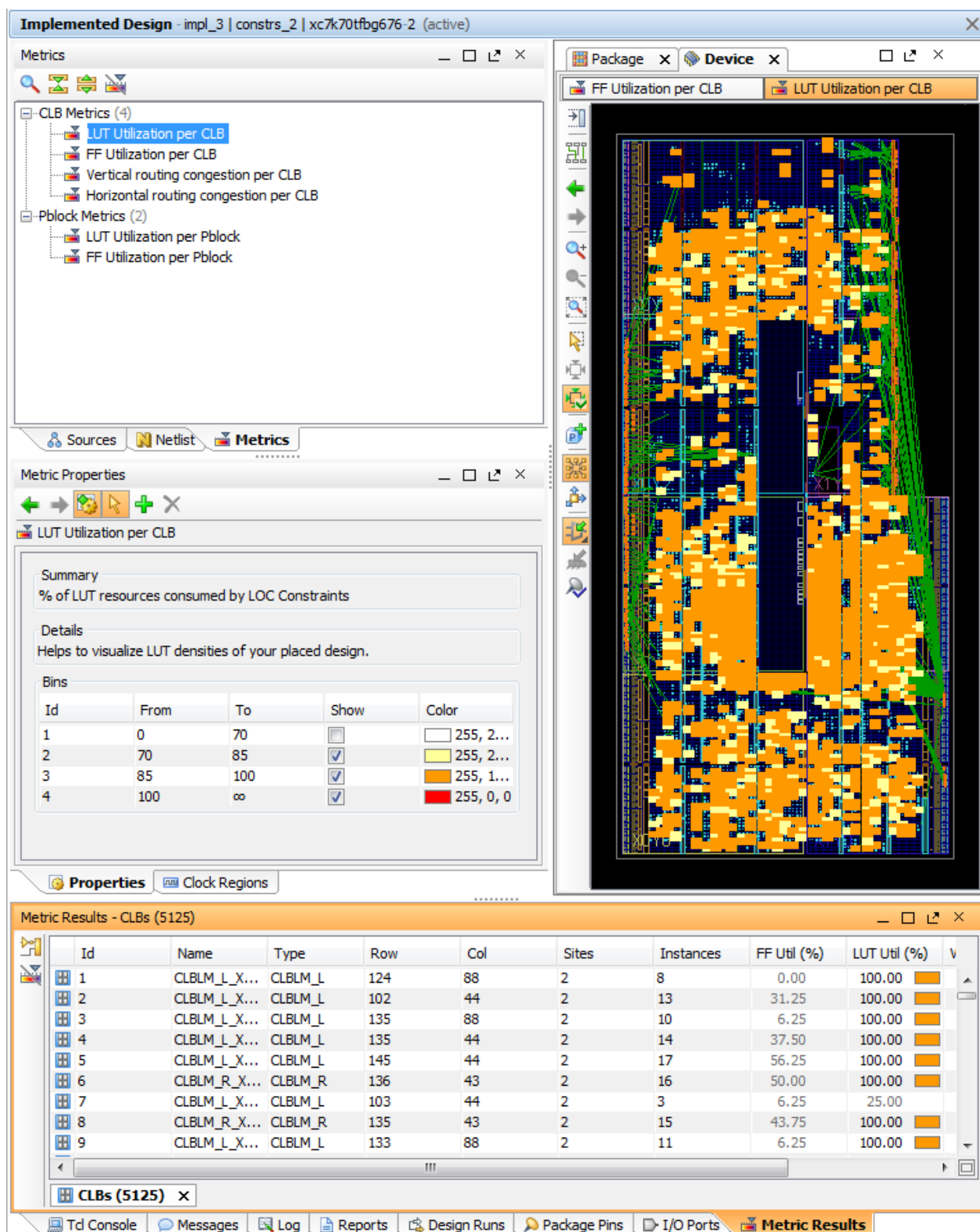


図 48：メトリック

配置済みデザインが必要なメトリック

次の 4 つのメトリックには、配置済みデザインが必要です。完全に配線されたデザインは必要ありません。

- CLB ごとの LUT の使用率
配置された LUT の使用率に基づいて、スライスを色分け表示します。
- CLB ごとのフリップフロップの使用率
配置されたフリップフロップの使用率に基づいて、スライスを色分け表示します。
- CLB ごとの垂直方向の配線の密集度
垂直方向の配線の使用状況をベスト ケースで予測した結果に基づいて、ファブリックを色分け表示します。
- CLB ごとの水平方向の配線の密集度
水平方向の配線の使用状況をベスト ケースで予測した結果に基づいて、ファブリックを色分け表示します。

配置のないネットリスト デザインでのメトリック

Pblock が存在する場合、2 つのメトリックがあります。これらは配置に依存しません。

- Pblock ごとの LUT の使用率
Pblock に含まれる LUT のスライスへの配置予測に基づいて、Pblock を色分け表示します。
- Pblock ごとのフリップフロップの使用率
Pblock に含まれるフリップフロップのスライスへの配置予測に基づいて、Pblock を色分け表示します。

図 48 に示すように、同時に複数のルールを使用できます。CLB ごとの LUT 使用率と CLB ごとのフリップフロップの使用率がオンになっています。

ヒント：デザインに使用率の高い部分や配線の密集度が高くなることが予測される部分がある場合は、RTL または配置制約を変更して、その部分でのロジックおよび配線の使用率を削減してください。

配置のハイライト

デザインの配置を確認する別の方法として、セル配置の解析があります。この解析には、[Highlight Primitives] コマンドを使用します。

1. [Netlist] ビューで解析する階層レベルを選択します。
2. 右クリックして [Highlight Primitives] をクリックし、色を選択します。
3. 複数の階層レベルを選択している場合は、[Cycle Colors] をクリックします。

[Device] ビューでプリミティブを構成するセルが色表示されます。

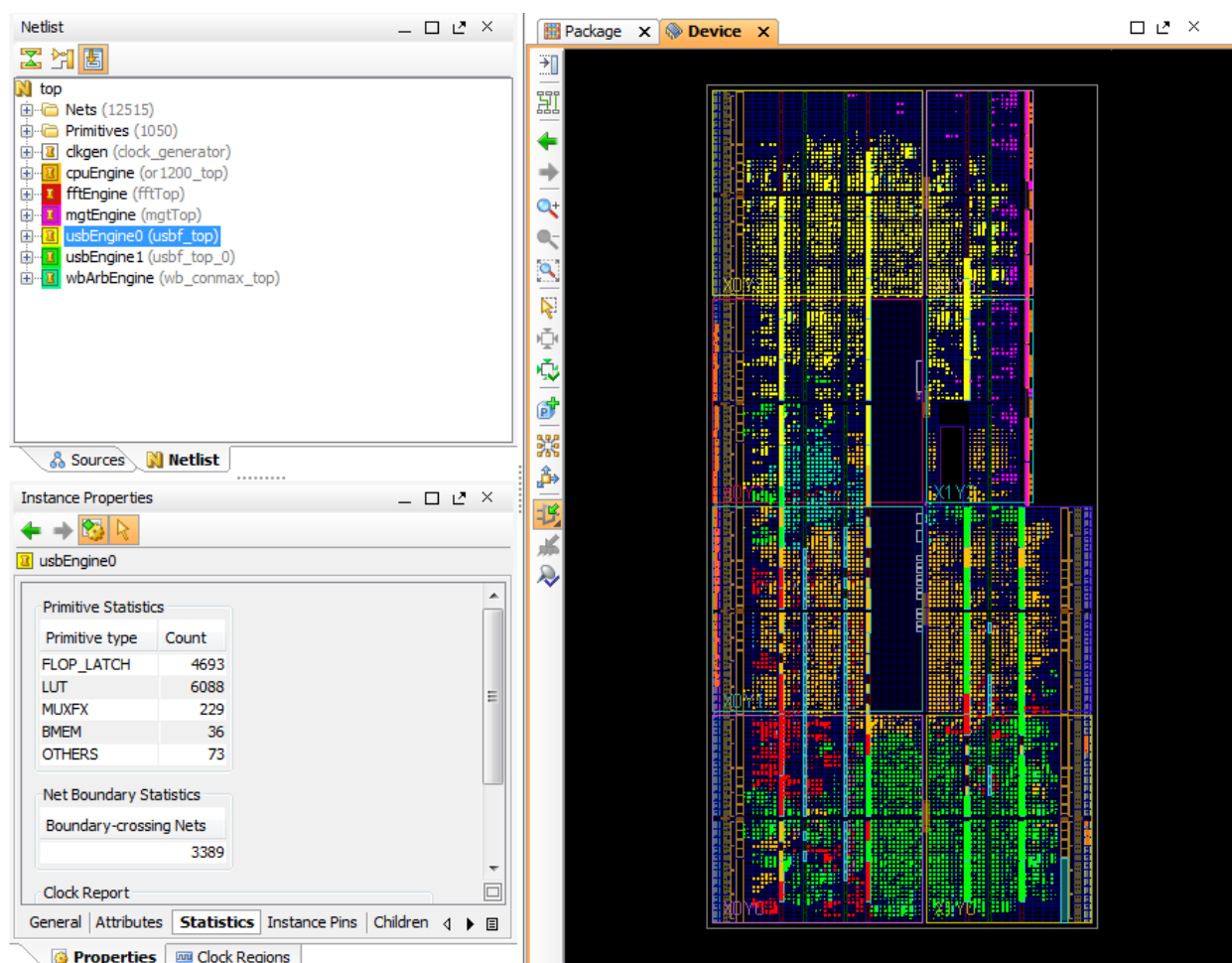


図 49：階層のハイライト

図 49 では、色分け表示により、UsbEngine0 (黄色) が次のようであることがわかります。

- ブロック RAM および DSP48 セルを使用しています。
- DSP 以外は、チップの上部のクロック領域内にあります。
- デザインのその他のロジック (セル) とそれほど混じり合っていない。

fftEngine (赤色) と cpuEngine (茶色) が混じり合っているのがわかります。この 2 つのブロックは、主に異なるリソースを使用します (一方は DSP48、もう一方はスライス)。これらが混じり合っていることで、デバイスが最適に使用されます。

接続の表示

接続に基づいてデザインを解析すると有益な場合があります。たとえば、ある入力、ブロック RAM、または DSP のバンクで駆動されるすべてのロジックの配置を確認する場合などです。これには、[Show Connectivity] コマンドを使用します。このコマンドは、複数のセルまたはネットをシードとして、別のタイプのオブジェクトを選択します。この機能を使用して、デザイン内のロジックのコーンを表示できます。

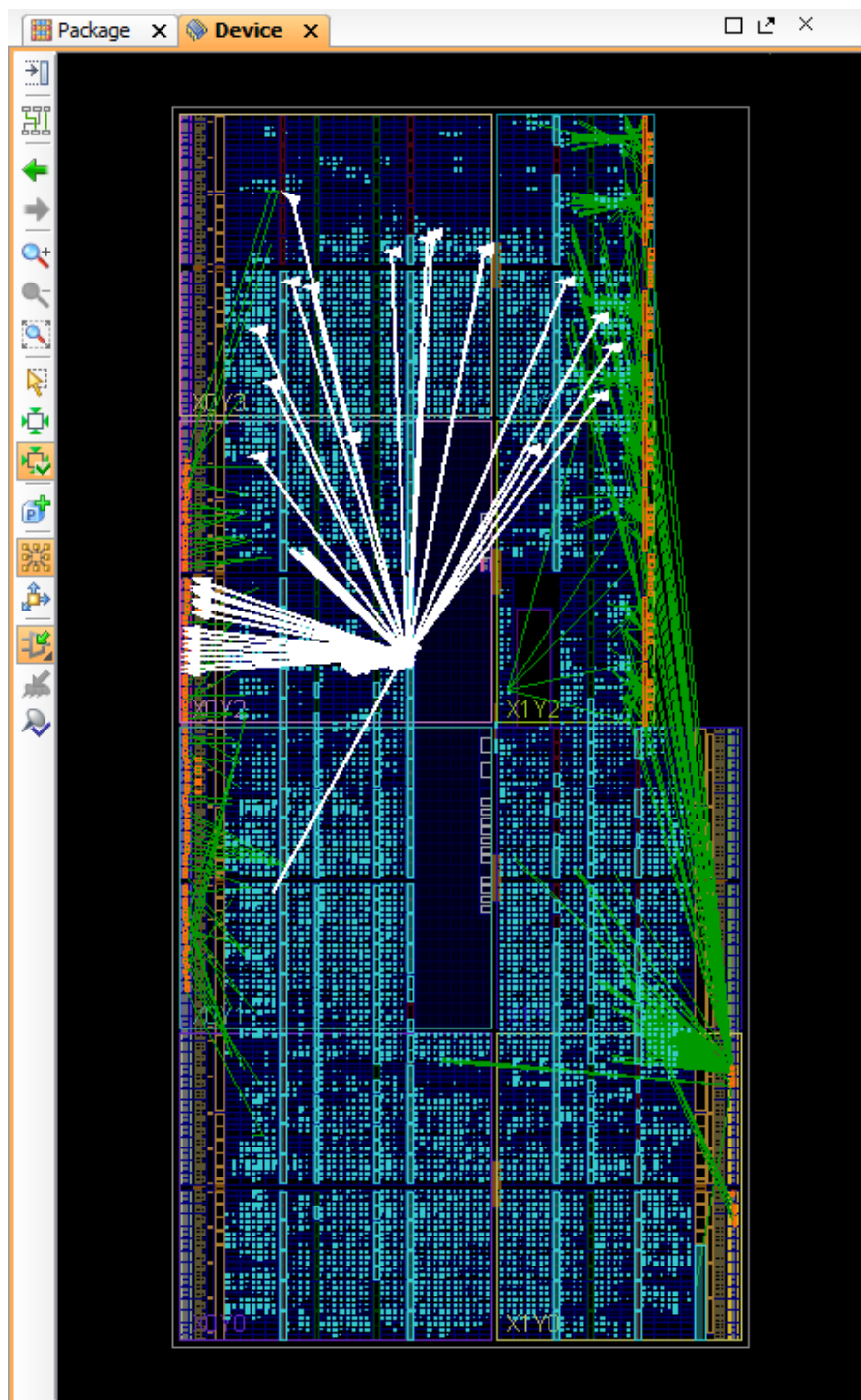


図 50 : 接続の表示

図 50 では、デバイス内の OBUF を含むロジックを駆動するブロック RAM が表示されています。合成プラグマにより、メモリの推論でブロック RAM 内に出力フリップフロップが配置されないようになっています。

