

Vivado Design Suite Tcl コマンド リファレンス ガイド

UG835 (v 2013.2) 2013 年 6 月 19 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 2002–2013 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. The PowerPC name and logo are registered trademarks of IBM Corp., and used under license. All other trademarks are the property of their respective owners.

本資料は英語版(v 2013.2)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。

概要

Vivado の Tcl 機能の概要

Tcl (Tool Command Language) は Vivado® ツール環境に統合されているスクリプト言語です。Tcl は、アプリケーション プログラム インターフェイスの半導体業界標準言語で、SDC (Synopsys® Design Constraints) に使用されます。

SDC は、Synopsis 社の Synplify やその他のベンダー ツールから FPGA 合成ツールへタイミング制約を渡すメカニズムで、タイミング制約の業界標準なので、スクリプト言語には Tcl が最も適しています。

Tcl を使用することで、自動スクリプトだけでなく、デザイン ツールに対するインタラクティブなクエリを実行できます。Tcl には、デザイン データベースからツール、デザイン設定、ステートなどに関する情報をインタラクティブに取り出す機能があります。たとえば、特定のタイミング解析レポートコマンドを検索したり、制約を適用してその直後にクエリを実行し、ツールの手順を再実行せずに動作が予測どおりかどうかを確認できます。

次のセクションでは、Vivado での基本的な Tcl 機能について説明します。

注記：このマニュアルは、Tcl 言語の包括的なりファレンスではなく、Vivado Design Suite の Tcl シェル特定の機能を説明するものです。Tcl プログラムの追加情報に関しては、参考資料を示します。

Vivado Design Suite の起動

異なる方法を使用して、Vivado Design Suite を起動し、ツールを実行できます。たとえば、非プロジェクト モードと呼ばれる Tcl スクリプト ベースのコンパイル スタイル方法を使用して、ソースおよびデザイン プロセスをユーザーが自分で管理できます。または、プロジェクト モードと呼ばれるプロジェクト ベースの方法を使用して、プロジェクトおよびプロジェクト ステータスによりデザイン プロセスおよびデザイン データが自動的に管理されるようにすることもできます。どちらの方法も、Tcl スクリプトを使用したバッチ モードで、または Vivado IDE でインタラクティブに実行できます。異なるデザイン フロー モードの詳細は、『Vivado Design Suite ユーザー ガイド：デザイン フロー の概要』(UG892) を参照してください。

Tcl

Tcl コマンドを直接使用する場合は、次のいずれかの方法を使用して Tcl コマンドでデザインを処理できます。

- ・ Vivado IDE 外で Vivado Design Suite Tcl シェルに個々の Tcl コマンドを入力する。
- ・ Vivado IDE の下部にある [Tcl Console] ビューに個々の Tcl コマンドを入力する。
- ・ Vivado Design Suite Tcl シェルから Tcl スクリプトを実行する。
- ・ Vivado IDE から Tcl スクリプトを実行する。

Vivado Design Suite Tcl シェルを起動するには、Linux コマンド プロンプトまたは Windows コマンド プロンプトに次のように入力します。

```
vivado -mode tcl
```

注記：Windows では、[スタート] → [すべてのプログラム] → [Xilinx Design Tools] → [Vivado 2013.x] → [Vivado 2013.x Tcl Shell] をクリックして起動することもできます。

Tcl および Tcl スクリプトの使用に関する詳細は、『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。Vivado ツールで Tcl を使用する手順ごとの詳細な説明は、『Vivado Design Suite チュートリアル：デザインフローの概要』(UG888) を参照してください。

Tcl

ツールを起動するときに Tcl スクリプトを指定することにより、Vivado ツールをバッチ モードで使用できます。Linux コマンド プロンプトまたは Windows コマンド プロンプトに次のように入力します。

```
vivado -mode batch -source <your_Tcl_script>
```

Vivado Design Suite Tcl シェルが開き、指定の Tcl スクリプトを実行して、スクリプトが完了すると閉じます。バッチ モードでは、複数の Tcl スクリプトをキューに入れ、夜間に複数のデザインに対して合成、シミュレーション、インプリメンテーションを実行し、翌朝結果を確認できます。

Vivado IDE

GUI を使用する場合は、Windows または Linux で Vivado IDE を起動します。Vivado IDE 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。

Vivado IDE は作業ディレクトリから起動してください。デフォルトでは、Vivado ジャーナル ファイル、ログ ファイル、生成レポート ファイルは、Vivado ツールが起動されたディレクトリに生成されます。これにより、起動ディレクトリに書き込まれるプロジェクト ファイル、ログ ファイル、ジャーナル ファイルが見つけやすくなります。

Windows OS で [スタート] → [すべてのプログラム] → [Xilinx Design Tools] → [Vivado 2013.x] → [Vivado 2013.x] をクリックします。

注記：または、Windows デスクトップの Vivado IDE のショートカットをダブルクリックします。

Linux OS では、コマンド プロンプトに次のコマンドを入力します。

```
vivado  
-or-  
vivado -mode gui
```

Vivado ツール コマンド ライン実行ファイルのヘルプを表示するには、次のように入力します。

```
vivado -help
```

Vivado ツールを Vivado Design Suite Tcl シェルから実行している場合は、Tcl シェルから **start_gui** コマンドを使用して直接 Vivado IDE を開くことができます。

Vivado IDE を閉じて Vivado Tcl シェルに戻るには、Vivado IDE で **stop_gui** コマンドを使用します。

Tcl

Vivado ツールを起動すると、デザイン セッション中に実行されるコマンドおよび操作の記録が `vivado.log` ファイルに記述されます。`vivado.jou` というファイルも生成され、セッション中に実行された Tcl コマンドのジャーナルのみが記述されます。このジャーナルファイルを基に、新しい Tcl スクリプトを『作成できます。

注記：このファイルのバックアップ バージョンが `vivado_<id>.backup.jou` で、以前に Vivado ツールを起動したときのセッションで実行した Tcl コマンドの詳細が保存されます。`<id>` は、ログ ファイルおよびジャーナルファイルの複数のバックアップ バージョンを保存するための固有の識別子です。

Tcl ヘルプ

Tcl の `help` コマンドを使用すると、サポートされる Tcl コマンドの概要が表示されます。

- `help` : Tcl コマンド カテゴリのリストを返します。

help

コマンド カテゴリは、File I/O などの特定のファンクションを実行するコマンド グループです。

- `help -category category` : 指定したカテゴリのコマンドのリストを返します。

help -category object

この例では、オブジェクトを処理する Tcl コマンドのリストが返されます。

- `help pattern` : 指定した検索パターンに一致するコマンドのリストを返します。この構文を使用すると、コマンド グループから特定のコマンドをすばやく検索できます。

help get_*

この例では、「get_」で始まる Tcl コマンドのリストが返されます。

- `help command` : 指定したコマンドに関する詳細情報を表示します。

help get_cells

この例では、`get_cells` コマンドの詳細が表示されます。

- `help -args command` : 指定のコマンドの簡単な説明、コマンド構文、および各引数の説明を表示します。

help -args get_cells

- `help -syntax command` : 指定のコマンドの構文を表示します。

help -syntax get_cells

Tcl 初期化スクリプト

Vivado ツールを起動すると、次の 2箇所で Tcl 初期化スクリプトが検索されます。

1. ツールのインストール ディレクトリ : `installdir/Vivado/version/scripts/init.tcl`
2. ローカルのユーザー ディレクトリ :
 - Windows 7 : `%APPDATA%/Roaming/Xilinx/Vivado/init.tcl`
 - Linux : `$HOME/.Xilinx/Vivado/init.tcl`

説明 :

`installdir` : Vivado Design Suite のインストール ディレクトリ

`init.tcl` がいずれかまたは両方の場所で見つかった場合、まず Vivado ツールのインストール ディレクトリにあるファイルが読み込まれ、次にホーム ディレクトリにあるファイルが読み込まれます。

- インストール ディレクトリにある `init.tcl` ファイルを使用すると、企業またはデザイnergループですべてのユーザーに対して共通の初期化スクリプトをサポートできます。インストール ディレクトリから Vivado ツールを起動すると、そのディレクトリの `init.tcl` スクリプトが使用されます。
- ホーム ディレクトリにある `init.tcl` ファイルを使用すると、各ユーザーがそれぞれコマンドを追加したり、デザイン要件を満たすためにツールのインストール ディレクトリに含まれるコマンドを変更できます。
- Vivado Design Suite のインストールでは `init.tcl` ファイルは提供されません。特定のニーズに合わせて `init.tcl` ファイルを作成し、インストール ディレクトリまたはホーム ディレクトリに配置してください。

この `init.tcl` スクリプトは標準の Tcl コマンド ファイルで、Vivado ツールでサポートされるどの Tcl コマンドも含めることができます。次の文を追加すると、`init.tcl` から別の Tcl スクリプト ファイルを読み込むことができます。

`source path_to_file /file_name.tcl`

注記 : または、コマンド ラインから Vivado Design Suite を起動する際に `-init` オプションを指定することもできます。詳細は、`vivado -help` と入力してください。

Tcl スクリプトの実行

Tcl スクリプトは、コマンド ライン オプションの 1つとして指定するか、GUI で指定します。Vivado 統合設計環境 (IDE) で Tcl スクリプトを実行するには、[Tools] → [Run Tcl Script] をクリックします。

コマンド ラインから Tcl スクリプトを実行するには、次のコマンドを使用します。

```
source file_name
```

Tcl スクリプトを Vivado IDE から実行すると、進捗状況バーが表示され、スクリプトが終了するまですべての IDE 操作が実行できなくなります。

ランタイム中にスクリプトの実行を一時停止する方法はないので、標準的な OS でのプロセスを停止する方法 (`kill`) で強制終了するしかありません。この場合、最後に保存した後の作業が失われます。

Tcl コンソールに「`help source`」と入力すると、`source` コマンドに関する情報を表示できます。

Tcl.pre および Tcl.post フック スクリプトの使用

Tcl フック スクリプトを使用すると、合成 run またはインプリメンテーション run、あるいはインプリメンテーションの任意の段階の前 (tcl.pre) および後 (tcl.post) にカスタム Tcl スクリプトを実行できます。run を起動すると、定義済みの Tcl スクリプトが使用され、選択したストラテジに基づいて標準デザインフローが実行されます。Tcl フック スクリプトにより、カスタム レポートを生成するなど、前後にプロセスを追加してこの標準フローをカスタマイズできます。Tcl フック スクリプトには、標準の Tcl スクリプトを使用する必要があります。

デザインフローの各段階の前後でフック スクリプトを実行できます。一般的に、次のような使用法があります。

- ・ カスタム レポート：タイミング、消費電力、リソース使用率、またはユーザー定義の Tcl レポート
- ・ 一時的なパラメーター設定を使用して問題回避
- ・ フローの一部でタイミング制約を変更
- ・ ある段階を複数回実行 (phys_opt_design を複数回呼び出すなど)
- ・ ネットリスト、制約、またはデバイス プログラムの変更

Tcl フック スクリプトの定義方法は、『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。

一般的な Tcl 構文のガイドライン

Tcl では、OS に関係なく Linux のファイル区切り文字 (/) が使用されます。

次のセクションでは、Vivado Design Suite で Tcl を使用する際の一般的な構文ガイドラインについて説明します。

eval コマンドの使用

Tcl コマンドを実行する際、Tcl コマンドで使用可能なまたは必須のコマンドライン引数の代わりに変数置換を使用できます。ただしこの場合、Tcl **eval** コマンドを使用してコマンドの一部として Tcl 変数を含めたコマンド ラインを評価する必要があります。

たとえば、help コマンドには **-category** オプションを使用すると、コマンド カテゴリの 1 つを指定できます。

```
help -category ipflow
```

コマンド カテゴリを保持する変数を定義できます。

```
set cat "ipflow"
```

説明：

- ・ **set**：変数を定義する Tcl キーワードです。
- ・ **cat**：定義される変数の名前です。
- ・ **"ipflow"**：変数に割り当てる値です。

変数は、Tcl コマンド内で評価できます。

```
eval help -category $cat
```

または

```
set cat "category ipflow"
eval help $cat
```

ダブルクオーテーション（“”）の代わりに波かっこ（{}）を使用しても同じ結果が得られます。

```
set runblocksOptDesignOpts { -sweep -retarget -propconst -remap }
eval opt_design $runblocksOptDesignOpts
```

Tcl コンソールに「**help eval**」と入力すると、**eval** コマンドに関する情報を表示できます。

特殊文字の使用

コマンドの引数に、Tcl で特別の意味を持つ特殊文字が含まれることがあります。その場合、Tcl で誤った処理が実行されないように引数を中かっこ（{}）で囲む必要があります。次に、よく使用される特殊文字の例を示します。

バス インデックス： 角かっこ [] は Tcl で特別な意味を持つので、角かっこを使用するインデックス付きバス（ビットまたは一部選択）は中かっこで囲む必要があります。たとえば、角かっこを使用してバスのインデックス 4 を Vivado 波形ビューアーに追加するには、次のコマンドを使用します。

```
add_wave {bus[4]}
```

バスのインデックスには丸かっこも使用できますが、丸かっこは Tcl で特別な意味はないので、中かっこは必要ありません。次に例を示します。

```
add_wave bus(4)
```

Verilog のエスケープ文字： Verilog の予約文字またはキーワードを含む Verilog 識別子は、Verilog ソースコードおよびシミュレータ コマンド ラインで冒頭にバックスラッシュ（¥）を、末尾にスペースを追加してエスケープ処理する必要があります。さらに Tcl コマンド ラインでは、このエスケープ処理された識別子を中かっこで囲む必要があります。

注記： 識別子に既に中かっこが含まれる場合、中かっこ内の中かっこも Tcl で予約文字として処理されるので、中かっこで囲む方法は使用できません。「VHDL 拡張識別子」に示す方法を使用する必要があります。

たとえば、「my wire」という名前のワイヤを Vivado 波形ビューアーに追加するには、コマンドを次のように記述する必要があります。

```
add_wave {\my wire }
```

注記： 最後の文字と閉じかっことの間にスペースを忘れずに追加してください。

Verilog ではどの識別子もエスケープ処理可能ですが、Tcl コマンド ラインでは、エスケープ処理の不要な識別子をエスケープ処理しないでください。たとえば、「w」という名前のワイヤを Vivado 波形ビューアーに追加する場合、Vivado シミュレータで次のコマンドは無効です。

```
add_wave {\w }
```

この識別子 `w` にはエスケープ処理は不要です。次のコマンドを使用する必要があります。

```
add_wave w
```

VHDL 拡張識別子：VHDL 拡張識別子には、Tcl の予約文字であるバックスラッシュ (`\`) が含まれます。バックスラッシュの後に閉じ中かっこがあると (`\$`)、Tcl で閉じ中かっこと解釈されるので、VHDL 拡張識子に中かっこを使用することはできません。各 Tcl 特殊文字の前にバックスラッシュを追加してください。たとえば、「`\$my sig`」という信号を波形ビューアーに追加するには、コマンドを次のように記述する必要があります。

```
add_wave \\$my\\ sig\\
```

注記：拡張識別子の一部である 2 つのバックスラッシュと、識別子の間のスペースの前にバックスラッシュが追加されています。

一般的な構文構造

Vivado Design Suite の Tcl コマンドの一般的な構文は、次のとおりです。

```
command [optional_parameters] required_parameters
```

コマンド構文は、アンダースコア (`_`) で区切られた「動詞 - 名詞」および「動詞 - 形容詞 - 名詞」の形になります。

コマンドは、関連するコマンドに同じ接頭辞が付けられ、分類されています。

- クエリを実行するコマンドには、通常 `get_` が接頭辞として付いています。
- 値やパラメーターを設定するコマンドには、通常 `set_` が接頭辞として付いています。
- レポートを生成するコマンドには、通常 `report_` が接頭辞として付いています。

これらのコマンドは、グローバル名前空間に属しており、コマンドに付属するサブコマンドはありません。

構文例

次は、**get_cells -help** コマンドを実行した例です。

```
get_cells

Description:
Get a list of cells in the current design

Syntax:
get_cells [-hsc arg] [-hierarchical] [-regexp] [-nocase] [-filter arg]
           [-of_objects args] [-match_style arg] [-quiet] [ patterns ]

Returns:
list of cell objects

Usage:
      Name      Optional  Default  Description
      -----
      -hsc      yes       /        Hierarchy separator
      -hierarchical yes      Search level-by-level in current instance
      -regexp    yes      Patterns are full regular expressions
      -nocase    yes      Perform case-insensitive matching (valid
                           only when -regexp specified)
      -filter    yes      Filter list with expression
      -of_objects yes      Get cells of these pins or nets
      -match_style yes     sdc     Style of pattern matching, valid values are
                           ucf, sdc
      -quiet     yes      Ignore command errors
      patterns   yes     *       Match cell names against patterns

Categories:
SDC, XDC, Object
```

不明コマンド

Tcl には、通常サポートされるビルトインコマンド、Tcl インタープリターに渡される Vivado 特有のコマンド、およびユーザー定義のプロシージャのリストが含まれます。

これらの既知のコマンドに含まれないコマンドは OS に送信され、exec コマンドからシェルで実行されます。これにより、OS 特有のシェルコマンドを実行できます。シェルコマンドがない場合、コマンドが見つからなかったことを示すエラー メッセージが表示されます。

戻りコード

Tcl コマンドの中には、オブジェクトのリストやコレクションなどの戻り値が出力されるものがあります。それ以外のコマンドでは、処理は実行されても、ユーザーが直接利用できるような値が返されるとは限りません。Tcl インターフェイスを統合したツールの一部には、コマンドでエラーのない場合は 0、エラーがある場合は 1 を返すものもあります。

Tcl コマンドまたはスクリプトのエラーを正しく処理するには、Tcl ビルトイン コマンドの `catch` を使用する必要があります。一般的には、`catch` コマンドと番号付き情報/警告/エラー メッセージに基づいて、Tcl スクリプトのフローで問題を評価します。

Vivado ツールの Tcl コマンドでは、コマンドの完了時に `TCL_OK` または `TCL_ERROR` が返され、標準の Tcl メカニズムによりグローバル変数 `$ERRORINFO` が設定されます。

`$ERRORINFO` 変数を使用する場合は、Tcl コンソールでエラーがレポートされた後に次を入力します。

puts \$ERRORINFO

これにより、エラーの詳細情報が表示されます。たとえば、次のコード例では Tcl スクリプト (`procs.tcl`) が使用されており、ユーザー定義の手順 (`loads`) が実行されます。数行トランスクриプト メッセージが表示された後、5 行目にエラーが表示されます。

```
Line 1: Vivado % source procs.tcl
Line 2: Vivado% loads
Line 3: Found 180 driving FFs
Line 4: Processing pin a_reg_reg[1]/Q...
Line 5: ERROR: [HD-Tcl 53] Cannot specify '-patterns' with '-of_objects'.
Line 6: Vivado% puts $errorInfo
Line 7: ERROR: [HD-Tcl 53] Cannot specify '-patterns' with '-of_objects'. While executing
"get_ports -of objects $pin" (procedure "my_report" line 6) invoked from within procs.tcl
```

Tcl スクリプトファイルの `catch` 節に **puts \$errorInfo** を追加し、エラーが見つかったときに詳細を表示するようにしたり、Tcl コンソールでエラーが発生したときに必要に応じて「`puts $errorInfo`」と入力して特定のエラーの詳細を表示できます。

上記のコード例では、6 行目に「**puts \$errorInfo**」と入力することで、7 行目にエラーの詳細情報が表示されています。

ファースト クラスの Tcl オブジェクトとその関係

Vivado Design Suite の Tcl コマンドを使用すると、ネットリスト、デバイス、プロジェクトのオブジェクト モデルに直接アクセスできます。これらオブジェクトは「ファースト クラス」と呼ばれ、単なる文字列記述ではなく、操作およびクエリが可能であることを意味します。例外もありますが、通常はオブジェクトとしてクエリを実行できます。これらのオブジェクトには、クエリ可能なプロパティが含まれ、ほかのオブジェクトを取得できる関係があります。

オブジェクト タイプと定義

Vivado Design Suite には多くのオブジェクト タイプがありますが、ここでは基本的なタイプの定義と説明を示します。最も基本的で重要なオブジェクト タイプは、デザイン ネットリストのエンティティに関連するもので、次のものがあります。

セル

セルは、プリミティブまたはネットリスト内の階層のいずれかのインスタンスです。これには、フリップフロップ、LUT、I/O バッファー、RAM、DSP のほか、ほかのセルのグループのラッパーである階層インスタンスが含まれます。

ピン

ピンはセル上の論理接続ポイントです。ピンにより、セル内部が抽象化されて使用しやすくなります。ピンは、階層またはプリミティブのセル上に存在します。ピンには、クロックピン、データピン、リセットピン、フリップフロップの出力ピンなどが含まれます。

ポート

ポートは、特殊なタイプの階層ピン、最上位ネットリスト オブジェクト、モジュールまたはエンティティ上のピンです。ポートは通常 I/O パッドに接続され、FPGA デバイス外部に接続されます。

ネット

ネットは、物理的に直接相互接続される 1 つのワイヤまたは複数のワイヤです。ネットは階層またはフラットにできますが、常に一連のピンがまとめて分類されます。

クロック

クロックは、デザイン内の順序ロジックに伝搬される周期的な信号です。クロックはプライマリクロックドメインにできるほか、DCM、PLL、MMCM などのクロック プリミティブで生成できます。クロックは UCF の TIMESPEC PERIOD 制約とほぼ同じで、スタティック タイミング解析アルゴリズムの基盤になっています。

オブジェクトのクエリ

ファーストクラス オブジェクトはすべて、通常次のように Tcl コマンド `get_` を使用してクエリできます。

`get_object_type pattern`

ここで `pattern` は検索パターンであり、必要に応じて階層区切り文字を使用して完全な名前を指定します。オブジェクトは通常、階層の各レベルで文字列パターンを一致させることによりクエリされます。検索パターンには次のようにワイルドカードも使用でき、オブジェクトを検索しやすくなっています。

`get_cells */inst_1`

このコマンドでは、最上位のすぐ下の階層で `inst_1` という名前のセルが検索されます。階層のすべてのレベルで同じパターンを繰り返し検索する場合は、次の構文を使用してください。

`get_cells -hierarchical inst_1`

このコマンドでは、`inst_1` に一致するインスタンスがすべての階層レベルで検索されます。

構文の詳細は、次のコマンドでヘルプ情報を参照してください。

- `help get_cells`
- `get_cells -help`

オブジェクト プロパティ

オブジェクトには、クエリを実行できるプロパティが含まれます。プロパティ名はオブジェクトタイプによって異なります。オブジェクトの特定のプロパティをクエリするには、次のコマンドを使用します。

get_property property_name object

次の例では、セル オブジェクトの lib_cell プロパティをクエリしており、指定のインスタンスがどの UniSim コンポーネントにマップされているかがわかります。

get_property lib_cell [get_cell inst_1]

指定したオブジェクトに使用可能なプロパティすべてを表示するには、**report_property** コマンドを使用します。

report_property [get_cells inst_1]

次の表に、特定のオブジェクトに対して返されるプロパティを示します。

特定のオブジェクトで返されるプロパティ

キー	値	データ型
bel	OLOGICE1.OUTFF	string
class	cell	string
ioib	TRUE	string
is_blackbox	0	bool
is_fixed	0	bool
is_partition	0	bool
is_primitive	1	bool
is_reconfigurable	0	bool
is_sequential	1	bool
lib_cell	FD	string
LOC	OLOGIC_X1Y27	string
name	error	string
primitive_group	FD_LD	string
primitive_subgroup	flop	string
site	OLOGIC_X1Y27	string
type	FD & LD	string
XSTLIB	1	bool

プロパティの中には、読み出し専用のものもあれば、ユーザー設定が可能なものもあります。UCF や HDL でアナート可能な属性にマップされるプロパティは、通常 Tcl コマンドの **set_property** でユーザーが設定できます。

set_property loc OLOGIC_X1Y27 [get_cell inst_1]

プロパティに基づいたフィルター処理

オブジェクトをクエリする `get_*` コマンドには、そのオブジェクトのプロパティ値に基づいてクエリをフィルター処理するオプションがあります。このオプションは、非常に優れたオブジェクトクエリコマンド機能です。たとえば、プリミティブタイプ FD のセルをすべてクエリするには、次のように入力します。

```
get_cells * -hierarchical -filter "lib_cell == FD"
```

また、`=~` 演算子を使用すると、文字列パターンでフィルター処理できます。たとえば、デザイン内のすべてのフリップフロップタイプをクエリするには、次のように入力します。

```
get_cells * -hierarchical -filter "lib_cell =~ FD*"
```

OR (||) や AND (&&) を使用すると、複数のプロパティフィルターを組み合わせて検索できます。次の例では、デザイン内のすべてのセルから、フリップフロップタイプで配置済みロケーション制約が設定されているものをクエリしています。

```
get_cells * -hierarchical -filter {lib_cell =~ FD* && loc != ""}
```

注記：この例では、フィルター オプションの値が “” ではなく、{} で囲まれています。これはインタープリターによるコマンド変換を回避する標準的な Tcl 構文で、これにより `loc` プロパティに空の文字列を渡すことができます。

オブジェクトのリスト

複数のオブジェクトを返すコマンドは、通常ネイティブ Tcl リストのようなコンテナー (コレクション) を返します。この機能により、多数の Tcl オブジェクト処理を大幅に最適化でき、ほかのツールで導入されている `foreach_in_collection` コマンドのような特殊な反復コマンドを必要としません。これは、Tcl ビルトインの `foreach` コマンドで処理されます。

リストが大きい場合、ログ ファイルと GUI の Tcl コンソールでの表示が少し異なります。通常、Tcl 変数を `get_*` コマンドの結果に設定すると、リスト全体がコンソールおよびログ ファイルに出力されます。リストが大きい場合、ツールのバッファーのメモリ オーバーフローを回避するため、コンソールおよびログに出力される際にリストの一部が省略されます。

コンソールとログ ファイルでリストが省略されている場合、最後のエレメントが「...」と表示されますが、変数代入での実際のリストは正しいもので、最後のエレメントはエラーではありません。次は、1 つのセルをクエリした場合と、デザインに含まれるすべてのセルをクエリした場合 (リストが大きい場合) の例です。

```
get_cells inst_1
inst_1
get_cells * -hierarchical
XST_VCC XST_GND error readIngressFifo wbDataForInputReg fifoSelect_0 fifoSelect_1 fifoSelect_2 fifoSelect_3 ...
%set x [get_cells * -hierarchical]
XST_VCC XST_GND error readIngressFifo wbDataForInputReg fifoSelect_0 fifoSelect_1 fifoSelect_2 fifoSelect_3 ...
%lindex $x end
bftClk_BUFGP/bufg
%llength $x
4454
```

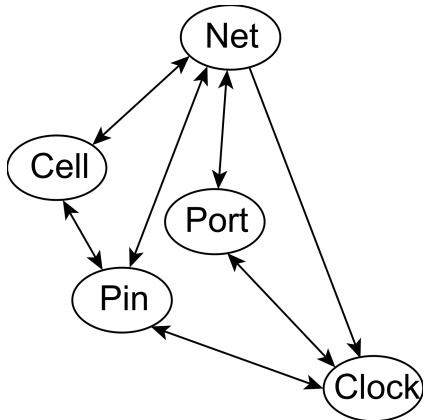
この例では、4000 セルすべてはコンソールに表示されず、「...」で省略されていますが、実際にはリストの最後のエレメントまで Tcl 変数に正しく含まれています。

オブジェクトの関係

関連するオブジェクトは、`get_*` コマンドに `-of` オプションを使用してクエリできます。たとえば、あるセル オブジェクトに接続されたピンのリストを取得するには、次のように入力します。

```
get_pins -of [get_cells inst_1]
```

次の図は、オブジェクトタイプとその関係を示します。オブジェクト間の矢印は、`get_*` コマンドで `-of` オプションを使用して、論理接続をたどって接続されているオブジェクトへの Tcl リファレンスを取得できることを示しています。



エラー、警告、クリティカル警告、および情報メッセージ

各コマンドの結果を示すメッセージは、ログ ファイルと、GUI を使用中であればコンソールにも表示されます。これらのメッセージには識別しやすいように番号が付いています。ログ ファイルでは、INFO、WARNING、CRITICAL WARNING、ERROR の後にサブシステム識別子と独自の番号が付きます。

次は、タイミング ライブラリを読み込んだ後に表示される INFO メッセージの例です。

```
INFO: [HD-LIB 1] Done reading timing library
```

このようなメッセージにより、ログ ファイルで特定の問題を検出しやすくなり、コマンド実行中の動作内容が理解しやすくなります。

通常、Tcl スクリプトからの Tcl コマンドでエラーが発生すると、続くコマンドの実行は停止されます。これは、回復不可能なエラー状況を避けるためです。これらのエラー状況を回避して続行させるための Tcl ビルトイン コマンドがあります。一般的な Tcl メカニズムを使用したエラー処理方法については、Tcl リファレンスで `catch` コマンドを参照してください。

Tcl コマンド リスト (カテゴリ別)

カテゴリ

- Board (ボード)
- ChipScope
- DRC
- FileIO (ファイル入力および出力)
- Floorplan (フロアプラン)
- GUIControl (GUI 制御)
- Hardware (ハードウェア)
- IPFlow (IP フロー)
- IPIntegrator (IP インテグレーター)
- Netlist (ネットリスト)
- Object (オブジェクト)
- PinPlanning (ピン配置)
- Power (電力)
- Project (プロジェクト)
- PropertyAndParameter (プロパティおよびパラメーター)
- Report (レポート)
- SDC
- Simulation (シミュレーション)
- SysGen (System Generator)
- Timing (タイミング)
- ToolLaunch (ツール起動)
- Tools (ツール)
- Waveform (波形)
- XDC
- XPS

Board (ボード)

- [current_board](#)
 - [get_board_interfaces](#)
 - [get_board_pins](#)
-

ChipScope

- [launch_chipscope_analyzer](#)
 - [launch_impact](#)
 - [write_chipscope_cdc](#)
-

DRC

- [add_drc_checks](#)
- [create_drc_check](#)
- [create_drc_ruledeck](#)
- [create_drcViolation](#)
- [delete_drc_check](#)
- [delete_drc_ruledeck](#)
- [get_drc_checks](#)
- [get_drc_ruledecks](#)
- [get_drc_vios](#)
- [remove_drc_checks](#)
- [report_drc](#)
- [reset_drc](#)
- [reset_drc_check](#)

FileIO (ファイル入力および出力)

- config_webtalk
- infer_diff_pairs
- pr_verify
- read_checkpoint
- read_csv
- read_edif
- read_ip
- read_saif
- read_schematic
- read_twx
- read_vcd
- read_verilog
- read_vhdl
- read_xdc
- write_bitstream
- write_bmm
- write_checkpoint
- write_chipscope_cdc
- write_csv
- write_debug_probes
- write_edif
- write_ibis
- write_schematic
- write_sdf
- write_verilog
- write_vhdl
- write_xdc

Floorplan (フロアプラン)

- [add_cells_to_pblock](#)
 - [create_pblock](#)
 - [delete_pblock](#)
 - [delete_rpm](#)
 - [get_pblocks](#)
 - [place_cell](#)
 - [place_pblocks](#)
 - [remove_cells_from_pblock](#)
 - [reset_ucf](#)
 - [resize_pblock](#)
 - [swap_locs](#)
 - [unplace_cell](#)
-

GUIControl (GUI 制御)

- [endgroup](#)
 - [get_selected_objects](#)
 - [highlight_objects](#)
 - [mark_objects](#)
 - [redo](#)
 - [select_objects](#)
 - [show_objects](#)
 - [show_schematic](#)
 - [start_gui](#)
 - [startgroup](#)
 - [stop_gui](#)
 - [undo](#)
 - [unhighlight_objects](#)
 - [unmark_objects](#)
 - [unselect_objects](#)
-

Hardware (ハードウェア)

- [close_hw](#)
- [close_hw_target](#)

- commit_hw_sio
- commit_hw_vio
- connect_hw_server
- create_hw_sio_link
- create_hw_sio_linkgroup
- create_hw_sio_scan
- current_hw_device
- current_hw_ila
- current_hw_ila_data
- current_hw_server
- current_hw_target
- disconnect_hw_server
- display_hw_ila_data
- display_hw_sio_scan
- get_hw_devices
- get_hw_ila_datas
- get_hw_ilas
- get_hw_probes
- get_hw_servers
- get_hw_sio_commons
- get_hw_sio_gtgroups
- get_hw_sio_gts
- get_hw_sio_iberts
- get_hw_sio_linkgroups
- get_hw_sio_links
- get_hw_sio_pll
- get_hw_sio_rx
- get_hw_sio_scans
- get_hw_sio_tx
- get_hw_targets
- get_hw_vios
- open_hw
- open_hw_target
- program_hw_devices
- read_hw_ila_data
- read_hw_sio_scan
- refresh_hw_device
- refresh_hw_server

- [refresh_hw_sio](#)
 - [refresh_hw_target](#)
 - [refresh_hw_vio](#)
 - [remove_hw_sio_link](#)
 - [remove_hw_sio_linkgroup](#)
 - [remove_hw_sio_scan](#)
 - [reset_hw_ilा](#)
 - [reset_hw_vio_activity](#)
 - [reset_hw_vio_outputs](#)
 - [run_hw_ilा](#)
 - [run_hw_sio_scan](#)
 - [stop_hw_sio_scan](#)
 - [upload_hw_ilा_data](#)
 - [wait_on_hw_ilा](#)
 - [wait_on_hw_sio_scan](#)
 - [write_hw_ilा_data](#)
 - [write_hw_sio_scan](#)
-

IPFlow (IP フロー)

- [copy_ip](#)
 - [create_ip](#)
 - [generate_target](#)
 - [get_ipdefs](#)
 - [get_ips](#)
 - [import_ip](#)
 - [open_example_project](#)
 - [read_ip](#)
 - [report_ip_status](#)
 - [reset_target](#)
 - [update_ip_catalog](#)
 - [upgrade_ip](#)
 - [validate_ip](#)
-

IPIntegrator (IP インテグレーター)

- [apply_bd_automation](#)
- [assign_bd_address](#)

- `close_bd_design`
- `connect_bd_intf_net`
- `connect_bd_net`
- `copy_bd_objs`
- `create_bd_addr_seg`
- `create_bd_cell`
- `create_bd_design`
- `create_bd_intf_net`
- `create_bd_intf_pin`
- `create_bd_intf_port`
- `create_bd_net`
- `create_bd_pin`
- `create_bd_port`
- `current_bd_design`
- `current_bd_instance`
- `delete_bd_objs`
- `disconnect_bd_intf_net`
- `disconnect_bd_net`
- `find_bd_objs`
- `generate_target`
- `get_bd_addr_segs`
- `get_bd_addr_spaces`
- `get_bd_cells`
- `get_bd_designs`
- `get_bd_intf_nets`
- `get_bd_intf_pins`
- `get_bd_intf_ports`
- `get_bd_nets`
- `get_bd_pins`
- `get_bd_ports`
- `group_bd_cells`
- `move_bd_cells`
- `open_bd_design`
- `regenerate_bd_layout`
- `replace_bd_cell`
- `save_bd_design`
- `ungroup_bd_cells`
- `upgrade_bd_cells`

- [validate_bd_design](#)
 - [write_bd_tcl](#)
-

Netlist (ネットリスト)

- [connect_net](#)
 - [create_cell](#)
 - [create_net](#)
 - [create_pin](#)
 - [disconnect_net](#)
 - [get_net_delays](#)
 - [remove_cell](#)
 - [remove_net](#)
 - [remove_pin](#)
 - [rename_ref](#)
 - [resize_net_bus](#)
 - [resize_pin_bus](#)
 - [tie_unused_pins](#)
-

Object (オブジェクト)

- [add_drc_checks](#)
- [create_drc_check](#)
- [create_drc_ruledeck](#)
- [current_board](#)
- [delete_drc_check](#)
- [delete_drc_ruledeck](#)
- [filter](#)
- [get_bel_pins](#)
- [get_bels](#)
- [get_board_interfaces](#)
- [get_board_pins](#)
- [get_boards](#)
- [get_cells](#)
- [get_clock_regions](#)
- [get_clocks](#)
- [get_debug_cores](#)
- [get_debug_ports](#)

- `get_delays`
- `get_designs`
- `get_drc_checks`
- `get_drc_ruledecks`
- `get_drc_vios`
- `get_files`
- `get_filesets`
- `get_generated_clocks`
- `get_gtbanks`
- `get_hw_devices`
- `get_hw_ilabatas`
- `get_hw_ilas`
- `get_hw_probes`
- `get_hw_servers`
- `get_hw_sio_commons`
- `get_hw_sio_gtgroups`
- `get_hw_sio_gts`
- `get_hw_sio_iberts`
- `get_hw_sio_linkgroups`
- `get_hw_sio_links`
- `get_hw_sio_pll`
- `get_hw_sio_rx`
- `get_hw_sio_scans`
- `get_hw_sio_tx`
- `get_hw_targets`
- `get_hw_vios`
- `get_interfaces`
- `get_io_standards`
- `get_jobanks`
- `get_ipdefs`
- `get_ip`
- `get_lib_cells`
- `get_lib_pins`
- `get_libs`
- `get_macros`
- `get_net_delays`
- `get_nets`
- `get_nodes`

- `get_package_pins`
 - `get_parts`
 - `get_path_groups`
 - `get_pblocks`
 - `get_pins`
 - `get_pips`
 - `get_ports`
 - `get_projects`
 - `get_property`
 - `get_runs`
 - `get_selected_objects`
 - `get_site_pins`
 - `get_site_pips`
 - `get_sites`
 - `get_slrs`
 - `get_tiles`
 - `get_timing_arcs`
 - `get_timing_paths`
 - `get_wires`
 - `list_property`
 - `list_property_value`
 - `remove_drc_checks`
 - `report_property`
 - `reset_drc_check`
 - `reset_property`
 - `set_property`
-

PinPlanning (ピン配置)

- `create_interface`
- `create_port`
- `delete_interface`
- `make_diff_pair_ports`
- `place_ports`
- `remove_port`
- `resize_port_bus`
- `set_package_pin_val`
- `split_diff_pair_ports`

Power (電力)

- [delete_power_results](#)
 - [power_opt_design](#)
 - [read_saif](#)
 - [read_vcd](#)
 - [report_power](#)
 - [report_power_opt](#)
 - [reset_default_switching_activity](#)
 - [reset_operating_conditions](#)
 - [reset_switching_activity](#)
 - [set_default_switching_activity](#)
 - [set_operating_conditions](#)
 - [set_power_opt](#)
 - [set_switching_activity](#)
-

Project (プロジェクト)

- [add_files](#)
- [archive_project](#)
- [close_design](#)
- [close_project](#)
- [copy_ip](#)
- [create_fileset](#)
- [create_project](#)
- [create_run](#)
- [current_board](#)
- [current_fileset](#)
- [current_project](#)
- [current_run](#)
- [delete_fileset](#)
- [delete_run](#)
- [find_top](#)
- [generate_target](#)
- [get_board_interfaces](#)
- [get_board_pins](#)
- [get_boards](#)

- get_files
- get_filesets
- get_ips
- get_projects
- get_runs
- help
- import_files
- import_ip
- import_synplify
- import_xise
- import_xst
- launch_runs
- list_targets
- lock_design
- make_wrapper
- move_files
- open_example_project
- open_io_design
- open_project
- open_run
- refresh_design
- reimport_files
- remove_files
- reorder_files
- report_compile_order
- reset_project
- reset_run
- reset_target
- save_constraints
- save_constraints_as
- save_project_as
- set_speed_grade
- update_compile_order
- update_design
- update_files
- wait_on_run

PropertyAndParameter (プロパティおよびパラメーター)

- [create_property](#)
 - [filter](#)
 - [get_param](#)
 - [get_property](#)
 - [list_param](#)
 - [list_property](#)
 - [list_property_value](#)
 - [report_param](#)
 - [report_property](#)
 - [reset_param](#)
 - [reset_property](#)
 - [set_param](#)
 - [set_property](#)
-

Report (レポート)

- [check_timing](#)
- [create_drcViolation](#)
- [create_slack_histogram](#)
- [delete_clock_networks_results](#)
- [delete_timing_results](#)
- [delete_utilization_results](#)
- [get_msg_config](#)
- [get_msg_count](#)
- [get_msg_limit](#)
- [report_carry_chains](#)
- [report_clock_interaction](#)
- [report_clock_networks](#)
- [report_clock_utilization](#)
- [report_clocks](#)
- [report_config_timing](#)
- [report_control_sets](#)
- [report_datasheet](#)
- [report_debug_core](#)
- [report_default_switching_activity](#)

- report_disable_timing
 - report_drc
 - report_environment
 - report_exceptions
 - report_high_fanout_nets
 - report_incremental_reuse
 - report_io
 - report_operating_conditions
 - report_param
 - report_phys_opt
 - report_power
 - report_property
 - report_pulse_width
 - report_route_status
 - report_ssn
 - report_switching_activity
 - report_timing
 - report_timing_summary
 - report_transformed_primitives
 - report_utilization
 - reset_drc
 - reset_msg_config
 - reset_msg_count
 - reset_msg_limit
 - reset_msg_severity
 - reset_ssn
 - reset_timing
 - set_msg_config
 - set_msg_limit
 - set_msg_severity
 - version
-

SDC

- all_clocks
- all_fanin
- all_fanout
- all_inputs

- all_outputs
- all_registers
- create_clock
- create_generated_clock
- current_design
- current_instance
- get_cells
- get_clocks
- get_hierarchy_separator
- get_nets
- get_pins
- get_ports
- get_timing_arcs
- get_timing_paths
- group_path
- report_operating_conditions
- reset_operating_conditions
- set_case_analysis
- set_clock_groups
- set_clock_latency
- set_clock_sense
- set_clock_uncertainty
- set_data_check
- set_disable_timing
- set_false_path
- set_hierarchy_separator
- set_input_delay
- set_load
- set_logic_dc
- set_logic_one
- set_logic_unconnected
- set_logic_zero
- set_max_delay
- set_max_time_borrow
- set_min_delay
- set_multicycle_path
- set_operating_conditions
- set_output_delay

- [set_propagated_clock](#)
 - [set_units](#)
-

Simulation (シミュレーション)

- [add_bp](#)
- [add_condition](#)
- [add_files](#)
- [add_force](#)
- [checkpoint_vcd](#)
- [close_saif](#)
- [close_sim](#)
- [close_vcd](#)
- [compile_simlib](#)
- [create_fileset](#)
- [current_scope](#)
- [current_sim](#)
- [current_time](#)
- [current_vcd](#)
- [data2mem](#)
- [delete_fileset](#)
- [describe](#)
- [flush_vcd](#)
- [get_objects](#)
- [get_scopes](#)
- [get_value](#)
- [import_files](#)
- [launch_modelsim](#)
- [launch_xsim](#)
- [limit_vcd](#)
- [log_saif](#)
- [log_vcd](#)
- [log_wave](#)
- [ltrace](#)
- [move_files](#)
- [open_saif](#)
- [open_vcd](#)
- [open_wave_database](#)

- `ptrace`
 - `read_saif`
 - `read_vcd`
 - `remove_bps`
 - `remove_conditions`
 - `remove_files`
 - `remove_forces`
 - `report_bps`
 - `report_conditions`
 - `report_drivers`
 - `report_objects`
 - `report_scopes`
 - `report_simlib_info`
 - `report_values`
 - `reset_simulation`
 - `restart`
 - `run`
 - `set_value`
 - `start_vcd`
 - `step`
 - `stop`
 - `stop_vcd`
 - `write_sdf`
 - `write_verilog`
 - `write_vhdl`
 - `xsim`
-

SysGen (System Generator)

- `create_sysgen`
- `make_wrapper`

Timing (タイミング)

- [check_timing](#)
 - [config_timing_analysis](#)
 - [config_timing_corners](#)
 - [delete_timing_results](#)
 - [get_net_delays](#)
 - [get_timing_arcs](#)
 - [get_timing_paths](#)
 - [report_config_timing](#)
 - [report_disable_timing](#)
 - [report_exceptions](#)
 - [report_timing](#)
 - [report_timing_summary](#)
 - [reset_timing](#)
 - [set_delay_model](#)
 - [set_disable_timing](#)
 - [update_timing](#)
-

ToolLaunch (ツール起動)

- [launch_chipscope_analyzer](#)
 - [launch_impact](#)
 - [launch_modelsim](#)
 - [launch_sdk](#)
 - [launch_xsim](#)
-

Tools (ツール)

- [link_design](#)
- [list_features](#)
- [load_features](#)
- [opt_design](#)
- [phys_opt_design](#)
- [place_design](#)
- [route_design](#)
- [synth_design](#)

Waveform (波形)

- [add_wave](#)
 - [add_wave_divider](#)
 - [add_wave_group](#)
 - [add_wave_marker](#)
 - [add_wave_virtual_bus](#)
 - [close_wave_config](#)
 - [create_wave_config](#)
 - [current_wave_config](#)
 - [get_wave_configs](#)
 - [open_wave_config](#)
 - [save_wave_config](#)
-

xdc

- [add_cells_to_pblock](#)
- [all_clocks](#)
- [all_cpus](#)
- [all_dsp](#)
- [all_fanin](#)
- [all_fanout](#)
- [all_ffs](#)
- [all_hsi](#)
- [all_inputs](#)
- [all_latches](#)
- [all_outputs](#)
- [all_rams](#)
- [all_registers](#)
- [create_clock](#)
- [create_generated_clock](#)
- [create_macro](#)
- [create_pblock](#)
- [create_property](#)
- [current_design](#)
- [current_instance](#)
- [delete_macros](#)

- [delete_pblock](#)
- [filter](#)
- [get_bel_pins](#)
- [get_bels](#)
- [get_cells](#)
- [get_clocks](#)
- [get_generated_clocks](#)
- [get_gtbanks](#)
- [get_hierarchy_separator](#)
- [get_iobanks](#)
- [get_macros](#)
- [get_nets](#)
- [get_nodes](#)
- [get_package_pins](#)
- [get_path_groups](#)
- [get_pblocks](#)
- [get_pins](#)
- [get_pips](#)
- [get_ports](#)
- [get_property](#)
- [get_site_pins](#)
- [get_site_pips](#)
- [get_sites](#)
- [get_slrs](#)
- [get_tiles](#)
- [get_timing_arcs](#)
- [get_wires](#)
- [group_path](#)
- [remove_cells_from_pblock](#)
- [resize_pblock](#)
- [set_case_analysis](#)
- [set_clock_groups](#)
- [set_clock_latency](#)
- [set_clock_sense](#)
- [set_clock_uncertainty](#)
- [set_data_check](#)
- [set_default_switching_activity](#)
- [set_disable_timing](#)

- `set_external_delay`
 - `set_false_path`
 - `set_hierarchy_separator`
 - `set_input_delay`
 - `set_input_jitter`
 - `set_load`
 - `set_logic_dc`
 - `set_logic_one`
 - `set_logic_unconnected`
 - `set_logic_zero`
 - `set_max_delay`
 - `set_max_time_borrow`
 - `set_min_delay`
 - `set_multicycle_path`
 - `set_operating_conditions`
 - `set_output_delay`
 - `set_package_pin_val`
 - `set_power_opt`
 - `set_propagated_clock`
 - `set_property`
 - `set_switching_activity`
 - `set_system_jitter`
 - `set_units`
 - `update_macro`
-

XPS

- `create_xps`
- `export_hardware`
- `generate_target`
- `get_boards`
- `launch_sdk`
- `list_targets`
- `make_wrapper`
- `reset_target`

Tcl コマンド リスト (アルファベット順)

この章では、すべての SDC および Tcl コマンドをアルファベット順にリストします。

add_bp

HDL ソースの指定した行にブレークポイントを追加します。

構文

```
add_bp [-quiet] [-verbose] file_name line_number
```

戻り値

指定のファイルの行にブレークポイントが設定されていなかった場合は新しいブレークポイント オブジェクト、ブレークポイントが設定されていた場合は既存のブレークポイント オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>file_name</i>	ブレークポイントを追加するファイルの名前を指定します。
<i>line_number</i>	指定のファイルのブレークポイントを設定する行番号を指定します。

カテゴリ

シミュレーション

add_cells_to_pblock

Pblock にセルを追加します。

構文

```
add_cells_to_pblock [-top] [-add_primitives] [-clear_locs] [-quiet]
[-verbose] pblock [cells...]
```

戻り値

なし

使用法

名前	説明
[-top]	最上位インスタンスを追加します。cells 引数または -add_primitives オプションと共に使用することはできません。cells 引数または -top オプションのどちらかを指定する必要があります。
[-add_primitives]	指定したインスタンスのすべてのプリミティブを Pblock に割り当てます。
[-clear_locs]	インスタンスのロケーション制約を削除します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
pblock	セルを追加する Pblock を指定します。
[cells]	追加するセルを指定します。-top オプションを使用している場合は指定できません。cells 引数または -top オプションのどちらかを指定する必要があります。

カテゴリ

フロアプラン、XDC

説明

指定したロジック インスタンスを Pblock に追加します。Pblock にセルを追加したら、**resize_pblock** コマンドを使用して Pblock を FPGA のファブリック上に配置できます。**resize_pblock** コマンドを使用すると、Pblock を手動で移動およびサイズ変更できます。

Pblock からインスタンスを削除するには、**remove_cells_from_pblock** コマンドを使用します。

引数

-top (オプション) : 最上位インスタンスを追加して、デザイン全体の Pblock を作成します。Pblock にオブジェクトを追加するには、cells または **-top** オプションのどちらかを使用する必要があります。

-add_primitives (オプション) : 指定したインスタンスのすべてのプリミティブを Pblock に割り当てます。ブロック モジュールを指定し、そのモジュールに含まれるすべてのインスタンスを自動的に指定の Pblock に追加できます。

注記 : **-top** オプションと共に使用することはできません。

-clear_locs (オプション) : 既に配置されているセルのインスタンス ロケーション制約を消去します。これにより、フロアプランするために新しい Pblock を定義する際に、セルの LOC 制約をリセットできます。このオプションを指定しないと、配置が指定されているインスタンスを Pblock に追加する場合に配置が解除されません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

pblock : Pblock の名前を指定します。

cells : 指定の Pblock に追加する 1 つまたは複数のセル オブジェクトを指定します。

注記 : **-top** を指定した場合、*cells* を指定することはできません。

例

次の例では、pb_cpuEngine という Pblock を作成し、cpuEngine モジュールに含まれるすべてのプリミティブを追加し、配置済みのインスタンスの配置制約を消去します。

```
create_pblock pb_cpuEngine
add_cells_to_pblock pb_cpuEngine [get_cells cpuEngine] -add_primitives -clear_locs
```

関連項目

- [get_pblocks](#)
- [place_pblocks](#)
- [remove_cells_from_pblock](#)
- [resize_pblock](#)

add_condition

Tcl コマンドを条件に基づいて実行します。

構文

```
add_condition [-name arg] [-radix arg] [-quiet]
[-verbose] condition_expression commands
```

戻り値

作成された条件オブジェクト

使用法

名前	説明
[-name]	条件の名前 (ラベル) を指定します。複数の条件に同じ名前を付けることはできません。名前を指定しない場合、デフォルト名 (condition<id>) が自動的に作成されます。
[-radix]	使用する基底を指定します。有効な値は、default、dec、bin、oct、hex、unsigned、ascii です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
condition_expression	条件式を指定します。条件式が満たされると、コマンドが実行されます。
commands	条件が満たされたときに実行するコマンドを指定します。

カテゴリ

シミュレーション

add_drc_checks

ルール デックに DRC ルール チェック オブジェクトを追加します。

構文

```
add_drc_checks [-of_objects args] [-regexp] [-nocase] [-filter arg]
-ruledesk arg [-quiet] [-verbose] [patterns]
```

戻り値

DRC チェック

使用法

名前	説明
[-of_objects]	指定した DRC ルール デックの DRC チェック オブジェクトを追加します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
-ruledesk	変更する DRC ルール デックを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	DRC チェック オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

DRC、オブジェクト

説明

指定した DRC ルール デック オブジェクトにデザイン ルール チェックを追加します。

ルール デックとは、デザイン ルール チェックのグループで、I/O 配置などの FPGA デザイン フローの異なる段階で **report_drc** コマンドにより実行されます。ツールには定義済みのルール デックが含まれていますが、**create_drc_ruledesk** コマンドを使用して新しいユーザー定義のルール デックを作成できます。

get_drc_ruledecks コマンドを使用すると、**report_drc** コマンドで使用可能な定義済みのルール デックを取得できます。

ルール デックには、ツールに含まれる定義済みのルール チェックまたは **create_drc_check** コマンドで作成したユーザー定義のルール チェックを追加できます。**get_drc_ruledecks** コマンドを使用すると、ルール デックに追加できるルール チェックを取得できます。

ルール デックからチェックを削除するには、**remove_drc_checks** コマンドを使用します。

注記：特定の DRC ルールを一時的にディスエーブルにするには、**set_property** コマンドを使用してルールの **IS_ENABLED** プロパティを **false** に設定します。これにより、ルール デックからルールを削除せずに、**report_drc** を実行したときにルールを除外できます。ルールをデフォルト設定に戻すには、**reset_drc_check** コマンドを使用します。

このコマンドを実行すると、ルール デックに追加されているデザイン ルール チェックのリストが返されます。

引数

-of_objects arg (オプション)：指定した DRC ルール デック オブジェクトのルール チェックを追加します。このオプションを使用すると、ルールがあるルール デックから別のルール デックにコピーされます。

-regexp (オプション)：検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイレンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記：Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション)：パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション)：結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、検索パターンにより返されるオブジェクトのリストに、指定したプロパティ値に基づくフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記：フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-ruledeck arg (必須)：指定のデザイン ルール チェックを追加するルール デックの名前を指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するデザイン ルール チェックをルール デックに追加します。デフォルトのパターンはワイルドカード (*) で、すべてのルール チェックが指定のルール デックに追加されます。複数のパターンを指定して、異なる検索条件に基づいてルール チェックを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション ("") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定した検索パターンに一致するルール チェックを `project_rules` ルール デックに追加しています。

```
add_drc_checks -ruledeck project_rules {*DCI* *BUF*}
```

次の例では、`placer+` というルール デックを作成し、`placer_checks` ルール デックに含まれるすべてのルール チェックを `placer+` ルール デックにコピーして、さらにルール チェックを追加しています。

```
create_drc_ruledeck placer+
add_drc_checks -of_objects [get_drc_ruledecks placer_checks] -ruledeck placer+
add_drc_checks -ruledeck placer+ *IO*
```

次の例では、重要度が警告のルール チェックのみを ルール デックに追加しています。

```
add_drc_checks -filter {SEVERITY == Warning} -ruledeck warn_only
```

関連項目

- [create_drc_check](#)
- [create_drc_ruledeck](#)
- [get_drc_checks](#)
- [get_drc_ruledecks](#)
- [list_property](#)
- [remove_drc_checks](#)
- [report_drc](#)
- [report_property](#)
- [reset_drc_check](#)
- [set_property](#)

add_files

アクティブなファイルセットにソースを追加します。

構文

```
add_files [-fileset arg] [-norecurse] [-scan_for_includes] [-quiet] [-verbose]
[files...]
```

戻り値

追加されたファイル オブジェクトのリスト

使用法

名前	説明
[-fileset]	ファイルセット名を指定します。
[-norecurse]	下位ディレクトリでは検索を実行しないよう指定します。
[-scan_for_includes]	ファイルセットの RTL ソースに含まれるファイルすべてをスキャンして追加します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[<i>files</i>]	追加するファイルおよびディレクトリ名を指定します。 -scan_for_includes を使用しない場合は指定する必要があります。

カテゴリ

プロジェクト、シミュレーション

説明

1 つ以上のソース ファイルまたは 1 つ以上のディレクトリ内のソース ファイルを、現在のプロジェクトの指定したファイルセットに追加します。

Vivado ツールでは、**add_files** コマンドでプロジェクトにファイルを追加したときにはファイルの内容は自動的には読み込まれず、必要なときに読み込まれます。たとえば、制約ファイルはデザインに追加されたときには読み込まれず、合成、タイミング、インプリメンテーションが実行されたときに読み込まれます。デザインにファイルを追加したときに読み込むには、**read_xxx** コマンドを使用してください。

注記： 非プロジェクト モードを使用しており、プロジェクトのソース ファイルを管理するプロジェクト ファイルがない場合は、**read_xxx** コマンドを使用してソース ファイル メモリ内のデザインに読み込む必要があります。非プロジェクト モードの詳細は、『Vivado Design Suite ユーザー ガイド：デザイン フローの概要』(UG892) を参照してください。

add_files コマンドは、ファイルをローカルのプロジェクト フォルダーにコピーし、指定したファイルセットにも追加する **import_files** コマンドとは異なり、指定したファイルセットを参照することによってのみファイルを追加します。

引数

-fileset name (オプション) : 指定したソース ファイルを追加するファイルセットを指定します。指定したファイルセットが存在しない場合は、エラーが返されます。ファイルセットを指定しない場合は、デフォルトでソース ファイルセットに追加されます。

-norecurse (オプション) : 指定したディレクトリの下位ディレクトリでコマンドを実行しないよう指定します。このオプションを指定しない場合、下位ディレクトリでもプロジェクトに追加可能なソース ファイルが検索されます。

-scan_for_includes (オプション) : Verilog ソース ファイルで `'include` 文を検索し、それらの参照ファイルも指定したファイルセットに追加します。デフォルトでは、`'include` ファイルは追加されません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

files (オプション) : ファイルセットに追加する 1 つまたは複数のファイルまたはディレクトリ名を指定します。ディレクトリ名を指定した場合は、そのディレクトリとそれに含まれる下位ディレクトリの有効なソース ファイルすべてがファイルセットに追加されます。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次の例では、`rtl.v` というファイルを現在のプロジェクトに追加しています。

```
add_files rtl.v
```

この例ではパスが指定されていないので、現在の作業ディレクトリで `rtl.v` ファイルが検索されます。ファイルセットは指定されていないので、ファイルはデフォルトでソース ファイルセットに追加されます。

次の例では、`top.ucf` というファイルを `constrs_1` 制約ファイルセットに追加し、`project_1` ディレクトリとその下位ディレクトリの有効なソース ファイルを追加しています。

```
add_files -fileset constrs_1 -quiet c:/Design/top.ucf c:/Design/project_1
```

また、`-quiet` オプションが使用されているので、コマンド ライン エラーは無視されます。

-norecurse オプションが指定されていれば、`project_1` ディレクトリの制約ファイルのみが追加され、下位ディレクトリは検索されません。

次の例では、既存の IP コア ファイルを現在のプロジェクトに追加しています。

```
add_files -norecurse C:/Data/ip/c_addsub_v11_0_0.xci
```

注記 : IP ファイルをローカル プロジェクト フォルダーにインポートするには、`import_ip` コマンドを使用します。

次の例では、既存のエンベデッド プロセッサ サブデザインを現在のプロジェクトに追加しています。

```
add_files C:/Data/dvi_tpg_demo_ORG/system.xmp
```

注記 : Xilinx Platform Studio (XPS) を使用してエンベデッド プロセッサを作成するには、**create_xps** を使用します。

次の例では、System Generator で作成した既存の DSP モジュールを現在のプロジェクトに追加しています。

```
add_files C:/Data/model1.mdl
```

注記 : System Generator を使用して DSP モジュールを生成するには、**create_sysgen** コマンドを使用します。

関連項目

- [create_sysgen](#)
- [create_xps](#)
- [import_files](#)
- [import_ip](#)
- [read_ip](#)
- [read_verilog](#)
- [read_vhdl](#)
- [read_xdc](#)

add_force

信号、ワイヤ、またはレジスタを特定の値に強制的に設定します。

構文

```
add_force [-radix arg] [-repeat_every arg] [-cancel_after arg] [-quiet]
[-verbose] hdl_object values...
```

戻り値

追加された force オブジェクト

使用法

名前	説明
<code>[-radix]</code>	使用する基数を指定します。有効な値は、default、dec、bin、oct、hex、unsigned、ascii です。
<code>[-repeat_every]</code>	指定の時間間隔で繰り返します。
<code>[-cancel_after]</code>	指定の時間後にキャンセルします。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>hdl_object</code>	force を追加するオブジェクトを指定します。
<code>values</code>	force に値と時間オフセットを追加します。「{value [time_offset] }」という形式を使用します。

カテゴリ

シミュレーション

説明

シミュレーション中に信号、ワイヤ、またはレジスタを特定の値に強制的に設定します。

add_force コマンドは、テストベンチまたはモジュール定義の Verilog **force**/**release** コマンドと同じように機能します。指定の期間、**-cancel_after** オプションで指定されたイベントまで、または **remove_forces** コマンドが使用されるまで、HDL オブジェクトを指定の値に保持します。

重要： テストベンチまたはモジュールの HDL オブジェクトに Verilog **force**/**release** 文がある場合、それらの文より Tcl **add_force** コマンドが優先されます。Tcl force が期限切れになるかまたは解除されると、シミュレーションで、Verilog force の実行も含め、HDL オブジェクトが通常の動作に戻ります。

このコマンドでは、作成された force オブジェクトの名前が返されるか、コマンドが正しく実行されなかった場合はエラーが返されます。返された force オブジェクトの名前は、**remove_forces** コマンドを使用する場合に重要であり、後で使用できるよう、下の例に示すように Tcl 変数に代入しておく必要があります。

引数

-radix arg (オプション) : *values* を指定する際の基数を指定します。サポートされる基数は、**default**、**dec**、**bin**、**oct**、**hex**、**unsigned**、**ascii** です。指定した HDL オブジェクトタイプに基数が定義されていない場合、デフォルトの基数は 2 進数 (bin) です。

-repeat_every arg (オプション) : **add_force** を指定の時間ごとに繰り返します。指定の *hdl_object* に対して反復 force を作成する場合に使用できます。

注記 : このオプションで指定する時間は、*values* を *{value time}* で指定した場合の時間よりも長くする必要があります。そうでないと、エラーが発生します。

-cancel_after arg (オプション) : **current_time** からの指定された時間後に force を解除します。これは、指定の時間後に **remove_forces** コマンドを実行した場合と同じです。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

hdl_object (必須) : 値を強制する 1 つの HDL オブジェクトの名前を指定します。オブジェクトは、名前または **get_objects** コマンドで指定します。有効なオブジェクトは信号、ワイヤ、およびレジスタです。

value (必須) : HDL オブジェクトを強制する値を指定します。1 つの値を指定でき、シミュレーション中、**-cancel_after** オプションまたは **remove_forces** コマンドを使用して解除されるまで、その値に保持されます。

指定の値は、*hdl_object* のタイプによって異なります。HDL オブジェクトタイプには、logic、浮動小数点、VHDL 列挙型、VHDL 整数があります。logic 以外では、-radix オプションは無視されます。

- logic は、実際の HDL オブジェクトタイプを示すのではなく、次のような VHDL std_logic と同様の値を持つオブジェクトを示します。
 - Verilog 暗示 4 値ビット型
 - VHDL ビットおよび std_logic 既定義型
 - std_logic のサブセットである VHDL 列挙型 (文字リテラル 0 および 1 を含む)
- logic 型では、値は基数によって異なります。
 - 指定の値のビット数が logic 型で予測されるビット数より少ない場合、値は予測される長さにあわせて 0 で拡張されます (符号拡張されない)。
 - 指定の値のビット数が logic 型で予測されるビット数より多い場合、MSB 側の余分なビットはすべて 0 である必要があり、そうでない場合は Vivado シミュレータでサイズ不一致エラーが返されます。
- 浮動小数点オブジェクトの値は浮動小数点値です。
- logic でない VHDL 列挙型の値は、値の列挙セットからのスカラー値 (文字の場合はシングル クオーテーションなし) です。
- VHDL 整数型の値は、その型で許容される範囲の符号付き 10 進整数です。

値は *{value time}* のペアで指定することもでき、この場合 HDL オブジェクトが現在の時間から *time* で指定された期間 *value* で指定された値に保持され、その後次のペアの *time* で指定された期間 *value* で指定された値に保持され、最後の *{value-time}* ペアまでそのように値が強制されます。

注記：最初の *{value time}* ペアでは *time* はオプションであり、指定されていない場合は 0 であると想定されます。その後の *{value time}* ペアでは、*time* は必須です。

{value time} ペアで指定される *time* は、現在のシミュレーション時間に対して定義され、**current_time** からの期間を示します。たとえば、現在のシミュレーション時間が 1000ns の場合、*time* を 20ns にすると、1000ns ~ 1020ns までの期間が定義されます。

Restriction *time* はシミュレーションのタイム ラインで増加するように指定する必要があり、繰り返すとエラーが発生します。

time は、現在のシミュレーションのデフォルト TIME_UNIT で定義するか、時間単位をスペースなしで含めて指定できます。有効な時間単位は fs、ps、ns、us、ms、s です。50 を指定すると、デフォルトで 50ns に設定されます。50ps を指定すると、50 ピコ秒に設定されます。

例

次の例では、リセット信号をデフォルトの基数を使用して 300ns 間 High に設定し、返された force オブジェクトの名前を後で force を解除するために使用できるように Tcl 変数に保存しています。

```
set for10 [ add_force reset 1 300 ]
```

次の例では、*{value time}* ペアの使用法を示しており、定期的に繰り返して、指定の時間後キャンセルします。

```
add_force mySig {0} {1 50} {0 100} {1 150} -repeat_every 200 -cancel_after 10000
```

注記 : この例では、最初の *{value time}* ペアでは時間は設定されていません。これは、指定の値 0 が時間 0 (**current_time**) に適用されることを示します。

関連項目

- [current_time](#)
- [get_objects](#)
- [remove_forces](#)

add_wave

新しい波形を追加します。

構文

```
add_wave [-into args] [-at_wave args] [-after_wave args] [-before_wave args]
[-reverse] [-radix arg] [-color arg] [-name arg] [-recursive] [-r] [-regexp]
[-nocase] [-quiet] [-verbose] items...
```

戻り値

新しい波形

使用法

名前	説明
[-into]	新しい波形オブジェクトを挿入する波形コンフィギュレーション、グループ、または仮想バスを指定します。
[-at_wave]	指定の波形オブジェクト、あるいはグループまたは仮想バスでない場合は指定の波形オブジェクトの後に新しい波形オブジェクトを挿入します。
[-after_wave]	指定の波形オブジェクトの後に新しい波形オブジェクトを挿入します。
[-before_wave]	指定の波形オブジェクトの前に新しい波形オブジェクトを挿入します。
[-reverse]	新しい波形オブジェクトの表示ビット順を逆にします。
[-radix]	新しい波形オブジェクトの表示基数を設定します。有効な値は、default、dec、bin、oct、hex、unsigned、ascii です。
[-color]	新しい波形オブジェクトの色を設定します。標準の色の名前を指定するか、RRGGBB 形式で指定します。
[-name]	1 つの新しい波形オブジェクトの表示名を設定します。
[-recursive]	デザイン オブジェクトがスコープの場合に、そのスコープ内のデザイン オブジェクトすべての波形オブジェクトを作成します。
[-r]	デザイン オブジェクトがスコープの場合に、そのスコープ内のデザイン オブジェクトすべての波形オブジェクトを作成します。
[-regexp]	<items> を正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
items	波形オブジェクトを作成する基になるデザイン オブジェクトを指定します。

カテゴリ

波形

説明

1 つまたは複数のデザインに基づく波形オブジェクトを作成します。

このコマンドを実行すると、新しく作成された波形オブジェクトの名前が返されます。

注記：このコマンドは、シミュレーションでのみ使用できます。少なくとも、シミュレーション プロジェクトの HDL オブジェクト(信号)を 1つ指定する必要があります。Vivado インターフェイスでは、オブジェクトは [Objects] ビューに表示されます。

引数

-into *wcfgGroupVbusObj*(オプション)：新しい波形オブジェクトを挿入する波形コンフィギュレーション、グループ、または仮想バスを指定します。*wcfgGroupVbusObj*がオブジェクトではなく文字列の場合は、現在の WCFG でその名前のグループが検索されます。その名前のグループが見つからない場合は、現在の WCFG の仮想バス名が検索されます。その名前の仮想バスも見つからない場合は、すべての WCFG オブジェクトの名前が検索されます。**-into** オプションでオブジェクトを指定しない場合は、現在の波形コンフィギュレーションに追加されます。

-at_wave *waveObj*(オプション)：指定の波形オブジェクトの位置に波形オブジェクトを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-after_wave *waveObj*(オプション)：指定の波形オブジェクトの後に波形オブジェクトを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-before_wave *waveObj*(オプション)：指定の波形オブジェクトの前に波形オブジェクトを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-reverse (オプション)：新しい波形オブジェクトの **IS_REVERSED** プロパティを **true** に設定します。

-radix *arg*(オプション)：新しい波形オブジェクトの基数プロパティを設定します。有効な値は、**default**、**dec**、**bin**、**oct**、**hex**、**unsigned**、**ascii** です。

-color *arg*(オプション)：新しい波形オブジェクトの色プロパティを設定します。定義済みの色の名前を指定するか、6 枠の RGB 形式 (RRGGBB) で指定します。

-name *arg*(オプション)：新しい波形オブジェクトの **DISPLAY_NAME** プロパティを設定します。複数の波形オブジェクトが作成されている場合は、エラーになります。

-recursive | -r (オプション)：*items* でスコープが指定されている場合、このスコープのすべてのサブスコープを追加します。

-regexp (オプション)：*items* が正規表現を使用して記述されていることを指定します。ザイリンクス Tcl コマンドでは、正規表現は常に検索文字列の先頭にアンカーされています。検索文字列の先頭に「.*」を追加して、検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記：Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション)：パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

items (必須) : 指定した HDL オブジェクトの波形を現在のシミュレーションに追加します。

例

次の例では、clk を既存の波形コンフィギュレーションに追加しています。

```
add_wave clk
clk
```

次の例では、rsb_design_testbench の dout_tvalid 信号を既存のシミュレーション波形コンフィギュレーションに追加しています。

```
add_wave dout_tvalid
/rsb_design_testbench/dout_tvalid
```

関連項目

- [add_wave_divider](#)
- [add_wave_group](#)
- [add_wave_marker](#)
- [add_wave_virtual_bus](#)

add_wave_divider

仕切りを追加します。

構文

```
add_wave_divider [-into args] [-at_wave args] [-after_wave args]
[-before_wave args] [-color arg] [-quiet] [-verbose] [name]
```

戻り値

新しい仕切り

使用法

名前	説明
[-into]	新しい仕切りを挿入する波形コンフィギュレーションまたはグループを指定します。
[-at_wave]	指定の波形オブジェクト、あるいはグループでない場合は指定の波形オブジェクトの後に新しい仕切りを挿入します。
[-after_wave]	指定の波形オブジェクトの後に新しい仕切りを挿入します。
[-before_wave]	指定の波形オブジェクトの前に新しい仕切りを挿入します。
[-color]	新しい仕切りの色を設定します。標準の色の名前を指定するか、RRGGBB 形式で指定します。デフォルトは default です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[name]	新しい仕切りの表示名を指定します。デフォルトは new_divider です。