固定および未固定のロジック

次の 2 種類の配置があります。

- ユーザーが配置したエレメント (オレンジ色) は固定されます。
 - 固定ロジックは、XDC に保存されます。
 - 固定ロジックには、通常 LOC 制約が設定されています。BEL 制約が設定されている場合もあります。
- ツールで配置されたエレメント (青色) は固定されません。

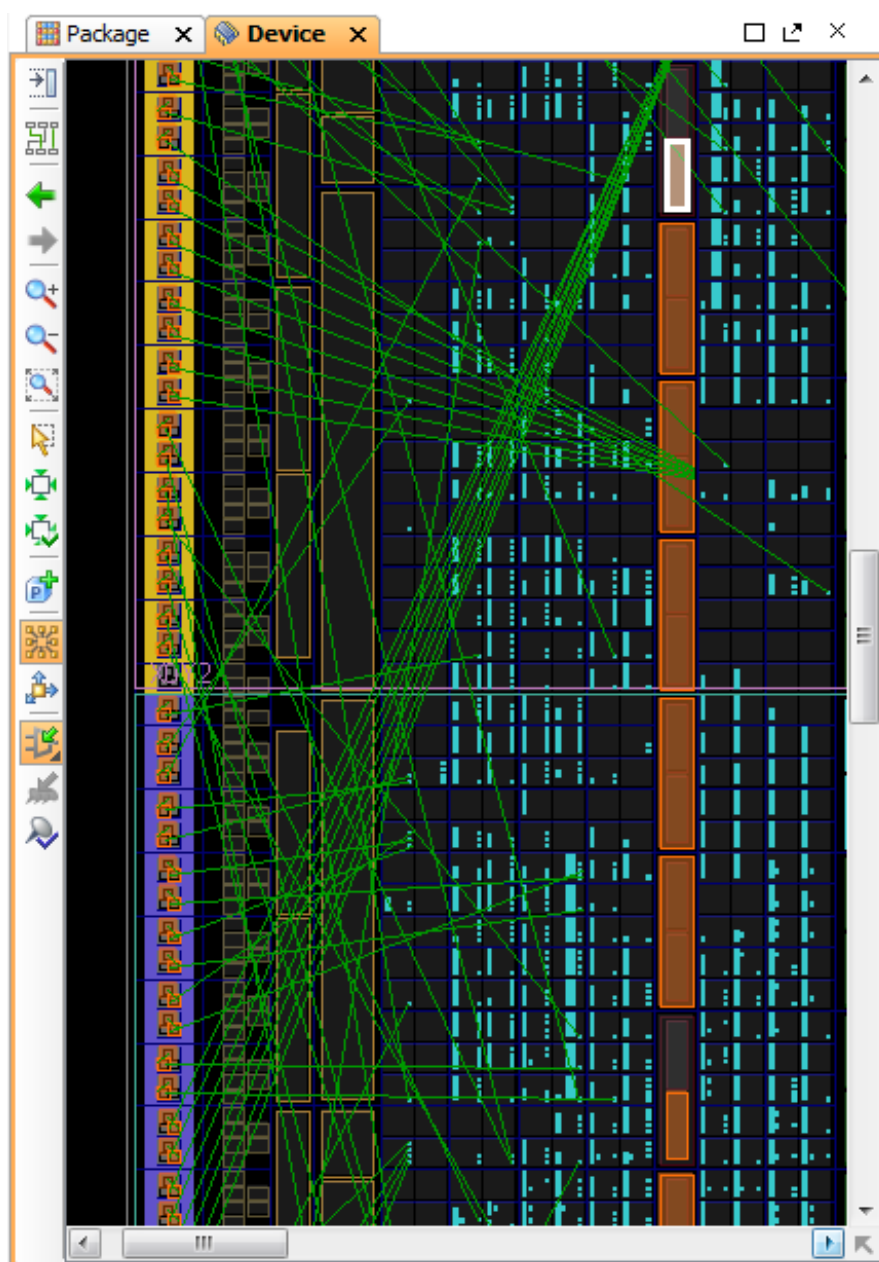


図 51：固定および未固定のロジック

I/O およびブロック RAM の配置は固定されます。スライス ロジックは固定されません。

ロジックを固定するには、次のいずれかを実行します。

- 配置を XDC に入力します。
- Vivado IDE GUI でロジックを右クリックし、[Fix Instances] をクリックします。
- Tcl コンソールで `is_loc_fixed` または `is_bel_fixed` プロパティを設定します。

XDC の制約は、次のようになります。

- フリップフロップに LOC および BEL を設定

```
set_property BEL DFF [get_cells
{cpuEngine/cpu_iwb_dat_i/buffer_fifo/infer_fifo.wr_addr_tmp_reg[9]}]
set_property LOC SLICE_X7Y103 [get_cells
{cpuEngine/cpu_iwb_dat_i/buffer_fifo/infer_fifo.wr_addr_tmp_reg[9]}]
```

フリップフロップの配置には、LOC のみを使用可能です。

- ブロック RAM

```
set_property LOC RAMB36_X0Y21 [get_cells
usbEngine1/dma_out/buffer_fifo/infer_fifo.block_ram_performance.fifo_ram_reg]
```

- I/O ピン

```
set_property PACKAGE_PIN E23 [get_ports {DataIn_pad_0_i[5]}]
```

クロスプローブ

Vivado 合成を使用して合成されたデザインでは、ネットリスト デザインがメモリに読み込まれていればソース ファイルにクロスプローブできます。

クロスプローブするには、次の手順に従います。

1. ゲートを選択します。
2. 右クリックして [Go to instantiation] をクリックします。

第 3 章 : インプリメンテーション結果の解析

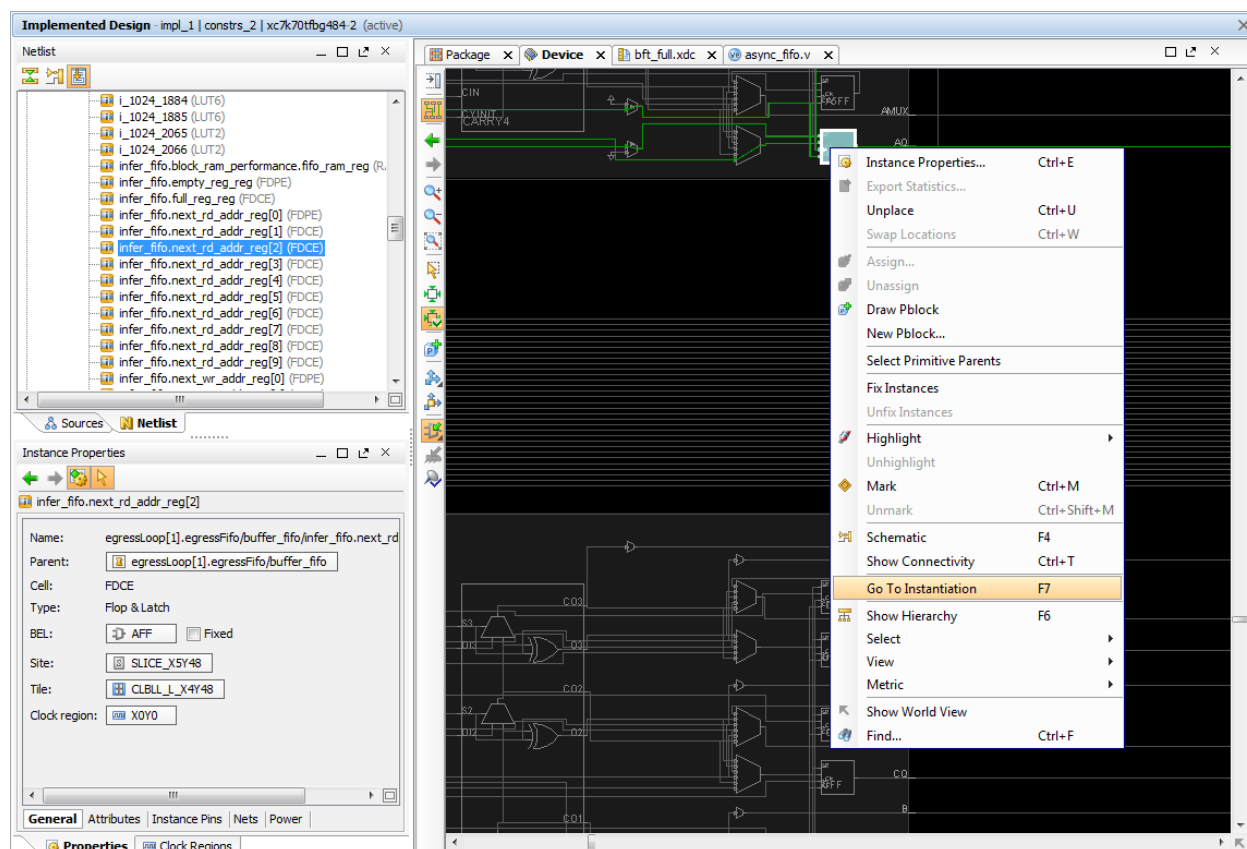


図 52 : ソースへのクロスプローブ

クロスプローブ機能を使用して、ネットリスト ゲートに関連するソースを特定します。合成での変換により、デザインのすべてのゲートをソースにクロスプローブできるとは限りません。