カテゴリ

波形

説明

波形ビューアーに仕切りを追加します。仕切りは、見やすくするために関連のオブジェクトのグループを仕切るために使用します。

仕切りは、指定の波形コンフィギュレーションまたは現在の波形コンフィギュレーションの指定の位置に追加できます。位置が指定されていない場合は、波形コンフィギュレーションの最後に追加されます。

このコマンドを実行すると、新しく作成された仕切りの名前が返されます。

注記： このコマンドは、シミュレーションでのみ使用できます。

引数

-into *wcfgGroupVbusObj*(オプション) : 仕切りを挿入する波形コンフィギュレーション、グループ、または仮想バスを指定します。*wcfgGroupVbusObj*がオブジェクトではなく文字列の場合は、現在の WCFG でその名前のグループが検索されます。その名前のグループが見つからない場合は、現在の WCFG の仮想バス名が検索されます。その名前の仮想バスも見つからない場合は、すべての WCFG オブジェクトの名前が検索されます。**-into** オプションでオブジェクトを指定しない場合は、現在の波形コンフィギュレーションに追加されます。

-at_wave *waveObj*(オプション) : 指定の波形オブジェクトの位置に仕切りを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-after_wave *waveObj*(オプション) : 指定の波形オブジェクトの後に仕切りを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-before_wave *waveObj*(オプション) : 指定の波形オブジェクトの前に仕切りを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-color *arg*(オプション) : 仕切りの色プロパティを設定します。定義済みの色の名前を指定するか、6 衍の RGB 形式 (RRGGBB) で指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name(オプション) : 指定した表示名の仕切りを作成します。デフォルト名は new_divider です。

例

次の例では、CLK 波形オブジェクトの後に Div1 という名前の仕切りを挿入します。

```
add_wave_divider -after_wave CLK Div1
```

関連項目

- [add_wave](#)
- [add_wave_group](#)
- [add_wave_marker](#)
- [add_wave_virtual_bus](#)

add_wave_group

グループを追加します。

構文

```
add_wave_group [-into args] [-at_wave args] [-after_wave args]
[-before_wave args] [-quiet] [-verbose] [name]
```

戻り値

新しいグループ

使用法

名前	説明
[-into]	新しいグループを挿入する波形コンフィギュレーションまたはグループを指定します。
[-at_wave]	指定の波形オブジェクト、あるいはグループでない場合は指定の波形オブジェクトの後に新しいグループを挿入します。
[-after_wave]	指定の波形オブジェクトの後に新しいグループを挿入します。
[-before_wave]	指定の波形オブジェクトの前に新しいグループを挿入します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[name]	新しいグループの表示名を指定します。デフォルトは new_group です。

カテゴリ

波形

説明

指定の波形コンフィギュレーションまたは現在の波形コンフィギュレーションに波形グループを作成します。波形オブジェクトと仕切りを波形グループに追加し、波形表示を見やすくなります。

波形グループは、指定の位置に挿入できます。位置が指定されていない場合は、指定の波形コンフィギュレーションの最後に追加されます。

このコマンドを実行すると、新しく作成された波形グループ オブジェクトの名前が返されます。

注記： このコマンドは、シミュレーションでのみ使用できます。

引数

-into *wcfgGroupVbusObj*(オプション) : 波形グループを挿入する波形コンフィギュレーション、グループ、または仮想バスを指定します。*wcfgGroupVbusObj*がオブジェクトではなく文字列の場合は、現在の WCFG でその名前のグループが検索されます。その名前のグループが見つからない場合は、現在の WCFG の仮想バス名が検索されます。その名前の仮想バスも見つからない場合は、すべての WCFG オブジェクトの名前が検索されます。

-into オプションでオブジェクトを指定しない場合は、現在の波形コンフィギュレーションに追加されます。

-at_wave *waveObj*(オプション) : 指定の波形オブジェクトの位置に波形グループを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-after_wave *waveObj*(オプション) : 指定の波形オブジェクトの後に波形グループを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-before_wave *waveObj*(オプション) : 指定の波形オブジェクトの前に波形グループを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (オプション) : 指定した表示名の波形グループを作成します。デフォルト名は new_group です。

例

次の例では、clk を既存の波形コンフィギュレーションに追加しています。

```
add_wave_group clk
group10
```

関連項目

- [add_wave](#)
- [add_wave_divider](#)
- [add_wave_marker](#)
- [add_wave_virtual_bus](#)

add_wave_marker

新しい波形マーカーを作成します。

構文

```
add_wave_marker [-into arg] [-name arg] [-quiet] [-verbose] [time] [unit]
```

戻り値

作成されたマーカー

使用法

名前	説明
[-into]	マーカーを作成する波形コンフィギュレーションを指定します。
[-name]	マーカーの名前を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[time]	新しいマーカーを指定した時間に作成します。デフォルトは 0 です。
[unit]	マーカーの時間の単位を指定します。有効な値は fs、ps、ns、us、ms、および s です。

カテゴリ

波形

説明

現在の波形コンフィギュレーションの指定の時間に指定した名前の波形マーカーを作成します。

このコマンドを実行しても、何も返されません。

注記： このコマンドは、シミュレーションでのみ使用できます。

引数

-into wcfg (オプション)：波形マーカーを挿入する波形コンフィギュレーションを指定します。**-into** オプションを指定しない場合、波形マーカーは現在の波形コンフィギュレーションに追加されます。

-name arg (オプション)：マーカーの名前を指定します。デフォルト名は **new_marker** です。

time (オプション)：マーカーを追加する波形上のシミュレーション時間指定します。デフォルト値は時間 0 です。

unit (オプション)：時間の単位を指定します。有効な値は **s**、**ms**、**us**、**ns**、および **ps** です。デフォルトは、指定の波形コンフィギュレーションで使用されている時間単位です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、既存の波形コンフィギュレーションの時間 500ns にマーカーを追加しています。

```
add_wave_marker 500 ns
```

関連項目

- [add_wave](#)
- [add_wave_divider](#)
- [add_wave_group](#)
- [add_wave_virtual_bus](#)

add_wave_virtual_bus

新しい仮想バスを追加します。

構文

```
add_wave_virtual_bus [-into args] [-at_wave args] [-after_wave args]
[-before_wave args] [-reverse] [-radix arg] [-color arg] [-quiet] [-verbose]
[name]
```

戻り値

新しい仮想バス

使用法

名前	説明
[-into]	新しい仮想バスを挿入する波形コンフィギュレーション、グループ、または仮想バスを指定します。
[-at_wave]	指定の波形オブジェクト、あるいはグループまたは仮想バスでない場合は指定の波形オブジェクトの後に新しい仮想バスを挿入します。
[-after_wave]	指定の波形オブジェクトの後に新しい仮想バスを挿入します。
[-before_wave]	指定の波形オブジェクトの前に新しい仮想バスを挿入します。
[-reverse]	新しい仮想バスの表示ビット順を逆にします。
[-radix]	新しい仮想バスの表示基數を設定します。有効な値は、default、dec、bin、oct、hex、unsigned、ascii です。
[-color]	新しい仮想バスの色を設定します。標準の色の名前を指定するか、RRGGBB 形式で指定します。デフォルトは default です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[name]	新しい仮想バスの表示名を指定します。デフォルトは new_virtual_bus です。

カテゴリ

波形

説明

新しい仮想バスを作成します。既存の波形の指定した位置またはデフォルトでは最後に、指定の名前の仮想バスが挿入されます。作成された仮想バス `vb###` が返されます。

注記：このコマンドは、シミュレーションでのみ使用できます。少なくとも、新しい仮想バスの名前 `name` を指定する必要があります。

引数

-into *wcfgGroupVbusObj*(オプション) : 仮想バスを挿入する波形コンフィギュレーション、グループ、または仮想バスを指定します。*wcfgGroupVbusObj*がオブジェクトではなく文字列の場合は、現在の WCFG でその名前のグループが検索されます。その名前のグループが見つからない場合は、現在の WCFG の仮想バス名が検索されます。その名前の仮想バスも見つからない場合は、すべての WCFG オブジェクトの名前が検索されます。**-into** オプションでオブジェクトを指定しない場合は、現在の波形コンフィギュレーションに追加されます。

-at_wave *waveObj*(オプション) : 指定の波形オブジェクトの位置に仮想バスを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-after_wave *waveObj*(オプション) : 指定の波形オブジェクトの後に仮想バスを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-before_wave *waveObj*(オプション) : 指定の波形オブジェクトの前に仮想バスを追加します。*waveObj*が文字列の場合は、波形オブジェクトの表示名として処理されます。

-reverse (オプション) : 新しい仮想バス オブジェクトの **IS_REVERSED** プロパティを **true** に設定します。

-radix *value* (オプション) : 新しい仮想バス オブジェクトの基底プロパティを設定します。有効な値は、**default**、**dec**、**bin**、**oct**、**hex**、**unsigned**、**ascii** です。

-color *arg*(オプション) : 新しい仮想バス オブジェクトの色プロパティを設定します。定義済みの色の名前を指定するか、6 衔の RGB 形式 (RRGGBB) で指定します。

-name *customName*(オプション) : 新しい仮想バス オブジェクトの **DISPLAY_NAME** プロパティを設定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、**dout_tvalid** という名前の仮想バスを既存の波形コンフィギュレーションに追加しています。

```
add_wave_virtual_bus dout_tvalid
    vbus200
```

関連項目

- [add_wave_divider](#)
- [add_wave_group](#)
- [add_wave_marker](#)
- [add_wave](#)

all_clocks

現在のデザインに含まれるすべてのクロックのリストを取得します。

構文

all_clocks [-quiet] [-verbose]

戻り値

クロック オブジェクトのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、XDC

説明

現在のデザインで宣言されたすべてのクロックのリストを返します。デザインの特定のクロックのリストを取得するには、**get_clocks** コマンドを使用します。

クロックを定義するには、**create_clock** または **create_generated_clock** コマンドを使用します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、サンプル CPU ネットリスト プロジェクトのすべてのクロックが表示されます。

```
% all_clocks
cpuClk wbClk usbClk phy_clk_pad_0_i phy_clk_pad_1_i fftClk
```

次の例では、**set_propagated_clock** コマンドをすべてのクロックに適用し、返されたリスト (**all_clocks**) を別のコマンドに渡しています。

```
% set_propagated_clock [all_clocks]
```

関連項目

- [create_clock](#)
- [create_generated_clock](#)
- [get_clocks](#)
- [set_propagated_clock](#)

all_cpus

現在のデザインの CPU セルのリストを取得します。

構文

all_cpus [-quiet] [-verbose]

戻り値

CPU セル オブジェクトのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

xdc

説明

現在のデザインの CPU セル オブジェクトのリストを返します。現在のデザインで宣言されたすべての CPU セル オブジェクトのリストを作成します。

all_cpus コマンドでは、デザインの最上位または現在のインスタンスのレベルにあるオブジェクトが返されます。デフォルトでは、デザインの最上位が現在のインスタンスとして定義されますが、これは **current_instance** コマンドを使用して変更できます。

注記：このコマンドを実行すると、CPU セル オブジェクトのリストが返されます。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインの CPU オブジェクトすべてが返されます。

all_cpus

次の例では、返されたリストを別のコマンドに渡しています。

```
set_false_path -from [all_cpus] -to [all_registers]
```

関連項目

- [all_dsp](#)
- [all_hsi](#)
- [all_registers](#)
- [current_instance](#)
- [get_cells](#)
- [set_false_path](#)

all_dsp

現在のデザインの DSP セルのリストを取得します。

構文

```
all_dsp [-quiet] [-verbose]
```

戻り値

DSP セル オブジェクトのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

xdc

説明

現在のデザインで宣言されたすべての DSP セル オブジェクトのリストを返します。

all_dsp コマンドでは、デザインの最上位または現在のインスタンスのレベルにあるオブジェクトが返されます。デフォルトでは、デザインの最上位が現在のインスタンスとして定義されますが、これは **current_instance** コマンドを使用して変更できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインで定義されたすべての DSP のリストが返されます。

```
all_dsp
```

次の例では、返されたリストを別のコマンドに渡しています。

```
set_false_path -from [all_dsps] -to [all_registers]
```

関連項目

- [all_cpus](#)
- [all_hsiros](#)
- [all_registers](#)
- [current_instance](#)
- [get_cells](#)
- [set_false_path](#)

all_fanin

指定したシンクのファンインのピンまたはセルのリストを取得します。

構文

```
all_fanin [-startpoints_only] [-flat] [-only_cells] [-levels arg]
[-pin_levels arg] [-trace_arcs arg] [-quiet] [-verbose] to
```

戻り値

セルまたはピン オブジェクトのリスト

使用法

名前	説明
[-startpoints_only]	タイミング開始点のみを検出します。
[-flat]	階層を無視します。
[-only_cells]	セルのみを検出します。
[-levels]	処理するセル レベルの最大数を指定します。0 以上の値を指定します。デフォルト値は 0 です。
[-pin_levels]	処理するピン レベルの最大数を指定します。0 以上の値を指定します。デフォルト値は 0 です。
[-trace_arcs]	トレースするネットワーク アークのタイプを指定します。有効な値は timing、enabled、all です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
to	シンク ピン、ポート、またはネットのリストを指定します。

カテゴリ

SDC、XDC

説明

指定したシンクのファンインのポート、ピン、またはセルのリストを返します。

all_fanin コマンドでは、デザインの最上位または現在のインスタンスのレベルにあるオブジェクトが返されます。デフォルトでは、デザインの最上位が現在のインスタンスとして定義されますが、これは **current_instance** コマンドを使用して変更できます。すべての階層レベルのファンインを取得するには、**-flat** オプションを使用します。

引数

-startpoints_only (オプション) : タイミング開始点のみを検出します。このオプションを使用すると、ファンイン ネットワークの中間点は返されません。このオプションは、シンクのプライマリドライバーを特定する場合に使用できます。

-flat (オプション) : デザインの階層を無視します。デフォルトでは、シンクと同じ階層レベルのオブジェクトのみが返されます。このオプションを使用すると、階層にかかわらず、シンクのファンイン ネットワークのすべてのオブジェクトが返されます。

-only_cells (オプション) : 指定したシンクのファンイン パスにあるセル オブジェクトのみを返します。ピンまたはポートは返しません。

-levels value (オプション) : 処理するセル レベルの最大数を指定します。デフォルト値は 0 です。

-pin_levels value (オプション) : 処理するピン レベルの最大数を指定します。デフォルト値は 0 です。

-trace_arcs value (オプション) : トレースするネットワーク アークのタイプを指定します。有効な値は timing、enabled、および all です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

to (必須) : ファンイン オブジェクトをレポートするピン、ポート、またはネットを指定します。

例

次の例では、led_pins 出力ポートのタイミング ファンインがリストされます。

```
all_fanin [get_ports led_pins[*]]
```

次の例では、指定のフリップフロップのクロック ピンからクロック ソース (この例の場合は MMCM 出力ピン) までをトレースします。

```
all_fanin -flat -startpoints_only [get_pins cmd_parse_i0/prescale_reg[7]/C]
```

関連項目

- [all_fanout](#)
- [current_instance](#)
- [get_cells](#)
- [get_pins](#)
- [get_ports](#)

all_fanout

指定したソースのファンアウトのピンまたはセルのリストを取得します。

構文

```
all_fanout [-endpoints_only] [-flat] [-only_cells] [-levels arg]
[-pin_levels arg] [-trace_arcs arg] [-quiet] [-verbose] from
```

戻り値

セルまたはピン オブジェクトのリスト

使用法

名前	説明
[-endpoints_only]	タイミング終点のみを検出します。
[-flat]	階層を無視します。
[-only_cells]	セルのみを検出します。
[-levels]	処理するセル レベルの最大数を指定します。0 以上の値を指定します。デフォルト値は 0 です。
[-pin_levels]	処理するピン レベルの最大数を指定します。0 以上の値を指定します。デフォルト値は 0 です。
[-trace_arcs]	トレースするネットワーク アークのタイプを指定します。有効な値は timing、enabled、all です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
from	ソース ピン、ポート、またはネットのリストを指定します。

カテゴリ

SDC、XDC

説明

指定したソースのファンアウトのポート、ピン、またはセルのリストを返します。

all_fanout コマンドでは、デザインの最上位または現在のインスタンスのレベルにあるオブジェクトが返されます。デフォルトでは、デザインの最上位が現在のインスタンスとして定義されますが、これは current_instance コマンドを使用して変更できます。すべての階層レベルのファンアウトを取得するには、-flat オプションを使用します。

引数

-endpoints_only (オプション) : タイミング終点のみを検出します。このオプションを使用すると、ファンアウト ネットワークの中間点は返されません。このオプションは、ドライバーのプライマリ ロードを特定する場合に使用できます。

-flat (オプション) : デザインの階層を無視します。デフォルトでは、シンクと同じ階層レベルのオブジェクトのみが返されます。このオプションを使用すると、階層にかかわらず、ドライバーのファンアウト ネットワークのすべてのオブジェクトが返されます。

-only_cells (オプション) : 指定したソースのファンアウト パスにあるセル オブジェクトのみを返します。

-levels value (オプション) : 処理するセル レベルの最大数を指定します。デフォルト値は 0 です。

-pin_levels value (オプション) : 処理するピン レベルの最大数を指定します。デフォルト値は 0 です。

-trace_arcs value (オプション) : トレースするネットワーク アークのタイプを指定します。有効な値は timing、enabled、および all です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

from (必須) : ファンアウト パスのオブジェクトをリストするソース ポート、ピン、またはネットを指定します。

例

次の例では、デザインのすべての入力ポートのファンアウトを取得しています。

```
all_fanout [all_inputs]
```

関連項目

- [all_fanin](#)
- [current_instance](#)
- [get_cells](#)
- [get_pins](#)
- [get_ports](#)

all_ffs

現在のデザインのフリップフロップ セルのリストを取得します。

構文

all_ffs [-quiet] [-verbose]

戻り値

フリップフロップ セル オブジェクトのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

xdc

説明

現在のデザインのフリップフロップ インスタンスすべてのリストを返します。

all_ffs コマンドでは、デザインの最上位または現在のインスタンスのレベルにあるオブジェクトが返されます。デフォルトでは、デザインの最上位が現在のインスタンスとして定義されますが、これは **current_instance** コマンドを使用して変更できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、デザインのすべてのフリップフロップを返し、fftEngine モジュールに含まれるフリップフロップをカウントしています。

```
current_instance
INFO: [Vivado 12-618] Current instance is the top level of design 'netlist_1'.
top
llength [all_ffs]
15741
current_instance fftEngine
fftEngine
llength [all_ffs]
1519
```

次の例では、指定したフリップフロップに設定されているプロパティが返されます。

```
report_property [lindex [all_ffs] 2 ]
```

関連項目

- [all_latches](#)
- [all_registers](#)
- [current_instance](#)
- [get_cells](#)
- [report_property](#)

all_hsios

現在のデザインの HSIO セルのリストを取得します。

構文

all_hsios [-quiet] [-verbose]

戻り値

HSIO セル オブジェクトのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

xdc

説明

現在のデザインで宣言されたすべての高速 I/O (HSIO) セル オブジェクトのリストを返します。これらの HSIO セル オブジェクトは、変数に代入するか、別のコマンドに渡すことができます。

all_hsios コマンドでは、デザインの最上位または現在のインスタンスのレベルにあるオブジェクトが返されます。デフォルトでは、デザインの最上位が現在のインスタンスとして定義されますが、これは **current_instance** コマンドを使用して変更できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインの HSIO オブジェクトすべてが返されます。

all_hsios

次の例では、返されたリストを直接別のコマンドに渡しています。

```
set_false_path -from [all_hsios] -to [all_registers]
```

関連項目

- [all_cpus](#)
- [all_dsp](#)s
- [all_registers](#)
- [get_cells](#)
- [set_false_path](#)

all_inputs

現在のデザインの入力ポートすべてのリストを取得します。

構文

```
all_inputs [-quiet] [-verbose]
```

戻り値

ポート オブジェクトのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、xdc

説明

現在のデザインの入力ポート オブジェクトすべてのリストを返します。

all_inputs コマンドでは、デザインの最上位または現在のインスタンスのレベルにあるオブジェクトが返されます。デフォルトでは、デザインの最上位が現在のインスタンスとして定義されますが、これは **current_instance** コマンドを使用して変更できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインの入力ポート オブジェクトすべてが返されます。

```
all_inputs
```

次の例では、返されたリストを別のコマンドに渡しています。

```
set_input_delay 5 -clock REFCLK [all_inputs]
```

関連項目

- [all_clocks](#)
- [all_outputs](#)
- [current_instance](#)
- [get_clocks](#)
- [get_ports](#)
- [set_input_delay](#)

all_latches

現在のデザインに含まれるラッチ セルすべてのリストを取得します。

構文

all_latches [-quiet] [-verbose]

戻り値

ラッチ セル オブジェクトのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

xdc

説明

現在のデザインで宣言されているすべてのラッチのリストを返します。

all_latches コマンドでは、デザインの最上位または現在のインスタンスのレベルにあるオブジェクトが返されます。デフォルトでは、デザインの最上位が現在のインスタンスとして定義されますが、これは **current_instance** コマンドを使用して変更できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインに含まれるすべてのラッチのリストが返されます。

all_latches

次の例では、返されたリストを別のコマンドに渡しています。

```
set_false_path -from [all_mults] -to [all_latches]
```

関連項目

- [all_ffs](#)
- [all_registers](#)
- [current_instance](#)
- [get_cells](#)
- [set_false_path](#)

all_outputs

現在のデザインの出力ポートすべてのリストを取得します。

構文

all_outputs [-quiet] [-verbose]

戻り値

ポート オブジェクトのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、xdc

説明

現在のデザインで宣言されたすべての出力ポート オブジェクトのリストを返します。

all_outputs コマンドでは、デザインの最上位または現在のインスタンスのレベルにあるオブジェクトが返されます。デフォルトでは、デザインの最上位が現在のインスタンスとして定義されますが、これは **current_instance** コマンドを使用して変更できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインの出力ポート オブジェクトすべてが返されます。

all_outputs

次の例では、デザインのすべての出力の出力遅延が設定されます。

```
set_output_delay 5 -clock REFCLK [all_outputs]
```

関連項目

- [all_inputs](#)
- [current_instance](#)
- [get_ports](#)
- [set_output_delay](#)

all_rams

現在のデザインの RAM セルのリストを取得します。

構文

all_rams [-quiet] [-verbose]

戻り値

RAM セル オブジェクトのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

xdc

説明

現在のインスタンスに含まれるすべての RAM セル オブジェクト (ブロック RAM、ブロック RAM FIFO、および分散 RAM を含む) のリストを返します。これらの RAM セル オブジェクトは、変数に代入するか、別のコマンドに渡すことができます。

all_rams コマンドでは、デザインの最上位または現在のインスタンスのレベルにあるオブジェクトが返されます。デフォルトでは、デザインの最上位が現在のインスタンスとして定義されますが、これは **current_instance** コマンドを使用して変更できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、デザインの RAM オブジェクトすべてが返されます。

all_rams

次の例では、現在のインスタンスを設定し、現在のインスタンスの階層レベルにあるすべての RAM オブジェクトを返しています。

```
current_instance usbEngine0
all_rams
```

関連項目

- [all_clocks](#)
- [all_cpus](#)
- [all_dspes](#)
- [all_fanin](#)
- [all_fanout](#)
- [all_ffs](#)
- [all_hsios](#)
- [all_inputs](#)
- [all_latches](#)
- [all_outputs](#)
- [all_registers](#)
- [current_instance](#)
- [get_cells](#)

all_registers

現在のデザインのレジスタ セルまたはピンのリストを取得します。

構文

```
all_registers [-clock args] [-rise_clock args] [-fall_clock args] [-cells]
[-data_pins] [-clock_pins] [-async_pins] [-output_pins] [-level_sensitive]
[-edge_triggered] [-no_hierarchy] [-quiet] [-verbose]
```

戻り値

セルまたはピン オブジェクトのリスト

使用法

名前	説明
[-clock]	指定したクロックが供給されるレジスタを取得します。
[-rise_clock]	クロックの立ち上がりエッジでトリガーされるレジスタを取得します。
[-fall_clock]	クロックの立ち下がりエッジでトリガーされるレジスタを取得します。
[-cells]	セルのリストを返します (デフォルト)。
[-data_pins]	レジスタのデータ ピンのリストを返します。
[-clock_pins]	レジスタのクロック ピンのリストを返します。
[-async_pins]	非同期プリセット/クリア ピンのリストを返します。
[-output_pins]	レジスタの出力ピンのリストを返します。
[-level_sensitive]	レベルで認識されるラッチのみを返します。
[-edge_triggered]	エッジでトリガーされるフリップフロップのみを返します。
[-no_hierarchy]	現在のインスタンスのみを検索します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、XDC

説明

現在のデザインに含まれるシーケンシャル レジスタ セルまたはレジスタ ピンのリストを返します。

返されるオブジェクトは、下記のオプションを使用することにより制限できます。指定のクロックに制限したり、指定のクロックの立ち上がりエッジまたは立ち下がりエッジでトリガーされるレジスタに制限したりすることができます。

また、ピン オプションのいずれかを使用することにより、レジスタ オブジェクトではなくレジスタ ピンを取得できます。

引数

-cells (オプション) : ピン オブジェクトではなくレジスタ セル オブジェクトを返します。これがデフォルトの動作です。

-clock args (オプション) : 指定したクロックのファンアウトにクロック ピンが含まれるレジスタすべてのリストを返します。

-rise_clock args (オプション) : 指定したクロックの立ち上がりエッジでトリガーされるレジスタのリストを返します。

-fall_clock args (オプション) : 指定したクロックの立ち下がりエッジでトリガーされるレジスタのリストを返します。

注記 : **-clock**、**-rise_clock**、および **-fall_clock** を同じコマンドで同時に使用しないでください。

-level_sensitive (オプション) : レベルで認識されるレジスタまたはラッチのリストを返します。

-edge_triggered (オプション) : エッジでトリガーされるレジスタまたはフリップフロップのリストを返します。

-data_pins (オプション) : デザインに含まれるすべてのレジスタまたは検索条件を満たすレジスタのデータ ピンのリストを返します。

-clock_pins (オプション) : 検索条件を満たすレジスタのクロック ピンのリストを返します。

-async_pins (オプション) : 検索条件を満たすレジスタの非同期ピンを返します。

-output_pins (オプション) : 検索条件を満たすレジスタの出力ピンのリストを返します。

注記 : ***_pins** オプションは個別に使用してください。複数のオプションを同時に使用すると、1 つのオプションのみが **-data_pins**、**-clock_pins**、**-async_pins**、**-output_pins** の優先順位で使用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、デザインの任意のクロックの立ち下がりエッジでトリガーされるレジスタのリストが返されます。

```
all_registers -fall_clock [all_clocks]
```

次の例では、返されたリストを別のコマンドに渡しています。

```
set_min_delay 2.0 -to [all_registers -clock CCLK -data_pins]
```

関連項目

- [all_clocks](#)
- [set_msg_limit](#)
- [set_min_delay](#)

apply_bd_automation

IP インテグレーター オブジェクトに対してオートメーション ルールを実行します。

構文

```
apply_bd_automation [-config args] -rule arg [-quiet] [-verbose] objects...
```

戻り値

実行結果 (正常に完了したか、エラーが発生したか)

使用法

名前	説明
<code>[-config]</code>	パラメーター値のペアをリストします。
<code>-rule</code>	ルール ID 文字列を指定します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>objects</code>	自動化ルールを実行するオブジェクトを指定します。

カテゴリ

IP インテグレーター

説明

IP インテグレーターには、**apply_bd_automation** コマンドを使用した設計アシスト機能が含まれており、選択した IP インテグレーター オブジェクトの周辺にあるほかの関連の IP インテグレーター セルを自動的にコンフィギュレーションしたり追加したりできます。設計アシスト機能の詳細は、『Vivado Design Suite ユーザー ガイド：IP を使用した設計』(UG896) または『Vivado Design Suite ユーザー ガイド：IP インテグレーターを使用した IP サブシステムの設計』(UG994) を参照してください。

現在のところ、セル、インターフェイス、ピン、ポートに対してブロック オートメーションおよびコネクション オートメーションがあります。ブロック オートメーション機能は、Zynq プロセッsing システム 7、MicroBlaze プロセッサ、AXI イーサネット、MIG メモリ コントローラなどの複雑なブロックに対して使用できます。

コネクション オートメーション機能は、異なるタイプの接続を自動化します。たとえば、スレーブ AXI-MM インターフェイスを接続する際、オートメーションにより関連のクロックとリセットピンが接続され、必要な場合はインターフェクトが作成されます。また、ボード レベルの接続では、関連のセルからのピンおよびインターフェイスを外部ポートおよびインターフェイスに接続し、これらの外部 I/O に適切なボード制約を適用します。

注記：この IP インテグレーター コマンドは、Vivado IDE の設計アシスト GUI から実行します。この機能は、Tcl スクリプトから直接使用するのではなく、Vivado IDE の IP インテグレーターから使用することをお勧めします。ユーザー スクリプトで使用するには、**write_bd_tcl** コマンドを使用して Tcl を出力します。

引数

-config args (オプション) : IP インテグレーター オブジェクトのコンフィギュレーション パラメーターとその値のリストを指定します。パラメータ名 (param) はクオーテーションを使用せずに指定し、値にはクオーテーションを付けて param と区別します。パラメーターと値はペアで指定します (param "value")。指定のオブジェクトに対して複数の param "value" ペアを指定するには、中かっこ {} で囲みます。

```
-config {local_mem "16KB" ecc "Basic" debug_module "Debug Only"}
```

-rule arg (必須) : 選択したオブジェクトに使用する定義されたオートメーション ルールを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須) : オートメーションを適用する IP インテグレーター オブジェクトを指定します。オブジェクトは、**get_bd_cells**、**get_bd_pins**、**get_bd_interface** などのコマンドで取得します。名前で参照することはできません。1 つの IP インテグレーター オブジェクトのみを指定し、**-rule** および **-config** オプションと互換している必要があります。

例

次の例では、ブロック オートメーションを MicroBlaze セルに指定したルールで適用し、指定のパラメーターを指定の値に設定します。

```
apply_bd_automation -rule xilinx.com:bd_rule:microblaze \
    -config {local_mem "16KB" ecc "Basic" debug_module "Debug Only" \
    axi_periph "1" axi_intc "1" clk "New Clocking Wizard (100 MHz)" } \
    [get_bd_cells /microblaze_1]
```

次の例では、コネクション オートメーション機能により、ボードに既知の互換インターフェイスがある場合にボード ルールを IP サブシステム ピンまたはインターフェイス (クロック インターフェイス オブジェクト) に適用します。最初の **get_board_interfaces** コマンドは、IP オブジェクトと互換性のあるターゲット ボード上のインターフェイスを返します。2 番目の **apply_bd_automation** コマンドは、クロック インターフェイスを選択したボード インターフェイスに接続します。

```
get_board_interfaces -filter "VLNV==[get_property VLNV \
    [get_bd_intf_pins clk_wiz_1/CLK_IN1_D]]"
sys_diff_clock
apply_bd_automation -rule xilinx.com:bd_rule:board \
    -config {Board_Interface "sys_diff_clock" } \
    [get_bd_intf_pins /clk_wiz_1/CLK_IN1_D]
```

次の例では、ターゲット ボードにクロック インターフェイスがない場合に、カスタム ボード ルールを IP サブシステム クロック インターフェイス オブジェクト CLK_IN1_D に適用します。

```
apply_bd_automation -rule xilinx.com:bd_rule:board \
    [get_bd_pins /clk_wiz_2/CLK_IN1_D]
```

次の例では、ボードにリセットインターフェイスがない場合に、カスタム ボード ルールを IP サブシステム リセット ピン `ext_reset_in` に適用します。

```
apply_bd_automation -rule xilinx.com:bd_rule:board \
    -config {rst_polarity "ACTIVE_HIGH" } \
    [get_bd_pins /proc_sys_reset_1/ext_reset_in]
```

関連項目

- [create_bd_cell](#)
- [create_bd_design](#)
- [get_bd_cells](#)
- [get_bd_intf_pins](#)
- [get_bd_pins](#)
- [get_board_interfaces](#)
- [write_bd_tcl](#)

archive_project

現在のプロジェクトのアーカイブを作成します。

構文

```
archive_project [-force] [-exclude_run_results] [-include_config_settings]
[-quiet] [-verbose] [file]
```

戻り値

True

使用法

名前	説明
[-force]	既存のアーカイブ ファイルを上書きします。
[-exclude_run_results]	アーカイブに run の結果を含めません。
[-include_config_settings]	アーカイブに現在のプロジェクトの環境コンフィギュレーション設定/ファイルを含めます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[file]	アーカイブ ファイル名を指定します。

カテゴリ

プロジェクト

説明

プロジェクトのアーカイブを作成して、プロジェクトのバックアップとして保存したり、リモート サイトに送信したりするために使用します。デザイン階層を解析し、必要なソース ファイル、インクルード ファイル、リモート ファイルがをライブラリ ディレクトリからコピーし、制約ファイルをコピーし、さまざまな合成、シミュレーション、インプリメンテーション run の結果をコピーしてから、プロジェクトの ZIP ファイルを作成します。

注記：プロジェクトをアーカイブする別の方法として、`write_project_tcl` を使用して、現在の形のプロジェクトを作成する Tcl スクリプトを作成することもできます。

引数

-force (オプション)：同じ名前の既存の ZIP ファイルを上書きします。ZIP ファイルが存在する場合に、**-force** が指定されていないと、エラー メッセージが表示されます。

-exclude_run_results (オプション)：合成またはインプリメンテーション run の結果を除外します。このコマンドにより、プロジェクト アーカイブのサイズを大幅に縮小できます。

-include_config_settings (オプション) : 初期化 Tcl コマンド (init.tcl) をアーカイブ ファイルの *project_name/config_settings* フォルダーに追加します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (オプション) : **archive_project** コマンドで作成する ZIP ファイルの名前を指定します。*file* を指定しない場合、プロジェクトと同じ名前の ZIP ファイルが作成されます。

例

次の例では、現在のプロジェクトのアーカイブが作成されます。

```
archive_project
```

注記 : ファイル名が指定されていないので、プロジェクトアーカイブの名前は *project_name.zip* となります。

次の例では、現在のプロジェクトとして *project_3* を指定し、*proj3.zip* というプロジェクトのアーカイブを作成しています。

```
current_project project_3
archive_project -force -exclude_run_results proj3.zip
```

注記 : **-force** を使用しているので、*proj3.zip* ファイルが既に存在している場合は上書きされます。

-exclude_run_results を使用しているので、合成 run またはインプリメンテーション run の結果はアーカイブに含まれません。プロジェクトで定義されたさまざまな run は含まれますが、その結果は含まれません。

関連項目

- [current_project](#)
- [write_project_tcl](#)

assign_bd_address

マップされていない IP に自動的にアドレスを割り当てます。

構文

```
assign_bd_address [-target_address_space arg] [-quiet] [-verbose] [objects...]
```

戻り値

新しくマップされたセグメント、エラーが発生した場合は ""

使用法

名前	説明
[-target_address_space]	セグメントを配置するターゲット アドレス空間を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	割り当てるオブジェクトを指定します。

カテゴリ

IP インテグレーター

説明

マップされていない IP アドレス セグメントを IP インテグレーター サブシステム デザインのアドレス空間に割り当てます。ターゲット アドレス空間が指定されていない場合、IP インテグレーターによりアドレス セグメントが使用可能なアドレス空間に自動的に割り当てられます。

このコマンドが正常に実行された場合は新しくマップされたアドレス セグメントが返され、正常に実行できなかつた場合はエラーが返されます。

引数

-target_address_space arg (オプション) : セグメントを配置するターゲット アドレス空間を指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects : ターゲット アドレス空間に割り当てるアドレス セグメント オブジェクトを指定します。

例

次の例では、指定のアドレス セグメント オブジェクトをアドレス空間に自動的に割り当てています。この例ではターゲット空間が指定されていないので、IP インテグレーターにより自動的に割り当てられます。

```
assign_bd_address [get_bd_addr_segs \
{/microblaze_1_local_memory/ilmb_bram_if_cntlr/SLMB/Mem }]
```

関連項目

- [create_bd_addr_seg](#)
- [get_bd_addr_segs](#)
- [get_bd_addr_spaces](#)

check_timing

デザインで発生する可能性のあるタイミング問題をチェックします。

構文

```
check_timing [-file arg] [-name arg] [-override_defaults args]
[-include args] [-exclude args] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-name]	GUI パネルに表示する結果の名前を指定します。
[-override_defaults]	デフォルトのタイミング チェックではなく、指定したチェックを実行します。
[-include]	デフォルトのタイミング チェックに加えて実行するチェックを指定します。
[-exclude]	デフォルトのタイミング チェックから指定のチェックを除外します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート、タイミング

説明

ポート、ピン、およびパスのデザイン エレメントを現在のタイミング制約に対してチェックします。**report_timing** コマンドを実行する前にデザイン データおよびタイミング制約に問題がないかどうかを確認する場合に使用します。**check_timing** コマンドはデフォルトタイミング チェックを実行し、検出された違反のサマリをレポートします。違反に関する詳細を取得するには、**-verbose** オプションを使用します。

注記：デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、GUI に表示したりファイルに書き込んだりすることもできます。

デフォルトのタイミング チェックは、次のとおりです。

- **no_clock** : クロックが供給されていないレジスタをレポートします。レジスタにクロックが供給されていない場合、データピンでレジスタ クロックピンに対するセットアップまたはホールド チェックが実行されません。
 - **unconstrained_internal_endpoints** : レジスタ データピンでの 制約が設定されていないタイミング パス エンドポイントをレポートします。エンドポイントがレジスタ データピンの場合、**create_clock** コマンドを使用して割り当てたクロックで制約されます。出力ポートでのエンドポイントは、**no_output_delay** チェックによりチェックおよびレポートされます。
 - **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。入力遅延は、**set_input_delay** コマンドを使用して指定できます。クロックに同期しない入力ポートでは、入力遅延はチェックされません。
 - **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。出力遅延は、**set_output_delay** コマンドを使用して指定できます。クロックに同期しない出力ポートでは、出力遅延はチェックされません。
 - **multiple_clock** : レジスタ クロックピンに複数のクロックが供給されていないことをチェックします。レジスタ クロックピンに複数のクロックが供給されている場合、解析にどのクロックを使用するかが不明になります。その場合、**set_case_analysis** コマンドを使用してレジスタ クロックピンに 1 つのクロックのみが伝搬されるようにしてください。
 - **generated_clocks** : 生成クロック ネットワーク内でループや循環定義がないかを確認します。生成クロックが別の生成クロックをソースとして生成されており、その 2 番目の生成クロックが最初の生成くろっくをソースとして生成されている場合にエラーを返します。
 - **loops** : デザインに組み合わせフィードバック ループがないことをチェックします。
 - **partial_input_delay** : 入力遅延制約が部分的に定義されている入力ポートをレポートします。入力ポートに **set_input_delay -max** または **set_input_delay -min** のどちらかのみを設定すると、部分的に定義された入力遅延が作成されます。この場合、入力ポートからのパスが制約されず、ポートのタイミングがチェックされないことがあります。**set_input_delay** を **-min** または **-max** を指定せずに設定した場合、ツールで最小遅延と最大遅延の両方が予測され、部分的に定義された入力遅延は作成されません。
- 注記** : クロックが供給されていない入力ポートは、部分的に定義された入力遅延があるかどうかはチェックされません。
- **partial_output_delay** : 出力遅延制約が部分的に定義されている出力ポートをレポートします。出力ポートに **set_output_delay -max** または **set_output_delay -min** のどちらかのみを設定すると、部分的に定義された出力遅延が作成されます。この場合、ポートに到達するパスが制約されず、ポートのタイミングがチェックされないことがあります。**set_output_delay** を **-min** または **-max** を指定せずに設定した場合、ツールで最小遅延と最大遅延の両方が予測され、部分的に定義された出力遅延は作成されません。

注記 : クロックが供給されていない出力ポートは、部分的に定義された出力遅延があるかどうかはチェックされません。

 - **unexpandable_clocks** : クロック間にパスが 1 つ以上ある場合に、お互いに対して周期が拡張可能でないクロックのセットをレポートします。1000 サイクル以内にソース クロックとデスティネーション クロックの間に倍数が見つからない場合、拡張不可能です。
 - **latch_loops** : デザインに組み合わせラッチ ループがないことをチェックします。

引数

-file *arg* (オプション) : 結果をディスク上の指定のファイルに書き込みます。デフォルトでは、このコマンドの結果は Tcl コンソールに表示されます。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-name *arg* (オプション) : GUI の [Timing] ビューに表示する結果の名前を指定します。

-override_defaults {*args*} (オプション) : デフォルトのタイミング チェックではなく、指定したチェックを実行します。上記のチェックから実行するチェックを指定します。

-include *args* (オプション) : デフォルト チェックに加えて実行するチェックを実行します。上記のチェックから実行するチェックを指定します。

-exclude *args* (オプション) : **check_timing** コマンドで実行されるデフォルト チェックから指定のチェックを除外します。上記のチェックから除外するチェックを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、デフォルトのタイミング チェックから指定のチェックが除外されて **check_timing** が実行されます。

```
check_timing -exclude {loops generated_clocks}
```

次の例では、multiple_clocks チェックのみを実行し、**-verbose** オプションで詳細な結果を表示しています。その後、**get_clocks** を使用して問題を調べています。

```
check_timing -verbose -override_defaults {multiple_clock}
Checking multiple_clock.
There are 2 register/latch pins with multiple clocks.
procEngine/mode_du/set_reg[0]/C
provEngine/mode_du/set_reg[1]/C
get_clocks -of_objects [get_pin procEngine/mode_du/set_reg[0]/C]
sysClk coreClk
```

関連項目

- [create_clock](#)
- [get_clocks](#)
- [report_timing](#)
- [set_case_analysis](#)
- [set_input_delay](#)
- [set_max_delay](#)
- [set_output_delay](#)

checkpoint_vcd

VCD チェックポイントを作成します (Verilog \$dumpall システム タスクと同等)。

構文

```
checkpoint_vcd [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

シミュレーション

説明

現在の HDL オブジェクト信号値を VCD (Value Change Dump) ファイルに挿入します。戻り値はありません。この Tcl コマンドは Verilog \$dumpall システム タスクと同等で、指定の信号の初期値を示します。

VCD は、ヘッダー情報、変数定義、HDL 信号値の変更の詳細を含む ASCII 形式のファイルです。VCD ファイルを使用すると、VCD ビューアーでシミュレーション結果を表示したり、デザインの消費電力予測を実行したりできます。VCD ファイル フォーマットの詳細は、Verilog ハードウェア記述言語の IEEE 規格 (IEEE Std 1364-2005) を参照してください。

checkpoint_vcd コマンドを実行する前に、open_vcd および log_vcd コマンドを実行する必要があります。checkpoint_vcd コマンドを実行したら、シミュレーションを実行して信号値をキャプチャします。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、set_msg_config コマンドで定義できます。

例

次の例では、指定の HDL オブジェクトの信号値を開いている VCD ファイルに記述しています。

```
checkpoint_vcd
```

関連項目

- [flush_vcd](#)
- [log_vcd](#)
- [open_vcd](#)

close_bd_design

デザインを閉じます。

構文

```
close_bd_design [-quiet] [-verbose] name
```

戻り値

デザイン オブジェクト、エラーが発生した場合は ""

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	閉じるデザインの名前を指定します。

カテゴリ

IP インテグレーター

説明

Vivado Design Suite の IP インテグレーター機能の指定した IP サブシステムを閉じます。

デザインが変更されている場合でも、閉じる前にデザインを保存するかどうか尋ねるメッセージは表示されません。**close_bd_design** コマンドを使用する前に、**save_bd_design** コマンドを実行して変更を保存しておく必要があります。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name : 閉じる IP サブシステム デザイン オブジェクトの名前を指定します。

例

次の例では、現在のプロジェクトの現在の IP サブシステム デザインを閉じています。

```
close_bd_design [current_bd_design]
```

関連項目

- [create_bd_design](#)
- [current_bd_design](#)
- [get_bd_designs](#)
- [open_bd_design](#)
- [save_bd_design](#)

close_design

現在のデザインを閉じます。

構文

```
close_design [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

説明

現在アクティブなデザインを閉じます。デザインが変更されている場合でも、閉じる前にデザインを保存するかどうか尋ねるメッセージは表示されません。**close_design** コマンドを使用する前に、**save_design** または **save_design_as** コマンドを実行して変更を保存しておく必要があります。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のプロジェクトが閉じられます。

```
close_design
```

注記 : 複数のデザインが開いている場合は、**close_design** コマンドを実行する前に **current_design** コマンドで現在のデザインを指定できます。

次の例では、現在のデザインを指定してから閉じています。

```
current_design rtl_1
close_design
```

rtl_1 デザインが現在のデザインとして指定され、**close_design** コマンドで閉じられます。

関連項目

[current_design](#)

close_hw

ハードウェア ツールを閉じます。

構文

```
close_hw [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ハードウェア

close_hw_target

ハードウェア ターゲットを閉じます。

構文

```
close_hw_target [-quiet] [-verbose] [hw_target]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_target]	ハードウェア ターゲットを指定します。デフォルトは、現在のハードウェア ターゲットです。

カテゴリ

ハードウェア

close_project

現在開いているプロジェクトを閉じます。

構文

```
close_project [-delete] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-delete]	プロジェクトをディスクからも削除します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

説明

現在開いているプロジェクトを閉じます。

引数

-delete (オプション) : プロジェクトを閉じた後、ハード ディスクからプロジェクト データを削除します。

注記 : このオプションを使用する際には、注意が必要です。プロジェクト データの削除を確認するメッセージは表示されません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、アクティブ プロジェクトが閉じられます。

```
close_project
```

次の例では、現在のプロジェクトが閉じられます。複数のプロジェクトが開いている場合は、現在のプロジェクトに対してのみ **close_project** コマンドが実行されます。現在のプロジェクトは、**current_project** コマンドで指定できます。

次の例では、**project_1** を現在のプロジェクトとして指定してから閉じ、コンピューターのハード ディスクから削除します。

```
current_project project_1
close_project -delete
```

注記 : **-delete** オプションを使用する際には、注意が必要です。プロジェクト データの削除を確認するメッセージは表示されません。

関連項目

[current_project](#)

close_saif

SAIF トグル情報を SAIF 出力ファイルに保存し、ファイルを閉じます。

構文

```
close_saif [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

[シミュレーション](#)

説明

開いている SAIF ファイルを閉じます。

Vivado シミュレータでは、**open_saif** を使用して一度に開くことのできる SAIF ファイルは 1 つのみです。別の SAIF ファイルを開くには、現在開いている SAIF ファイルを閉じる必要があります。

このコマンドが正常に実行された場合は何も返されず、正常に実行できない場合はエラーが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次に例を示します。

```
close_saif
```

関連項目

- [log_saif](#)
- [open_saif](#)

close_sim

Vivado を終了せずに現在のシミュレーションをアップロードします。

構文

```
close_sim [-force] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-force]	変更が失われる場合でも、強制的にシミュレーションを閉じます。デフォルトでは、変更がある場合は波形コンフィギュレーションは閉じず、エラー メッセージが表示されます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

シミュレーション

説明

現在の Vivado シミュレーションを閉じます。

注記： このコマンドでは、サードパーティ シミュレータを閉じることはできません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記： コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記： メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のプロジェクトのシミュレーションを閉じています。

```
close_sim
```

関連項目

- [current_sim](#)
- [launch_xsim](#)

close_vcd

VCD 情報を VCD 出力ファイルに保存し、ファイルを閉じます。

構文

```
close_vcd [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

[シミュレーション](#)

説明

開いている VCD (Value Change Dump) ファイルを閉じます。

Vivado シミュレータでは、一度に開くことのできる VCD ファイルは 1 つのみです。別の VCD ファイルを開くには、現在開いている VCD ファイルを閉じる必要があります。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在の VCD オブジェクトを閉じています。

```
close_vcd
```

関連項目

[open_vcd](#)

close_wave_config

波形コンフィギュレーションを閉じます。

構文

```
close_wave_config [-force] [-quiet] [-verbose] [wcfgobj]
```

戻り値

なし

使用法

名前	説明
[-force]	変更が失われる場合でも、強制的に波形コンフィギュレーションを閉じます。デフォルトでは、変更がある場合は波形コンフィギュレーションは閉じず、エラー メッセージが表示されます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[wcfgobj]	指定の波形コンフィギュレーション オブジェクトを閉じ、破棄します。指定しない場合は、現在の波形コンフィギュレーションを閉じます。デフォルトは NULL です。

カテゴリ

波形

commit_hw_sio

現在のハードウェア オブジェクトのプロパティの変更を確定します。任意のハードウェア オブジェクトを入力できます。少なくとも 1 つのオブジェクトが必要です。

構文

```
commit_hw_sio [-quiet] [-verbose] hw_objects
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hw_objects</i>	ハードウェア オブジェクトを指定します。

カテゴリ

ハードウェア

commit_hw_vio

ハードウェア VIO プローブの OUTPUT_VALUE プロパティ値を VIO コアに書き込みます。

構文

```
commit_hw_vio [-quiet] [-verbose] hw_objects...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hw_objects</i>	ハードウェア VIO およびハードウェア プローブ オブジェクトのリストを指定します。

カテゴリ

ハードウェア

compile_simlib

シミュレーションライブラリをコンパイルします。

構文

```
compile_simlib [-cfg_file] [-directory arg] [-exclude_sublib]
[-exclude_superseded] [-family arg] [-force] [-language arg] [-library arg]
[-precompiled_directory arg] [-simulator arg] [-simulator_exec_path arg]
[-source_library_path arg] [-32bit] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-cfg_file]	コンフィギュレーション ファイルをデフォルト設定で生成します。デフォルトでは compile_simlib.cfg ファイルが生成されます。
[-directory]	コンパイルした結果を保存するディレクトリ パスを指定します。デフォルトは . です。
[-exclude_sublib]	EDK の .pao ファイルで定義されるサブライブラリをコンパイルで除外します (EDK ライブラリのみ)。
[-exclude_superseded]	使用されていない EDK のライブラリをコンパイルで除外します (EDK ライブラリのみ)。
[-family]	デバイス アーキテクチャを選択します。デフォルトは all です。
[-force]	コンパイル済みライブラリを上書きします。
[-language]	ライブラリをコンパイルする言語を指定します。デフォルトは all です。
[-library]	コンパイルするライブラリを選択します。デフォルトは all です。
[-precompiled_directory]	以前に compile_simlib でコンパイルしたライブラリが存在する既存ディレクトリを指定します。
[-simulator]	指定したシミュレータ用にライブラリをコンパイルします。
[-simulator_exec_path]	指定したディレクトリからシミュレータの実行ファイルを使用します。
[-source_library_path]	環境変数 XILINX_PLANAHEAD (Vivado 用) または XILINX_EDK (EDK 用) で指定されているデフォルト パスを検索する前に、指定したディレクトリでライブラリソース ファイルを検索します。
[-32bit]	32 ビットのコンパイルを実行します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

シミュレーション

説明

ザイリンクス シミュレーション ライブラリをコンパイルします。

ザイリンクスの HDL ベースのシミュレーション ライブラリは、Vivado シミュレータおよび ISim 用にあらかじめコンパイルされています。**compile_simlib** コマンドは、ほかのシミュレータ用にシミュレーション ライブラリをコンパイルする際に使用します。ライブラリは、シミュレーション モデルをアップデートし、新しいバージョンのシミュレータをサポートするため、ツール リリースごとにコンパイルし直す必要があります。

プロジェクトを開いた状態でこのコマンドを実行すると、下に説明するコマンドのデフォルト設定ではなく、そのプロジェクトで指定されたデバイス ファミリ、ターゲット言語、およびライブラリ設定がデフォルト値として使用されます。デフォルト設定は、コマンドの実行時にオプションを設定することで変更できます。

このコマンドを実行すると、コンパイルされたライブラリに関する情報が返されます。

引数

-cfg_file (オプション) : `compxlib.cfg` というコンフィギュレーション ファイルが現在のディレクトリに存在しない場合に作成します。

-directory arg (オプション) : コンパイルした結果を保存するディレクトリ パスを指定します。

注記 : デフォルトでは、ライブラリは現在の作業ディレクトリまたはツールを起動したディレクトリに保存されます。

-exclude_sublib (オプション) : EDK .pao ファイルで指定されているサブライブラリをコンパイルから除外します。このオプションは、EDK ライブラリにのみ使用できます。サブライブラリの詳細は、『エンベデッド システム ツール リファレンス マニュアル』(UG111) を参照してください。

-exclude_superseded (オプション) : 置き換えられた以前の EDK ライブラリをコンパイルから除外します。このオプションは、EDK ライブラリにのみ使用できます。置き換えられた以前のライブラリの詳細は、『エンベデッド システム ツール リファレンス マニュアル』(UG111) を参照してください。

-family arg (オプション) : 選択されたライブラリを指定のデバイス ファミリ用にコンパイルします。デフォルトでは、すべてのデバイス ファミリが生成されます。次のデバイス ファミリを指定できます。

- `virtex7` (Virtex®-7)
- `kintex7` (Kintex®-7)
- `kintex7l` (Kintex-7 低消費電力)
- `artix7` (Artix®-7)
- `artix7l` (Artix-7 低消費電力)
- `zynq` (Zynq®-7000 EPP)

-force (オプション) : コンパイル済みの現在のライブラリを上書きします。

-language [verilog | vhdl | all] (オプション) : ライブラリを指定の言語用にコンパイルします。このオプションを指定しない場合、**-simulator** で指定したシミュレータに基づく言語に設定されます。混合言語シミュレータの場合は、Verilog および VHDL ライブラリがコンパイルされます。

-library *arg* (オプション) : コンパイルするライブラリを指定します。有効な値は次のとおりです。

- all
- unisim
- simprim
- xilinxcorelib
- edk

複数のライブラリを指定するには、各ライブラリに -lib オプションを個別に使用します。次に例を示します。

```
... -library unisim -library simprim ...
```

注記 : EDK ライブラリを指定した場合 (-lib edk)、EDK ライブラリは UNISIM および SIMPRIM に依存しているので、ISE® ライブラリもすべてコンパイルされます。

-precompiled_directory *arg* (オプション) : コンパイル済みライブラリのディレクトリを指定します。

-simulator *arg* (オプション) : 指定したシミュレータ用にライブラリをコンパイルします。有効な値は、次のとおりです。

- modelsim
- questasim
- ies (Linux のみ)
- vcs_mx (Linux のみ)
- riviera
- Active-HDL

-simulator_exec_path *arg* (オプション) : シミュレータの実行ファイルのディレクトリを指定します。このオプションは、ターゲット シミュレータが \$PATH または %PATH% 環境変数で指定されていない場合や、\$PATH または %PATH% 環境変数で指定されているパスとは別のパスを指定する場合に使用します。

-source_library_path *arg* (オプション) : 環境変数 (\$XILINX、\$XILINX_PLANAHEAD、または \$XILINX_EDK) で指定されているデフォルト パスを検索する前に、指定したディレクトリでライブラリ ソース ファイルを検索します。

注記 : このオプションは、ザイリンクス テクニカル サポートから指示された場合以外は使用しないでください。

-32bit (オプション) : デフォルトの 64 ビットのコンパイルではなく、32 ビットのコンパイルを実行します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、Virtex-7 デバイスを使用するデザインを ModelSim (VHDL) でシミュレーションするために UNISIM および SIMPRIM ライブラリをコンパイルしています。

```
compile_simlib -simulator modelsim -family virtex7 -library unisim \
    -library simprim -language vhdl
```

関連項目

[launch_modelsim](#)

config_timing_analysis

タイミング解析の一般設定を指定します。

構文

```
config_timing_analysis [-enable_input_delay_default_clock arg]
[-enable_preset_clear_arcs arg] [-disable_flight_delays arg] [-quiet]
[-verbose]
```

戻り値

なし

使用法

名前	説明
[-enable_input_delay_default_clock]	内部定義されたクロックからの SDC のクロックなし入力遅延をイネーブルにします。有効な値は true、false です。このオプションは UCF 制約ではサポートされていません。
[-enable_preset_clear_arcs]	非同期プリセットまたはクリアタイミング アークを介するタイム パスをイネーブルにします。有効な値は true、false です。
[-disable_flight_delays]	I/O の計算にパッケージ遅延を追加しません。有効な値は true、false です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

タイミング

説明

タイミング解析の一般的な機能を設定します。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-enable_input_delay_default_clock [true | false] (オプション) : タイミング解析で内部定義されたクロックからのクロックなし入力遅延をイネーブルにします。有効な値は true または false で、デフォルト値は false です。

-enable_preset_clear_arcs [true | false] (オプション) : 非同期プリセットまたはクリアタイミング アークを介するタイム パスをイネーブルにします。有効な値は true または false で、デフォルト値は false です。

-disable_flight_delays [true | false] (オプション) : true の場合、I/O の計算にパッケージ遅延を追加しません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、タイミング解析中にパッケージ遅延を無視するよう指定しています。

```
config_timing_analysis -disable_flight_delays true
```

関連項目

- [config_timing_corners](#)
- [report_timing](#)

config_timing_corners

シングル/マルチ コーナーのタイミング解析を設定します。

構文

```
config_timing_corners [-corner arg] [-delay_type arg] [-setup] [-hold]
[-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-corner]	変更するタイミング コーナーの名前を指定します。設定可能な値は、Slow、Fast です。
[-delay_type]	指定したタイミング コーナーを解析するパス遅延のタイプを指定します。有効な値は、none、max、min、min_max です。
[-setup]	セットアップ解析のタイミング コーナーをイネーブルにします (-delay_type max と同じ)。
[-hold]	ホールド解析のタイミング コーナーをイネーブルにします (-delay_type min と同じ)。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

タイミング

説明

シングル/マルチ コーナー タイミング解析でのスローおよびファーストタイミング コーナーを設定します。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-corner [Slow | Fast] (オプション)：設定するタイミング コーナーの名前を指定します。有効な値は、Slow または Fast です。

注記：コーナー名では大文字/小文字が区別されます。

-delay_type value (オプション)：指定したタイミング コーナーを解析するパス遅延のタイプを指定します。有効な値は max、min、および min_max です。

-setup (オプション) : 指定のタイミングコーナーに対してセットアップ解析を指定します。これは **-delay_type max** と同じです。

-hold (オプション) : 指定のタイミングコーナーに対してホールド解析を指定します。これは **-delay_type min** と同じです。

注記 : **-setup** と **-hold** の両方を指定すると、**-delay_type min_max** と同じになります。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、セットアップ解析およびホールド解析の両方でスロー タイミングコーナーを設定します。

```
config_timing_corners -corner Slow -setup -hold
config_timing_corners -corner Slow -delay_type min_max
```

注記 : 上記のどちらでも、同じ結果が得られます。

次の例では、最小遅延解析でファースト タイミングコーナーを設定します。

```
config_timing_corners -corner Fast -delay_type min
```

関連項目

- [config_timing_analysis](#)
- [report_timing](#)

config_webtalk

ソフトウェア、IP、およびデバイスの使用統計をザイリンクスに送信する WebTalk をイネーブル/ディスエーブルにします。

構文

```
config_webtalk [-info] [-user arg] [-install arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-info]	現在 WebTalk がイネーブルかディスエーブルかを表示します。
[-user]	現在のユーザーに対して WebTalk をイネーブル/ディスエーブルにします。イネーブルにする場合は on、ディスエーブルにする場合は off に設定します。デフォルトでは設定されていません。
[-install]	現在のインストールのすべてのユーザーに対して WebTalk をイネーブル/ディスエーブルにします。イネーブルにする場合は on、ディスエーブルにする場合は off に設定します。off に設定した場合、個々のユーザーが -user オプションを使用して WebTalk をイネーブルにすることはできません。このオプションを使用するには、管理者権限が必要な場合があります。デフォルトでは設定されていません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ファイル入力および出力

説明

WebTalk はデザイン データを収集するためのザイリンクスツールの機能であり、ザイリンクス FPGA デバイス、ツール、および IP がどのように使用されているかをザイリンクスが理解するのに役立ちます。

このコマンドは、現在のユーザーおよびツール インストールにおける WebTalk 機能の現在の設定を返します。また、ツール、IP、およびデバイスの使用統計をザイリンクスに送信する WebTalk の機能をイネーブル/ディスエーブルにすることもできます。WebPACK ライセンスを使用してビットストリームを生成する場合以外は、WebTalk をディスエーブルにするとデータは送信されません。

WebTalk をイネーブルにするかどうかはユーザーの自由ですが、WebPACK を使用している場合は常にイネーブルになっており、常にデータが送信されます。WebPACK ライセンスを使用してビットストリームを生成する場合、ユーザーおよびインストールのプリファレンスは無視されます。

注記 : WebPACK に含まれるデバイスを使用していて WebPACK ライセンスが存在する場合、常に WebPACK ライセンスが使用されます。これを変更するには、アンサー 34746 を参照してください。

引数

-info (オプション) : 現在の WebTalk 設定を返します。WebTalk の設定は、ユーザーおよびインストールの両方の設定によります。どちらかの設定がディスエーブルになっていると、WebTalk はディスエーブルになります。

-user arg (オプション) : 現在のユーザーに対して WebTalk をイネーブルまたはディスエーブルにします。

-install arg (オプション) : 現在のツールインストールに対して WebTalk をイネーブルまたはディスエーブルにします。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、WebTalk の現在の設定が返されます。

```
config_webtalk -info
INFO: [Coretcl-120] Webtalk has been disabled by the current user.
INFO: [Coretcl-123] Webtalk has been enabled for the current installation.
INFO: [Coretcl-110] This combination of user/install settings means that WebTalk is currently disabled.
```

次の例では、現在のユーザーに対して WebTalk をイネーブルにしています。

```
config_webtalk -user on
```

connect_bd_intf_net

インターフェイス ポートとインターフェイス ピンのリストを接続します。

構文

```
connect_bd_intf_net [-intf_net arg] [-quiet] [-verbose] object1 object2
```

戻り値

正常に実行された場合は 0、それ以外の場合はエラー

使用法

名前	説明
[-intf_net]	すべてのオブジェクトを接続する 1 つのインターフェイス ネットを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
object1	接続するインターフェイス ポートまたはインターフェイス ピンの名前を指定します。
object2	接続するインターフェイス ポートまたはインターフェイス ピンの名前を指定します。

カテゴリ

IP インテグレーター

説明

IP インテグレーター セルのインターフェイス ピンをほかのインターフェイス ピンまたは外部インターフェイス ポートに接続します。インターフェイスとは、IP インテグレーター サブシステム デザインで共通の機能を共有する信号をグループ化したものです。

このコマンドは、**-intf_net** オプションで指定された名前のインターフェイス ネットを作成して、指定の名前の既存のインターフェイス ネットに接続するか、または名前が指定されていない場合は名前を割り当てます。

このコマンドを実行すると、接続されたインターフェイス ネット オブジェクトが返されるか、正常に実行されなかつた場合はエラーが返されます。

引数

-intf_net arg (オプション) : **create_bd_intf_net** で作成済みの既存のインターフェイス ネット名を指定するか、作成する新しいインターフェイス ネットを指定します。名前を指定しない場合、自動的にネットに名前が付けられます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

object1 (必須) : ネットを接続する最初のピンまたはポートを指定します。

object2 (必須) : ネットを接続する 2 番目のピンまたはポートを指定します。

例

次の例では、IP インテグレーター コアのインターフェイス ピンをサブシステム デザインのインターフェイス ポートに接続しています。

```
connect_bd_intf_net [get_bd_intf_pins clk_wiz_1/CLK_IN1_D] \
[get_bd_intf_ports /diff_clock_rtl]
```

関連項目

- [create_bd_cell](#)
- [create_bd_intf_net](#)

connect_bd_net

ポートとピン オブジェクトのリストを接続します。

構文

```
connect_bd_net [-net arg] [-quiet] [-verbose] objects...
```

戻り値

正常に実行された場合は 0、それ以外の場合はエラー

使用法

名前	説明
[-net]	すべてのオブジェクトを接続する 1 つのネットを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>objects</i>	ネットに接続するオブジェクトを指定します。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインにブロック ダイアグラム ポートおよびピン オブジェクトのリストを接続する新しいネットを作成するか、既存のネットを指定のピンおよびポートに接続します。

-net オプションを指定しない場合、指定したオブジェクトを接続する新しいネットが作成されます。**-net** を使用すると、指定のネットが必要に応じて接続または作成されます。

接続するポートおよびピン オブジェクトを指定するには、**get_bd_ports** および **get_bd_pins** コマンドを使用します。

このコマンドを使用すると、サブシステム デザインの異なる階層レベルにあるピンまたはポートを接続できますが、その場合は **-net** オプションは指定できません。**-net** を指定すると、接続が 1 つのネットではなく複数のネットになります。

このコマンドを実行すると、接続された IP インテグレーター サブシステム デザインのネット オブジェクトが返されるか、正常に実行されなかった場合はエラーが返されます。

引数

-net *arg* (オプション) : 現在の IP サブシステム デザインに 1 つのネットを作成します。

注記 : 接続するオブジェクトが同じ階層レベルにない場合は指定しないでください。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects : 接続する現在の IP インテグレーター サブシステム デザインのポートおよびピンのリストを指定します。

例

次の例では、IP サブシステム デザインの異なる階層レベルにある 2 つのピンを接続しています。

```
connect_bd_net [get_bd_pins /vidOut_1/locked] [get_bd_pins /newMod1/bridge_1/fid]
```

注記 : /vidOut_1/locked と /newMod1/bridge_1/fid はサブシステム デザインの異なる階層レベルにあるので、**-net** オプションは指定しません。この場合、階層をまたいで接続するために複数のネットが作成されます。

関連項目

- [create_bd_net](#)
- [disconnect_bd_net](#)
- [get_bd_pins](#)
- [get_bd_ports](#)

connect_debug_port

ネットとピンをデバッグ ポート チャネルに接続します。

構文

```
connect_debug_port [-channel_start_index arg] [-quiet] [-verbose] port nets...
```

戻り値

なし

使用法

名前	説明
<code>[-channel_start_index]</code>	チャネル インデックスからネットを接続します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>port</code>	デバッグ ポート名を指定します。
<code>nets</code>	ネットまたはピンのリストを指定します。

カテゴリ

説明

ネットリスト デザインからの信号を Vivado ラボ ツール デバッグ コアのポートに接続します。信号は、ポートの特定のチャネル インデックスに接続するか、ポートで使用可能なチャネルに接続できます。

ポートに接続する信号が多すぎたり、接続をサポートするだけのチャネルがない場合は、エラー メッセージが表示されます。

デバッグ コアにポートを追加するには `create_debug_port` コマンド、既存ポートで使用可能なチャネルを増加するには `set_property port_width` コマンドを使用します。例を参照してください。

ポートから信号の接続を解除するには、`disconnect_debug_port` コマンドを使用します。

デバッグ コアを定義して接続すると、デバッグ コアをブロックとしてインプリメントし、ネットリスト デザインに含めることができます。コアをインプリメントするには、`implement_debug_core` コマンドを使用します。

引数

`-channel_start_index arg` (オプション) : 接続に使用するチャネル インデックスを指定します。複数の信号を指定した場合、このオプションで指定したチャネル インデックス番号から接続が追加されます。チャネル インデックスの番号は 0 から開始します。

注記 : このオプションを指定しない場合、最初に使用可能なチャネル インデックスに接続されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

port (必須) : 信号を接続するポートの名前を指定します。ポートは、`core_name/port_name` で指定する必要があります。

nets (必須) : 指定したデバッグ ポートに接続する、ネットリスト デザインからのネット名のリストを指定します。

例

次の例では、myCore デバッグ コアに新しい PROBE ポートを作成し、ポートの PORT_WIDTH プロパティを増加して接続される信号数を受信できるようにし、信号を 3 番目のチャネル位置 (インデックス 2) からポートに接続します。

```
create_debug_port myCore PROBE
set_property PORT_WIDTH 8 [get_debug_ports myCore/PROBE1]
connect_debug_port myCore/PROBE1 [get_nets [list m0_ack_o m0_cyc_i m0_err_o \
m0_rty_o m0_stb_i m0_we_i ]] -channel_start_index 2
```

注記 : ポートで使用可能なチャネルに接続するネット数が多すぎると、エラー メッセージが表示され、ポートは接続されません。

関連項目

- [create_debug_port](#)
- [disconnect_debug_port](#)
- [get_debug_ports](#)
- [get_nets](#)
- [implement_debug_core](#)
- [set_property](#)

connect_hw_server

ハードウェア サーバーへの接続を開きます。

構文

```
connect_hw_server [-host arg] [-port arg] [-password arg] [-launch arg]
[-quiet] [-verbose]
```

戻り値

ハードウェア サーバー

使用法

名前	説明
[-host]	サーバーのホスト名を指定します。デフォルトは localhost です。
[-port]	サーバーのポート番号を指定します。デフォルトは 60001 です。
[-password]	サーバーのパスワードを指定します。デフォルトはなしです。
[-launch]	サーバー プロセスを起動します。デフォルトは No です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ハードウェア

connect_net

ネットをピンまたはポートに接続します。

構文

```
connect_net [-hier] [-basename arg] -net arg -objects args [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-hier]	階層接続を可能にし、必要に応じてネットおよびピンを作成します (-basename を参照)。
[-basename]	階層接続を実行する場合に必要なネット/ピンのベース名を指定します (-hier を参照)。デフォルト値は、接続されるネットの名前に基づきます (-net を参照)。
-net	接続するネットを指定します。
-objects	接続するピンまたはポートのリストを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ネットリスト

説明

指定のネットを、開いている合成済みデザインまたはインプリメント済みデザインのネットリストに含まれる 1 つ以上のピンまたはポートに接続します。

connect_net コマンドは、デザインの階層レベルをまたがるネットも接続します。この際、必要に応じてピンおよび階層ネットが追加されます。追加されたネットおよびピンには、識別しやすいようにカスタム ベース名を付けることもできますが、指定しない場合は Vivado ツールで自動的にベース名が割り当てられます。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソースファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザイン チェックポイントとして保存するか、**write_*** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリストファイルにエクスポートできます。

注記： エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

引数

-hier (オプション) : 階層の異なるレベルにあるネットを接続します。

注記 : **-hier** を指定しない場合は、階層ピンは接続されず、警告メッセージが表示されます。

-basename arg (オプション) : 階層レベルをまたがる指定のネットを接続するのに必要な階層ネットまたはピンに使用するカスタム名を指定します。このオプションを使用しない場合、自動的に接続されるネットに基づくベース名が付けられます。

-net arg (必須) : 接続するネットを指定します。

注記 : **create_net** コマンドの **-from** および **-to** オプションを使用してバスを作成できますが、バスの各ビットを **connect_net** コマンドを使用してそれぞれ接続する必要があります。

-objects args (必須) : ネットを接続するピンまたはポートのリストを指定します。ネットは 1 つ以上のピンまたはポートオブジェクトに接続できます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、ポートを作成し、myDMA インスタンスにピンを作成し、myEnable というネットを作成した後、そのネットを作成したポートおよびピンに接続しています。