配線の表示

[Device] ビューのツールバーの [Routing Resources] ボタンをオンにすると、配線リソースが表示されます。

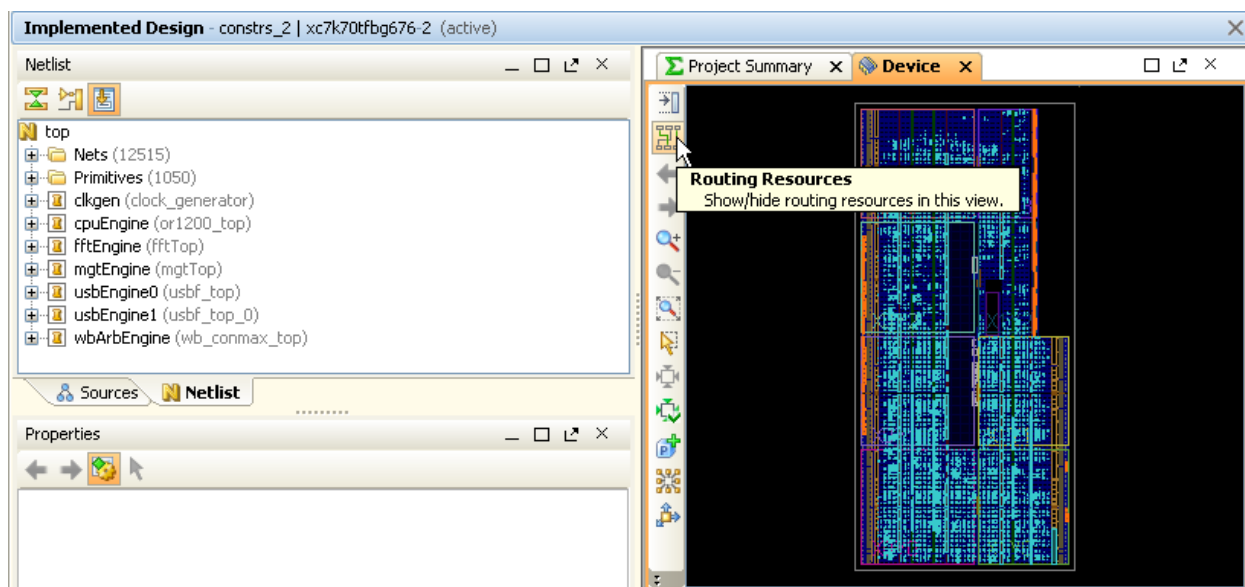


図 53 : 配線の表示

配線配置の表示

配置配線は、ズームレベルによって表示が異なります。

- 縮小表示した場合
- 拡大表示した場合

2 種類の表示を使用することにより、ランタイムとメモリの使用量を最小限に抑えながら、すべてのサイズでデザインの詳細が表示されるようになっています。

縮小した場合の配線配置の表示

縮小すると、表示は簡略化されたものになります。この表示には、次のような特徴があります。

- 配線が凝縮されたものになります。
- 特定の領域を通過する配線の数によって線の太さが変わります。

配置も同様に、各タイルを表すブロックに、配置されているロジックが表示されます。タイルに含まれるロジックが多いほど、タイルを表すブロックが大きくなります。

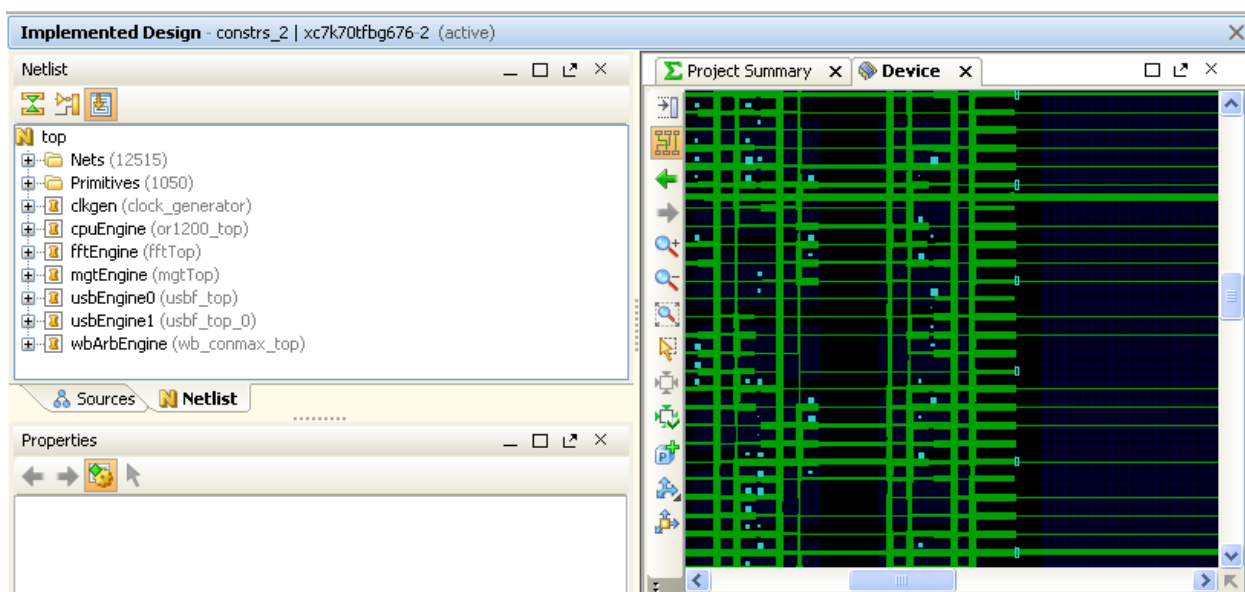


図 54：簡略表示

拡大した場合の配線配置の表示

拡大すると、実際のロジックセルおよび配線が表示されます。

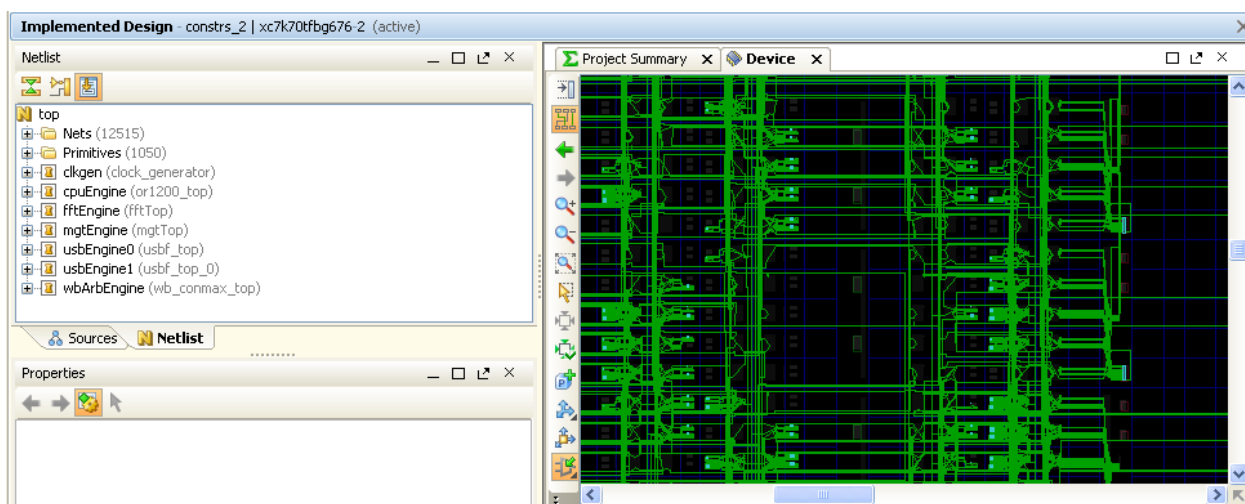


図 55：詳細表示

表示オプション

[Device] ビューをカスタマイズして、デバイスおよびデザインをさまざまな方法で表示できます。これらは、ツールバーの [Device View Options] をクリックして設定します。

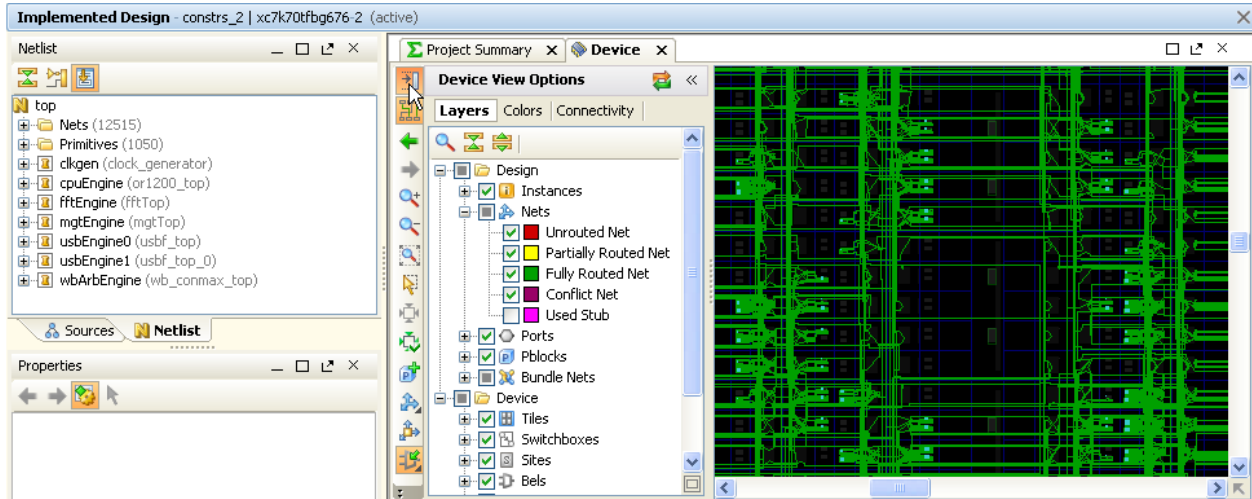


図 56 : [Device View Options] の [Layers] タブ

デザインおよびデバイスの異なるリソースの表示/非表示を切り替えたり、表示色を変更したりできます。

[Device] ビューのナビゲーション

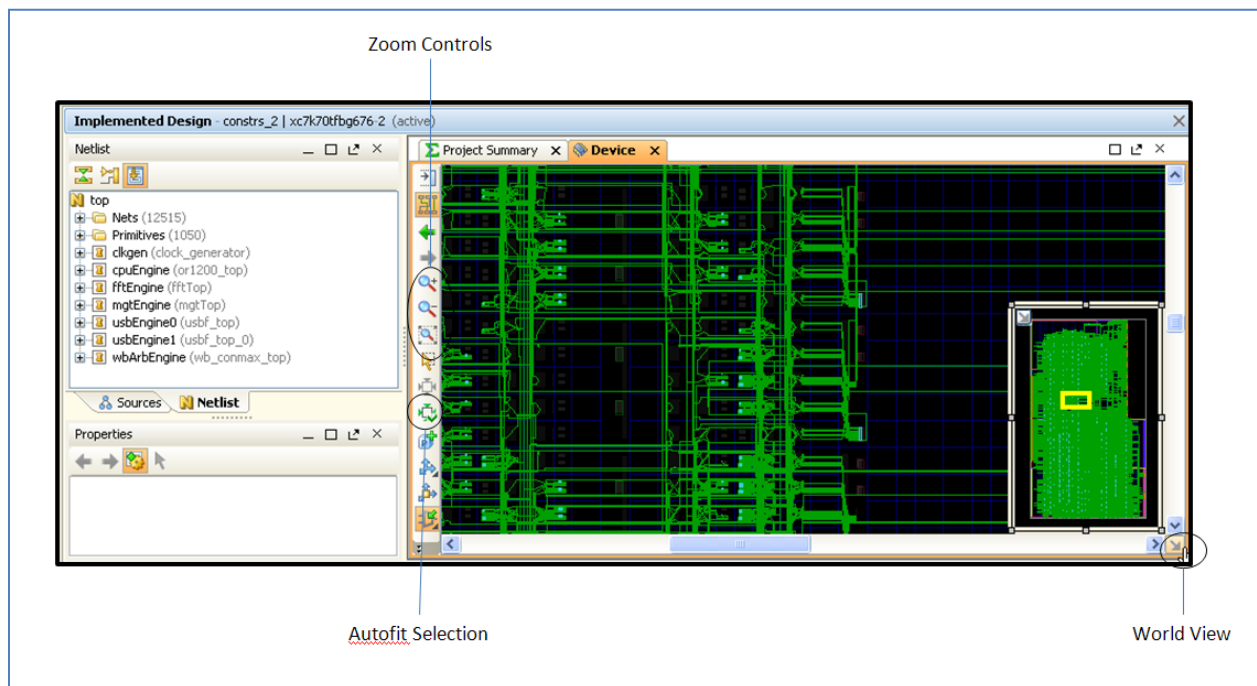


図 57 : [Device] ビューのナビゲーション

[Device] ビューは、次の機能を使用してナビゲートできます。

- ズーム操作

標準的な拡大 ([Zoom In])、縮小 ([Zoom Out])、および全体表示 ([Zoom Fit])

- [Auto Fit Selection]

[Device] 以外のビューで選択されたオブジェクトを自動的に拡大表示します。クロスプローブに便利です。

- [World] ビュー

[Device] ビューに現在表示されている部分がデバイス全体のどこかを表示します。[World] ビューに表示される黄色のボックスをドラッグまたはサイズ変更すると、[Device] ビューの表示を移動したりサイズ変更したりできます。

- Ctrl キー

Ctrl キーを押しながらドラッグすると、表示箇所を移動できます。

タイミング解析

Vivado™ Design Suite には、レポートを生成するコマンドが複数あり、デザインがタイミング制約を満たしており、アプリケーション ボードに読み込む準備ができたかを確認できます。タイミング サマリ レポートはタイミングの最終確認レポートであり、ISE™ Design Suite の TRCE レポートに相当します。すべてのタイミング チェックの概要が示され、タイミング問題の解析およびデバッグを開始するのに十分な情報が提供されます。詳細は、第 1 章「[デザイン解析](#)」を参照してください。

このレポートは、ビューに表示するか、ファイルに保存するか、ログ ファイルに出力できます。タイミング サマリ レポートでタイミングが満たされていない、不足している制約があるなどの結果が示された場合、レポートのさまざまなセクションに示される詳細を確認して、問題によってはより詳細な解析を実行します。ほかのタイミング レポートでは、特定の状況または特定のロジックに対するより詳細な情報が提供されます。

タイミングの最終確認

タイミング解析の詳細を説明する前に、デザインがハードウェアで実行する準備ができたことが、タイミング レポートのどの部分に示されるかを知っておくことが重要です。デザインが完全に配置配線されたら、インプリメンテーション結果を解析してタイミングの最終確認をすることは必須です。デフォルトでは、Vivado Design Suite でプロジェクトを使用する場合、テキスト形式のタイミング サマリ レポートが自動的に生成されます。このレポートは、インプリメンテーション後にメモリにデザイン チェックポイントを読み込んで、インタラクティブなバージョンを生成することもできます。

タイミングの最終確認には、次の 2 つの基準があります。

- デザインが完全に制約されている

デザインが完全に制約されているかを確認するには、[Check Timing] セクションを参照します。次を確認します。

- 定数でないクロック ピンに定義されたクロックが供給されている (no_clock チェック)。
- すべての内部パス エンドポイントにタイミングが設定されている (unconstrained_endpoint チェック)。
- 入力ポートおよび出力ポートがすべて制約されている (no_input_delay, no_output_delay チェック)。

一部の不足制約はそのまま放置できますが、デザインの最終確認の質は下がります。ほかのチェックの結果も正しいことを調べ、最終確認に影響していないことを確認してください。

- デザインのタイミングが満たされている

- トータル ネガティブ スラック (TNS) が 0ns である必要があります。これは、最大遅延解析に対応します。
- トータル ホールド スラック (THS) が 0ns である必要があります。これは、最小遅延解析に対応します。
- トータル パルス幅スラック (TPWS) が 0ns である必要があります。これは、ISE のコンポーネント ピン スイッチ制限と同等です。最大および最小遅延解析の両方で実行されます。

TNS、THS、および TPWS の合計が、ISE の最終的なタイミング スコアに相当します。

第 4 章 : タイミング解析

次の図に、タイミングが満たされていることを確認するために最初に見る必要のある情報を緑で囲んで示します。

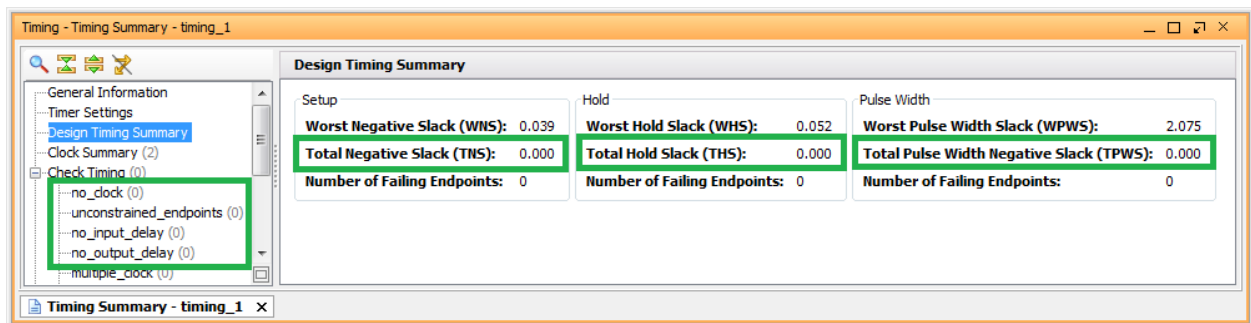


図 58 : タイミング サマリレポートでのタイミングの最終確認

次の図に、テキスト形式のレポートの [Check Timing] セクションを示します。

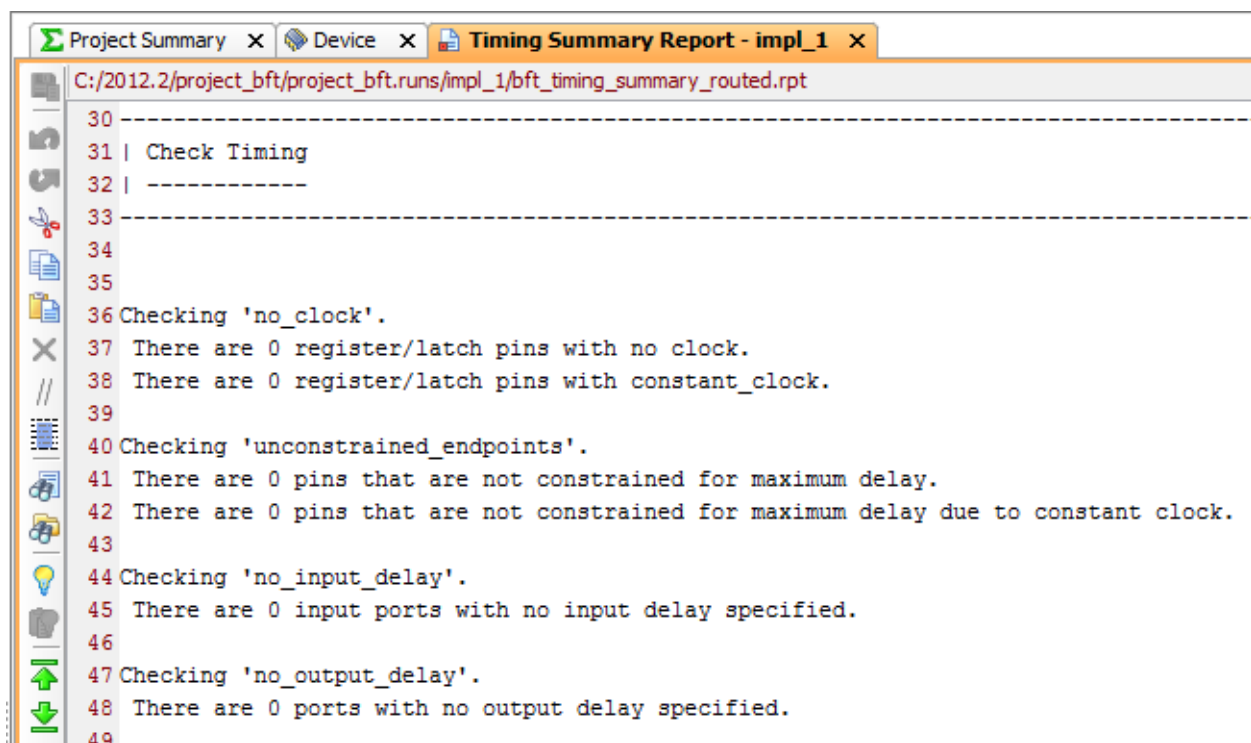


図 59 : テキスト形式のタイミング サマリレポートの [Check Timing] セクション

次の図に、テキスト形式のレポートの [Design Timing Summary] セクションを示します。

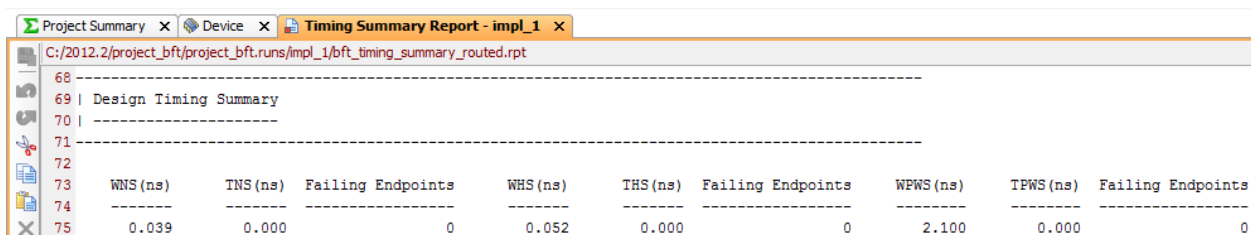


図 60 : テキスト形式のタイミング サマリレポートの [Design Timing Summary] セクション

Vivado IDE では、スラック違反は赤で表示され、不足している制約は特定の色で表示されます。タイミング違反を調べるには、まず次のセクションを参照します。

- [Intra-Clock Paths]、[Inter-Clock Paths]、および [Path Groups] (**async_default**) セクションには、セットアップ/リカバリ違反およびホールド/リムーバル違反に関する情報が示されます。
- [Intra-Clock Paths] セクションには、パルス幅チェック違反の詳細が示されます。

[Timing] ビューの詳細は、第 1 章「デザイン解析」の「タイミング サマリレポート」を参照してください。

違反のみを表示するには、ツールバーの [Show only failing checks]  をクリックします。

デフォルトでは、各解析で各クロック ペアに対して N 個のワースト パスの詳細が表示されます。GUI のデフォルトでは N = 10、report_timing_summary コマンドのデフォルトは N = 1 です。パスをダブルクリックすると、そのタイミング パスの詳細情報が新しいビューに表示されます。レポートされているパスの数が十分でない場合は、レポートされるパス数を増加して [Report Timing Summary] を再実行するか（同等の Tcl コマンド : report_timing_summary -max_paths N）、特定のクロック ペアまたは特定のタイミング パスに対して [Report Timing] を実行します。

タイミング パス レポートの解釈

タイミング サマリ レポートでパスをダブルクリックしたときに表示されるタイミング パス レポートには、タイミング違反の原因を理解するのに必要な情報がすべて含まれています。このレポートは、次に説明する 4 つのセクションに分かれています。

タイミング パス サマリ

タイミング パスに関する重要な情報を示します。タイミング パスの詳細を解析せずに、違反の原因を知ることができます。スラック、パスの要件、データパス遅延、セル遅延、配線遅延、クロック スキュー、およびクロックのばらつきが示されます。セルの配置に関する情報は含まれません。タイミング制約およびタイミング解析で使用する用語、およびスラックおよびパスの要件の決定方法は、『Vivado Design Suite ユーザー ガイド：制約の使用』(UG903) の「タイミング解析」を参照してください。

次の図に、テキスト形式のレポートのタイミング パス サマリ ヘッダーを示します。

```
Slack (MET) :          1.690ns
Source:          ct0/xOutReg_reg/CLK
                  (rising edge-triggered cell DSP48E1 clocked by bftClk
                  {rise@0.000ns fall@2.500ns period=5.000ns})
Destination:     transformLoop[0].ct0/xOutReg_reg/C[36]
                  (rising edge-triggered cell DSP48E1 clocked by bftClk
                  {rise@0.000ns fall@2.500ns period=5.000ns})

Path Group:      bftClk
Path Type:       Max at Slow Process Corner
Requirement:     5.000ns
Data Path Delay:  2.646ns  (logic 0.555ns (20.975%)  route 2.091ns (79.025%))
Logic Levels:    1  (LUT2=1)
Clock Path Skew:  -0.103ns (DCD - SCD + CPR)
  Destination Clock Delay (DCD):      3.797ns
  Source Clock Delay (SCD):            4.123ns
  Clock Pessimism Removal (CPR):       0.223ns
Clock Uncertainty: 0.561ns ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE + UU
  Total System Jitter (TSJ):           0.071ns
  Total Input Jitter (TIJ):             0.100ns
  Discrete Jitter (DJ):                 0.000ns
  Phase Error (PE):                    0.000ns
  User Uncertainty (UU):                0.500ns
```

図 61: テキスト形式のタイミング パス サマリ

次の図に、Vivado IDE でのタイミング パス サマリ ヘッダーを示します。

Summary	
Name	Path 1
Slack	1.690ns
Source	ct0/xOutReg_reg/CLK (rising edge-triggered cell DSP48E1 clocked by bftClk {rise@0.000ns fall@2.500ns period=5.000ns})
Destination	transformLoop[0].ct0/xOutReg_reg/C[36] (rising edge-triggered cell DSP48E1 clocked by bftClk {rise@0.000ns fall@2.500ns period=5.000ns})
Path Group	bftClk
Path Type	Max at Slow Process Corner
Requirement	5.000ns
Data Path Delay	2.646ns (logic 0.555ns (20.975%) route 2.091ns (79.025%))
Logic Levels	1 (LUT2=1)
Clock Path Skew	-0.103ns
Clock Uncertainty	-0.561ns

図 62 : Vivado IDE のタイミング パス サマリ

タイミング パス サマリ ヘッダーには、次の情報が含まれます。

- **Slack** : 正のスラック値は、パスがタイミング制約から算出されたパスの要件を満たしていることを示します。スラックを求める式は、解析によって異なります。
 - 最大遅延解析 (セットアップ/リカバリ) :

$$\text{スラック} = \text{データ所要時間} - \text{データ到着時間}$$
 - 最小遅延解析 (ホールド/リムーバル) :

$$\text{スラック} = \text{データ到着時間} - \text{データ所要時間}$$

データ所要時間とデータ到着時間は、タイミング パス レポートの別のセクションに表示されます。
- **Source** : パスの開始点およびデータを送信するソース クロックを示します。開始点は通常、シーケンシャル セルのクロック ポートまたは入力ポートです。2 行目がある場合は、プリミティブおよびクロック ピンの認識されるエッジが示されます。クロック名とクロック エッジの定義 (波形および周期) も示されます。
- **Destination** : パスの終点およびデータを受信するデスティネーション クロックを示します。終点は通常、デスティネーション シーケンシャル セルの入力データ ピンまたは出力ポートです。2 行目がある場合は、プリミティブおよびクロック ピンの認識されるエッジが示されます。クロック名とクロック エッジの定義 (波形および周期) も示されます。
- **Path Group** : パスの終点が属するタイミング グループを示します。これは通常デスティネーション クロックにより定義されるグループですが、非同期タイミング チェック (リカバリ/リムーバル) では ****async_default**** タイミング グループにまとめられます。ユーザー定義のグループもここに表示されます。これらは、レポート生成で便利です。
- **Path Type** : このパスで実行された解析のタイプを示します。**Max** はデータ パス遅延の算出に最大遅延値が使用されたことを示します。セットアップ解析およびリカバリ解析に対応します。**Min** はデータ パス遅延の算出に最小遅延値が使用されたことを示します。ホールド解析およびリムーバル解析に対応します。また、使用されたコーナー (Slow または Fast) も示されます。
- **Requirement** : タイミング パス要件を示します。通常、開始点と終点が同じクロックで制御されている場合、セットアップ/リカバリ解析では 1 クロック周期、ホールド/リムーバル解析では **0ns** です。パスが 2 つのクロックの間にある場合、要件はソース クロック エッジとデスティネーション クロック エッジの最小の正の差に対応します。この値は、複数サイクル、最大遅延、最小遅延などのタイミング例外制約に置き換えられます。タイミング制約からタイミング パス要件を求める方法の詳細は、『Vivado Design Suite ユーザー ガイド : 制約の使用』(UG903) を参照してください。
- **Data Path Delay** : パスのロジック セクション間で累積された遅延を示します。クロックがデータとして使用されている場合を除き、クロック遅延は含まれません。遅延のタイプは、[Path Type] と同じです。
- **Logic Levels** : パスのデータ セクションに含まれている、開始セルと終了セルを除く各種プリミティブの段数を示します。

- **Clock Path Skew** : ソース クロックの送信エッジとデスティネーション クロックの受信エッジ間の挿入遅延差およびクロックの不必要に悪い見積もり部分の削除を示します。
 - **Destination Clock Delay (DCD)** : デスティネーション クロックのソース ポイントからパスの終点までに累積される遅延を示します。最大遅延解析 (セットアップ/リカバリ) では最小セルおよびネット遅延値が使用され、最小遅延解析 (ホールド/リムーバル) では最大遅延値が使用されます。
 - **Source Clock Delay (SCD)** : クロックのソース ポイントからパスの開始点までに累積される遅延を示します。最大遅延解析 (セットアップ/リカバリ) では最大セルおよびネット遅延値が使用され、最小遅延解析 (ホールド/リムーバル) では最小遅延値が使用されます。
 - **Clock Pessimism Removal (CPR)** : ソースおよびデスティネーション クロックが共通回路上であっても異なる遅延でレポートされることにより発生する、追加のクロック スキューの絶対値を示します。この不必要に悪い見積もり部分の削除すると、共通回路上にあるソース クロックとデスティネーション クロックの間にはスキューは発生しません。配線済みのデザインでは、最後の共通クロック ツリー ノードは通常クロック ネットで 사용되는配線リソースにあり、パスの詳細にはレポートされません。
- **Clock Uncertainty** : クロック エッジのペア間での時間の変動量の合計を示します。ばらつきは、算出されるクロック ジッター (システムおよび個別)、一部のハードウェア プリミティブにより発生する位相エラー、デザイン制約でユーザーにより指定されるクロックのばらつき (`set_clock_uncertainty`) で構成されます。ユーザーが指定するクロックのばらつきは、Vivado タイミング エンジンで算出されるばらつきに加算されます。
 - **Total System Jitter (TSJ)** : ソース クロックおよびデスティネーション クロックの両方に適用されるシステム ジッターを示します。システム ジッターは、`set_system_jitter` XDC コマンドを使用してグローバルに変更できます。仮想クロックは理想的なクロックなので、システム ジッターはありません。システム ジッターの詳細は、『Vivado Design Suite ユーザー ガイド : 制約の使用』(UG903) を参照してください。
 - **Total Input Jitter (TIJ)** : ソース クロックおよびデスティネーション クロックの両方に適用される入力ジッターを示します。`set_input_jitter` XDC コマンドを使用すると、各プライマリ クロックの入力ジッターを個別に指定できます。生成クロックの入力ジッターは、マスター クロック ジッターおよび伝搬されるクロック リソースに基づいて Vivado IDE タイミング エンジンで算出されます。デフォルトでは、仮想クロックは理想的なクロックであり、ジッターはありません。ジッターの詳細は、『Vivado Design Suite ユーザー ガイド : 制約の使用』(UG903) を参照してください。
 - **Discrete Jitter (DJ)** : MMCM や PLL などのハードウェア プリミティブにより追加されるジッター量を示します。この値は、これらのセルのコンフィギュレーションに基づいて Vivado タイミング エンジンにより算出されます。
 - **Phase Error (PE)** : MMCM や PLL などのハードウェア プリミティブにより追加される、2 つのクロック信号間の位相の変動量を示します。この値は、これらのセルのコンフィギュレーションに基づいて Vivado タイミング エンジンにより算出されます。
 - **User Uncertainty (UU)** : `set_clock_uncertainty` XDC コマンドで指定された追加のばらつきを示します。このコマンドの使用方法は、『Vivado Design Suite ユーザー ガイド : 制約の使用』(UG903) を参照してください。