```
create_port -direction IN enableIn
create_pin -direction IN myDMA/en
create_net myEnable
connect_net -net myEnable -objects {enableIn myDMA/en}
```

次の例では、32 ビットのバスポート、ピン、およびネットを作成し、loop コマンドを使用して各ビットを 1 つずつ接続します。

```
create_port -from 0 -to 31 -direction IN dataIN
create_pin -from 0 -to 31 -direction IN myDMA/data
create_net -from 0 -to 31 dataBus
for {set x 0} {$x<32} {incr x} {
    connect_net -net dataBus[$x] -objects {dataIN[$x] myDMA/data[$x] } }
```

注記 : dataBus を接続しようとすると、ネットが見つからないというエラーが発生します。バスの各ビットを個別に接続する必要があります。

関連項目

- [create_net](#)
- [create_pin](#)
- [create_port](#)
- [disconnect_net](#)
- [remove_net](#)
- [resize_net_bus](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

copy_bd_objs

オブジェクトのコピーを作成し、指定の階層セルに追加します。

構文

```
copy_bd_objs [-prefix arg] [-from_design arg] [-quiet]
[-verbose] parent_cell objects...
```

戻り値

正しく処理された場合は 0、エラーが発生した場合は ""

使用法

名前	説明
[-prefix]	セルに追加する接頭辞を指定します。
[-from_design]	元のオブジェクトを含むデザインを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
parent_cell	親セルを指定します。
objects	コピーするオブジェクトを指定します。

カテゴリ

IP インテグレーター

説明

開いているサブシステム デザインの IP インテグレーター オブジェクトを 2 番目のサブシステム デザインにコピーします。選択したオブジェクトは、現在のサブシステム デザインの最上位または既存の階層モジュール内にコピーできます。

`get_bd_cells` などのコマンドは現在のサブシステム デザインに対して実行されるので、下の例に示すように、コピーするオブジェクトを Tcl 変数に保存する必要があります。現在のサブシステム デザインをソース デザインに設定し、コピーするオブジェクトのグループを選択して、Tcl 変数に保存します。その後、`current_bd_design` を使用して現在のサブシステム デザインをターゲット デザインに変更し、選択したオブジェクトをコピーします。この場合、`-from_design` オプションを使用する必要があります。

このコマンドは、現在のサブシステム デザインのある階層レベルにあるオブジェクトを別の階層レベルにコピーするためにも使用できます。この場合は、`-from_design` オプションを使用する必要はありません。

このコマンドが正常に実行された場合は 0 が返され、正常に実行されなかった場合はエラーが返されます。

引数

`-prefix arg` (オプション) : 階層モジュールにコピーするセルに適用する接頭辞を指定します。

-from_design *arg* (オプション) : 指定するオブジェクトが存在する IP インテグレーター サブシステム デザインの名前を指定します。デザインは IP インテグレーターで開いておく必要があります。**-from_design** を指定しない場合、オブジェクトは **current_bd_design** コマンドで指定されている現在のデザインで検索されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

parent_cell (必須) : 指定のオブジェクトのコピー先となる階層モジュールの名前を指定します。現在のサブシステム デザインの最上位には / を指定できます。

objects (必須) : 指定の *parent_cell* にコピーするセルおよびネットのリストを **get_bd_cells** および **get_bd_nets** コマンドで指定します。

例

次の例では、現在のサブシステム デザインを設定し、オブジェクトのグループを選択して、Tcl 変数に保存した後、現在のサブシステム デザインを変更し、選択したオブジェクトを現在のデザインの最上位にコピーしています。

```
current_bd_design myDesign
myDesign
set copyObjs [get_bd_cells {vidOut_1 bridge_1}]
/vidOut_1 /bridge_1
current_bd_design design_1
design_1
copy_bd_objs -from_design myDesign / $copyObjs
0
```

注記 : **get_bd_cells** コマンドでは現在のサブシステム デザインからのセルのみが返されるので、**current_bd_design** でターゲット デザインを変更する前に、Tcl 変数にこれらのオブジェクトを保存します。

関連項目

- [current_bd_design](#)
- [get_bd_cells](#)
- [get_bd_nets](#)

copy_ip

既存の IP をコピーします。

構文

```
copy_ip -name arg [-dir arg] [-quiet] [-verbose] objects...
```

戻り値

プロジェクトに追加された IP ファイル オブジェクト

使用法

名前	説明
-name	コピーの IP の名前を指定します。
[-dir]	プロジェクト外で作成し、管理するリモート IP へのディレクトリ パスを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>objects</i>	コピーする IP を指定します。

カテゴリ

プロジェクト、IP フロー

説明

現在のプロジェクトにインスタンシエートされている IP コアのコピーを作成します。

引数

-name *arg* (必須) : 作成する新しい IP の名前を指定します。

-dir *arg* (オプション) : 新しく作成した IP を保存する、ローカル プロジェクト外のディレクトリへのパスを指定します。指定するディレクトリは、既に存在している必要があります。ディレクトリが存在しない場合は、エラーが返されます。

注記 : ディレクトリを指定しない場合、新しい IP は ローカル プロジェクト ディレクトリ構造の *project_name.srcs/sources_1/ip* に追加されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記: メッセージの制限は、**set_msg_config** コマンドで定義できます。

object (必須) : コピーする IP オブジェクトを指定します。**copy_ip** コマンドでは、一度に 1 つの IP コアのみコピーできます。IP は **get_ips** コマンドで指定する必要があります。名前で指定することはできません。

例

次の例では、現在のプロジェクトにインスタンシエートされている FIFO コアをコピーし、指定のディレクトリに保存しています。

```
copy_ip -name newFIFO -dir C:/Data/new_IP [get_ips oldFIFO]
```

関連項目

- [create_ip](#)
- [get_ips](#)
- [import_ip](#)
- [read_ip](#)

create_bd_addr_seg

新しいセグメントを作成します。

構文

```
create_bd_addr_seg -range arg -offset arg [-quiet] [-verbose]
[parent_addr_space] [slave_segment] name
```

戻り値

新しく作成されたセグメント オブジェクト、エラーが発生した場合は ""

使用法

名前	説明
-range	セグメントの範囲 (4096、4K、16M、1G など) を指定します。
-offset	セグメントのオフセット (0x00000000 など) を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[parent_addr_space]	セグメントの親アドレス空間を指定します。
[slave_segment]	作成したセグメントのスレーブ セグメントを指定します。
name	作成するセグメントの名前を指定します。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインに新しいアドレス セグメント オブジェクトを作成します。

このコマンドが正常に実行された場合は新しく作成されたセグメント オブジェクトが返され、正常に実行されなかった場合は何も返されません。

引数

-range arg (必須) : 作成するアドレス セグメントの範囲またはサイズを整数または 16 進数値で指定します。範囲は 2 のべき数で表されたビット数 (4096 など) または割り当てるメモリ量 (4K、16M、1G など) で指定します。

-offset arg (必須) : アドレス セグメントのオフセットを指定します。整数または 16 進数値 (0x00000000 など) で指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

parent_addr_space (必須) : セグメントの親アドレス空間を定義します。

slave_segment (必須) : 作成するマスター セグメントのスレーブ セグメントを定義します。

name (必須) : 作成するアドレス セグメントの名前を指定します。

例

次の例では、MicroBlaze コアのデータおよび命令アドレス空間のアドレス セグメントを作成しています。

```
create_bd_addr_seg -range 0x10000 -offset 0x41200000 \
[get_bd_addr_spaces microblaze_1/Data] \
[get_bd_addr_segs microblaze_1_axi_intc/s_axi/Reg] SEG1
create_bd_addr_seg -range 0x40000000 -offset 0x0 \
[get_bd_addr_spaces microblaze_1/Instruction] \
[get_bd_addr_segs microblaze_1_local_memory/ilmb_bram_if_cntlr/SLMB/Mem] SEG1
```

関連項目

- [get_bd_addr_segs](#)
- [get_bd_addr_spaces](#)

create_bd_cell

IP カタログから IP セルを追加するか、新規階層ブロックを作成します。

構文

```
create_bd_cell [-vlnv arg] [-type arg] [-quiet] [-verbose] name
```

戻り値

新しく作成されたセル オブジェクト、エラーが発生した場合はなし

使用法

名前	説明
[-vlnv]	IP カタログから追加する IP セルの Vendor:Library:Name:Version を指定します。
[-type]	作成するセルのタイプを指定します。有効な値は IP または hier で、デフォルトは IP です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	作成するセルの名前を指定します。

カテゴリ

IP インテグレーター

説明

IP インテグレーター カタログからセルを現在のサブシステム デザインに追加するか、サブシステム デザインに追加する新しい階層モジュールを作成します。

カタログから IP コアを追加する場合は、**-vlnv** オプションが必要です。

新しい階層モジュールを作成する場合は、**-type hier** オプションが必要です。

このコマンドが正常に実行された場合は新しく作成されたセル オブジェクトの名前が返され、正常に実行されなかった場合は何も返されません。

引数

-vlnv arg (オプション) : IP インテグレーター カタログから追加するセルの Vendor:Library:Name:Version 属性を指定します。VLAN 属性は、IP インテグレーター カタログ内のオブジェクトを識別します。新しい階層モジュールを作成する場合は、このオプションは必要ありません。

注記 : IP インテグレーター カタログからの IP の **-vlnv** プロパティは、Vivado Design Suite インストール ディレクトリの `data/ip/xilinx` にあるファイルを参照します。

-type [IP | hier] (オプション) : セルが IP であるか階層モジュールであるかを指定します。デフォルトでは、IP カタログから IP セルが作成されます。

注記 : **-vlnv** および **-type** はオプションですが、どちらかを指定する必要があります。IP インテグレーター カタログから追加する IP コアを指定する場合は **-vlnv**、新しい階層モジュールを作成する場合は **-type hier** を使用します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name : 現在の IP サブシステム デザインに追加する IP セルの名前を指定します。

例

次の例では、IP インテグレーター カタログから AXI FIFO コアを現在のサブシステム デザインに指定した名前で追加しています。

```
create_bd_cell -vlnv xilinx.com:ip:axi_fifo_mm_s:4.0 axi_fifo_1
```

注記 : **-vlnv** オプションで IP インテグレーター カタログから追加するコアを指定しています。

次の例では、新しい階層モジュール myModule1 を作成し、前の例の AXI FIFO を新しいモジュール内に移動します。その後、myModule1 をサブシステム デザインの現在のインスタンスとして設定し、新しいモジュール myModule2 を作成して、現在のインスタンスに追加します。最後に、現在のインスタンスをサブシステム デザインの最上位に戻します。

```
create_bd_cell -type hier myModule1
/myModule1
move_bd_cells /myModule1 [get_bd_cells /axi_fifo_1]
/myModule1
current_bd_instance /myModule1
/myModule1
create_bd_cell -type hier myModule2
/myModule1/myModule2
current_bd_instance
/
```

関連項目

- [copy_bd_objs](#)
- [current_bd_instance](#)
- [move_bd_cells](#)

create_bd_design

新規デザインと、同じ名前の最上位セルを作成します。

構文

```
create_bd_design [-quiet] [-verbose] name
```

戻り値

新しく作成されたデザイン オブジェクト、エラーが発生した場合は ""

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	作成するデザインの名前を指定します。

カテゴリ

IP インテグレーター

説明

現在のプロジェクトに追加したり、Vivado Design Suite の IP インテグレーター機能で使用する新規 IP サブシステム デザインを作成します。

空の IP サブシステム モジュールが作成され、現在のプロジェクトのソース ファイルに追加されます。サブシステム モジュールおよびファイルは、指定の *name* で、現在のプロジェクトの次の場所に作成されます。

project_name/project_name.srcs/sources_1/bd/name/name.bd

このコマンドが正しく実行されると、ファイル パスと IP サブシステム デザインの名前が返されます。コマンドが正しく実行されなかった場合はエラーが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name : 作成する IP サブシステム デザインの名前を指定します。

例

次の例では、design_1 という新しい空の IP サブシステム モジュールを作成して現在のプロジェクトに作成し、プロジェクトの sources ディレクトリに design_1.bd というファイルを作成しています。

```
create_bd_design design_1
```

関連項目

- [close_bd_design](#)
- [current_bd_design](#)
- [open_bd_design](#)
- [save_bd_design](#)

create_bd_intf_net

新規インターフェイス ネットを作成します。

構文

```
create_bd_intf_net [-quiet] [-verbose] name
```

戻り値

新しく作成されたインターフェイス ネット オブジェクト、エラーが発生した場合は ""

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	作成する intf_net の名前を指定します。

カテゴリ

IP インテグレーター

説明

IP サブシステム デザインの新しい IP インテグレーター インターフェイスを作成します。

このコマンドが正常に実行された場合は新しく作成されたインターフェイス ネット オブジェクトが返され、正常に実行されなかった場合は何も返されません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name : 作成するインターフェイス ネットの名前を指定します。

例

次の例では、現在のサブシステム デザインにインターフェイス ネットを作成しています。

```
create_bd_intf_net diff_clock_rtl
```

関連項目

[connect_bd_intf_net](#)

create_bd_intf_pin

新規インターフェイス ピンを作成します。

構文

```
create_bd_intf_pin -vlnv arg -mode arg [-quiet] [-verbose] name
```

戻り値

新しく作成されたインターフェイス ピン オブジェクト、エラーが発生した場合は “”

使用法

名前	説明
<code>-vlnv</code>	バスの VLNV を指定します。
<code>-mode</code>	バスインターフェイスのモードを指定します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>name</code>	作成するインターフェイス ピンの名前を指定します。

カテゴリ

IP インテグレーター

説明

IP インテグレーター階層モジュールに新しいインターフェイス ピンを作成します。IP インテグレーター インターフェイスとは、共通の機能を共有する信号をグループ化したもので、関連の機能を共有する個別の信号およびバスの両方を含めることができます。たとえば、AXI4-Lite マスターは、多数の個別の信号と複数のバスを含むインターフェイスです。

1 つの接続ピンまたは標準バス ピンを作成するには、`create_bd_pin` コマンドを使用します。

インターフェイス ピンは、互換性のあるほかのインターフェイス ピンまたはインターフェイス ポートに接続します。インターフェイス ピンは、階層モジュール内ではモジュールの外部に接続するためポートとして追加され、階層モジュール上ではピンとして追加されます。

`current_bd_instance` コマンドを使用して、階層モジュールを IP インテグレーター サブシステム デザインの現在のインスタンスとして指定する必要があります。`create_bd_intf_pin` コマンドは、現在のインスタンスに対して実行されます。

このコマンドが正常に実行された場合は新しく作成されたインターフェイス ピン オブジェクトの名前が返され、正常に実行されなかった場合はエラーが返されます。

引数

-vlnv arg (必須) : サブシステム デザインに追加するインターフェイス ピン オブジェクトの Vendor:Library:Name:Version (VLAN) 属性を指定します。VLAN 属性は、IP インテグレーター カタログ内のオブジェクトを識別します。

注記 : インターフェイス ピンおよびポートの **-vlnv** プロパティは、Vivado Design Suite インストール ディレクトリの `data/ip/interfaces` にあるファイルを参照します。たとえば、`-vlnv xilinx.com:interface:lmb_rtl:1.0` は Vivado Design Suite インストール ディレクトリの `data/ip/interfaces/lmb_v1_0` にあります。

-mode (必須) : インターフェイス ピンのモードを指定します。有効な値は Master、Slave、System、MirroredMaster、MirroredSlave、MirroredSystem、Monitor です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

name (必須) : 現在のインスタンスに追加するインターフェイス ピンの名前を指定します。

例

次の例では、階層モジュール newMod1 をIP インテグレーター サブシステム デザインの現在のインスタンスとして設定し、そのモジュールに新しいインターフェイス ピンを作成しています。

```
current_bd_instance [get_bd_cells /newMod1]
create_bd_intf_pin -mode Slave -vlnv xilinx.com:user:dma_rtl:1.0 data_in
```

関連項目

- [connect_bd_intf_net](#)
- [create_bd_intf_pin](#)
- [create_bd_port](#)
- [get_bd_intf_ports](#)

create_bd_intf_port

新規インターフェイス ポートを作成します。

構文

```
create_bd_intf_port -vlnv arg -mode arg [-quiet] [-verbose] name
```

戻り値

新しく作成されたインターフェイス ポート オブジェクト、エラーが発生した場合は ""

使用法

名前	説明
<code>-vlnv</code>	バスの VLNV を指定します。
<code>-mode</code>	バスインターフェイスのモードを指定します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>name</code>	作成するポートの名前を指定します。

カテゴリ

IP インテグレーター

説明

IP サブシステム デザインの新しいインターフェイス ポートを作成します。IP インテグレーター インターフェイスとは、共通の機能を共有する信号をグループ化したので、関連の機能を共有する個別の信号およびバスの両方を含めることができます。たとえば、AXI4-Lite マスターは、多数の個別の信号と複数のバスを含むインターフェイスです。

1 つの接続ポートまたは共通バス ポートを作成するには、`create_bd_port` コマンドを使用します。

このコマンドが正常に実行された場合は新しく作成されたインターフェイス ポート オブジェクトの名前が返され、正常に実行されなかった場合は何も返されません。

引数

`-vlnv arg` (必須) : サブシステム デザインに追加するインターフェイス ポート オブジェクトの Vendor:Library:Name:Version (VLNV) 属性を指定します。VLNV 属性は、IP インテグレーター カタログ内のオブジェクトを識別します。

注記 : インターフェイス ピンおよびポートの `-vlnv` プロパティは、Vivado Design Suite インストール ディレクトリの `data/ip/interfaces` にあるファイルを参照します。たとえば、`-vlnv xilinx.com:interface:lmb_rtl:1.0` は Vivado Design Suite インストール ディレクトリの `data/ip/interfaces/lmb_v1_0` にあります。

-mode (必須) : インターフェイス ピンのモードを指定します。有効な値は Master、Slave、System、MirroredMaster、MirroredSlave、MirroredSystem、Monitor です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : サブシステム デザインに追加するインターフェイス ポートの名前を指定します。

例

次の例では、新しい IP インテグレーター インターフェイス ポートを作成し、現在のサブシステム デザインに追加しています。

```
create_bd_intf_port -vlnv xilinx.com:interface:diff_clock_rtl:1.0 \
-mode Slave diff_clock_rtl
```

関連項目

- [connect_bd_intf_net](#)
- [create_bd_intf_pin](#)
- [create_bd_port](#)
- [get_bd_intf_ports](#)

create_bd_net

新規ネットを作成します。

構文

```
create_bd_net [-quiet] [-verbose] name
```

戻り値

新しく作成されたネット オブジェクト、エラーが発生した場合は ""

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	作成するネットの名前を指定します。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインに新しいネットを作成します。

このコマンドが正常に実行された場合は新しく作成されたネット オブジェクトが返され、正常に実行されなかった場合は何も返されません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

***name* (必須)** : 作成するネットの名前を指定します。ネット名は最上位から指定するか、名前のみ (net1) で指定するか、階層ネット名 (cell1/cellA/net1) を指定してデザイン階層内で指定できます。

例

次の例では、新しいネットを作成しています。

```
create_bd_net net1
```

関連項目

- [connect_bd_intf_net](#)
- [connect_bd_net](#)
- [create_bd_cell](#)
- [create_bd_intf_net](#)
- [create_bd_intf_pin](#)
- [create_bd_intf_port](#)
- [create_bd_pin](#)
- [create_bd_port](#)
- [current_bd_design](#)

create_bd_pin

新規ピンを作成します。

構文

```
create_bd_pin [-from arg] [-to arg] -dir arg [-type arg] [-quiet]
[-verbose] name
```

戻り値

新しく作成されたピン オブジェクト、エラーが発生した場合は ""

使用法

名前	説明
[-from]	開始インデックスを指定します。デフォルトは指定なしです。
[-to]	終了インデックスを指定します。デフォルトは指定なしです。
-dir	ピンの方向を指定します。
[-type]	ピンのタイプを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	作成するピンの名前を指定します。

カテゴリ

IP インテグレーター

説明

IP インテグレーター階層モジュールに追加する新しいピンを作成します。

このコマンドが正常に実行された場合は新しく作成されたピン オブジェクトの名前が返され、正常に実行されなかった場合はエラー メッセージが返されます。

引数

-from arg (オプション) : 標準バスピンの開始インデックスを指定します。1 ビットピンの場合は指定しません。
-to arg (オプション) : 標準バスピンの終了インデックスを指定します。1 ビットピンの場合は指定しません。
-dir [I | O | IO] (必須) : ピンの方向を指定します。入力ピンの場合は I、出力ピンの場合は O、双方向ピンの場合は IO に設定します。
-type arg (オプション) : ピンのタイプを指定します。クロックピンの場合は CLK、リセットピンの場合は RST、クロック イネーブルピンの場合は CE、割り込みピンの場合は INTR、データピンの場合は DATA に設定します。ピンのタイプを指定しない場合、未定義 (UNDEF) になります。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 作成するサブシステム ピンの名前を指定します。名前は、ピンを追加する階層モジュールを基準に参照されます (**/modName/pinname**)。

例

次の例では、現在の IP インテグレーター サブシステム デザインの指定したモジュールに新しい入力ピンを作成しています。

```
create_bd_pin -dir I -type rst /newMod1/rst
/newMod1/rst
```

関連項目

- [create_bd_cell](#)
- [create_bd_intf_pin](#)
- [create_bd_intf_port](#)
- [create_bd_port](#)

create_bd_port

IP サブシステム デザインの新しいポートを作成します。

構文

```
create_bd_port [-from arg] [-to arg] -dir arg [-type arg] [-quiet]
[-verbose] name
```

戻り値

新しく作成されたポート オブジェクト、エラーが発生した場合はなし

使用法

名前	説明
[-from]	開始インデックスを指定します。デフォルトは指定なしです。
[-to]	終了インデックスを指定します。デフォルトは指定なしです。
-dir	ポートの方向を指定します。有効な値は I、O、IO です。
[-type]	ポートのタイプを指定します。有効な値は clk、ce、rst、intr、data です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	作成するポートの名前を指定します。

カテゴリ

IP インテグレーター

説明

IP インテグレーター サブシステム デザインに追加する新しいポートを作成します。ポートは、サブシステム デザインの外部信号への接続です。

このコマンドが正常に実行された場合は新しく作成されたポート オブジェクトの名前が返され、正常に実行されなかった場合はエラー メッセージが返されます。

引数

- from arg (オプション) : 標準バス ポートの開始インデックスを指定します。1 ビット ポートの場合は指定しません。
- to arg (オプション) : 標準バス ポートの終了インデックスを指定します。1 ビット ポートの場合は指定しません。
- dir [I | O | IO] (必須) : ポートの方向を指定します。入力ポートの場合は I、出力ポートの場合は O、双方向ポートの場合は IO に設定します。

-type arg (オプション) : ポートのタイプを指定します。クロック ポートの場合は CLK、リセット ポートの場合は RST、クロック イネーブル ポートの場合は CE、割り込みポートの場合は INTR、データ ポートの場合は DATA に設定します。ポートのタイプを指定しない場合、未定義 (UNDEF) になります。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 作成するサブシステム ポートの名前を指定します。

例

次の例では、現在の IP インテグレーター サブシステム デザインに新しい双方向バス ポートを作成しています。

```
create_bd_port -from 0 -to 32 -dir IO -type data addr
/addr
```

関連項目

- [create_bd_cell](#)
- [create_bd_intf_pin](#)
- [create_bd_intf_port](#)
- [create_bd_port](#)

create_cell

現在のデザインにセルを作成します。

構文

```
create_cell -reference arg [-black_box] [-quiet] [-verbose] cells...
```

戻り値

なし

使用法

名前	説明
<code>-reference</code>	セルが参照するライブラリ セルまたはデザインを指定します。
<code>[-black_box]</code>	ブラック ボックス インスタンスを作成します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>cells</code>	作成するセルの名前を指定します。

カテゴリ

ネットリスト

説明

現在の合成済みデザインまたはインプリメント済みデザインのネットリストにセルを追加します。

新しいセル インスタンスは、デザインの最上位、またはデザインのモジュール内の階層に追加できます。インスタンスは、ライブラリまたはデザイン ソース ファイルから既存のセルを参照するか、作成されていないセルを参照するブラック ボックス インスタンスを追加することもできます。

注記：セルをライブラリ マクロまたはマクロ プリミティブに追加することはできません。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソース ファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、`write_checkpoint` コマンドを使用してデザイン チェックポイントとして保存するか、`write_*` コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリスト ファイルにエクスポートできます。

注記：エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

このコマンドを実行すると、作成したセル インスタンスの名前が返されます。

引数

`-reference arg` (必須)：新しく作成するセル インスタンスが参照するライブラリ セルまたはソース ファイル モジュールを指定します。

-black_box (オプション) : 指定した参照セルのブラックボックスインスタンスを定義します。このオプションは、トップダウン設計手法において、参照セルがまだ存在しておらず、セルのブラックボックスインスタンスを作成する場合に使用します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

cells (必須) : 作成するセルのインスタンス名を指定します。インスタンス名は、デザインの最上位からの階層名で指定できます。この場合、階層インスタンス名に階層区切り文字を使用する必要があります。現在の階層区切り文字を確認するには、**get_hierarchy_separator** コマンドを使用します。

例

次の例では、or1200_cpu モジュールの 3 つのセルインスタンスを指定の名前で作成しています。

```
create_cell -reference or1200_cpu myCell1 myCell2 myCell3
```

次の例では、まず階層区切り文字を設定し、階層インスタンス名を指定して参照セルのブラックボックスインスタンスを作成しています。

```
set_hierarchy_separator |
create_cell -reference dmaBlock -black_box usbEngine0|myDMA
```

注記 : **-black_box** を使用しているときに **-reference** セルが既に存在していると、エラーが返されます。

関連項目

- [remove_cell](#)
- [set_hierarchy_separator](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

create_clock

クロック オブジェクトを作成します。

構文

```
create_clock -period arg [-name arg] [-waveform args] [-add] [-quiet]
[-verbose] [objects]
```

戻り値

新しいクロック オブジェクト

使用法

名前	説明
-period	クロック周期を指定します。0 より大きい値を設定します。
[-name]	クロック名を指定します。
[-waveform]	クロック エッジを指定します。
[-add]	ソース オブジェクトの既存クロックに追加します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	クロック ソース ポート、ピン、またはネットのリストを指定します。

カテゴリ

SDC、XDC

説明

指定した周期または波形でクロック オブジェクトを作成します。このコマンドは、タイミング エンジンでクロック エッジの遅延伝搬の開始点として使用されるプライマリ クロックを定義します。定義したクロックは、既存のクロックの定義に追加したり、既存のクロックを上書きできます。

デザイン内にソースがない仮想クロックも作成できます。仮想クロックは、入力および出力遅延を設定する際の時間の基準として使用できますが、デザインに物理的には存在しません。

クロックは、既存の物理クロックから、そのマスター クロックのプロパティの多くを継承して生成することも可能です。既存の物理クロックからクロックを生成するには、**create_generated_clock** コマンドを使用します。

注記：このコマンドでは、作成したクロック オブジェクトの名前が返されます。

引数

-period *arg* (必須) : 作成するクロック オブジェクトのクロック周期を指定します。0 より大きい値を設定する必要があります。デフォルト値は 10.0 です。

注記 : 時間の単位は、**set_units** コマンドで指定します。デフォルトの時間の単位はナノ秒 (ns) で、精度は 1 ピコ秒 (ps) です。

-name *arg* (オプション) : 作成するクロック オブジェクトの名前を指定します。名前を指定しない場合は、指定されたソース オブジェクト (*objects*) に基づいてシステムにより名前が生成されます。ソース オブジェクト *objects* を指定せずに **-name** オプションを使用すると、デザインの物理的なソースに関連付けられていない仮想クロックを作成できます。

-waveform *arg1 arg2 ...* (オプション) : 作成するクロックの、1 クロック サイクル内での立ち上がりエッジと立ち下がりエッジの時間をナノ秒で指定します。波形の立ち上がりエッジと立ち下がりエッジの両方を表すエッジを、偶数個指定する必要があります。最初に指定した時間が最初の立ち上がり遷移、2 番目に指定した時間が立ち下がりエッジとなります。立ち下がりエッジの値が立ち上がりエッジの値より小さい場合、立ち下がりエッジが立ち上がりエッジの前に発生することを意味します。

注記 : 波形を指定しない場合、デフォルトの波形は立ち上がりエッジが時間 0.0、立ち下がりエッジが指定した周期の 1/2 (**-period/2**) で発生するよう設定されます。

-add (オプション) : 異なるクロック波形で同時解析を実行する場合に、同じソースに複数のクロックを定義します。追加するクロックを **-name** を使用して指定する必要があります。このオプションを使用しないと、自動的に名前が割り当てられ、同じ名前のクロックが存在する場合に上書きされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (オプション) : 指定したクロックのソースとなるポート、ピン、またはネットを指定します。既にクロックが指定されているソース オブジェクトにクロックを指定した場合、**-add** オプションを使用しなければ元のクロックが新しいクロックで上書きされます。クロック オブジェクトを接続する *objects* を指定しない場合、クロックは仮想クロックとして作成されます。

注記 : 指定したネットの最初のドライバーがクロックのソースとして使用されます。

例

次の例では、bftClk という物理クロックを作成し、クロック周期を設定しています。

```
create_clock -name bftClk -period 5.000 [get_ports bftClk]
```

注記 : この例でオブジェクトを定義する **get_ports** コマンドを使用しない場合、物理クロックではなく仮想クロックが作成されます。

次の例では、入力ポート clk_pin_p に周期が 10ns、立ち上がりエッジが 2.4ns、立ち下がりエッジが 7.4ns の clk というクロックを作成しています。

```
create_clock -name clk -period 10.000 -waveform {2.4 7.4} [get_ports clk_pin_p]
```

次の例では、クロックソースを指定していないので、仮想クロックが作成されます。

```
create_clock -name virtual_clock -period 5.000
```

次の例では、立ち下がりエッジが 2ns、立ち上がりエッジが 7ns のクロックを作成しています。

```
create_clock -name clk -period 10.000 -waveform {7 2} [get_ports clk_pin_p]
```

関連項目

- [all_clocks](#)
- [create_generated_clock](#)
- [get_clocks](#)
- [report_clocks](#)
- [report_clock_interaction](#)
- [report_clock_networks](#)
- [report_clock_utilization](#)
- [set_clock_groups](#)
- [set_clock_latency](#)
- [set_clock_uncertainty](#)
- [set_input_delay](#)
- [set_output_delay](#)
- [set_propagated_clock](#)
- [set_units](#)

create_debug_core

新しい Integrated Logic Analyzer デバッグ コアを作成します。

構文

```
create_debug_core [-quiet] [-verbose] name type
```

戻り値

新しいデバッグ コア オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	新しいデバッグ コア インスタンスの名前を指定します。
<i>type</i>	新しいデバッグ コア インスタンスのタイプを指定します。

カテゴリ

説明

新しい Integrated Logic Analyzer (ILA) デバッグ コア labtools_ila_v2 を作成し、現在のプロジェクトのネットリスト デザインに追加します。ILA デバッグ コアは、デバッグ目的でネットを接続するポートを定義します。Vivado ラボ ツール デバッグ コアの使用方法の詳細は、『Vivado Design Suite ユーザー ガイド：プログラムおよびデバッグ』(UG908) を参照してください。

注記： デバッグ コアは、ツールで開いているネットリスト デザインにのみ追加できます。

デフォルトでは、ILA コアは CLK ポートおよび PROBE ポートと共に作成されます。CLK ポートはILA コアのクロック ドメインを定義し、そのドメインに共通の信号をプローブできます。CLK ポートでサポートされるクロック信号は 1 つだけなので、クロック ドメインごとに別のデバッグ コアを作成する必要があります。PROBE ポートは、MARK_DEBUG プロパティでデバッグ用にマークされたネットのプローブ ポイントを提供します。PROBE ポートには複数のチャネルがあり、1 つのILA コアから複数のネットをプローブできます。

create_debug_port コマンドを使用して既存のILA コアに新しいポートを追加し、**connect_debug_port** コマンドを使用してそのポートに信号を接続できます。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記： コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : プロジェクトに追加する ILA デバッグ コアの名前を指定します。

type (必須) : 挿入するデバッグ コアのタイプを指定します。Vivado ツールで現在サポートされているのは labtools_ila_v2 デバッグ コアのみです。ILA デバッグ コアは接続されたネットに別のロードを追加するだけで、その他の変更はありません。

注記 : ILA コアをプロジェクトに追加すると、1 つまたは複数の ILA コアのコンテナーとして Debug Hub コア (labtools_xsdbmasterlib_v2) が 1 つ追加されます。ただし、Debug Hub コアをプロジェクトに直接追加することはできません。

例

次の例では、ネットリスト デザインを開き、新しい ILA デバッグ コアを作成します。

```
open_netlist_design -name netlist_1
create_debug_core myCore labtools_ila_v2
```

デバッグ コアのプロパティは、次の例のように **set_property** コマンドを使用するとカスタマイズできます。

```
set_property C_DATA_DEPTH 2048 [get_debug_cores myCore]
```

次の例では、MARK_DEBUG プロパティを使用してネットのシーケンスをデバッグするようマークし、新しいデバッグ コアを作成して CLK ポートを適切なクロックドメインに接続し、デバッグ ネットをコア上の PROBE ポートに割り当てます。

```
set_property MARK_DEBUG true [get_nets [list {control_reg[0]} {control_reg[1]} \
{control_reg[2]} {control_reg[3]} {control_reg[4]} {control_reg[5]} \
{control_reg[6]} {control_reg[7]} {control_reg[8]} {control_reg[9]} \
{control_reg[10]} {control_reg[11]} {control_reg[12]} {control_reg[13]} \
{control_reg[14]} {control_reg[15]} {control_reg[16]} {control_reg[17]} \
{control_reg[18]} {control_reg[19]} {control_reg[20]} {control_reg[21]} \
{control_reg[22]} {control_reg[23]} {control_reg[24]} {control_reg[25]} \
{control_reg[26]} {control_reg[27]} {control_reg[28]} {control_reg[29]} \
{control_reg[30]} {control_reg[31]}]]
create_debug_core u_il_0 labtools_ila_v2
set_property port_width 1 [get_debug_ports u_il_0/CLK]
connect_debug_port u_il_0/CLK [get_nets [list wbClk ]]
set_property port_width 32 [get_debug_ports u_il_0/PROBE0]
connect_debug_port u_il_0/PROBE0 [get_nets [list {control_reg[0]} {control_reg[1]}\
{control_reg[2]} {control_reg[3]} {control_reg[4]} {control_reg[5]} \
{control_reg[6]} {control_reg[7]} {control_reg[8]} {control_reg[9]} \
{control_reg[10]} {control_reg[11]} {control_reg[12]} {control_reg[13]} \
{control_reg[14]} {control_reg[15]} {control_reg[16]} {control_reg[17]} \
{control_reg[18]} {control_reg[19]} {control_reg[20]} {control_reg[21]} \
{control_reg[22]} {control_reg[23]} {control_reg[24]} {control_reg[25]} \
{control_reg[26]} {control_reg[27]} {control_reg[28]} {control_reg[29]} \
{control_reg[30]} {control_reg[31]} ]]
```

関連項目

- [connect_debug_port](#)
- [create_debug_port](#)
- [delete_debug_core](#)
- [get_debug_cores](#)
- [implement_debug_core](#)
- [report_debug_core](#)
- [report_property](#)
- [set_property](#)

create_debug_port

新規デバッグ ポートを作成します。

構文

```
create_debug_port [-quiet] [-verbose] name type
```

戻り値

新しいデバッグ ポート オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	デバッグ コア インスタンスの名前を指定します。
<i>type</i>	デバッグ ポートのタイプを指定します。

カテゴリ

説明

既存の Vivado ラボ ツール デバッグ コアに追加する新しいポートを定義します。このポートは ILA コアへの接続 ポイントで、デバッグ目的でデザインのネットに接続するために使用されます。

create_debug_core コマンドを使用して新しいデバッグ コアを作成すると、デフォルトで CLK と PROBE ポートが含まれますが、トリガー入力/出力ポートタイプも追加できます。ポートのタイプと目的の詳細は、『Vivado Design Suite ユーザー ガイド：プログラムおよびデバッグ』(UG908) を参照してください。

ポートに接続ポイントを複数設定し、デバッグ用に複数のネットをサポートできます。デフォルトでは、新しいポートの幅は 1 で定義されるので、接続できるネットは 1 つのみです。PROBE ポートの幅を変更して複数の信号をサポートできるようにするには、**set_property port_width** コマンドを使用します (例を参照)。

注記 : CLK 、TRIG_IN、TRIG_IN_ACK、TRIG_OUT、および TRIG_OUT_ACK ポートの幅は 1 にしか設定できません。

信号をポートに接続するには **connect_debug_port** コマンド、接続を解除するには **disconnect_debug_port** コマンドを使用します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記: メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 新しいポートを接続する ILA デバッグ コアの名前を指定します。デバッグ コアは、**create_debug_core** で既に作成され、存在している必要があります。

type (必須) : 挿入するデバッグ ポートのタイプを指定します。サポートされるポートタイプは、次のとおりです。

- CLK : ILA デバッグ コアをクロックドメインに接続するクロック ポートを定義します。各デバッグ コアには CLK ポートを 1 つしか含めることができず、各 CLK ポートは 1 つのクロックドメインにしか接続できないので、異なるクロックドメインからの信号をプローブするには複数の ILA コアを使用する必要があります。
- PROBE : MARK_DEBUG プロパティでデバッグ用にマークされた信号に接続するためのプローブ ポイントを提供します。ILA デバッグ コアには複数の PROBE ポートを含めることができます、コアにポートを追加したときに Vivado ツールにより自動的に番号が付けられます。各 PROBE ポートには 1 つまたは複数のチャネル (接続ポイント) を含めることができます、PORT_WIDTH プロパティで定義します。
- TRIG_IN/TRIG_IN_ACK および TRIG_OUT/TRIG_OUT_ACK : ILA コアの PROBE 入力における特定の比較条件を検出するために使用される ILA プローブ トリガー コンパレーターです。TRIG_IN と TRIG_IN_ACK、TRIG_OUT と TRIG_OUT_ACK は、ポートペアとしてデバッグ コアに追加する必要があります。詳細は、『Vivado Design Suite ユーザー ガイド：プログラムおよびデバッグ』(UG908) を参照してください。

例

次の例では、新しいデバッグ コアを作成して PROBE ポートを追加し、ポート幅を 8 に設定して、その PROBE ポートに信号を接続しています。

```
create_debug_core myCore labtools_ila_v2
create_debug_port myCore PROBE
set_property PORT_WIDTH 8 myCore/PROBE1
connect_debug_port -channel_start_index 1 myCore/PROBE1 \
{m1_cyc_i m1_ack_o m1_err_o m1_rty_o}
```

注記: ILA コアにはデフォルトで CLK ポートおよび PROBE ポートが含まれるので、新しく追加した PROBE ポートには番号が付けられます (PROBE1)。

関連項目

- [connect_debug_port](#)
- [create_debug_core](#)
- [disconnect_debug_port](#)
- [set_property](#)

create_drc_check

ユーザー定義の DRC ルールを作成します。

構文

```
create_drc_check [-hiername arg] -name arg [-desc arg] [-msg arg]
                  -rule_body arg [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-hiername]	ルールの階層名を指定します。DRC UI パネルを作成した場合、この名前がメニュー階層にルールを配置するために使用されます。メニュー階層を分離するには、ピリオド(.) を使用します。これはオプションであり、デフォルトは User Defined です。
-name	DRC ルールの名前を指定します。PREFIX-id という形式で指定します (PREFIX は 4 ~ 6 文字の略称、id は特定のルールを識別する整数)。類似したルールには同じ略称を使用し、固有の id で識別するようにします。
[-desc]	DRC ルールの簡単な説明を指定します。これはオプションであり、デフォルトは User rule - default description です。
[-msg]	DRC ルールの詳細な説明を指定します。置換キーを含めることができます。使用可能な置換キーは、%MSG_STRING、%NETLIST_ELEMENT、%SITE_GROUP、%CLOCK_REGION、%BANK です。
-rule_body	ルールの本体を表す文字列を指定します。評価する Tcl プロシージャ名または Tcl コードの文字列を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

DRC、オブジェクト

説明

report_drc コマンドを実行したときに使用するユーザー定義の DRC ルール チェック (drc_check) を作成します。

ユーザー定義ルール チェックには固有の名前または略称を指定でき、オプションで複数のルールを階層にグループ化したり、ルールの説明や違反が発生したときの汎用プレースホルダー メッセージを指定したり、report_drc コマンドで実行するデザインルール チェックに関連付ける Tcl コードを参照できます。

このコマンドで指定された汎用プレースホルダー メッセージは、Tcl チェッカー プロシージャおよび **create_drcViolation** コマンドで検出されたデザイン オブジェクトと違反に関連する特定の情報に置き換えられます。

プロセスは次のとおりです。

- ユーザー定義ルールをチェックする際に適用するメソッドと、ルールでチェックするオブジェクトを定義する Tcl チェッカー プロシージャを記述します。Tcl チェッカー プロシージャは別の Tcl スクリプトで定義し、**report_drc** を実行する前に **source** コマンドで読み込む必要があります。
- Tcl チェッカーの **create_drcViolation** を使用して、デザインでルールをチェックした際に検出される違反を特定します。
- create_drcCheck** コマンドを使用して、**-rule_body** で指定した Tcl チェッカー プロシージャを呼び出すユーザー定義 DRC ルール チェックを定義します。
- create_drcRuleDeck** コマンドを使用してルール デックを作成し、**add_drcChecks** コマンドを使用してユーザー定義ルール チェックをルール デックに追加します。
- ルール デックまたはユーザー定義ルール チェックを指定して **report_drc** を実行し、違反がないかどうかをチェックします。

指定の名前の **drcCheck** が既に存在する場合は、エラーが返されます。既存の **drcCheck** を変更または上書きする場合は、**delete_drcCheck** コマンドを使用してまずチェックを削除してください。

DRC ルール チェック オブジェクトには **is_enabled** プロパティがあり、**set_property** コマンドを使用して TRUE または FALSE に設定できます。新しいルール チェックを作成すると、**is_enabled** プロパティはデフォルトで TRUE に設定されます。**report_drc** を実行したときにルール チェックが使用されないようにするには、**is_enabled** プロパティを FALSE に設定します。これにより、新しい DRC チェックを作成し、**add_drcChecks** を使用してルール デックに追加した場合に、そのチェックをルール デックから削除せずにイネーブルにしたりディスエーブルにしたりできます。

各ユーザー定義 DRC ルール チェックには **USER_DEFINED** プロパティが設定されており、ユーザー定義ルール チェックを簡単に特定および選択できます。

引数

-hiername arg (オプション) : 新しいルールのグループを指定します。デフォルトは User Defined です。このオプションで指定した値は、GUI での DRC ルールのリストにおける第一階層となります。新しく作成されたすべての DRC チェックは、all 階層にも追加され、**report_drc** コマンドによりデフォルトで使用されます。

-name arg (必須) : デザイン ルールの名前を指定します。これは、**-rule_body** で指定された Tcl チェッカー プロシージャの **create_drcViolation** コマンドで使用された名前と一致している必要があります。この名前が DRC レポートに関連の違反と共に表示されます。名前は、4 ~ 6 文字のルール グループの略称に、同じグループのルールを識別する ID をつけたものにする必要があります (ABCD-1, ABCD-23 など)。

-desc arg (オプション) : ルールの簡単な説明を指定します。デフォルトは User Rule です。この説明は、GUI で DRC ルールをリストしたときに表示されます。DRC レポートおよびサマリでも使用されます。

-msg arg (オプション) : ルールの違反が検出されたときに表示するメッセージを指定します。メッセージには、ルールの違反で検出されたデザイン エレメントで動的に置換するプレースホルダーを含めることができます。実際のデザイン データは、**report_drc** を実行したときにメッセージに挿入されます。各置換キーには、長い形と短縮形があります。有効な置換キーは次のとおりです。

- %MSG_STRING (%STR) : **create_drcViolation** コマンドで **-msg** オプションを使用して違反用に定義したメッセージ文字列。

注記 : **create_drcCheck** コマンドで **-msg** を指定しない場合、%STR がデフォルト メッセージとなり、**create_drcViolation** の **-rule_body** で定義されたメッセージが DRC レポートに渡されます。

- %NETLIST_ELEMENT (%ELG) : セル、ピン、ポート、ネットなどのネットリスト エレメント
- %SITE_GROUP (%SIG) : デバイス サイト
- %CLOCK_REGION (%CRG) : クロック領域
- %BANK (%PBG) : パッケージ I/O バンク

-rule_body arg (必須) : ルール チェック機能を定義する Tcl プロシージャの名前を指定します。Tcl プロシージャを **-rule_body** オプションに含めるか、別の Tcl スクリプトで定義して、ツールを起動したときまたは **report_drc** コマンドを実行する前に **source** コマンドで読み込みます。

Tcl チェッカー プロシージャでは、**create_drcViolation** コマンドを使用して、デザイン ルール違反に関連するデザイン エレメントを含む DRC 違反オブジェクトを作成できます。**-msg** で定義されたメッセージの置換キーは、違反オブジェクトからのデザイン エレメントで置換されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、RAMW-1 という名前のデザイン ルール チェックを定義しています。階層名および説明を定義し、デフォルトの重要度である警告を使用し、チェックを実行したときに **dataWidthCheck** プロシージャが呼び出されるようにします。

```
create_drcCheck -name {RAMW-1} -hiername {RAMB} \
    -desc {Data Width Check} -rule_body dataWidthCheck
```

次の Tcl スクリプトは **dataWidthCheck** プロシージャを定義しています。このプロシージャは、RAMW-1 チェックの **-rule_body** オプションで呼び出されます。この Tcl スクリプト ファイルは、**report_drc** コマンドを実行する前に **source** コマンドを使用して読み込む必要があります。

```
# This is a simplistic check -- report BRAM cells with WRITE_WIDTH_B wider than 36.
proc dataWidthCheck {} {
    # list to hold violations
    set vios {}
    # iterate through the objects to be checked
    foreach bram [get_cells -hier -filter {PRIMITIVE_SUBGROUP == bram}] {
        set bwidth [get_property WRITE_WIDTH_B $bram]
        if { $bwidth > 36 } {
            # define the message to report when violations are found
            set msg "On cell %ELG, WRITE_WIDTH_B is $bwidth"
            set vio [ create_drcViolation -name {RAMW-1} -msg $msg $bram ]
            lappend vios $vio
        }
    }
    if {[llength $vios] > 0} {
        return -code error $vios
    } else {
        return {}
    }
}
create_drc_check -name {RAMW-1} \
    -hiername {RAMB Checks} \
    -desc {Data Width Check} \
    -rule_body dataWidthCheck
```

注記：スクリプト ファイルには、Tcl チェッカー プロシージャと、**report_drc** コマンドで使用するためにそれを定義する **create_drc_check** コマンドの両方を含めることができます。この場合、Tcl スクリプト ファイルを読み込んだときに **dataWidthCheck** プロシージャと RAMW-1 デザイン ルール チェックの両方が読み込まれます。

関連項目

- [add_drc_checks](#)
- [create_drc_ruledeck](#)
- [create_drcViolation](#)
- [delete_drc_check](#)
- [get_drc_checks](#)
- [get_drc_vios](#)
- [report_drc](#)

create_drc_ruleddeck

1 つまたは複数のユーザー定義 DRC ルール デック オブジェクトを作成します。

構文

```
create_drc_ruleddeck [-quiet] [-verbose] ruledecks ...
```

戻り値

DRC ルール デック

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
ruledecks	作成するルール デックの名前を指定します。

カテゴリ

DRC、オブジェクト

説明

report_drc コマンドを実行したときに使用する 1 つまたは複数のユーザー定義ルール デックを作成します。

DRC ルール デックはデザインルール チェックのグループで、I/O 配置や配置などの FPGA デザインフローの異なる段階で report_drc コマンドにより実行されます。ツールには、定義済みのルール デックが含まれています。get_drc_ruledecks コマンドを使用すると、定義済みのルール デックを取得できます。

このコマンドで作成したルール デックは空で、チェックは含まれません。add_drc_checks コマンドを使用して、ルール デックにルール チェックを追加する必要があります。ルール デックからチェックを削除するには、remove_drc_checks コマンドを使用します。ルール デックに含めることができるデザイン ルール チェックのリストを取得するには、get_drc_checks コマンドを使用します。

このコマンドを実行すると、作成された DRC ルール チェックのリストが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、set_msg_config コマンドで定義できます。

ruledecks (必須) : 作成する 1 つまたは複数のユーザー定義 DRC ルール デックの名前を指定します。

例

次の例では、2 つの DRC ルール デック オブジェクトを作成しています。

```
create_drc_ruledeck my_rules project_rules
```

関連項目

- [add_drc_checks](#)
- [delete_drc_ruledeck](#)
- [get_drc_checks](#)
- [get_drc_ruledecks](#)
- [remove_drc_checks](#)
- [report_drc](#)

create_drcViolation

DRC 違反を作成します。

構文

```
create_drcViolation -name arg [-severity arg] [-msg arg] [-quiet] [-verbose]
[objects...]
```

戻り値

なし

使用法

名前	説明
-name	DRC ルールの名前を指定します。通常、4 ~ 6 文字で指定します。
[-severity]	DRC ルールの重要度を指定します。デフォルトは WARNING です。設定可能な値は、FATAL、ERROR、CRITICAL_WARNING、WARNING、ADVISORY です。
[-msg]	DRC ルールのメッセージ文字列を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	クエリするセル、ポート、ピン、ネット、クロック領域、サイト、パッケージ バンクを指定します。

カテゴリ

[DRC、レポート](#)

説明

report_drc コマンドでのレポート用に DRC 違反オブジェクトを作成し、その違反に関連するデザイン オブジェクトのリストを制御します。

create_drcViolation コマンドは、**create_drcCheck** コマンドで作成されたユーザー定義デザイン ルール チェックのチェック機能を定義および実行する Tcl チェッカー プロシージャの一部として指定されます。違反オブジェクトは、そのデザイン ルール 違反が検出されるたびに Tcl チェッカー プロシージャにより作成されます。

プロセスは次のとおりです。

- ユーザー定義ルールをチェックする際に適用するメソッドと、ルールでチェックするオブジェクトを定義する Tcl チェッカー プロシージャを記述します。Tcl チェッカー プロシージャは別の Tcl スクリプトで定義し、**report_drc** を実行する前に **source** コマンドで読み込む必要があります。
- Tcl チェッカーの **create_drcViolation** を使用して、デザインでルールをチェックした際に検出される違反を特定します。
- create_drcCheck** コマンドを使用して、**-rule_body** で指定した Tcl チェッカー プロシージャを呼び出すユーザー定義 DRC ルール チェックを定義します。
- create_drcRuleDeck** コマンドを使用してルール デックを作成し、**add_drcChecks** コマンドを使用してユーザー定義ルール チェックをルール デックに追加します。
- ルール デックまたはユーザー定義ルール チェックを指定して **report_drc** を実行し、違反がないかどうかをチェックします。

違反は **report_drc** コマンドでレポートされ、違反オブジェクトは **get_drc_vios** コマンドで返されます。

引数

-name arg (必須) : 違反に関連するデザイン ルール チェックの名前を指定します。これは、**-rule_body** で関連の Tcl チェッカー プロシージャを呼び出す **create_drcViolation** コマンドで使用された名前と一致している必要があります。create_drcViolation コマンドからのメッセージは、同じ **-name** で drcCheck に渡されます。

-severity arg (オプション) : 違反の重要度を指定します。ユーザー定義 DRC のデフォルトの重要度は WARNING です。有効な値は、次のとおりです。

- FATAL
- ERROR
- "CRITICAL WARNING"
- WARNING
- ADVISORY

注記 : SEVERITY は DRC 違反オブジェクトに関連付けられた DRC ルールのプロパティとして保存されます。重要度を変更するには、関連付けられている DRC チェックに対して **set_property** コマンドを使用します。

-msg arg (オプション) : **create_drcCheck** で定義したプレースホルダー メッセージで使用されている汎用文字列変数 (\$STR) を置換する違反特定のメッセージを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (オプション) : Tcl チェッカー プロシージャで検出された違反に関連するセル、ポート、ピン、ネット、クロック領域、サイト、パッケージ I/O バンク オブジェクトで、DRC オブジェクトのプレースホルダー メッセージに同じ **-name** で置換されるものを指定します。デザイン オブジェクトは、メッセージの置換キーに次のように対応します。

- %ELG : セル、ポート、ピン、ネットなどのネットリスト エレメント
- %CRG : クロック領域
- %SIG : デバイス サイト
- %PBG : パッケージ I/O バンク

注記 : `create_drcViolation` コマンドで渡される *objects* の順序とタイプが、`create_drcCheck` の **-msg** の指定と一致している必要があります。

例

次の Tcl スクリプトは `dataWidthCheck` プロシージャを定義しています。このプロシージャは、RAMW-1 チェックの **-rule_body** オプションで呼び出されます。この Tcl スクリプト ファイルは、`report_drc` コマンドを実行する前に `source` コマンドを使用して読み込む必要があります。

この Tcl チェッカー プロシージャの機能は、次のとおりです。

- 違反を格納するために作成されるリスト変数 (`$vios`)
- 違反が検出されるたびに違反オブジェクトが作成され、リスト変数に追加されます。
- `$msg` 文字列のプレースホルダー キー `%ELG` は、違反に関連する `$bram` セルに置換されます。
- 違反が検出されると (`$vios > 0`)、`dataWidthCheck` プロシージャによりエラー コードが返され、`report_drc` コマンドにチェックの結果を知らせます。
- 違反のリストは戻りコードと共に渡され、違反は `report_drc` でレポートされます。

```
# This is a simplistic check -- report BRAM cells with WRITE_WIDTH_B wider than 36.
proc dataWidthCheck {} {
    # list to hold violations
    set vios {}
    # iterate through the objects to be checked
    foreach bram [get_cells -hier -filter {PRIMITIVE_SUBGROUP == bram}] {
        set bwidth [get_property WRITE_WIDTH_B $bram]
        if { $bwidth > 36} {
            # define the message to report when violations are found
            set msg "On cell %ELG, WRITE_WIDTH_B is $bwidth"
            set vio [ create_drcViolation -name {RAMW-1} -msg $msg $bram ]
            lappend vios $vio
        }
    }
    if {[llength $vios] > 0} {
        return -code error $vios
    } else {
        return {}
    }
}
create_drcCheck -name {RAMW-1} \
    -hiername {RAMB Checks} \
    -desc {Data Width Check} \
    -rule_body dataWidthCheck
```

注記 : スクリプト ファイルには、Tcl チェッカー プロシージャと、`report_drc` コマンドで使用するためにそれを定義する `create_drcCheck` コマンドの両方を含めることができます。この場合、Tcl スクリプト ファイルを読み込んだときに `dataWidthCheck` プロシージャと RAMW-1 デザイン ルール チェックの両方が読み込まれます。

関連項目

- [add_drc_checks](#)
- [create_drc_ruledeck](#)
- [create_drc_check](#)
- [get_drc_checks](#)
- [get_drc_vios](#)
- [report_drc](#)
- [set_property](#)

create_fileset

新規ファイルセットを作成します。

構文

```
create_fileset [-constrset] [-simset] [-blockset] [-clone_properties arg]
-define_from arg [-quiet] [-verbose] name
```

戻り値

新しいファイルセット オブジェクト

使用法

名前	説明
<code>[-constrset]</code>	ファイルセットを制約ファイルセットとして作成します (デフォルト)。
<code>[-simset]</code>	ファイルセットをシミュレーション ソース ファイルセットとして作成します。
<code>[-blockset]</code>	ファイルセットをブロック ソース ファイルセットとして作成します。
<code>[-clone_properties]</code>	指定したファイルセットからプロパティをコピーします。
<code>-define_from</code>	ブロックセットの最上位となるソース ファイルセットのモジュール名を指定します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>name</code>	作成するファイルセットの名前を指定します。

カテゴリ

プロジェクト、シミュレーション

説明

プロジェクトで新しいファイルセットを定義します。

ファイルセットは、プロジェクト内で特定の機能を持つファイルのリストです。1 つ以上の制約ファイルは制約ファイルセット (`-constrset`)、1 つ以上のシミュレーション テストベンチはシミュレーション ファイルセット (`-simset`) で指定できます。`create_fileset` コマンドを使用する際に指定できるファイルセット オプションは 1 つのみです。タイプを指定しない場合、デフォルトで制約ファイルセットが作成されます。

`create_fileset` コマンドを実行すると、新しく作成されたファイルセットの名前が返されるか、`-quiet` オプションが指定されていなければエラー メッセージが表示されます。

引数

-constrset (オプション) : 1 つ以上の制約ファイルを含める制約ファイルセットを作成します。**-constrset** および **-simset** オプションのどちらも指定しない場合、デフォルトで制約ファイルセットが作成されます。

-simset (オプション) : 1 つ以上のシミュレーション ソース ファイルを含めるシミュレーション ファイルセットを作成します。ファイルセットのタイプには、**-constrset** または **-simset** のいずれか 1 つしか設定できません。両方とも指定した場合は、エラー メッセージが表示されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 作成するファイルセットの名前を指定します。

例

次の例では、constraints2 という名前の新しい制約ファイルセットを作成しています。

```
create_fileset -constrset -quiet constraints2
```

注記 : **-quiet** オプションが指定されているので、指定したファイルセットの作成中にエラーが発生しても、エラー メッセージは表示されません。

次の例では、sim_1 という名前の新しいシミュレーション ファイルセットを作成しています。

```
create_fileset -simset sim_1
```

ファイルは、**add_files** コマンドを使用して新しく作成したファイルセットに追加できます。

関連項目

- [add_files](#)
- [current_fileset](#)

create_generated_clock

既存のクロックからクロック オブジェクトを生成します。

構文

```
create_generated_clock [-name arg] [-source args] [-edges args]
[-divide_by arg] [-multiply_by arg] [-combinational] [-duty_cycle arg]
[-edge_shift args] [-add] [-master_clock arg] [-quiet] [-verbose] objects
```

戻り値

新しいクロック オブジェクト

使用法

名前	説明
[-name]	生成クロックの名前を指定します。
[-source]	マスター クロック ソース オブジェクト ピン/ポートを指定します。
[-edges]	エッジを指定します。
[-divide_by]	周期の分周係数を指定します。1 以上の値を指定します。デフォルト値は 0 です。
[-multiply_by]	周期の過倍係数を指定します。1 以上の値を指定します。デフォルト値は 0 です。
[-combinational]	組み合わせロジックを介して 1 で分周されるクロックを作成します。
[-duty_cycle]	周期過倍のデューティ サイクルを指定します。有効な値は 0.0 ~ 100.0 で、デフォルト値は 0.0 です。
[-edge_shift]	エッジ シフトを指定します。
[-add]	ソース オブジェクトの既存クロックに追加します。
[-master_clock]	マスター ピンに複数のクロックがある場合に、使用するクロックを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	クロック ソース ポート、ピン、またはネットのリストを指定します。

カテゴリ

SDC、XDC

説明

デザインに含まれる既存の物理クロック オブジェクトから新しいクロックを生成します。

注記： このコマンドでは、作成したクロック オブジェクトの名前が返されます。

引数

-name arg (オプション) : 指定のオブジェクトに作成する生成クロックの名前を指定します。名前を指定しない場合、生成クロックの名前はそれが割り当てられるオブジェクトの名前になります。複数のオブジェクトに割り当てる場合は、リストの最初のオブジェクトの名前になります。

-source arg (オプション) : 新しいクロックのマスターとなるクロックのピンまたはポートを指定します。マスター クロックには、仮想クロックではなく定義済みの物理クロックを使用する必要がありますが、プライマリ クロックまたは別の生成クロックを使用できます。ソース ピンまたはポートに複数のクロックが定義されている場合は、**-master_clock** オプションを使用して生成クロックを定義するのに使用するソース上のクロックを指定する必要があります。

-edges arg (オプション) : 生成クロックの遷移を定義するマスター クロックのエッジを指定します。生成クロックの遷移ポイント (1, 2, 3) を、マスター クロックのエッジ カウントで指定します。生成クロックの遷移のシーケンスは、クロックの周期とデューティ サイクルを定義します。位置 1 が生成クロックの最初の立ち上がりエッジを定義し、位置 2 は生成クロックの最初の立ち下がりエッジとなるのでデューティ サイクルを定義し、位置 3 は生成クロックの 2 番目の立ち上がりエッジとなるのでクロック周期を定義します。複数のエッジ番号は波かっこ {} で囲みます。エッジ番号の指定方法は、例を参照してください。

-divide_by arg (オプション) : マスター クロックの周波数を分周する値を指定し、生成クロック オブジェクトの周波数を設定します。1 以上の整数を指定する必要があります。

-multiply_by arg (オプション) : マスター クロックの周波数を倍する値を指定し、生成クロック オブジェクトの周波数を設定します。1 以上の整数を指定する必要があります。

-combinational (オプション) : **-divide_by** 1 のクロックを生成する組み合わせパスを定義します。

-duty_cycle arg (オプション) : **-multiply_by** オプションと共に使用し、生成クロックのデューティ サイクルを新しいクロック周期のパーセントで指定します。有効な値は 0.0 ~ 100 です。

-edge_shift arg (オプション) : 生成クロックのエッジを、マスター クロックを基準に指定の値だけシフトします。エッジシフトの指定方法は、例を参照してください。

-add (オプション) : 生成クロック オブジェクトを *objects* で指定した既存のクロック グループに追加します。

注記 : **-add** を使用する場合は、**-master_clock** と **-name** オプションを指定する必要があります。

-master_clock arg (オプション) : ソース ピンまたはポートに複数のクロックがある場合に、新しいクロック オブジェクトの生成に使用するマスター クロックを指定します。

注記 : **-master_clock** を使用する場合は、**-add** と **-name** オプションを指定する必要があります。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須) : 生成クロックを割り当てるピンまたはポート オブジェクトを指定します。指定したオブジェクトに既にクロックが定義されている場合は、**-add** オプションを使用して新しい生成クロックが追加されるようにし、オブジェクト上の既存のクロックが削除されないようにします。

例

次の例では、指定のクロックピン上のマスタークロックを分周してクロックを生成しています。**-name** が指定されていないので、生成クロックの名前は割り当てるピンと同じになります。

```
create_generated_clock -divide_by 2 -source [get_pins clkgen/sysClk] fftEngine/clk
```

次の例では、指定のソースクロックから CLK1 という生成クロックを定義し、生成クロックの遷移ポイントとして使用するマスタークロックのエッジを指定し、エッジを指定の量シフトしています。**-edges** オプションでソースクロックの 2 番目のエッジを生成クロックの最初のクロックエッジ、ソースクロックの 3 番目のエッジを生成クロックの最初の立ち下がりエッジ、ソースクロックの 8 番目のエッジを生成クロックの 2 番目の立ち上がりエッジに指定しています。これらの値により、生成クロックの周期はソースクロックのエッジ 2 からエッジ 8 までの時間に定義され、デューティサイクルはソースクロックのエッジ 2 からエッジ 3 までの時間の周期に対するパーセントに定義されます。また、生成クロックの各エッジは指定の量シフトされます。

```
create_generated_clock -name CLK1 -source CMB/CLKIN -edges {2 3 8} \
-edge_shift {0 -1.0 -2.0} CMB/CLKOUT
```

注記：生成クロックの波形は、**-edges** オプションで定義された遷移に基づいて繰り返されます。

次の例では、MUX の出力から 2 つの生成クロックを作成します。クロックを割り当てるオブジェクトに複数のクロックが割り当てられているので、**-master_clock** を使用して使用するクロックを指定し、**-add** を使用してフリップフロップの Q ピンに生成クロックを割り当て、**-name** を使用して生成クロックの名前を定義しています。

```
create_generated_clock -source [get_pins muxOut] -master_clock M_CLKA \
-divde_by 2 -add -name gen_CLKA [get_pins flop_Q]
create_generated_clock -source [get_pins muxOut] -master_clock M_CLKB \
-divde_by 2 -add -name gen_CLKB [get_pins flop_Q]
```

関連項目

- [check_timing](#)
- [create_clock](#)
- [get_generated_clocks](#)
- [get_pins](#)
- [set_clock_latency](#)
- [set_clock_uncertainty](#)
- [set_propagated_clock](#)

create_hw_sio_link

ハードウェア RX と TX エンドポイントの間に新しいリンクを作成します。ハードウェア TX または RX エンドポイントを少なくとも 1 つ指定する必要があります。どちらかがない場合、エンドポイントは不明として処理されます。不明のエンドポイントは、リンク プロパティで名前を変更できます。

構文

```
create_hw_sio_link [-quiet] [-verbose] [hw_sio_rx] [hw_sio_tx]
```

戻り値

新しいハードウェア SIO リンク

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_sio_rx]	RX エンドポイントを指定します。デフォルトはなしです。
[hw_sio_tx]	TX エンドポイントを指定します。デフォルトはなしです。

カテゴリ

ハードウェア

create_hw_sio_linkgroup

新しいハードウェア SIO リンク グループを作成します。

構文

```
create_hw_sio_linkgroup [-quiet] [-verbose] hw_sio_links
```

戻り値

新しいハードウェア SIO リンク グループ

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hw_sio_links</i>	ハードウェア SIO リンクを指定します。

カテゴリ

ハードウェア

create_hw_sio_scan

新しいハードウェア SIO スキャンを作成します。

構文

```
create_hw_sio_scan [-quiet] [-verbose] scan_type [hw_sio_rx]
```

戻り値

新しいハードウェア SIO スキャン

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
scan_type	スキャンタイプを指定します。有効な値は 1d_bathtub、2d_full_eye です。
[hw_sio_rx]	スキャンを実行する RX エンドポイントを指定します。デフォルトはなしです。

カテゴリ

ハードウェア

create_interface

I/O ポート インターフェイスを作成します。

構文

```
create_interface [-parent arg] [-quiet] [-verbose] name
```

戻り値

新しいインターフェイス オブジェクト

使用法

名前	説明
[-parent]	新しいインターフェイスを割り当てる親インターフェイスを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	新しい I/O ポートインターフェイスの名前を指定します。

カテゴリ

ピン配置

説明

スカラー ポートまたは差動 I/O ポートをグループ化するための新しいインターフェイスを作成します。

引数

-parent *arg* (オプション) : 新しいインターフェイスを割り当てる親インターフェイスを指定します。

注記 : 指定した親インターフェイスが存在しない場合、エラー メッセージが表示されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 作成する I/O ポートインターフェイスの名前を指定します。

例

次の例では、新しい USB インターフェイスが作成されます。

```
create_interface USB0
```

次の例では、指定した親インターフェイスにイーサネットインターフェイスが作成されます。

```
create_interface -parent Top_Int ENET0
```

関連項目

- [delete_interface](#)
- [create_port](#)
- [make_diff_pair_ports](#)
- [place_ports](#)
- [remove_port](#)
- [set_package_pin_val](#)
- [split_diff_pair_ports](#)

create_ip

コンフィギュラブル IP のインスタンスを作成して、ファイルセットに追加します。

構文

```
create_ip [-vlnv arg] -module_name arg [-dir arg] [-vendor arg]
[-library arg] [-name arg] [-version arg] [-quiet] [-verbose]
```

戻り値

追加されたファイル オブジェクトのリスト

使用法

名前	説明
[-vlnv]	新しい IP の作成元となる IP カタログの VLNV 文字列 (Vendor:Library:Name:Version) を指定します。
-module_name	プロジェクトに追加する新しい IP の名前を指定します。
[-dir]	プロジェクト外で作成し、管理するリモート IP へのディレクトリ パスを指定します。
[-vendor]	IP ベンダーの名前を指定します。
[-library]	IP ライブラリの名前を指定します。
[-name]	IP 名を指定します。
[-version]	IP のバージョン番号を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

IP フロー

説明

IP カタログからのコンフィギュラブル IP の XCI ファイルを作成し、現在のプロジェクトのソース ファイルに追加します。ネットリストに IP コアのインスタンスを作成するには、作成した IP ソース オブジェクトを HDL ファイルにインスタンシエートする必要があります。

同じコアのインスタンスが複数必要な場合は、HDL デザインにコアのモジュールを必要な数だけインスタンシエートします。同じ IP コアを異なるコンフィギュレーションで使用する場合は、**create_ip** コマンドを使用して別の IP ソース オブジェクトを作成する必要があります。

create_ip は、現在の IP カタログから IP コアをインポートするために使用します。カタログに IP を追加せずに XCI および XCO ファイルを直接読み込むには、**import_ip** コマンドを使用します。

このコマンドは IP 生成プロセスと、インポートされた IP コア ファイルのパスと名前を返します。

注記 : IP コアは Vivado にネイティブであり、Vivado ツールでカスタマイズおよび再生成できます。**convert_ip** コマンドを使用すると、レガシ IP を Vivado でサポートされているネイティブ IP に変換できます。

引数

-vlnv arg (オプション) : 新しい IP の作成元となる IP カタログの VLVN 文字列を指定します。VLNV は Vendor:Library:Name:Version 形式の文字列で、カタログ内の IP を識別します。VLNV 文字列は IP コアの IPDEF プロパティにマップされています。

注記 : **-vlnv** を指定するか、**-vendor**、**-library**、**-name**、および **-version** をすべて指定します。

-module_name arg (必須) : プロジェクトに追加する新しい IP インスタンスの名前を指定します。

-dir arg (オプション) : IP コア ファイルを保存するディレクトリを指定します。このオプションを指定しない場合、IP コア ファイル (.xci, .ngc, .veo など) は *project_name.srcs* ディレクトリの階層に保存されます。

-vendor arg (オプション) : IP を作成したベンダーの名前を指定します。

-library arg (オプション) : コアの追加元の IP ライブラリを指定します。

-name arg (オプション) : カタログ内での IP コアの名前を指定します。

-version arg (オプション) : IP コアのバージョン場号を指定します。

注記 : **-vlnv** を指定するか、**-vendor**、**-library**、**-name**、および **-version** をすべて指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、**-vlnv** 文字列で指定した IP コアをインポートし、現在のプロジェクトでのモジュール名を指定しています。

```
create_ip -vlnv xilinx.com:ip:c_addsub:11.0 -module_name test_addr
```