タイミング パス サマリには、タイミング制約、レポートされるパス、およびターゲット デバイスによって、次の 3 つの行が表示されることもあります。

- **Inter-SLR penalty** : 7 シリーズ SSI デバイスのみで、SLR の境界をまたがるパスを確実にレポートするために必要な追加マージンを示します。
- **Input Delay** : `set_input_delay` XDC コマンドにより入力ポートに指定された入力遅延値を示します。入力ポートから開始しないパスは含まれません。
- **Output Delay** : `set_output_delay` XDC コマンドにより出力ポートに指定された出力遅延値を示します。出力ポートで終了しないパスは含まれません。
- **Timing Exception** : パスに適用されるタイミング例外を示します。優先順位の最も高い例外のみがタイミング パス要件に影響するので、優先順位の最も高い例外のみが示されます。タイミング例外およびその優先順位の詳細は、『Vivado Design Suite ユーザー ガイド : 制約の使用』(UG903) を参照してください。

タイミング パスの詳細

レポートの後半には、パスが伝搬されるセル、ピン、ポート、およびネットの詳細が表示されます。3 つのセクションに分かれています。

- **Source Clock Path** : ソース クロックがソース ポイントからデータパスの開始点まで伝搬されるパスを示します。このセクションは、入力ポートから開始するパスには含まれません。
- **Data Path** : データが開始点から終点まで伝搬されるパスを示します。
- **Destination Clock Path** : デスティネーションクロックがソース ポイントからデータパスの終点のクロック ピンまで伝搬されるパスを示します。

[Source Clock Path] および [Data Path] セクションには、常に同じタイプの遅延 (セットアップ/リカバリ解析では最大遅延、ホールド/リムーバル解析では最小遅延) がレポートされます。データの送信エッジ時間から開始し、ソース クロックおよびデータ パスを介して累積される遅延も同じです。最終的な累積遅延値は、データ到着時間と呼ばれます。

デスティネーション クロック パスは、常にソース クロックおよびデータ パスと逆の遅延でレポートされます。初期の累積遅延値は、データ受信エッジがデスティネーション クロックのソース ポイントから出発した時間です。最終的な累積遅延値は、データ所要時間と呼ばれます。

レポートの最後の 3 行は、スラックの算出方法を示します。

- **最大遅延解析 (セットアップ/リカバリ) :**
 $\text{スラック} = \text{データ所要時間} - \text{データ到着時間}$
- **最小遅延解析 (ホールド/リムーバル) :**
 $\text{スラック} = \text{データ到着時間} - \text{データ所要時間}$

次の図に、テキスト形式のレポートのソース クロック、データ、およびデスティネーション クロック パスの部分を示します。パスには 5ns の周期制約が適用されているので、ソース クロックの送信エッジは 0ns で開始し、デスティネーション クロックの受信エッジは 5ns で開始します。

Location	Delay type	Incr(ns)	Path(ns)	Netlist Resource(s)

	(clock bftClk rise edge)	0.000	0.000	r
W17		0.000	0.000	r bftClk
	net (fo=0)	0.000	0.000	bftClk
W17	IBUF (Prop_ibuf_I_O)	0.868	0.868	r bftClk_IBUF_inst/O
	net (fo=1, routed)	1.901	2.769	bftClk_IBUF
BUFGCTRL_X0Y0	BUFG (Prop_bufg_I_O)	0.093	2.862	r bftClk_IBUF_BUGF_inst/O
	net (fo=746, routed)	1.261	4.123	bftClk_IBUF_BUGF
DSP48_X2Y30				r ct0/xOutReg_reg/CLK

DSP48_X2Y30	DSP48E1 (Prop_dsp48e1_CLK_P[15])	0.348	4.471	r ct0/xOutReg_reg/P[15]
	net (fo=1, routed)	0.555	5.026	arnd2/ct0/P[15]
SLICE_X47Y78	LUT2 (Prop_lut2_I_O)	0.043	5.069	r arnd2/ct0/i_271/O
	net (fo=66, routed)	1.536	6.605	rnd2_3[0][15]
DSP48_X1Y28				r transformLoop[0].ct0/xOutReg_reg/C[36]
DSP48_X1Y28	DSP48E1 (Setup_dsp48e1_CLK_C[36])	0.164	6.769	transformLoop[0].ct0/xOutReg_reg

	(clock bftClk rise edge)	5.000	5.000	r
W17		0.000	5.000	r bftClk
	net (fo=0)	0.000	5.000	bftClk
W17	IBUF (Prop_ibuf_I_O)	0.782	5.782	r bftClk_IBUF_inst/O
	net (fo=1, routed)	1.787	7.569	bftClk_IBUF
BUFGCTRL_X0Y0	BUFG (Prop_bufg_I_O)	0.083	7.652	r bftClk_IBUF_BUGF_inst/O
	net (fo=746, routed)	1.145	8.797	bftClk_IBUF_BUGF
DSP48_X1Y28				r transformLoop[0].ct0/xOutReg_reg/CLK
	clock pessimism	0.223	9.020	
	clock uncertainty	-0.561	8.458	

	required time		8.458	
	arrival time		-6.769	

	slack		1.690	

図 63 : テキスト形式レポートのタイミング パスの詳細

図 64 に、図 63 を Vivado IDE で表示した例を示します。

Source Clock Path				
Delay Type	Delay	Cumulative	Location	Logical Resource
(clock bftClk rise edge)	(r) 0.000	0.000		
net (fo=0)	(r) 0.000	0.000	W17	bftClk
IBUF (Prop_ibuf_I_O)	(r) 0.868	0.868	W17	bftClk_IBUF_inst/O
net (fo=1, routed)	1.901	2.769		bftClk_IBUF
BUFG (Prop_bufg_I_O)	(r) 0.093	2.862	BUFGCTRL_X0Y0	bftClk_IBUF_BUGF_inst/O
net (fo=746, routed)	1.261	4.123		bftClk_IBUF_BUGF
			DSP48_X2Y30	ct0/xOutReg_reg/CLK
Data Path				
Delay Type	Delay	Cumulative	Location	Logical Resource
DSP48E1 (Prop_dsp48e1_CLK_P[15])	(r) 0.348	4.471	DSP48_X2Y30	ct0/xOutReg_reg/P[15]
net (fo=1, routed)	0.555	5.026		arnd2/ct0/P[15]
LUT2 (Prop_lut2_I0_O)	(r) 0.043	5.069	SLICE_X47Y78	arnd2/ct0/i_271/O
net (fo=66, routed)	1.536	6.605		rnd2_3[0][15]
			DSP48_X1Y28	transformLoop[0].ct0/xOutReg_reg/C[36]
DSP48E1 (Setup_dsp48e1_CLK_C[36])	0.164	6.769	DSP48_X1Y28	transformLoop[0].ct0/xOutReg_reg
Arrival Time		6.769		
Destination Clock Path				
Delay Type	Delay	Cumulative	Location	Logical Resource
(clock bftClk rise edge)	(r) 5.000	5.000		
net (fo=0)	(r) 0.000	5.000	W17	bftClk
IBUF (Prop_ibuf_I_O)	(r) 0.782	5.782	W17	bftClk_IBUF_inst/O
net (fo=1, routed)	1.787	7.569		bftClk_IBUF
BUFG (Prop_bufg_I_O)	(r) 0.083	7.652	BUFGCTRL_X0Y0	bftClk_IBUF_BUGF_inst/O
net (fo=746, routed)	1.145	8.797		bftClk_IBUF_BUGF
			DSP48_X1Y28	transformLoop[0].ct0/xOutReg_reg/CLK
clock pessimism	0.223	9.020		
clock uncertainty	-0.561	8.458		
Required Time		8.458		

図 64 : Vivado IDE のタイミング パスの詳細

パスの情報は、5 つの列で示されます。

- [Location] : デバイス上でセルまたはポートが配置されている位置を示します。
- [Delay Type] : Unisim プリミティブとパスが通過するタイミング アークを示します。ネットの場合、ファンアウト (fo) およびそのステータスを示します。ネットのステータスは、次のとおりです。
 - Unplaced : ドライバーおよびロードは配置されていません。
 - Estimated : ドライバーまたはロード、あるいはその両方が配置されています。部分的に配線されているネットも Estimated と示されます。
 - Routed : ドライバーおよびロードの両方が配置され、ネットが完全に配線されています。
- [Incr(ns)] (テキスト レポート) / [Delay] (IDE レポート) : Unisim プリミティブのタイミング アークまたはネットに関連するインクリメンタル遅延値を示します。入力/出力遅延、クロックのばらつきなどの制約を示すこともあります。
- [Path(ns)] (テキスト レポート) / [Cumulative] (IDE レポート) : パスの各セグメントまでの累積遅延を示します。値は、前の行までの累積値と、その行のインクリメント遅延を加算したものです。
- [Netlist Resource(s)] (テキスト レポート) / [Logical Resource] (IDE レポート) : 通過したネットリスト オブジェクトの名前を示します。

各インクリメンタル遅延は、立ち上がりエッジ (r) または立ち下がりエッジ (f) に関連付けられています。初期のエッジは、パスに選択されている送信エッジまたは受信エッジにより決定されますが、タイミング アークの特性によって、パス上のセルで変換されることがあります。たとえば、インバーターの入力の立ち上がりエッジは、出力では立ち下がりエッジになります。必要以上に厳しいタイミング パス要件が、ソース クロック ツリーまたはデスティネーション クロック ツリーによるクロック エッジの反転によるものである場合があります。

クロージャ テクニック

この章では、次を含むタイミング クロージャの手法を説明します。

- タイミング制約の調整
- フロアプラン
- インプリメンテーション ツールのエフォートレベルの増加
- デザインのタイミングを自動または手動で向上させる

入力制約およびソースのチェック

デザインおよびタイミング入力が適切であることを確認します。

- ネットリストが質のよいものであることを確認します。

Vivado™ 合成を使用する場合は、合成特定のタイミング制約を XDC ファイルに追加します。合成はタイミングドリブンであり、タイミングを満たしやすくなるようにゲートが調整されます。

- [Schematic] または [Clock Networks] ビューでクロック ツリーを確認します。

クロック ツリーが適切なものであることを確認します。BUFG が別の BUFG や LUT を駆動していると、クロック スキューが大きくなります。追加のクロック スキューによりホールド タイミング制約が厳しくなり、システムのタイミングが悪化することがあります。配置 DRC でクロック ツリーの問題が警告として表示されることがあります。

- デザインのピン配置およびフロアプランを実行する際、クロックリソースを考慮します。

7 シリーズ FPGA デバイスのクロック領域では、領域ごとに 12 個のグローバル クロックがサポートされます。クロック ツリーの配置には、その他の制限もあります。

- クロック周期がデザインで必要なものであるかどうかを確認します。

デザインの制約を厳しくしすぎると、不必要に厳しいタイミング制約のためにランタイムが長くなります。タイミングエラーが発生したり、消費電力が大きくなる原因にもなります。不必要に厳しい制約が満たされない場合は、実際の制約を試してみてください。Vivado IDE では、配置配線を変更せずにタイミング制約を変更できます。実際のクロック周期で `report_timing_summary` を再実行します。

- タイミングを満たしていないパスが複数サイクル パスまたはフォルス パスでないかを確認します。

XDC では、クロックドメインをまたがるパスが本物のパスであると想定されます。これは、ISE および UCF とは異なります。制御ロジックの構造により到達不可能なパスがある可能性もあります。フォルス パス タイミング制約を使用してください。

ステート マシンで駆動されるクロック イネーブルなどの制御構造では、複数サイクル パスが生成されます。データがソースからデスティネーションに到達するのに複数のクロック周期が必要です。複数サイクル タイミング制約を使用してください。

- インプリメンテーション後に、タイミング制約を変更してタイミングを再実行できます。タイミング制約の変更がデザインのタイミングにどのように影響するかを調べるのに、インプリメンテーションを再実行する必要はありません。制約を変更してタイミングを確認し、必要に応じてタイミング制約を調整していきます。

- タイミング制約を変更してもタイミングが満たされない場合は、インプリメンテーションを再実行します。インプリメンテーションはタイミング ドリブンであり、問題のある箇所に焦点が置かれます。配線後のデザインのトータル ホールド スラック (THS) が大きかった場合は、インプリメンテーションを再実行する必要があります。

ヒント: 新しい制約を保存するのに [Save Constraints As] を使用し、元の制約を保持しておくくと便利です。

- タイミング パスを確認します。

クロック スキューおよびジッターが適切であることを確認します。

- ロジックを確認します。

周期に対するロジック遅延を確認し、ロジック遅延の周期に占める割合が高い場合は、再合成してロジック遅延を削減する必要があります。ロジック遅延を削減するには、次の 2 つの方法があります。

- RTL を記述し直して配線遅延を削減します。

ファンアウトの大きいネットがある場合は、ドライバーを複製してみてください。その場合、配置を考慮してください。各ドライバーがローカルのロードと接続されるようにし、ロードが分散しないようにします。

- 配線遅延のタイミング パスに占める割合が高い場合は、フロアプランを考慮します。

目的は、配線遅延を削減して、クリティカル パスのタイミングを改善することです。フロアプランによりクリティカル パスを構成するロジックは変更されません。フロアプランの際は、デザインのブロック同士の接続を確認してください。RTL の構造を変更した方がよい場合もあります。合成時には、配置はわかっていません。合成でファンアウトの大きいネットが複製される場合、配置は考慮されません。手動で複製して配線し直した方が、デザインが向上する場合があります。たとえば、2 つのメモリ インターフェイスがあり、チップの反対側の角に配置される場合、RTL ソースで制御信号を複製すると有益です。合成属性を使用して、論理的に等価なレジスタが最適化で削除されないように設定する必要がある場合もあります。

ツールのエフォート レベルの増加

インプリメンテーション ツールのエフォート レベルを増加してみます。インプリメンテーションに時間をかけるだけで、デザインのタイミングが満たされる場合があります。HighEffort ストラテジを使用してみてください。デフォルトでは、インプリメンテーション run は中程度のエフォート レベルで実行されます。place_design および route_design コマンドに -effort_level high オプションを使用すると、ランタイムは長くなりますが、別の方法でタイミングを満たすよう試みられます。

HighEffortPhysSynth ストラテジを使用してみてください。エフォート レベルが増加され、オプションの phys_opt_design が実行されます。このストラテジでは、ランタイムは増加しますが、タイミングは向上します。

[Flow] → [Create Runs] をクリックして、異なるストラテジを使用した複数の run を作成し、実行してみてください。

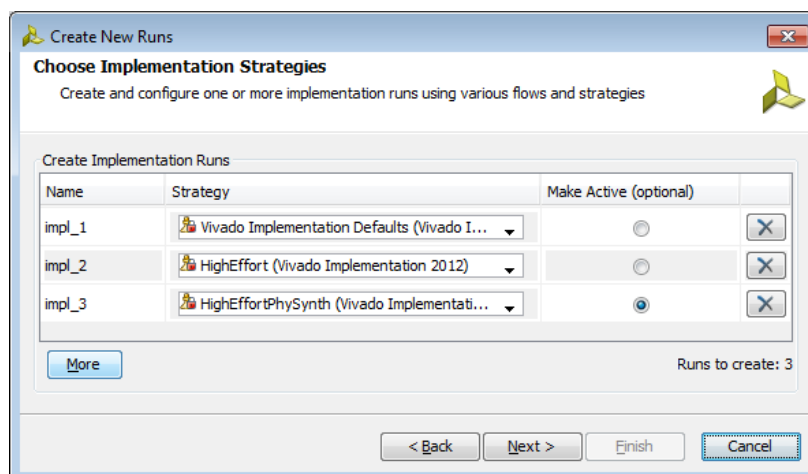


図 65 : 複数の run

インプリメンテーション結果を比較してデザインに最適なストラテジを見つけ、その後の実行ではそのストラテジを使用します。

ヒント : 最適なストラテジは、デザイン間およびツール リリース間で異なる場合があります。

手動フロアプラン

フロアプランがデザインのタイミングを満たすのに役立つ場合があります。フロアプランは、次のような場合に実行します。

- タイミングが満たされる場合と満たされない場合がある。
- タイミングが満たされることがない。

フロアプランは、複数の設計者と作業する場合や、一貫性が重要な場合にも有益です。

フロアプランでは、配線遅延を削減することにより、セットアップ遅延 (TNS WNS) を向上できます。インプリメンテーションでは、セットアップ、ホールド、およびコンポーネント スイッチ制限がレポートされますが、フロアプランで向上できるのはセットアップ スラックのみです。

手動フロアプランは、ネットリストが階層がある方が簡単です。合成でネットリスト全体がフラット化されると、デザイン解析に時間がかかります。階層ネットリストが生成されるように、合成オプションを設定します。

- XST
 - `-keep_hierarchy=no` (デフォルト) および `-netlist_hierarchy = rebuilt`
 - PlanAhead™ Defaults ストラテジ
- Vivado 合成
 - `-flatten_hierarchy = rebuilt`
 - Vivado Synthesis Defaults ストラテジ

論理パスが複雑な大型の階層ブロックは、解析が困難です。異なる論理構造が別の下位階層に含まれているデザインの方が解析は簡単です。階層モジュールのすべての出力にレジスタを付けることを考慮してください。複数の階層ブロックを通過するパスの配置を解析するのは困難です。

フロアプランの基礎

すべてのデザインでタイミングが満たされるとは限りません。ツールがソリューションを見つけられるよう、指示する必要があります。フロアプランを使用すると、高レベルの階層レイアウトや詳細なゲート配置を指定できます。

配線遅延を削減したり、クリティカルでないブロックのロジック集積度を増加するなどしてパフォーマンスを向上するため、フロアプランすることを考慮してみてください。ロジック集積度は、チップ上にロジックがどれだけ密にパックされるかを示します。

フロアプランを実行すると、クロック周波数および結果の一貫性を向上できます。

フロアプランには複数の取り組み方があり、それぞれに利点と欠点があります。

詳細なゲートレベルのフロアプラン

詳細なゲートレベルのフロアプランでは、個々のロジック エLEMENTをデバイスの特定のサイトに配置します。

詳細なゲートレベルのフロアプランの利点

- ネットを手動で配線します。
- デバイスの最大限のパフォーマンスを引き出せる可能性があります。

詳細なゲートレベルのフロアプランの欠点

- 時間がかかります。
- デバイスおよびデザインに関する豊富な知識が必要です。
- ネットリストが変更された場合、フロアプランを実行し直すことが必要な場合があります。

推奨：詳細なゲートレベルのフロアプランは、最終的な手段としてください。

情報の再利用

タイミングが満たされたデザインの情報を再利用します。

この手法は、デザインのタイミングが満たされたり満たされなかったりする場合に使用します。

情報を再利用するには、次の手順に従います。

- 1. 2 つのインプリメンテーション run を開きます。
 - タイミングが満たされている run
 - タイミングが満たされていない run

ヒント：複数のモニターを使用している場合、[Open Implementation in New Window] コマンドを使用してデザインを別のウィンドウで開きます。詳細は、「[複数のビュー](#)」を参照してください。

- 2. 2 つのデザインの違いを見つけます。
 - report_timing_summary で、タイミングが満たされていないパスを特定します。
 - タイミングが満たされているデザインで、report_timing を min_max モードで使用して、同じパスのタイミング解析を実行します。

- 3. タイミング結果を比較します。
 - クロック スキュー
 - ロジック遅延
 - 配置
 - 配線遅延
- 4. パスのエンドポイント間のロジック遅延に差がある場合は、合成 run に戻ります。

エレメントの配置の確認

デザインのエレメントの配置を確認します。

2 つの I/O レポートを比較して、I/O の配置と I/O 規格を確認します。すべての I/O が配置されていることを確認します。次の図に示すように検索を実行すると、配置が固定されていない I/O を検索できます。

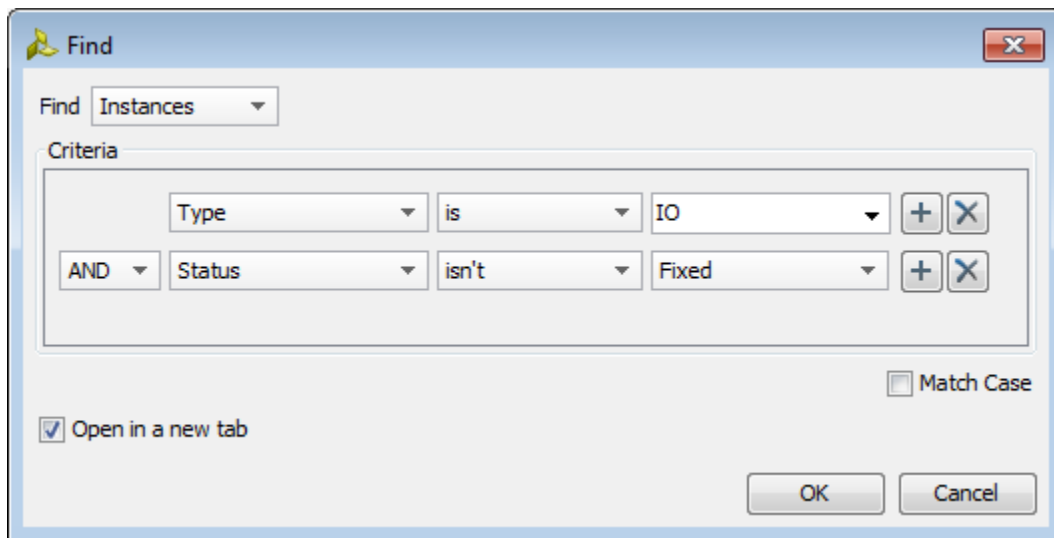


図 66 : 配置が固定されていない I/O

すべてのクロック プリミティブを、タイミングが満たされた run に基づいて配置することを考慮します。次の図に示すように、クロック使用量レポートにクロック ツリー ドライバーの配置がリストされます。

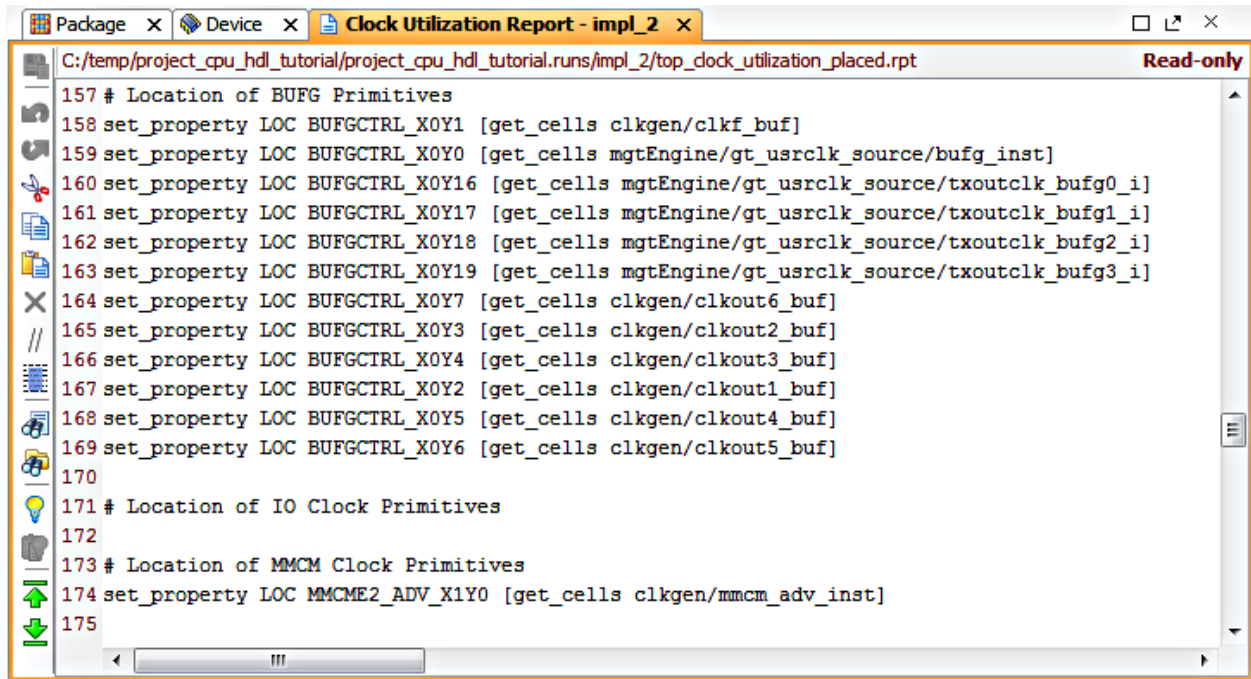


図 67 : クロックのロケーション

LOC 制約は、XDC 制約ファイルにコピーできます。

多くのデザインでは、ブロック RAM および DSP の配置を再利用することにより、タイミングを満たすことができます。
[Edit] → [Find] を使用して、インスタンスをリストします。