次の例では、Vivado で **-vendor**、**-library**、**-name**、**-version** を指定して IP ブロックを作成し、モジュール名を指定しています。IP を作成したら、**set_property** コマンドを使用して IP の属性をカスタマイズし、インスタンシエーション テンプレートおよび合成ターゲットを生成します。

```
create_ip -name c_addsub -version 11.0 -vendor xilinx.com -library ip \
    -module_name c_addsub_v11_0_0
set_property -name CONFIG.Component_Name -value {c_addsub_v11_0_0} \
    -objects [get_ips c_addsub_v11_0_0]
set_property -name CONFIG.A_Width -value {32} \
    -objects [get_ips c_addsub_v11_0_0]
set_property -name CONFIG.B_Width -value {32} \
    -objects [get_ips c_addsub_v11_0_0]
set_property -name CONFIG.Add_Mode -value {Add_Subtract} \
    -objects [get_ips c_addsub_v11_0_0]
set_property -name CONFIG.C_In -value {true} \
    -objects [get_ips c_addsub_v11_0_0]
generate_target {instantiation_template synthesis} \
    [get_files C:/Data/c_addsub_v11_0_0/c_addsub_v11_0_0.xci \
    -of_objects [get_filesets sources_1]]
```

関連項目

- [generate_target](#)
- [import_ip](#)
- [upgrade_ip](#)
- [validate_ip](#)

create_macro

マクロを作成します。

構文

```
create_macro [-quiet] [-verbose] name
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	作成するマクロの名前を指定します。

カテゴリ

xdc

説明

セルを相対配置するためのマクロを作成します。

マクロは、リソース効率を改善し内部接続を高速にするため、関連のセルをグループ化して一緒に配置するために主に使用します。**create_macro** コマンドを使用すると、RTL ソースファイルで RLOC 制約を使用して定義されている RPM と同様に、合成済みデザインまたはインプリメント済みデザインで **place_design** で相対配置されるようマクロを定義できます。相対配置マクロの定義方法は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

マクロを作成したら、**update_macro** コマンドを使用して特定のセルをマクロに割り当てることができます。定義済みのマクロを変更するには、**delete_macro** コマンドを使用してマクロを削除してからマクロを再作成し、新しい内容でマクロをアップデートする必要があります。既存のマクロを単に上書きすることはできません。

定義済みのマクロを削除するには、**delete_macro** を使用します。デザインで現在定義されているマクロをリストするには、**get_macros** を使用します。

このコマンドを実行しても、何も返されません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 作成するマクロの名前を指定します。

例

次の例では、usbMacro1 というマクロを作成しています。

```
create_macro usbMacro1
```

関連項目

- [delete_macros](#)
- [get_macros](#)
- [place_design](#)
- [update_macro](#)

create_net

現在のデザインにネットを作成します。

構文

```
create_net [-from arg] [-to arg] [-quiet] [-verbose] nets...
```

戻り値

なし

使用法

名前	説明
[-from]	開始バス インデックスを指定します。
[-to]	終了バス インデックスを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
nets	作成するネットの名前を指定します。

カテゴリ

ネットリスト

説明

開いている合成済みデザインまたはインプリメント済みデザインのネットリストにネットを追加します。

注記：ネットをライブラリ マクロまたはマクロ プリミティブに追加することはできません。

ネットはデザインの最上位に作成するか、階層ネット名を指定して任意の階層レベルに作成できます。

バス ネットは、バス インデックス値に正または負の値を使用することにより、増加するバス インデックスまたは減少するバス インデックスで作成できます。

新しいネットは、作成したときにはネットリスト内では接続されていません。**connect_net** コマンドを使用して接続する必要があります。接続されているネットの接続を解除するには **disconnect_net** コマンドを使用し、ネットリストから削除するには **remove_net** コマンドを使用します。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソースファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザイン チェックポイントとして保存するか、**write_*** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリスト ファイルにエクスポートできます。

注記：エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

引数

-from *arg* (オプション) : 新しいバスの開始インデックスを指定します。

-to *arg* (オプション) : 新しいバスの終了インデックスを指定します。

注記 : **-from** または **-to** を片方だけ指定すると、**-from** または **-to** で指定したインデックス値の 1 ビット バスが作成されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

nets (必須) : 作成するネットの名前を指定します。ネット名は最上位から指定するか、名前のみ (net1) で指定するか、階層ネット名 (cell1/cellA/net1) を指定してデザイン階層内で指定できます。

例

次の例では、現在の合成済みデザインまたはインプリメント済みデザインに 24 ビット バスを作成しています。

```
create_net tempBus -from 23 -to 0
```

関連項目

- [connect_net](#)
- [create_pin](#)
- [create_port](#)
- [disconnect_net](#)
- [get_nets](#)
- [remove_net](#)
- [resize_net_bus](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

create_pblock

新しい Pblock を作成します。

構文

```
create_pblock [-quiet] [-verbose] name
```

戻り値

新しい Pblock オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	新しい Pblock の名前を指定します。

カテゴリ

XDC、フロアプラン

説明

フロアプラン用にロジック インスタンスを追加する Pblock を定義します。

Pblock にロジック エレメントを追加するには **add_cells_to_pblock** コマンド、Pblock を FPGA ファブリック上に配置するには **resize_pblocks** コマンドを使用します。**resize_pblock** コマンドを使用すると、Pblock を手動で移動およびサイズ変更できます。

最初の例に示すように、**-parent** を使用して、階層フロアプラン用に 1 つの Pblock を別の Pblock にネストできます。また、2 番目の例に示すように、**set_property** コマンドを使用して PARENT プロパティを設定すると、既存の Pblock を別の Pblock の中にネストできます。

引数

-parent arg (オプション) : ネストされた Pblock を作成する親 Pblock の名前を指定します。親を指定しない場合、デフォルトで Root が親に指定され、Pblock がデザインの最上位に配置されます。**get_pblocks** コマンドを使用すると、現在定義されている Pblock で親として指定できるものをレポートできます。

注記 : 指定した親が存在しない場合、エラー メッセージが表示されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 作成する Pblock の名前を指定します。

例

次の例では、Vid_Array という Pblock の中に Video1 という Pblock を作成します。

```
create_pblock -parent Vid_Array Video1
```

次の例では、cpu1、cpu2、cpuEngine という 3 つの Pblock を作成した後、set_property コマンドを使用して cpuEngine の中に cpu1 および cpu2 をネストします。

```
create_pblock cpu1
create_pblock cpu2
create_pblock cpuEngine
set_property PARENT cpuEngine [get_pblocks {cpu1 cpu2}]
```

関連項目

- [add_cells_to_pblock](#)
- [get_pblocks](#)
- [place_pblocks](#)
- [resize_pblock](#)
- [set_property](#)

create_pin

現在のデザインにピンを作成します。

構文

```
create_pin [-from arg] [-to arg] -direction arg [-quiet] [-verbose] pins...
```

戻り値

なし

使用法

名前	説明
[-from]	開始バス インデックスを指定します。
[-to]	終了バス インデックスを指定します。
-direction	ピンの方向を指定します。有効な値は IN、OUT、INOUT です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>pins</i>	作成するピンの名前を指定します。

カテゴリ

ネットリスト

説明

開いている合成済みデザインまたはインプリメント済みデザインのネットリストにシングル ピンまたはバス ピンを追加します。ピン名だけでなく、方向やバス幅などの属性も指定できます。

バスピンは、バスインデックス値に正または負の値を使用することにより、増加するバスインデックスまたは減少するバスインデックスで作成できます。

ピンは、既存のセル インスタンスに作成するか、**create_port** コマンドを使用して最上位ピンとして作成する必要があります。ピン名の一部としてセルのインスタンス名を指定しない場合、エラーが返されます。

注記：ピンをライブラリ マクロまたはマクロ プリミティブに追加することはできません。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソースファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザイン チェックポイントとして保存するか、**write_*** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリスト ファイルにエクスポートできます。

注記：エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

引数

-from *arg* (オプション) : バス ピンの開始インデックスを指定します。

-to *arg* (オプション) : バス ピンの終了インデックスを指定します。

-direction (必須) : ピンの方向を指定します。有効な値は IN、OUT、INOUT です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

pins (必須) : 作成するピンの名前を指定します。ピン名は、ピンを割り当てるセル インスタンスを基準に階層で指定する必要があります。デザインの最上位に作成されたピンはポートであり、**create_port** コマンドで作成する必要があります。

例

次の例では、cpuEngine モジュールに指定した名前の入力ピンを作成しています。

```
create_pin -direction IN cpuEngine/inPin
```

次の例では、まず階層区切り文字を設定し、参照セルのブラック ボックス インスタンスを作成して、そのインスタンスに 24 ビットの双方向バスを作成しています。

```
set_hierarchy_separator |
create_cell -reference dmaBlock -black_box usbEngine0|myDMA
create_pin -direction INOUT -from 0 -to 23 usbEngine0|myDMA|dataBus
```

関連項目

- [create_cell](#)
- [create_net](#)
- [create_port](#)
- [connect_net](#)
- [disconnect_net](#)
- [remove_cell](#)
- [remove_pin](#)
- [resize_pin_bus](#)
- [set_hierarchy_separator](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

create_port

スカラー ポートまたはバス ポートを作成します。

構文

```
create_port -direction arg [-from arg] [-to arg] [-diff_pair]
[-interface arg] [-quiet] [-verbose] name [negative_name]
```

戻り値

作成されたポート オブジェクトのリスト

使用法

名前	説明
<code>-direction</code>	ポートの方向を指定します。有効な値は IN、OUT、INOUT です。
<code>[-from]</code>	新しいバスの開始インデックスを指定します。
<code>[-to]</code>	新しいバスの終了インデックスを指定します。
<code>[-diff_pair]</code>	ポートの差動ペアを作成します。
<code>[-interface]</code>	ポートを割り当てるインターフェイスを指定します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>name</code>	ポート名を指定します。
<code>[negative_name]</code>	差動ペアの N 側の名前を指定します (オプション)。

カテゴリ

ピン配置

説明

方向、幅、シングルエンドか差動かなどのパラメーターを指定してポートを作成し、オプションで既存のインターフェイスに割り当てます。新しいポートは、デザイン階層の最上位に追加されます。

バス ポートは、バス インデックス値に正または負の値を使用することにより、増加するバス インデックスまたは減少するバス インデックスで作成できます。

`create_port` コマンドを使用すると、I/O 配置プロジェクトで新しいポートを作成したり、開いている合成済みデザインまたはインプリメント済みデザインのネットリストを編集中に新しいポートを作成できます。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソースファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザイン チェックポイントとして保存するか、**write_*** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリスト ファイルにエクスポートできます。

注記：エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

引数

-direction (必須)：ポートの方向を指定します。有効な値は IN、OUT、INOUT です。

-from arg (オプション)：新しいバスの開始インデックスを指定します。負のインデックス値から開始することも可能です。

-to arg (オプション)：新しいバスの終了インデックスを指定します。負のインデックス値で終了することも可能です。

-diff_pair (オプション)：ポートを差動ペアとして作成します。P 側と N 側の両方のポートが作成されます。*name* のみを指定した場合、P 側のポートの名前が *name* になり、N 側のポートは *name_N* という名前になります。*name* と *negative_name* の両方を指定した場合、P 側のポートの名前が *name* になり、N 側のポートの名前が *negative_name* になります。

-interface arg (オプション)：ポートを割り当てるインターフェイスを指定します。

注記：**create_interface** コマンドを使用してインターフェイスを定義しておく必要があります。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記：コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須)：作成するポートの名前を指定します。**-diff_pair** を指定した場合、*name* は P 側のポートの名前となり、N 側のポートの名前は *name_N* となります。

negative_name (オプション)：**-diff_pair** を使用する場合に P 側のポートの名前を指定します。この場合、*name* が P 側のポートの名前になり、*negative_name* が N 側のポートの名前になります。

例

次の例では、PORT0 という名前の入力ポートを作成しています。

```
create_port -direction IN PORT0
```

次の例では、まず Group1 という新しいインターフェイスを作成し、その後そのインターフェイスを使用して 4 ビットの差動ペア出力バスを作成しています。バス ポートが差動ペアとして定義されており、*name* のみが指定されているので、N 側のポートの名前は D_BUS_N になります。

```
create_interface Group1
create_port -direction OUT -from 0 -to 3 -diff_pair -interface Group1 D_BUS
```

注記：このコマンドにより、D_BUS[0]、D_BUS_N[0]、D_BUS[1]、D_BUS_N[1]、D_BUS[2]、D_BUS_N[2]、D_BUS[3]、D_BUS_N[3] という 8 本のポートが作成されます。

次のコマンドでは、*name* のみを指定し、data および data_N という差動ペア出力ポートを作成しています。

```
create_port -direction OUT -diff_pair data
```

次のコマンドでは、*name* と *negative_name* の両方を指定し、data_P および data_N という差動ペア出力ポートを作成しています。

```
create_port -direction OUT -diff_pair data_P data_N
```

関連項目

- [create_interface](#)
- [make_diff_pair_ports](#)
- [place_ports](#)
- [remove_port](#)
- [resize_port_bus](#)
- [split_diff_pair_ports](#)

create_project

新しいプロジェクトを作成します。

構文

```
create_project [-part arg] [-force] [-ip] [-quiet] [-verbose] [name] [dir]
```

戻り値

新しいプロジェクト オブジェクト

使用法

名前	説明
[-part]	ターゲット パーツを指定します。
[-force]	既存のプロジェクトディレクトリを上書きします。
[-ip]	IP 管理プロジェクトでの GUI のデフォルト動作を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[name]	プロジェクト名を指定します。
[dir]	プロジェクトファイルを保存するディレクトリを指定します。デフォルトは . です。

カテゴリ

プロジェクト

説明

指定したディレクトリにプロジェクト ファイル (.xpr) を作成します。

引数

-part arg (オプション) : プロジェクトで使用するザイリンクス パーツを指定します。これは、プロジェクトの作成後に変更できます。-part オプションを指定しない場合、デフォルトのパーツが使用されます。

-force (オプション) : 既存のプロジェクトを上書きします。指定した dir に指定のプロジェクト名が既に存在する場合、-force オプションを使用して既存のプロジェクトを上書きする必要があります。

注記 : 既存プロジェクトを現在ツールで開いている場合、新しいプロジェクトでディスクの既存プロジェクトが上書きされますが、両方のプロジェクトがツールで開いたままになります。この場合、**create_project** を実行する前に **close_project** コマンドを実行しておくことをお勧めします。

-ip (オプション) : IP 管理フローで IP カタログの IP を表示およびカスタマイズし、コンフィギュレーション済みの IP のリポジトリを管理するためのプロジェクトを作成します。IP 管理フローの詳細は、『Vivado Design Suite ユーザー ガイド : IP を使用した設計』(UG896) を参照してください。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (オプション) : この引数はパラメーターナー名を必要としませんが、*dir* よりも前に記述する必要があります。パラメーターがないので、最初の引数は *name*、2 つ目の引数は *dir* として認識されます。プロジェクトファイル *name.xpr* およびプロジェクトデータフォルダー *name.data* が作成され、両方とも指定したディレクトリ *dir* に配置されます。

注記 : 作成されるプロジェクトファイルは、デフォルトでは RTL ソース ファイルです。**set_property** コマンドを使用して DESIGN_MODE プロパティを設定し、RTL ソース プロジェクトから、たとえば I/O ピン配置などの別のプロジェクト タイプに変更する必要があります。

dir (オプション) : 新しいプロジェクトファイルを保存するディレクトリ名を指定します。指定したディレクトリが存在しない場合は、その名前の新しいディレクトリが作成されます。ディレクトリを完全なパスで指定すると、その指定したパス名が使用されます。*dir* をパスなしで指定すると、現在の作業ディレクトリまたはツールの起動ディレクトリでそのディレクトリが検索されるか、作成されます。

注記 : プロジェクトを GUI モードで作成すると、ディレクトリ名 *dir* にファイル名 *name* を付けた *dir/name* という名前のプロジェクトディレクトリが作成され、新しいプロジェクトファイルとプロジェクトデータ フォルダーが保存されます。

例

次の例では、myDesigns というディレクトリに Project1 というプロジェクトを作成しています。

```
create_project Project1 myDesigns
```

注記 : *dir* にはフォルダーナー名のみが指定されているので、プロジェクトは現在の作業ディレクトリかツールの起動ディレクトリに作成されます。

次の例では、C:/Designs の FPGA というディレクトリに Proj1 というプロジェクトを作成しています。指定したディレクトリに同名のプロジェクトが既に存在する場合は、それが上書きされます。2 行目と 3 行目では、**-force** の位置を変更しても問題ないことを示しています。

```
create_project Proj1 C:/Designs/FPGA -force
-or-
create_project Proj1 -force C:/Designs/FPGA
-or-
create_project -force Proj1 C:/Designs/FPGA
```

注記 : どの例でも、キーワードのない最初の引数は *name* 変数として、2 つ目の引数は *dir* 変数として認識されます。

次の例では、IP 管理フロー用のプロジェクトを指定したディレクトリに作成しています。

```
create_project -ip manageIP C:/Data
```

次の例では、pin_project という新規プロジェクトを作成し、**design_mode** プロパティを I/O ピン配置プロジェクトに設定し、最後に I/O ピン配置デザインを開いています。

```
create_project pin_project C:/Designs/PinPlanning
set_property design_mode PinPlanning [current_fileset]
open_io_design -name io_1
```

関連項目

- [current_project](#)
- [set_property](#)
- [open_io_design](#)

create_property

オブジェクトのクラスのプロパティを作成します。

構文

```
create_property [-type arg] [-quiet] [-verbose] name class
```

戻り値

作成されたプロパティ、エラーが発生した場合は ""

使用法

名前	説明
[-type]	作成するプロパティのタイプを指定します。有効な値は string、int、long、double、bool です。デフォルトは string です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	作成するプロパティの名前を指定します。
class	プロパティを作成するオブジェクト タイプを指定します。有効な値は、design、net、cell、pin、port、pblock です。

カテゴリ

プロパティおよびパラメーター、XDC

説明

指定したオブジェクトのクラスに対し、指定したタイプのプロパティを指定した名前で作成します。作成したプロパティは、**set_property** コマンドで指定したクラスのオブジェクトに割り当てすることができますが、そのクラスのすべてのオブジェクトに自動的には関連付けられません。

プロパティをそのオブジェクトに割り当てないと、**report_property -all** コマンドを使用しても、指定したクラスのオブジェクトに対して新しく作成したプロパティはレポートされません。

引数

-type arg(オプション)：作成するプロパティのタイプを指定します。次のプロパティタイプを指定できます。

- **string**：新しいプロパティを文字列値で定義します。**-type** を使用しない場合、これがデフォルトです。
- **int**：新しいプロパティを長整数値で定義します。**int** プロパティ タイプに 10 進数値を指定すると、エラーメッセージが表示されます。
- **double**：新しいプロパティを浮動小数点値で定義します。
- **bool**：新しいプロパティをブール値 (1 = True、0 = False) で定義します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 定義するプロパティの名前を指定します。大文字/小文字が区別されます。

class (必須) : 新しいプロパティを割り当てるオブジェクトのクラスを指定します。指定したクラスのオブジェクトはすべて、新しく定義したプロパティに割り当てられます。有効な値は、design、net、cell、pin、port、pblock です。

例

次の例では、セル オブジェクトに対して PURPOSE というプロパティを定義します。

```
create_property PURPOSE cell
```

注記 : **-type** は指定されていないので、値は文字列になります。

次の例では、整数値を指定する COUNT というピン プロパティを作成します。

```
create_property -type int COUNT pin
```

関連項目

- [get_property](#)
- [list_property](#)
- [list_property_value](#)
- [report_property](#)
- [reset_property](#)
- [set_property](#)

create_run

現在のプロジェクトの合成またはインプリメンテーション run を定義します。

構文

```
create_run [-constrset arg] [-parent_run arg] [-part arg] -flow arg
[-strategy arg] [-quiet] [-verbose] name
```

戻り値

run オブジェクト

使用法

名前	説明
[-constrset]	使用する制約ファイルセットを指定します。
[-parent_run]	新しいインプリメンテーション run に関連付ける合成 run を指定します。
[-part]	ターゲット パーツを指定します。
-flow	フロー名を指定します。
[-strategy]	run に適用するストラテジを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	新規 run の名前を指定します。

カテゴリ

プロジェクト

説明

合成またはインプリメンテーション run を定義します。run の属性は、**config_run** コマンドを使用して設定できます。

引数

-constrset arg (オプション) : 合成またはインプリメンテーション run で使用する制約セットを指定します。

-parent_run arg : インプリメンテーション run でインプリメントする合成 run を指定します。RTL ソース プロジェクトの場合、インプリメンテーション run に -parent_run を指定する必要があります。合成 run には必要ありません。ネットリスト ベースのプロジェクトの場合、インプリメンテーション run を定義するのに -parent_run オプションは必要ありません。

-part partName (オプション) : run に使用するザイリンクス パーツを指定します。-part オプションを指定しない場合は、プロジェクトに定義されているデフォルト パーツが使用されます。

-flow arg (必須) : 合成ツール ({Vivado Synthesis 2012}) またはインプリメンテーション ツール ({Vivado Implementation 2012}) のツール フローおよびリリース バージョンを指定します。

-strategy arg (オプション) : 合成またはインプリメンテーション run で使用するストラテジを指定します。ツールには、ユーザー定義のカスタム ストラテジも含め、多くのストラテジが含まれています。使用可能な合成およびインプリメンテーション ストラテジについては、該当するユーザー ガイドを参照してください。ストラテジを指定しない場合、[Synthesis Defaults] または [Implementation Defaults] が使用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 設定する合成 run またはインプリメンテーション run の名前を指定します。

例

次の例では、Vivado 合成ツール フローを使用する synth_1 という名前の run を作成しています。

```
create_run -flow {Vivado Synthesis 2013} synth_1
```

注記 : この場合、合成 run でデフォルトの sources_1、constrs_1、プロジェクトのデフォルト パーツが使用されます。また、これは合成 run なので、**-parent_run** オプションは必要ありません。

次の例では、Vivado インプリメンテーション 2013 ツール フローを使用するインプリメンテーション run を作成し、先ほどの例で作成した synth_1 という合成 run に関連付けています。

```
create_run impl_2 -parent_run synth_1 -flow {Vivado Implementation 2013}
```

注記 : この例では、合成済み RTL ソースをインプリメントしているので、**-parent_run** オプションを使用する必要があります。

関連項目

- [current_run](#)
- [launch_runs](#)
- [set_property](#)

create_slack_histogram

スラックヒストグラムを作成します。

構文

```
create_slack_histogram [-to args] [-delay_type arg] [-num_bins arg]
[-slack_less_than arg] [-slack_greater_than arg] [-group args]
[-report_unconstrained] [-significant_digits arg] [-scale arg] [-name arg]
[-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-to]	範囲の終わりのクロックを指定します。
[-delay_type]	パス遅延のタイプを指定します。有効な値は max、min、min_max で、デフォルトは max です。
[-num_bins]	bin の最大数を指定します。1 以上の値を指定します。デフォルト値は 10 です。
[-slack_less_than]	この値よりも小さいスラックのパスを表示します。デフォルトは 1e+30 です。
[-slack_greater_than]	この値よりも大きいスラックのパスを表示します。デフォルトは -1e+30 です。
[-group]	指定のグループのパスのみをレポートします。
[-report_unconstrained]	制約の設定されていないエンドポイントをレポートします。
[-significant_digits]	有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 3 です。
[-scale]	ヒストグラムを描画する尺度のタイプを指定します。有効な値は linear または logarithmic で、デフォルトは linear です。
[-name]	GUI パネルに表示する結果の名前を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

パスをスラック値ごとにグループ化し、グラフィカルに表示するスラックヒストグラムを作成します。

グラフィカル スラックヒストグラムを表示するにはツールを GUI モードで実行している必要があり、GUI での表示名を指定するには **-name** オプションを使用します。

引数

- to arg** (オプション) : クロック名を指定し、そのクロックドメインで終了するパスを解析します。
 - delay_type arg** (オプション) : スラックレポートを作成する際に解析するパス遅延のタイプを指定します。有効な値は min、max、および min_max です。デフォルトは max です。
 - num_bins arg** (オプション) : 結果を分割するスラック ビンの数を指定します。ビンの数により、ヒストグラムの粒度が決まります。算出されたスラック値の範囲が指定したビン数に均等に分割され、スラック値に応じてパスがグループ化されます。1 以上の値を指定でき、デフォルト値は 10 です。
 - slack_less_than arg** (オプション) : 算出されたスラック値が指定した値より小さいパスのスラックをレポートします。**-slack_greater_than** と共に使用すると、スラック値の範囲を指定できます。
 - slack_greater_than arg** (オプション) : 算出されたスラック値が指定した値より大きいパスのスラックをレポートします。**-slack_less_than** と共に使用すると、スラック値の範囲を指定できます。
 - group args** (オプション) : 指定したパス グループのパスのスラックをレポートします。現在定義されているパス グループを確認するには、**get_path_groups** コマンドを使用します。
 - report_unconstrained** (オプション) : 制約が適用されていないパスの遅延スラックをレポートします。デフォルトでは、制約が適用されていないパスは解析されません。
 - significant_digits arg** (オプション) : 出力結果の有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 3 です。
 - scale [linear | logarithmic]** (オプション) : スラックヒストグラムで使用する Y 軸の尺度を指定します。logarithmic に設定すると、大きく異なる値の表示がスムーズになります。デフォルトは linear です。
 - name arg** (オプション) : GUI で表示する場合の結果の名前を指定します。指定した名前の結果が既に開いている場合は、その結果が上書きされます。
 - quiet** (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。
- 注記** : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。
- verbose** (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。
- 注記** : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインのスラックヒストグラムをデフォルト設定で作成し、結果を指定した名前で GUI に表示しています。

```
create_slack_histogram -name slack1
```

関連項目

- [delete_timing_results](#)
- [get_path_groups](#)
- [report_timing](#)

create_sysgen

ザイリンクス System Generator 用の DSP ソースを作成し、ソース ファイルセットに追加します。

構文

```
create_sysgen [-quiet] [-verbose] name
```

戻り値

新しいサブモジュールの名前

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	サブモジュールの名前を指定します。

カテゴリ

System Generator

説明

現在のプロジェクトで使用する DSP サブモジュールを作成し、ソース ファイルとして追加します。

このコマンドを実行すると、System Generator for DSP が起動します。System Generator はザイリンクスの DSP デザイン ツールで、RTL ソース ファイル、Simulink® および MATLAB® ソフトウェア モデル、および DSP システムの C/C++ コンポーネントを 1 つのシミュレーションおよびインプリメンテーション環境にまとめることができます。

ツールの機能の詳細は、『System Generator for DSP 入門ガイド』(UG639) を参照してください。

System Generator からの既存の DSP モジュール ファイル (.mdl) を **add_files** コマンドを使用して現在のプロジェクトに追加することもできます。

このコマンドを実行すると、作成されプロジェクトに追加された DSP モジュールの名前が返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 作成し、プロジェクトに追加する DSP モジュールの名前を指定します。

例

次の例では、System Generator を起動し、指定の DSP モジュールを定義および設定しています。

```
create_sysgen DSP_mod1
```

関連項目

- [add_files](#)
- [generate_target](#)
- [list_targets](#)
- [make_wrapper](#)

create_wave_config

新しい波形コンフィギュレーションを作成します。

構文

```
create_wave_config [-quiet] [-verbose] [name]
```

戻り値

新しい波形コンフィギュレーション

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[name]	作成する波形コンフィギュレーションの名前を指定します。指定しない場合は、デフォルト名で作成されます。波形コンフィギュレーションを表示する新しい波形ウィンドウが作成され、現在の波形ウィンドウとなります。

カテゴリ

波形

create_xps

XPS 用のエンベデッド ソースを作成し、ソース ファイルセットに追加します。

構文

```
create_xps [-quiet] [-verbose] name
```

戻り値

作成されたソース ファイルの名前

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	ソース ファイルの名前を指定します。

カテゴリ

XPS

説明

現在のプロジェクトで使用するエンベデッド プロセッサ ソースを作成し、ソース ファイルとして追加します。

このコマンドを実行すると、Xilinx Platform Studio (XPS) が起動し、エンベデッド プロセッサ システムのハードウェア部分を設計できます。XPS では、マイクロプロセッサ、ペリフェラル、およびそれらのコンポーネントのインターフェクトを定義および設定できます。XPS を閉じると、作成されたエンベデッド プロセッサ サブデザインのファイルがローカルのプロジェクトディレクトリ (*project_name.srcts/sources_1/edk/name*) に保存され、ソース ファイルとして追加されます。

XPS の機能の詳細は、『EDK コンセプト、ツール、テクニック』(UG683) を参照してください。

XPS からの既存の Xilinx Microprocessor Project (.xmp) ファイルを **add_files** コマンドを使用して現在のプロジェクトに追加することもできます。

このコマンドを実行すると、作成されたエンベデッド プロセッサ サブデザインの名前が返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 作成し、プロジェクトに追加するエンベデッド プロセッサ サブデザインの名前を指定します。

例

次の例では、XPS を起動し、指定のエンベデッド プロセッサ サブデザインを定義および設定しています。

```
create_xps xpsTest1
```

関連項目

- [add_files](#)
- [generate_target](#)
- [list_targets](#)
- [make_wrapper](#)

current_bd_design

現在のデザインを設定または取得します。

構文

```
current_bd_design [-quiet] [-verbose] [design]
```

戻り値

現在のデザイン オブジェクト、エラーが発生した場合は ""

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[design]	設定する現在のデザインの名前を指定します。

カテゴリ

IP インテグレーター

説明

Vivado Design Suite の IP インテグレーター機能で使用する現在の IP サブシステム デザインを定義するか、アクティブ プロジェクトの現在のデザインを返します。

IP インテグレーターで実行するデほとんどの Tcl コマンドおよびデザインの変更では、現在の IP サブシステム デザインおよび現在の IP サブシステム インスタンスがターゲットとなります。現在の IP サブシステム インスタンスを指定するには、**current_bd_instance** コマンドを使用します。

get_bd_designs コマンドを使用すると、アクティブ プロジェクトで開いている IP サブシステム デザインのリストが返されます。

IP インテグレーターのすべての Tcl コマンドをリストするには、Vivado Design Suite Tcl シェルで次のコマンドを実行します。

```
load_features IPIntegrator
help -category IPIntegrator
```

注記： **load_features** コマンドは、IP インテグレーター機能が現在 Vivado Design Suite に読み込まれていない場合にのみ必要です。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

design (オプション) : IP インテグレーターで現在のデザインとして設定する IP サブシステム デザインの名前を指定します。*design* を指定しない場合、アクティブ プロジェクトの現在の IP サブシステム デザインが返されます。

例

次の例では、IP サブシステム デザインを現在のデザインに設定しています。

```
current_db_design design_1
```

関連項目

- [current_bd_instance](#)
- [get_bd_designs](#)
- [open_bd_design](#)

current_bd_instance

現在のセル インスタンスを設定または取得します。

構文

```
current_bd_instance [-quiet] [-verbose] [instance]
```

戻り値

現在のセル インスタンス オブジェクト、エラーが発生した場合は ""

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[instance]	設定する現在のセル インスタンスの名前を指定します。

カテゴリ

IP インテグレーター

説明

current_bd_instance コマンドで定義された IP インテグレーター サブシステム デザインの、現在の階層セル インスタンスを設定または取得します。現在のインスタンスは、サブシステム デザイン階層の最上位を基準としています。

このコマンドが正常に実行された場合は現在の IP インテグレーター セル インスタンス オブジェクトが返され、正常に実行されなかった場合は何も返されません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

instance (オプション) : サブシステム デザインの現在のインスタンスとして設定する IP インテグレーター階層セルの名前を指定します。

- ・ *instance* は、現在のインスタンスとして定義されているインスタンスを基準に、階層区切り文字を使用して指定します。
- ・ 現在のインスタンスに対して 1 つ上の階層に移動するには、「..」を使用します。
- ・ *instance* を指定しない場合、現在のインスタンスがサブシステム デザイン階層の最上位モジュールにリセットされます。
- ・ *instance* に「.」を指定した場合、現在のインスタンスの名前が返され、インスタンスは変更されません。

例

次の例では、サブシステム デザインの現在のインスタンスを指定のモジュールに設定しています。

```
current_bd_instance module2
```

次の例では、現在のインスタンスが返されます。

```
current_bd_instance .
```

次の例では、サブシステム デザインの現在のインスタンスを階層の最上位にリセットしています。

```
current_instance  
/
```

関連項目

[current_bd_design](#)

current_board

現在のボード オブジェクトを取得します。

構文

```
current_board [-quiet] [-verbose]
```

戻り値

現在のボード オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

オブジェクト、プロジェクト、ボード

説明

現在のプロジェクトで使用されているボードを返します。

Vivado Design Suite インストール エリアの `data/boards` フォルダーにあるボード ファイル `board.xml` には、ボード属性に関する情報が保存されています。ボードは、ザイリンクス デバイスを含むシステム全体を表しており、クロック制約、I/O ポート割り当て、サポートされるインターフェイスなど、FPGA デザインの主要な部分を定義するのに役立ちます。

ボードは、プロジェクトを定義する際にターゲットリファレンス デザインの一部として指定するか、下の例に示すように現在のプロジェクトの BOARD プロパティを設定することにより指定するか、Vivado IDE の [Project Settings] ダイアログ ボックスで [Project device] を選択することにより指定します。プロジェクト設定の詳細は、『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。

重要： `set_property` コマンドでボードを指定すると、ターゲット デバイスも BOARD プロパティプロパティに合わせて変更されます。

`current_board` コマンドを実行すると、現在のボードの `Vendor:Library:Name:Version` 属性が返されます。プロジェクトで TRD とボードではなくザイリンクス FPGA デバイスがターゲットに指定されている場合、BOARD プロパティが定義されていない場合は、何も返されません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のプロジェクトの BOARD プロパティを設定し、プロジェクトで使用されるボードをレポートしています。

```
set_property board xilinx.com:kintex7:kc705:1.0 [current_project]
current_board
  xilinx.com:kintex7:kc705:1.0
```

次の例では、BOARD プロパティを設定した結果、ターゲットデバイスが変更されたことを示しています。ターゲットデバイスは自動的に変更され、警告メッセージが表示されます。

```
set_property board xilinx.com:artix7:ac701:1.0 [current_project]
WARNING: [Project 1-153] The current project part 'xc7k325tffg900-2' does
not match with the 'XILINX.COM:ARTIX7:AC701:1.0' board part settings. The
project part will be reset to 'XILINX.COM:ARTIX7:AC701:1.0' board part.
INFO: [Project 1-152] Project part set to artix7 (xc7a200tfg676-2)
```

注記 : **report_property** コマンドを使用すると、**current_project** の BOARD および PART プロパティを確認できます。

関連項目

- [current_project](#)
- [get_board_interfaces](#)
- [get_board_pins](#)
- [get_boards](#)
- [report_property](#)
- [set_property](#)

current_design

現在のデザインを設定または取得します。

構文

```
current_design [-quiet] [-verbose] [design]
```

戻り値

デザイン オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[design]	設定する現在のデザインの名前を指定します。

カテゴリ

SDC、XDC

説明

現在のデザインを定義するか、アクティブ プロジェクトの現在のデザインの名前を返します。

ほとんどの Tcl コマンド、ツールで実行したデザインのおよび制約の変更では、現在のデザインおよび現在のインスタンスがターゲットとなります。現在のインスタンスを指定するには、**current_instance** コマンドを使用します。

get_designs コマンドを使用するとアクティブ プロジェクトで開いているデザインのリストを取得でき、**get_projects** コマンドを使用すると開いているプロジェクトのリストを取得できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

design (オプション) : 現在のデザインとして設定するデザインの名前を指定します。design を指定しない場合、アクティブ プロジェクトの現在のデザインが返されます。