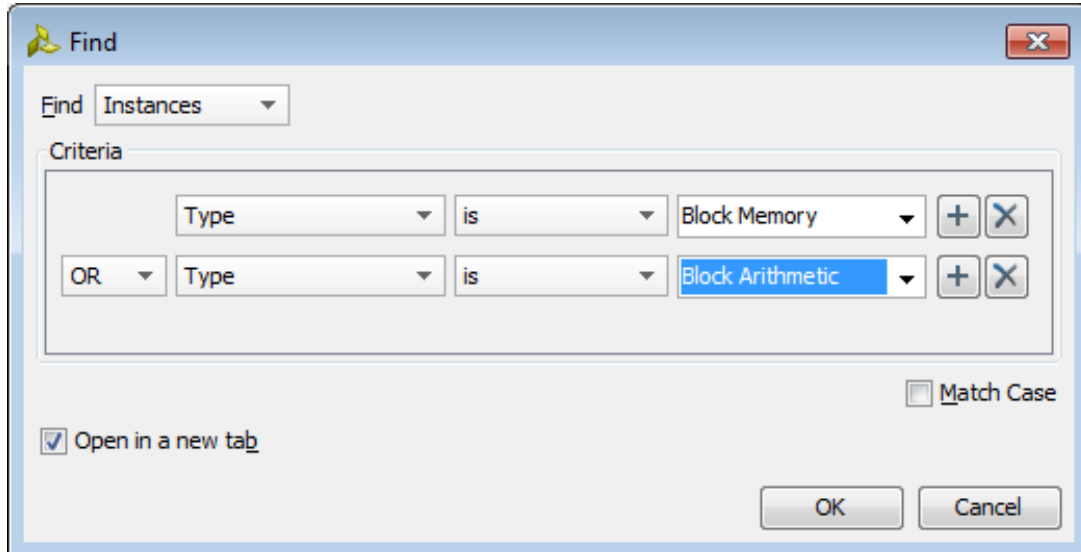


図 68 : ブロック RAM または DSP を検索

XDC に配置制約を追加して、ロジックを固定します。

1. [Find Results] ビューでマクロを選択します。
2. 右クリックします。
3. [Fix Instances] をクリックします。

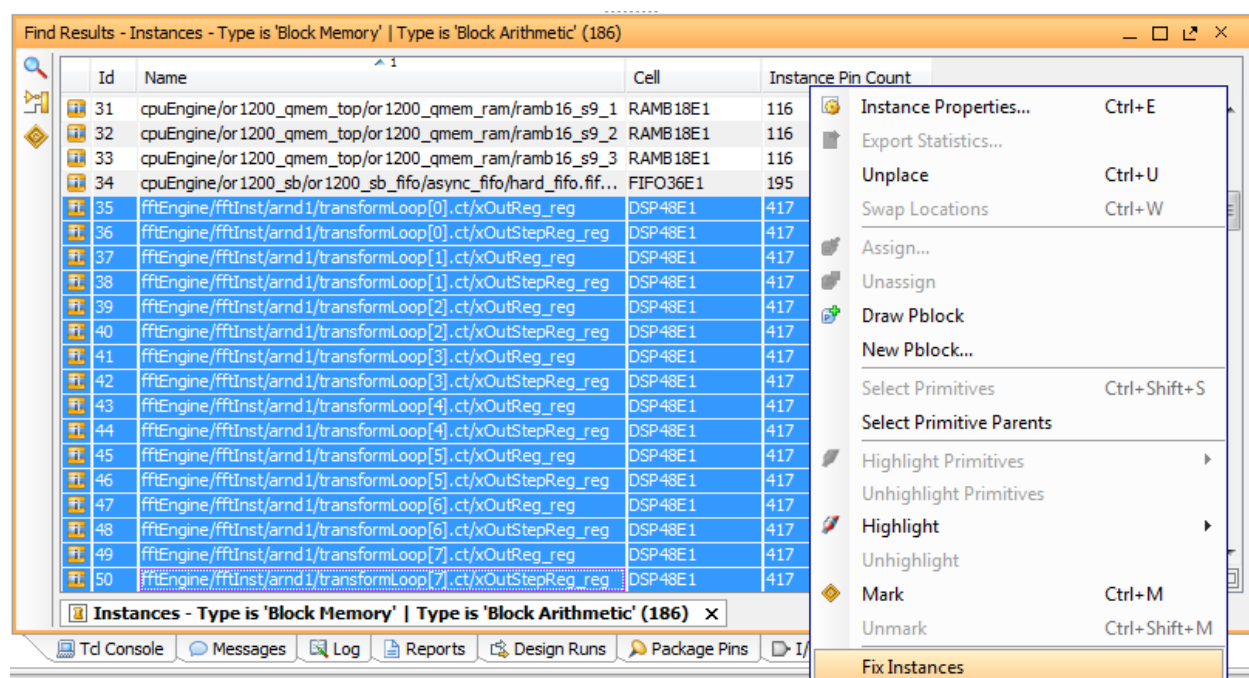


図 69 : 固定するロジックの選択

推奨 : 配置を固定する前に、階層名に基づいて配置を解析し、ハイライトしてください。

次のものの配置を再利用するのが比較的簡単です。

- I/O
- グローバル クロック リソース
- ブロック RAM
- DSP マクロ

配置を再利用すると、ネットリストのリビジョン間での結果がより一貫したものになります。これらのプリミティブの名前は通常変化しないので、配置を保持するのは簡単です。

ヒント : 汎用スライス ロジックの配置を再利用しないでください。デザインの変更する予定の部分は、配置を再利用しないでください。

ゲートレベルのフロアプラン

タイミングが満たされておらず、ネットリストまたは制約の変更が好ましくないデザインでは、ゲートのフロアプランを考慮します。

推奨 : ゲートレベルのフロアプランの前に、階層フロアプランを試してみてください。

階層フロアプラン

階層フロアプランでは、チップ上の領域に階層レベルを配置できます。この領域が、配置でのガイドラインとなります。配置ツールにより、詳細な配置が決定されます。

階層フロアプランは、ゲートフロアプランよりも短時間で実行できます。適切なフロアプランにより、タイミングを改善できます。フロアプランは、デザインが変更されても保持されます。

階層レベルが、ゲートのコンテナとなります。通常、ネットリストが変更されても機能します。

ゲートの手動配置

ゲートの手動配置では、デバイスの最大限のパフォーマンスを引き出せる可能性があります。この手法は通常、デザインの小さい部分でのみ使用されます。たとえば、高速 I/O インターフェイス周辺のロジックや、ブロック RAM および DSP を配置する場合などです。手動での配置には時間がかかります。

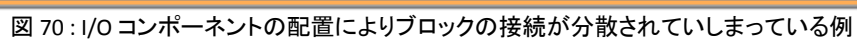
すべてのフロアプラン手法には、多大なエンジニアリング時間がかかります。作業を繰り返して調整する必要がある場合もあります。ゲート名が変更された場合、フロアプラン制約もアップデートする必要があります。

階層フロアプランは、次のようになります。

- クリティカル パスを含む下位階層を特定します。
- 最上位フロアプランを使用して、どこに配置するかを判断します。
- 個々のゲートは、インプリメンテーションにより配置されます。
- ゲートおよびタイミング パスに関する包括的な知識があります。
- 通常、適切な詳細配置が実行されます。

フロアプランの際は、最終的なピン配置を念頭においてください。I/O が固定されていると有益です。I/O はフロアプランを開始する際のアンカー ポイントとなります。I/O に接続されているロジックは、固定されているピンの近くに移動されます。

ヒント : I/O に接続されているブロックは、I/O の近くに配置してください。ピン配置によりブロックの I/O への接続が分散されている場合は、ピン配置または RTL を変更することを考慮してください。



上図のようなフロアプランは、タイミングには有益ではありません。ブロックを分離するか、ソースコードを変更するか、ブロック RAM および DSP のみを制約することを考慮してください。

上記の Pblock は、次の XDC 制約で表されます。

```
create_Pblock Pblock_fftEngine
add_cells_to_Pblock [get_Pblocks Pblock_fftEngine] [get_cells -quiet [list
usbEngine1]]
resize_Pblock [get_Pblocks Pblock_fftEngine] -add {SLICE_X8Y105:SLICE_X23Y149}
resize_Pblock [get_Pblocks Pblock_fftEngine] -add {DSP48_X0Y42:DSP48_X1Y59}
resize_Pblock [get_Pblocks Pblock_fftEngine] -add {RAMB18_X0Y42:RAMB18_X1Y59}
resize_Pblock [get_Pblocks Pblock_fftEngine] -add {RAMB36_X0Y21:RAMB36_X1Y29}
```

階層レベルを Pblock に割り当てている行があります。4 つのリソースタイプがあり、それぞれグリッドがあります。グリッドで制約されていないロジックは、デバイスのどこにでも配置できます。階層レベルのブロック RAM のみを制約するには、ほかの Pblock グリッドをディセーブルにします。

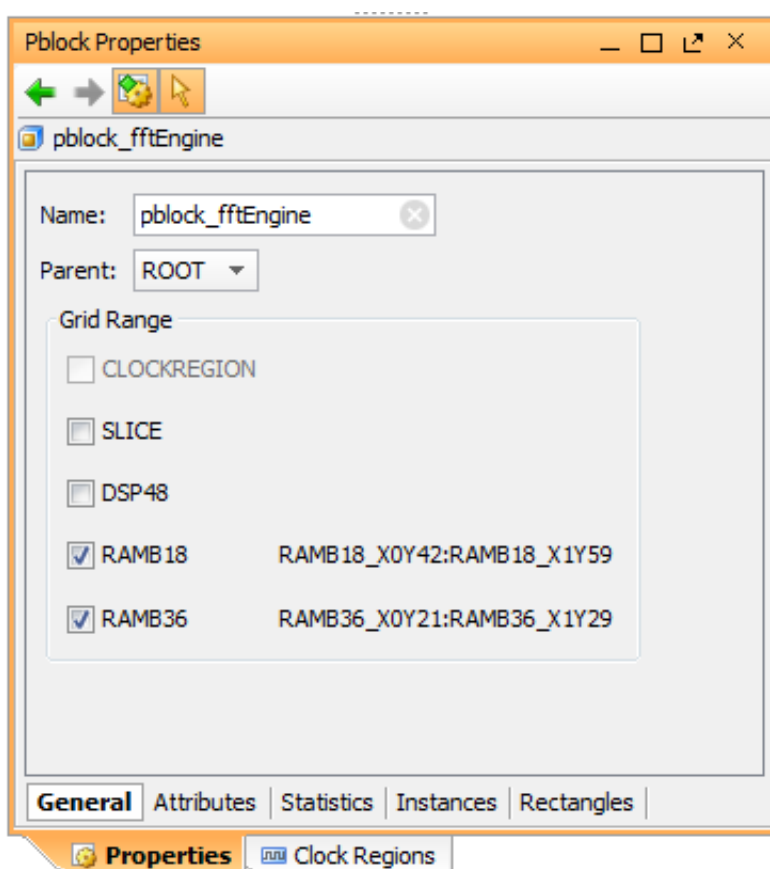


図 71 : Pblock グリッド

結果の XDC 制約では、Pblock が簡略化されます。

```
create_Pblock Pblock_fftEngine
add_cells_to_Pblock [get_Pblocks Pblock_fftEngine] [get_cells -quiet [list
usbEngine1]]
resize_Pblock [get_Pblocks Pblock_fftEngine] -add {RAMB18_X0Y42:RAMB18_X1Y59}
resize_Pblock [get_Pblocks Pblock_fftEngine] -add {RAMB36_X0Y21:RAMB36_X1Y29}
```

ブロック RAM がデバイスに制約されますが、スライス ロジックは I/O 内に配置できます。

ヒント: ロジックをフロアプランする際は、階層を中央の Config ブロックをまたがるように配置しないでください。

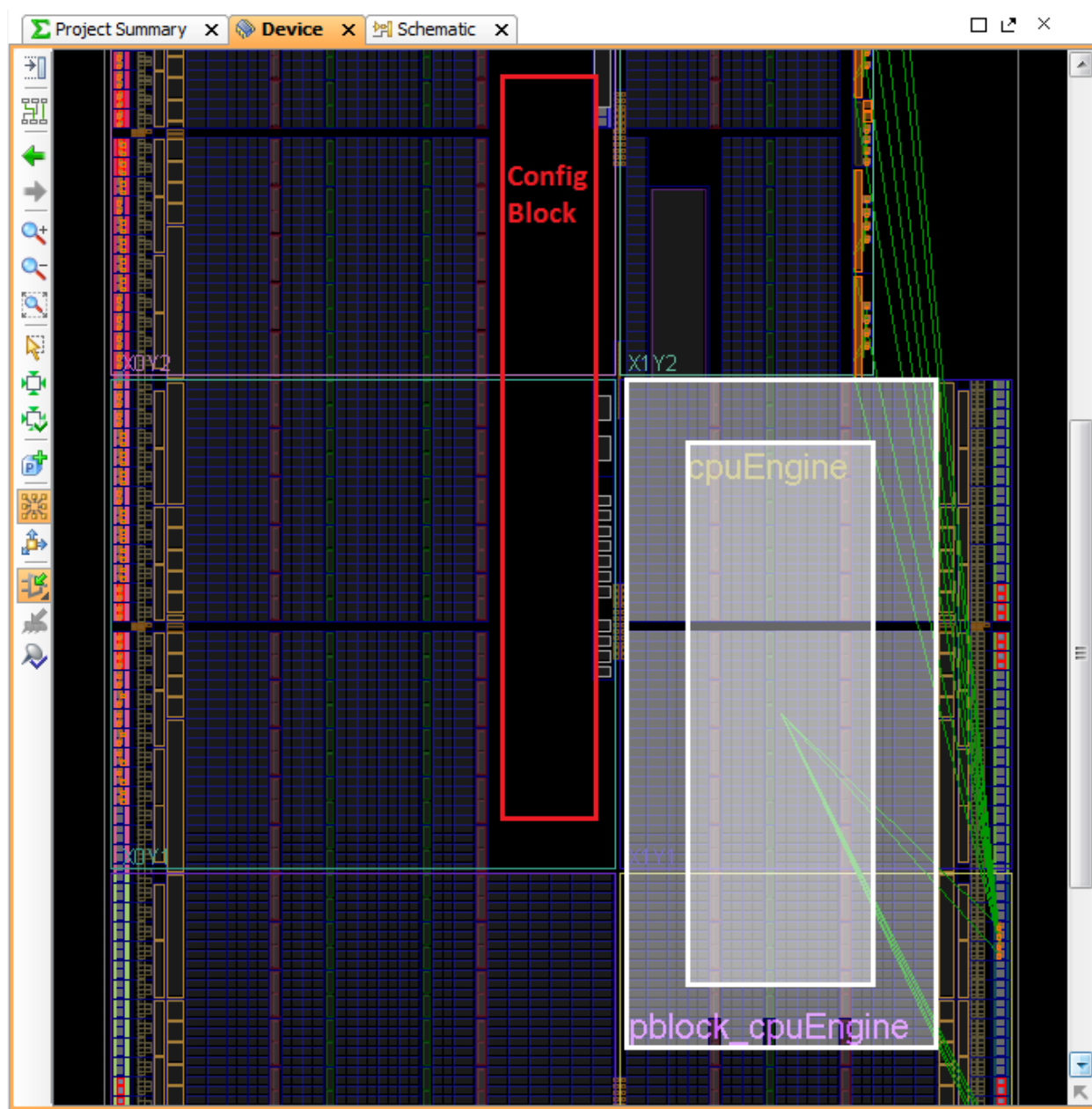


図 72 : Config ブロックをまたがるように配置しない

推奨 : config ブロックをまたがると、配線が増加します。Pblock を config ブロックの左右どちらかに配置してください。

スタックド シリコン インターコネクト (SSI)

スタックド シリコン インターコネクト (SSI) パーツでは、追加の考慮事項があります。SSI パーツは複数のダイ (SLR (Super Logic Region)) で構成されており、インターポーザーにより結合されています。インターポーザーの接続は、SLL (Super Long Line) と呼ばれます。1 つの SLR から別の SLR への移動には多少の遅延があります。

デザインの設計、ピン配置、フロアプランの際には、SLR を考慮してください。クリティカル タイミング パスは 1 つの SLR 内に収め、SLL を介する接続を最小限に抑えてください。

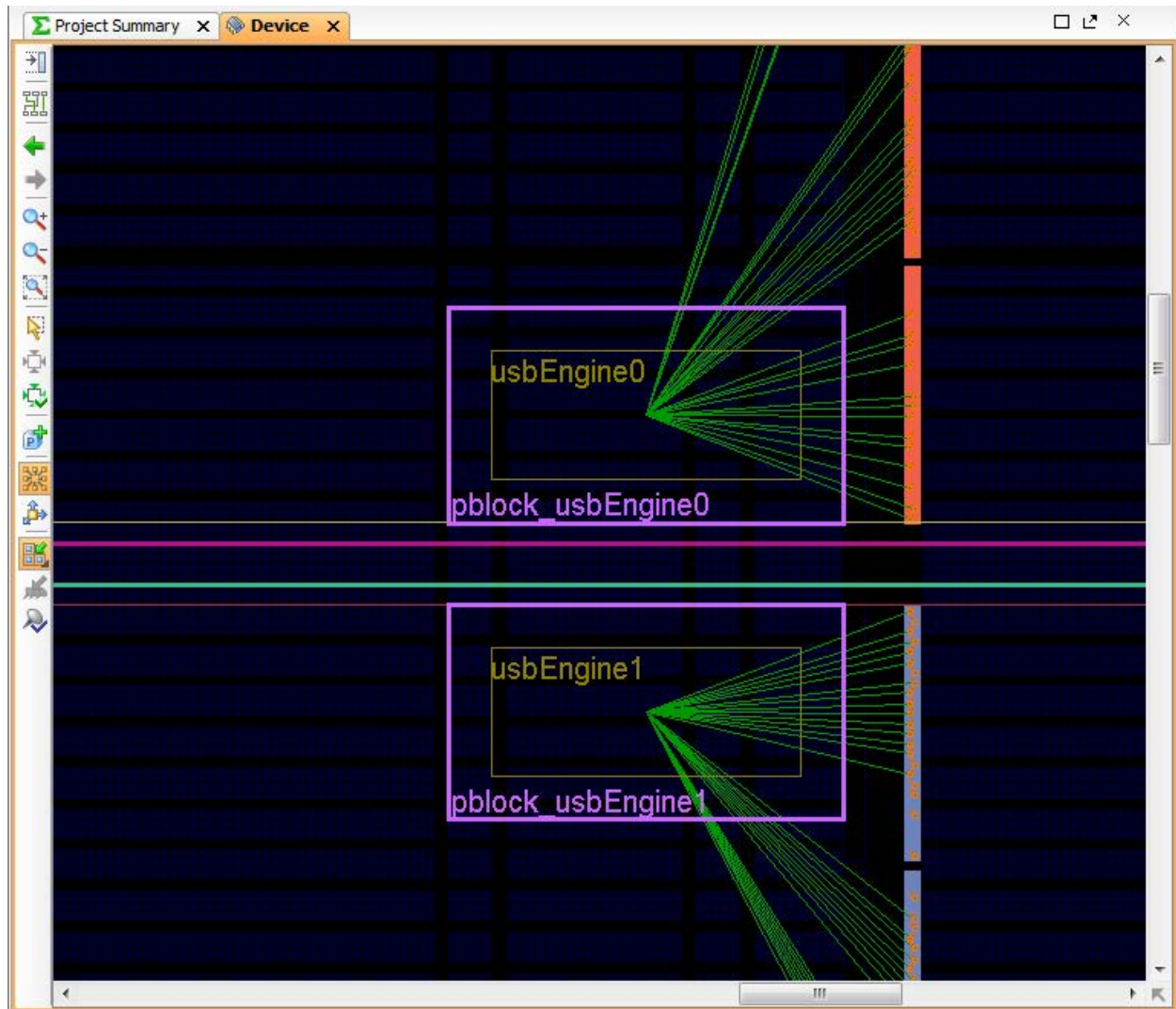


図 73 : SLL を介する接続を最小限に抑える

I/O は、関連する I/O インターフェイスと同じ SLR に配置されます。SSI パーツのロジックを配置する場合は、クロックの配置も考慮する必要があります。

推奨： 分割を実行する前に、ロジックの配置を自動で実行してみてください。自動配置を解析することにより、考慮していなかったフロアプラン方法が見つかることがあります。

配線の変更

結果によっては、配線の変更が必要な場合があります。[Device] ビューを使用して、個々のネットに対して配線の解除 ([Unroute])、配線([Route])、配線の固定 ([Fix Routing]) を実行できます。これらのコマンドは、ネットのポップアップメニューから実行できます。

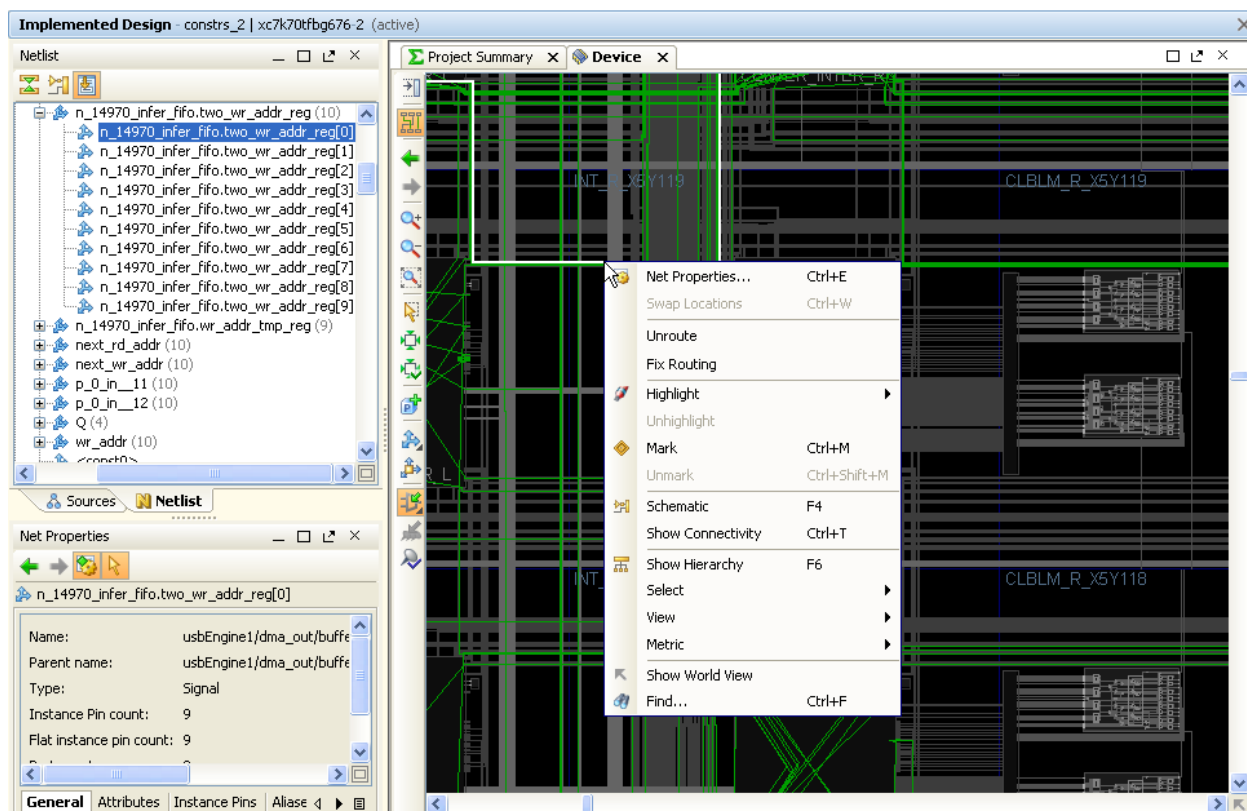


図 74 : ネットのポップアップ メニュー

ネットに関連するコマンドのみが表示されるので、すべてのネットにすべてのコマンドが表示されるわけではありません。

ロジックの変更

読み取り専用でない論理オブジェクトのプロパティは、インプリメンテーション後に Vivado IDE GUI または Tcl で変更できます。[Properties] ビューの [Attributes] タブを使用します。

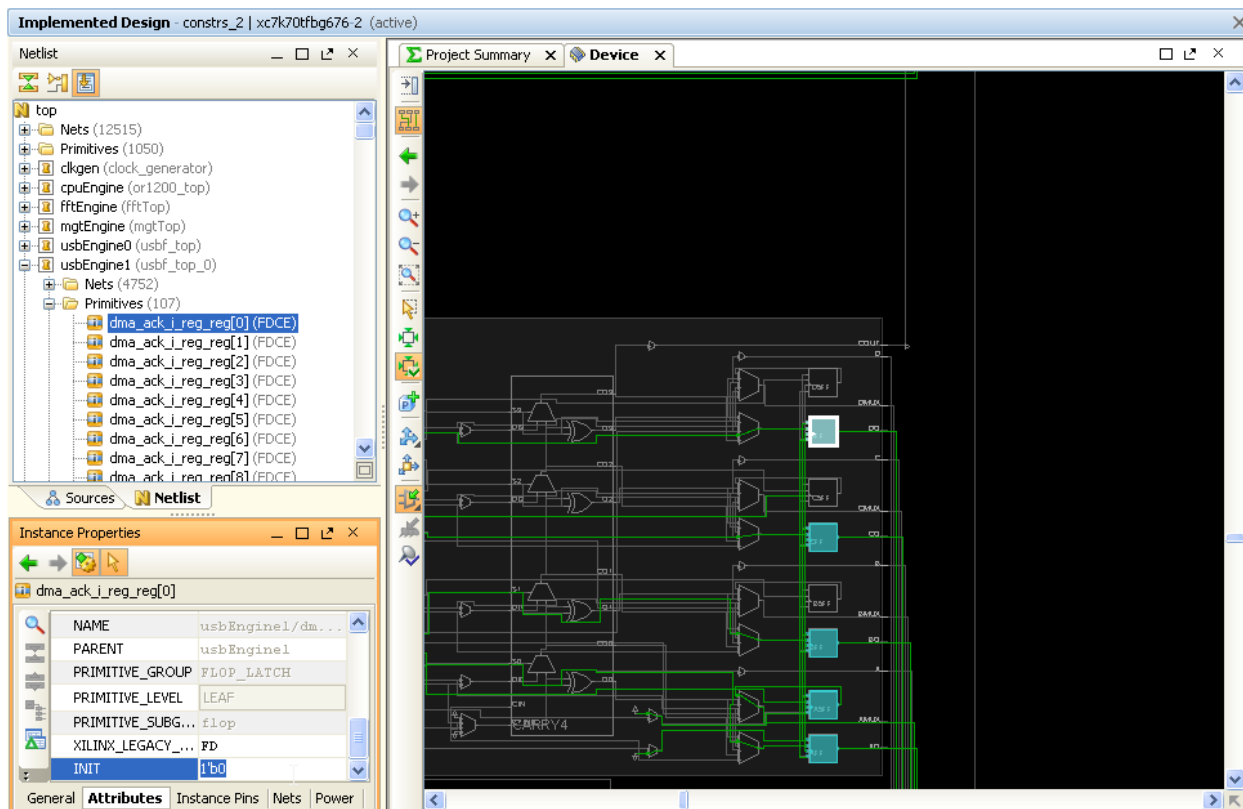


図 75 : プロパティの変更

変更可能なプロパティには、ブロック RAM の INIT、MMCM のクロックを変更するプロパティなどがあります。LUT オブジェクトの INIT を変更するためのダイアログ ボックスもあり、LUT 論理式を指定して適切な INIT 値がツールにより設定されるようにすることができます。

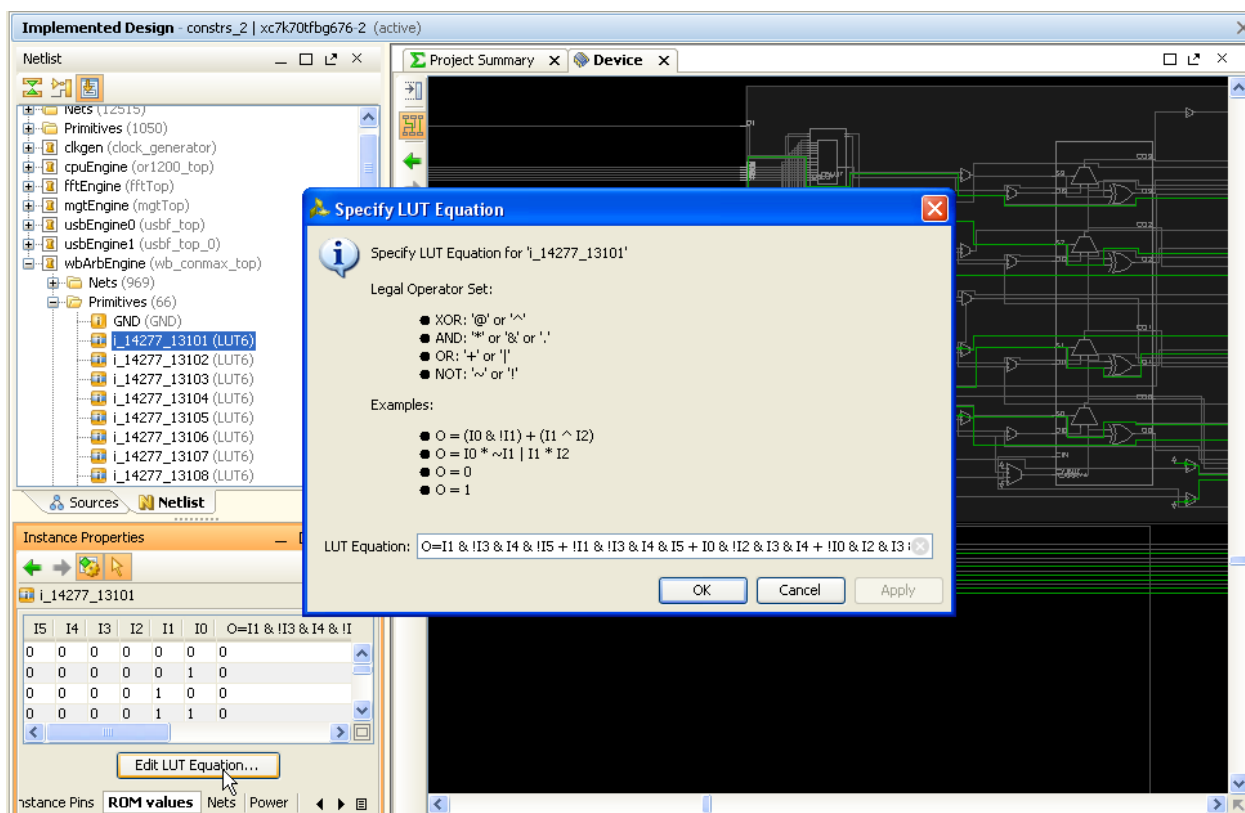


図 76 : LUT 論理式の指定

変更の使用

変更後は、チェックポイントを使用してメモリ内のデザインを保存することが推奨されます。論理的な変更は、論理デザインにバックアノテートされないので、次回の実行に適用されるようにするには、ソースまたは XDC を変更する必要があります。

その他のリソース

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、次のザイリンクス サポート サイトを参照してください。

<http://japan.xilinx.com/support>

ザイリンクス資料で使用する用語集は、次を参照してください。

<http://japan.xilinx.com/company/terms.htm>

ソリューション センター

デバイス、ツール、IP のサポートについては、[ザイリンクス ソリューション センター](#)を参照してください。トピックには、デザイン アシスタンス、アドバイザリ、トラブルシュートヒントなどが含まれます。

リファレンス

このガイドの補足情報は、次のサイトにリストされている資料を参照してください。

Vivado™ Design Suite 2012.2 資料ページ

http://japan.xilinx.com/support/documentation/dt_vivado_vivado2012-2.htm