例

次の例では、デザイン rtl_1 を現在のデザインに設定しています。

```
current_design rtl_1
```

関連項目

- [current_instance](#)
- [get_designs](#)
- [get_projects](#)

current_fileset

現在のファイルセットを取得するか (任意のタイプ)、現在のファイルセットを設定します (シミュレーション ファイルセットのみ)。

構文

```
current_fileset [-constrset] [-simset] [-quiet] [-verbose] [fileset...]
```

戻り値

現在のファイルセット (デフォルトでは現在のソース ファイルセット)

使用法

名前	説明
[-constrset]	現在の制約ファイルセットを取得します。
[-simset]	現在アクティブなシミュレーション ファイルセットを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[fileset]	現在の (アクティブな) シミュレーション ファイルセットを指定します (オプション)。

カテゴリ

プロジェクト

説明

現在のプロジェクトでアクティブなソース、制約、またはシミュレーション ファイルセットの名前を取得します。

オプションを指定せずに使用すると、sources_1 がアクティブ ファイルセットとして指定され、返されます。

このコマンドは、現在のシミュレーション ファイルセットを設定するのにも使用できます。

注記： 合成 run またはインプリメンテーション run でのアクティブ制約セットを定義するには、**set_property CONSTRSET** を使用します。

引数

-constrset (オプション)： 現在アクティブな制約セットを返します。

-simset (オプション)： 現在アクティブなシミュレーション ファイルセットを設定または返します。

-quiet (オプション)： コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記： コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

fileset (オプション) : アクティブにするシミュレーション ファイルセットの名前を指定します。この引数は、複数のファイルセットを含むプロジェクトでアクティブなシミュレーション ファイルセットを設定します。*fileset* を指定しない場合、sources_1 ファイルセットがアクティブなファイルセットとして返されます。

例

次の例では、現在アクティブな制約ファイルセットの名前が返されます。

```
current_fileset -constrset
```

次の例では、*sim_2* をアクティブなシミュレーション セットとして設定しています。

```
current_fileset -simset sim_2
```

関連項目

- [create_fileset](#)
- [delete_fileset](#)
- [get_filesets](#)

current_hw_device

現在のハードウェア デバイスを取得または設定します。

構文

```
current_hw_device [-quiet] [-verbose] [hw_device]
```

戻り値

ハードウェア デバイス

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_device]	現在のハードウェア デバイスとして設定するハードウェア デバイスを指定します。

カテゴリ

ハードウェア

current_hw_ilab

現在のハードウェア ILA を取得または設定します。

構文

```
current_hw_ilab [-quiet] [-verbose] [hw_ilab]
```

戻り値

ハードウェア ILA

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_ilab]	ハードウェア ILA を指定します。

カテゴリ

ハードウェア

current_hw_ilab_data

現在のハードウェア ILA データを取得または設定します。

構文

```
current_hw_ilab_data [-quiet] [-verbose] [hw_ilab_data]
```

戻り値

ハードウェア ILA データ

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_ilab_data]	ハードウェア ILA データを指定します。

カテゴリ

ハードウェア

current_hw_server

現在のハードウェア サーバーを取得または設定します。

構文

```
current_hw_server [-quiet] [-verbose] [hw_server]
```

戻り値

ハードウェア サーバー

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_server]	ハードウェア サーバーを指定します。

カテゴリ

ハードウェア

current_hw_target

現在のハードウェア ターゲットを取得または設定します。

構文

```
current_hw_target [-quiet] [-verbose] [hw_target]
```

戻り値

ハードウェア ターゲット

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_target]	ハードウェア ターゲットを指定します。

カテゴリ

ハードウェア

current_instance

現在のインスタンスを設定または取得します。

構文

```
current_instance [-quiet] [-verbose] [instance]
```

戻り値

インスタンス名

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[instance]	インスタンス名を指定します。

カテゴリ

SDC、XDC

説明

デザイン階層の現在のインスタンスを指定のインスタンス セルまたは最上位モジュールに設定します。デフォルトでは、**current_instance** は **current_design** の最上位モジュールを指定します。これは、インスタンシエートされたセル オブジェクトではありません。**current_instance** をデザインでのインスタンシエートされたセル オブジェクトに設定することも可能です。

最上位モジュールはインスタンシエートされたオブジェクトではないので、このコマンドを実行すると、オブジェクトではなく現在のインスタンスの名前が返されます。

ほとんどのコマンドおよびデザインの変更では、現在のデザインおよび現在のインスタンスがターゲットとなります。現在のデザインを指定するには、**current_design** コマンドを使用します。

instance は現在定義されている現在のインスタンスを基準に指定し、階層区切り文字を使用してインスタンス パスを定義します。現在の階層区切り文字を確認するには、**get_hierarchy_separator** コマンドを使用します。

インスタンス パスの 1 つ上の階層に移動するには、「..」を使用します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記: メッセージの制限は、**set_msg_config** コマンドで定義できます。

instance (オプション) : 現在のデザインの現在のインスタンスとして設定するインスタンスの名前を指定します。

- *instance* は、現在のインスタンスとして定義されているインスタンスを基準に、階層区切り文字を使用して指定します。
- 現在のインスタンスに対して 1 つ上の階層に移動するには、「..」を使用します。
- *instance* を指定しない場合、現在のインスタンスがデザイン階層の最上位モジュールにリセットされます。
- *instance* に「.」を指定した場合、現在のインスタンスの名前が返され、インスタンスは変更されません。

例

次の例では、現在のインスタンスを現在のデザインの最上位モジュールに設定します。

```
current_instance
  INFO: [Vivado 12-618] Current instance is the top level of design 'netlist_1'.
  top
```

次の例では、まず階層区切り文字を設定し、その後現在定義されている現在のインスタンスを基準に現在のインスタンスを設定します。

```
set_hierarchy_separator |
current_instance ..|cpu_iwb_dat_o|buffer_fifo
```

次の例では、現在定義されている現在のインスタンスの名前が返されます。

```
current_instance .
cpuEngine|cpu_iwb_dat_o|buffer_fifo
```

関連項目

- [current_design](#)
- [get_hierarchy_separator](#)
- [set_hierarchy_separator](#)

current_project

現在のプロジェクトを設定または取得します。

構文

```
current_project [-quiet] [-verbose] [project]
```

戻り値

現在のまたは新しく設定されたプロジェクト オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[project]	現在のプロジェクトとして設定するプロジェクトの名前を指定します。

カテゴリ

プロジェクト

説明

現在のプロジェクトを指定するか、プロジェクトを指定しない場合は現在のプロジェクトの名前を返します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

project (オプション) : 現在のプロジェクトとして設定するプロジェクトの名前を指定します。このコマンドを **close_project** コマンドの前に使用し、特定のプロジェクトをアクティブにしてから、プロジェクトを閉じます。

例

次の例では、project_2 を現在のプロジェクトとして設定しています。

```
current_project project_2
```

このコマンドにより、現在のプロジェクトがすべてのツールコマンドの対象となります。GUI モードでは、プロジェクト間の GUI を切り替えると現在のプロジェクトが自動的に定義されます。

次の例では、現在のプロジェクト名が返されます。

```
current_project
```

注記： 戻り値は、プロジェクトファイルの名前やパスではなく、プロジェクトの名前です。

関連項目

- [close_project](#)
- [current_design](#)

current_run

現在の run を設定または取得します。

構文

```
current_run [-synthesis] [-implementation] [-quiet] [-verbose] [run]
```

戻り値

run オブジェクト

使用法

名前	説明
[-synthesis]	現在の合成 run を設定または取得します。
[-implementation]	現在のインプリメンテーション run を設定または取得します。 -synthesis を指定しない場合、これがデフォルト設定です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[run]	現在の run として設定する run を指定します。

カテゴリ

プロジェクト

説明

現在の合成またはインプリメンテーション run を定義するか、現在の run の名前を返します。現在の run とは、合成またはインプリメントコマンドを実行したときに自動的に選択される run のことです。

`get_runs` コマンドを使用すると、現在のデザインで定義されている run のリストを取得できます。

引数

-synthesis (オプション) : 現在の合成 run を設定または返します。

-implementation (オプション) : 現在のインプリメンテーション run を設定または返します。これが、**-synthesis** または **-implementation** が指定されていない場合のデフォルトです。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

run (オプション) : 現在の run として設定する合成 run またはインプリメンテーション run の名前を指定します。

例

次の例では、synth_1 という run を現在の run として定義しています。

```
current_run synth_1
```

注記 : この場合、run の名前が指定されており、ツールで特定の run 名が認識されるので、**-synthesis** および **-implementation** オプションは必要ありません。

次のコマンドでは、現在のインプリメンテーション run の名前が返されます。

```
current_run -implementation -quiet
```

関連項目

- [create_run](#)
- [get_runs](#)
- [launch_runs](#)

current_scope

現在のスコープを取得するか、現在のスコープを設定します。

構文

```
current_scope [-quiet] [-verbose] [hdl_scope]
```

戻り値

現在のスコープ

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hdl_scope]	HDL スコープを指定します。デフォルトは NULL です。

カテゴリ

シミュレーション

current_sim

現在のシミュレーション オブジェクトを設定するか、現在のシミュレーション オブジェクトを取得します。

構文

```
current_sim [-quiet] [-verbose] [simulationObject]
```

戻り値

現在のシミュレーション オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[simulationObject]	現在のシミュレーション オブジェクトに設定するシミュレーション オブジェクトを指定します。デフォルトは NULL です。

カテゴリ

シミュレーション

説明

現在の Vivado シミュレーションの名前を返します。

このコマンドは、Vivado シミュレータを起動した後に使用できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のシミュレーションの名前が返されます。

```
current_sim
```

関連項目

- [close_sim](#)
- [launch_xsim](#)

current_time

現在のシミュレーション時間をレポートします。

構文

```
current_time [-quiet] [-verbose]
```

戻り値

現在のシミュレーション時間

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

シミュレーション

current_vcd

現在の VCD オブジェクトを設定するか、現在の VCD オブジェクトを取得します。

構文

```
current_vcd [-quiet] [-verbose] [VCDObject]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[VCDObject]	VCD オブジェクトを指定します。デフォルトは NULL です。

カテゴリ

シミュレーション

説明

現在の VCD (Value Change Dump) を定義するか、現在のシミュレーションにおける現在の VCD オブジェクトの名前を返します。

現在の VCD オブジェクトが存在するようにするためには、**open_vcd** コマンドを使用して VCD ファイルを開き、VCD オブジェクトに割り当てる必要があります。

このコマンドを実行すると、現在の VCD オブジェクトが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

VCDObject (オプション) : 現在のオブジェクトとして設定する VCD オブジェクトの名前を指定します。VCDObject を指定しない場合、アクティブ シミュレーションの現在の VCD オブジェクトが返されます。

例

次の例では、指定した VCD オブジェクトを現在の VCD オブジェクトとして設定しています。

```
current_vcd vcd2
```

関連項目

[open_vcd](#)

current_wave_config

現在の波形コンフィギュレーションを取得するか、波形コンフィギュレーションを指定した場合は現在の波形コンフィギュレーションを設定します。

構文

```
current_wave_config [-quiet] [-verbose] [wcfgObj]
```

戻り値

新しいまたは現在の波形コンフィギュレーション オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[wcfgObj]	指定した波形コンフィギュレーションを現在の波形コンフィギュレーションに設定します。デフォルトは、現在の波形コンフィギュレーションです。

カテゴリ

波形

data2mem

データをメモリに変換します。

構文

```
data2mem [-bd arg] [-bm arg] [-bt arg] [-bx arg] [-d arg] [-i] [-o arg]
[-ofmt arg] [-p arg] [-u] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-bd]	入力 ELF または MEM ファイルを指定します。
[-bm]	入力ブロック RAM メモリ マップ (BMM) ファイルを指定します。
[-bt]	入力ビットストリーム (BIT) ファイルを指定します。
[-bx]	HDL シミュレーション用の個々のメモリ デバイス MEM ファイルを出力するパスを指定します。
[-d]	ダンプ モードを指定します。デフォルトは r です。
[-i]	BMM ファイルで定義されたアドレス空間の範囲外の ELF または MEM データを無視します。
[-o]	出力ファイルを指定します。
[-ofmt]	出力フォーマット (.ucf、.v、.vhd、.bit、.dmp) を指定します。
[-p]	ターゲット パーツを指定します。
[-u]	アドレス空間にデータが転送されていない場合でも、すべてのアドレス空間に対して -o で指定した出力テキスト ファイルをアップデートします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

シミュレーション

delete_bd_objs

指定のオブジェクト削除します。

構文

```
delete_bd_objs [-quiet] [-verbose] objects...
```

戻り値

オブジェクトが正しく削除された場合は Pass

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	削除するオブジェクトを指定します。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインから指定のオブジェクトを削除します。

オブジェクトは **delete_bd_objs** コマンドに直接渡す必要があります。オブジェクト名で指定することはできません。たとえばピンを指定する場合、ピン名ではなく **get_bd_pins** を使用します。

このコマンドが正常に実行された場合は 0 が返され、正常に実行されなかった場合はエラーが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects : 現在の IP インテグレーター サブシステム デザインから削除するオブジェクトのリストを指定します。

例

次の例では、現在のサブシステム デザインからさまざまなオブジェクトを削除しています。

```
delete_bd_objs [get_bd_nets /Net] [get_bd_nets /vidout_1_vtg_ce] \
[get_bd_nets /newMod1/aclk_1] [get_bd_ports /addr] [get_bd_cells /vidOut_1]
```

次の例では同じオブジェクトを削除していますが、複数の delete_bd_objs コマンドを使用し、オブジェクトをタイプ別にグループ化して削除するオブジェクトを明確にしています。

```
delete_bd_objs [get_bd_nets /Net] [get_bd_nets /vidout_1_vtg_ce] \
[get_bd_nets /newMod1/aclk_1]
delete_bd_objs [get_bd_ports /addr]
delete_bd_objs [get_bd_cells /vidOut_1]
```

関連項目

- [get_bd_addr_segs](#)
- [get_bd_addr_spaces](#)

delete_clock_networks_results

メモリからクロック ネットワーク結果のセットを消去します。

構文

```
delete_clock_networks_results [-quiet] [-verbose] name
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	消去する結果の名前を指定します。

カテゴリ

レポート

delete_debug_core

デバッグ コアを削除します。

構文

```
delete_debug_core [-quiet] [-verbose] cores...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>cores</i>	削除するデバッグ コアを指定します。

カテゴリ

説明

現在のプロジェクトから **create_debug_core** コマンドで追加された Vivado ラボ ツール デバッグ コアを削除します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

cores (必須) : 現在のプロジェクトから削除するデバッグ コアを 1 つ以上指定します。

例

次の例では、現在のプロジェクトから myCore デバッグ コアを削除しています。

```
delete_debug_core myCore
```

次の例では、現在のプロジェクトからすべてのデバッグ コアを削除しています。

```
delete_debug_core [get_debug_cores]
```

注記: `get_debug_cores` コマンドを実行すると、すべてのコアがデフォルトで返されます。

関連項目

- [create_debug_core](#)
- [get_debug_cores](#)

delete_debug_port

デバッグ ポートを削除します。

構文

```
delete_debug_port [-quiet] [-verbose] ports...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
ports	削除するデバッグ ポートを指定します。

カテゴリ

説明

現在のプロジェクトに含まれる Vivado ラボ ツール デバッグ コアからポートを削除します。デバッグ ポートからの信号は、**disconnect_debug_port** で削除するか、このコマンドでポートと共に削除できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

ports (必須) : コアから削除するデバッグ ポートを core_name/port_name の形式で指定します。

例

次の例では、myCore デバッグ コアから DATA ポートを削除しています。

```
delete_debug_port myCore/DATA
```

注記 : ILA ポートには、最低 1 つの CLK ポートと 1 つの TRIG ポートが必要なので、削除できないポートもあります。

次の例では、myCore デバッグ コアからトリガー ポート (TRIG) が削除されます。

```
delete_debug_port [get_debug_ports myCore/TRIG*]
```

注記：この例の場合、ILA コアには少なくとも 1 つの TRIG ポートが必要なので、myCore からすべての TRIG ポートが削除されるわけではありません。このコマンドでは、最後のポートを除き、TRIG0 からすべての TRIG ポートが削除されます。

関連項目

- [disconnect_debug_port](#)
- [get_debug_ports](#)

delete_drc_check

ユーザー定義 DRC チェックを削除します。

構文

```
delete_drc_check [-quiet] [-verbose] name...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	削除するチェックの名前を指定します。これは通常 PREFIX-id という形式で指定します (PREFIX は 4 ~ 6 文字の略称、id はルールを識別する整数)。正しい名前を確認するには、get_drc_checks を使用します。削除できるのは、ユーザー定義ルールのみです。

カテゴリ

DRC、オブジェクト

説明

現在のプロジェクトからユーザー定義デザイン ルール チェックを削除します。ユーザー定義デザイン ルール チェックは、**create_drc_checks** コマンドを使用して作成します。

注記：ツールであらかじめ定義されているルール チェックは削除できません。

ルール チェックを一度削除すると、復元するできません。undo コマンドは機能しません。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 現在のプロジェクトから削除するユーザー定義デザインルール チェックの名前を指定します。

例

次の例では、指定したデザインルール チェックが削除されます。

```
delete_drc_check LJH-1
```

関連項目

[create_drc_check](#)

delete_drc_ruleddeck

1 つまたは複数のユーザー定義 DRC ルール デック オブジェクトを削除します。

構文

```
delete_drc_ruleddeck [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose]
[patterns]
```

戻り値

DRC ルール デック

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	DRC ルール デック オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

DRC、オブジェクト

説明

現在のプロジェクトから 1 つまたは複数のユーザー定義デザインルール デック オブジェクトを削除します。削除するルール デックは空である必要はありません。一度削除すると、復元するできません。undo コマンドは機能しません。

注記：ツールであらかじめ定義されているルール デックは削除できません。

ルール デックとは、デザインルール チェックのグループで、I/O 配置などの FPGA デザインフローの異なる段階で **report_drc** コマンドにより実行されます。ツールには定義済みのルール デックが含まれていますが、**create_drc_ruleddeck** コマンドを使用して新しいユーザー定義のルール デックを作成できます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、検索パターンにより返されるオブジェクトのリストに、指定したプロパティ値に基づくフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する DRC ルール デックを削除します。デフォルトのパターンはワイルドカード (*) で、現在のプロジェクトのユーザー定義ツール デックがすべて削除されます。複数のパターンを指定して、異なる検索条件に基づいてルール デックを削除できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、現在のプロジェクトからすべてのユーザー定義ルール デックを削除しています。

```
delete_drc_ruledesk
```

関連項目

- [create_drc_ruledesk](#)
- [list_property](#)
- [report_property](#)

delete_fileset

ファイルセットを削除します。

構文

```
delete_fileset [-merge arg] [-quiet] [-verbose] fileset
```

戻り値

なし

使用法

名前	説明
[-merge]	削除するファイルセットからファイルを結合するファイルセットを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
fileset	削除するファイルセットを指定します。

カテゴリ

プロジェクト、シミュレーション

説明

指定したファイルセットを削除します。ただし、ファルセットが削除できない場合でも、それを示すメッセージは表示されません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

fileset (必須) : 削除するファイルセットの名前を指定します。最後の制約ファイルセットまたはシミュレーションファイルセットは削除されません。その場合、エラー メッセージは表示されません。

例

次の例では、sim_2 というファイルセットを現在のプロジェクトから削除しています。

```
delete_fileset sim_2
```

注記：ファイルセットとそのファイルすべてがプロジェクトから削除されます。ファイルは、ハード ドライブからは削除されません。

関連項目

- [create_fileset](#)
- [current_fileset](#)

delete_interface

I/O ポート インターフェイスをプロジェクトから削除します。

構文

```
delete_interface [-all] [-quiet] [-verbose] interfaces...
```

戻り値

なし

使用法

名前	説明
[-all]	インターフェイスに付属するすべてのポートおよびバスも削除します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>interfaces</i>	削除する I/O ポート インターフェイスを指定します。

カテゴリ

ピン配置

説明

既存のインターフェイスおよびオプションでそのインターフェイスに関連付けられているポートおよびバスをすべて削除します。

引数

-all (オプション) : 指定のインターフェイスに関連付けられているポート、バス、およびネストされたインターフェイスをすべて削除します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

interfaces (必須) : 削除するインターフェイスの名前を指定します。

例

次の例では、指定のインターフェイスと、関連付けられているポートおよびバスがすべて削除されます。

```
delete_interface USB0
```

関連項目

- [create_interface](#)
- [create_port](#)

delete_macros

マクロのリストを削除します。

構文

```
delete_macros [-quiet] [-verbose] macros
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
macros	削除するマクロを指定します。

カテゴリ

xdc

説明

マクロを削除します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 削除するマクロの名前を指定します。

例

次の例では、usbMacro1 というマクロを削除しています。

```
delete_macros usbMacro1
```

関連項目

- [remove_cell](#)
- [set_hierarchy_separator](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

delete_pblock

Pblock を削除します。

構文

```
delete_pblock [-hier] [-quiet] [-verbose] pblocks...
```

戻り値

なし

使用法

名前	説明
[-hier]	Pblock の子ブロックすべて削除します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>pblocks</i>	削除する Pblock を指定します。

カテゴリ

フロアプラン、XDC

説明

現在のプロジェクトから指定した Pblock を削除します。Pblock は、**create_pblock** コマンドを使用して作成します。

引数

-hier (オプション) : 指定した Pblock にネストされている Pblock も削除します。**-hier** オプションを指定せずに親 Pblock を削除した場合、ネストされていた Pblock は 1 つ上のレベルに移動されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

pblocks (必須) : 削除する Pblock を 1 つ以上指定します。

例

次の例では、指定した Pblock と、その中にネストされている Pblock が削除されます。

```
delete_pblock -hier cpuEngine
```

関連項目

[create_pblock](#)

delete_power_results

指定した名前の消費電力予測結果をメモリから削除します。

構文

```
delete_power_results -name arg [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
-name	消去する結果の名前を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

電力

説明

特定の結果セットの消費電力解析結果を削除します。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-name arg (必須)：消去する結果の名前を指定します。名前は、ユーザーが定義したものか、**report_power** コマンドを実行したときに自動的に定義されたものです。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、消費電力解析を実行し、その結果を消去しています。

```
report_power -name my_set
delete_power_results -name my_set
```

関連項目

- [power_opt_design](#)
- [report_power](#)
- [reset_switching_activity](#)
- [set_switching_activity](#)

delete_rpm

RPM を削除します。

構文

```
delete_rpm [-quiet] [-verbose] rpm
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
rpm	削除する RPM を指定します。

カテゴリ

フロアプラン

説明

デザインから指定した相対配置マクロ (RPM) を削除します。

RPM とは、複数のロジック エレメント (FFS、LUT、CY4、RAM など) を 1 つのセット (U_SET、H_SET、および HU_SET) にまとめたものです。セット内での各エレメントの配置は、相対ロケーション制約 (RLOC) により同じセット内のほかのエレメントに相対して設定されます。RLOC 制約の設定されたロジック エレメントと共通セット名は、RPM 内で関連付けられます。これらの制約の定義方法は、『制約ガイド』(UG625) を参照してください。

デザインから削除できるのは、ユーザー定義の RPM のみです。階層またはネットリストにより定義された RPM をこのコマンドで削除することはできません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

rpm (必須) : 削除する RPM を指定します。

例

次の例では、デザインから指定した RPM (cs_ila_0/U0) を削除しています。

```
delete_rpm cs_ila_0/U0
```

delete_run

既存の run を削除します。

構文

```
delete_run [-noclean_dir] [-quiet] [-verbose] run
```

戻り値

なし

使用法

名前	説明
[-noclean_dir]	すべての出力ファイルおよびディレクトリをディスクから削除しません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
run	変更する run を指定します。

カテゴリ

プロジェクト

説明

プロジェクトから run を削除し、指定しない場合は、ハードドライブのプロジェクトディレクトリからも run の結果を削除します。

引数

-noclean_dir : ハードドライブから run 結果を削除しないように指定します。run はプロジェクトからは削除されますが、run ファイルはプロジェクトディレクトリに残ったままになります。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

run (必須) : プロジェクトから削除する合成 run またはインプリメンテーション run の名前を指定します。

例

次の例では、first_pass という run をプロジェクトから削除しています。

```
delete_run first_pass
```

注記：この例の場合、run 結果もハード ドライブのプロジェクト ディレクトリから削除されます。

次の例では、first_pass という run は削除されますが、run 結果はハード ドライブに残ります。

```
delete_run -noclean_dir first_pass
```

関連項目

- [create_run](#)
- [current_run](#)

delete_timing_results

メモリからタイミング結果のセットを消去します。

構文

```
delete_timing_results [-type arg] [-quiet] [-verbose] name
```

戻り値

なし

使用法

名前	説明
[-type]	消去するタイミング結果のタイプを指定します。有効な値は、timing_path、slack_histogram、clock_interaction、check_timing、pulse_width、timing_summary です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	消去する結果の名前を指定します。

カテゴリ

レポート、タイミング

説明

特定の結果セットから指定のタイミング結果を消去します。タイミング レポートのタイプとタイミング レポートの名前を指定する必要があります。これら両方を指定しないと、エラーが発生します。

引数

-type arg (オプション) : 消去するタイミング結果のタイプを指定します。指定可能なタイプは次のとおりです。

- timing_path : 指定した **report_timing** レポートを削除します。
- slack_histogram : 指定した **create_slack_histogram** レポートを削除します。
- clock_interaction : 指定した **report_clock_interaction** レポートを削除します。
- check_timing : 指定した **check_timing** レポートを削除します。
- pulse_width : 指定した **report_pulse_width** レポートを削除します。
- timing_summary : 指定した **report_timing_summary** レポートを削除します。

注記 : デフォルトは timing_path で、**report_timing** コマンドで生成されたレポートが削除されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

-name *arg* (必須) : 消去するタイミング結果の名前を指定します。

例

次の例では、メモリから指定した結果を消去しています。

```
delete_timing_results -type clock_interaction -name clkNets
```

関連項目

- [check_timing](#)
- [create_slack_histogram](#)
- [report_clock_interaction](#)
- [report_pulse_width](#)
- [report_timing](#)
- [report_timing_summary](#)

delete_utilization_results

指定した名前の使用率結果をメモリから削除します。

構文

```
delete_utilization_results -name arg [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
-name	消去する結果の名前を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

指定の結果セットから指定の使用率結果を消去します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

-name arg (必須) : 消去する結果の名前を指定します。

例

次の例では、メモリから指定した結果を消去しています。

```
delete_utilization_results -name SSO1
```

関連項目

[report_utilization](#)

describe

HDL オブジェクト (変数、信号、ワイヤ、またはレジスタ) のタイプおよび宣言情報を表示します。

構文

```
describe [-quiet] [-verbose] hdl_object
```

戻り値

選択したオブジェクトの説明

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hdl_object</i>	説明を表示するHDL オブジェクトまたは HDL スコープを指定します。

カテゴリ

シミュレーション

disconnect_bd_intf_net

インターフェイス ネットの接続を解除します。

構文

```
disconnect_bd_intf_net [-quiet] [-verbose] intf_net objects...
```

戻り値

正常に実行された場合は 0、それ以外の場合はエラー

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>intf_net</i>	オブジェクトが接続されているインターフェイス ネットを指定します。
<i>objects</i>	インターフェイス ネットから接続を解除するオブジェクトを指定します。

カテゴリ

IP インテグレーター

説明

IP インテグレーター サブシステム デザインの 1 つのインターフェイス ネットの接続を指定のオブジェクトから解除します。インターフェイスとは、IP インテグレーター サブシステム デザインで共通の機能を共有する信号をグループ化したものです。

このコマンドでは、IP サブシステム デザインのピンまたはポートから、ネット全体を削除せずに、指定のインターフェイス ネットを解除できます。ネット全体を削除するには、**delete_bd_objs** コマンドを使用します。

このコマンドが正常に実行された場合は 0 が返され、正常に実行されなかった場合はエラーが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

intf_net (必須) : 指定のオブジェクトから接続を解除する IP サブシステム デザインの 1 つのインターフェイス ネットを指定します。ネットは、名前で指定するか、または **get_bd_intf_nets** コマンドで 1 つのオブジェクトを返すことにより指定します。

objects (必須) : ネットの接続を解除するインターフェイス ピンまたはポート オブジェクトのリストを指定します。インターフェイス ピンおよびポートは、**get_bd_intf_pins** または **get_bd_intf_ports** コマンドで返すことにより指定する必要があります。名前で指定することはできません。

例

次の例では、IP サブシステム デザインの唯一のインターフェイス ネットの接続を、接続されているすべてのインターフェイス ピンおよびポート オブジェクトから解除します。

```
disconnect_bd_intf_net [get_bd_intf_nets] \
[get_bd_intf_pins -of_objects [get_bd_intf_nets]] \
[get_bd_intf_ports -of_objects [get_bd_intf_nets]]
```

注記 : この例では、デザインにはインターフェイス ネットが 1 つしかありません。複数のインターフェイス ネットが指定されると、**disconnect_bd_intf_net** コマンドからエラーが返されます。

関連項目

- [connect_bd_intf_net](#)
- [connect_bd_net](#)
- [delete_bd_objs](#)
- [disconnect_bd_intf_net](#)
- [get_bd_nets](#)
- [get_bd_pins](#)
- [get_bd_ports](#)

disconnect_bd_net

オブジェクトからネットの接続を解除します。

構文

```
disconnect_bd_net [-quiet] [-verbose] net objects...
```

戻り値

正常に実行された場合は 0、それ以外の場合はエラー

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
net	オブジェクトが接続されているネットを指定します。
objects	ネットから接続を解除するオブジェクトを指定します。

カテゴリ

IP インテグレーター

説明

IP インテグレーター サブシステム デザインの 1 つのネットの接続を指定のオブジェクトから解除します。

このコマンドでは、IP サブシステム デザインのピンまたはポートから、ネット全体を削除せずに、指定のネットを解除できます。ネット全体を削除するには、**delete_bd_objs** コマンドを使用します。

このコマンドが正常に実行された場合は 0 が返され、正常に実行されなかった場合はエラーが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

net (必須) : 指定のオブジェクトから接続を解除する IP サブシステム デザインの 1 つのネットを指定します。ネットは、名前で指定するか、または **get_bd_nets** コマンドで 1 つのオブジェクトを返すことにより指定します。

objects (必須) : ネットの接続を解除するピンまたはポート オブジェクトのリストを指定します。ピンおよびポートは、**get_bd_pins** または **get_bd_ports** コマンドで返すことにより指定する必要があります。名前で指定することはできません。

例

次の例では、名前で指定したネットの接続を指定したピン オブジェクトから解除しています。

```
disconnect_bd_net /vidout1_locked [get_bd_pins {vidOut1/locked newMod1/t1}]
```

関連項目

- [connect_bd_intf_net](#)
- [connect_bd_net](#)
- [delete_bd_objs](#)
- [disconnect_bd_intf_net](#)
- [get_bd_nets](#)
- [get_bd_pins](#)
- [get_bd_ports](#)

disconnect_debug_port

デバッグ ポート チャネルからネットとピンの接続を解除します。

構文

```
disconnect_debug_port [-channel_index arg] [-quiet] [-verbose] port
```

戻り値

なし

使用法

名前	説明
[-channel_index]	ネット接続を解除するチャネル インデックスを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
port	デバッグ ポート名を指定します。

カテゴリ

説明

デバッグ ポートから信号の接続を解除します。

ネットリスト デザインからの信号は、**connect_debug_port** コマンドを使用して ILA デバッグ コアのポートに接続されます。

ポートは単に接続を解除するだけでなく、**delete_debug_port** コマンドを使用してデバッグ コアから削除することもできます。

デバッグ コアのポート名を取得する必要がある場合は、**get_debug_ports** コマンドを使用するとコアのポートすべてをリストできます。プロジェクト内のコアと特定のパラメーターをすべてリストするには、**report_debug_core** コマンドを使用します。

引数

-channel_index value (オプション) : 接続を解除するポートのチャネル インデックスを指定します。

注記 : このオプションを使用しない場合、ポート全体の接続が解除されます。

port (必須) : 接続を解除するデバッグ コアのポート名を指定します。ポート名は、core_name/port_name という形式で指定する必要があります。例を参照してください。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、myCore の PROBE1 ポートから指定したチャネル インデックスのみの接続を解除しています。

```
disconnect_debug_port -channel_index 2 myCore/PROBE1
```

次の例のように -channel_index を指定しない場合、指定したポートのすべてのチャネルの接続が解除されます。

```
disconnect_debug_port myCore/PROBE1
```

関連項目

- [connect_debug_port](#)
- [delete_debug_port](#)
- [get_debug_ports](#)
- [report_debug_core](#)

disconnect_hw_server

ハードウェア サーバーへの接続を閉じます。

構文

```
disconnect_hw_server [-quiet] [-verbose] [hw_server]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_server]	ハードウェア サーバーを指定します。デフォルトは、現在のハードウェア サーバーです。

カテゴリ

ハードウェア

disconnect_net

ピンまたはポートからネットの接続を解除します。

構文

```
disconnect_net [-prune] -net arg -objects args [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-prune]	disconnect_net の実行により未接続となったピンおよびポートを削除します。
-net	接続を解除するネットを指定します。
-objects	接続を解除するピンまたはポートのリストを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ネットリスト

説明

開いている合成済みデザインまたはインプリメント済みデザインのネットリストに含まれる 1 つ以上のピンまたはポートから、指定のネットの接続を解除します。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソース ファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザイン チェックポイントとして保存するか、**write_*** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリスト ファイルにエクスポートできます。

注記： エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

引数

-prune (オプション)： 指定のネットの接続を解除したことにより未接続となったピンおよびポートを削除します。

-net arg (必須)： 接続を解除するネットを指定します。

注記： **create_net** コマンドの **-bus_from** および **-bus_to** オプションを使用してバスを作成できますが、バスの各ビットを **disconnect_net** コマンドを使用してそれぞれ接続解除する必要があります。

-objects args (必須)： ネットの接続を解除するピンまたはポート オブジェクトのリストを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、dataBus の指定したビットの接続を解除しています。

```
disconnect_net -net dataBus[1] -objects {dataIN[1] myDMA/data[1]}
```

関連項目

- [connect_net](#)
- [remove_net](#)
- [create_net](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

display_hw_ilab_data

ハードウェア ILA データをビューアーに表示します。

構文

```
display_hw_ilab_data [-wcfg arg] [-reset] [-quiet] [-verbose] [hw_ilab_data ...]
```

戻り値

なし

使用法

名前	説明
[-wcfg]	代替波形コンフィギュレーション ファイルを使用します。
[-reset]	波形コンフィギュレーション ファイルをデフォルトのコンフィギュレーションにリセットします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_ilab_data]	ハードウェア ILA データ オブジェクトのリストを指定します。デフォルトは、現在のハードウェア ILA データです。

カテゴリ

ハードウェア

display_hw_sio_scan

既存のハードウェア SIO スキャンを表示します。

構文

```
display_hw_sio_scan [-quiet] [-verbose] hw_sio_scans
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hw_sio_scans</i>	ハードウェア SIO スキャンを指定します。

カテゴリ

ハードウェア

endgroup

グループ単位で実行を取り消し/やり直しできるコマンド シーケンスを終了します。

構文

endgroup [-quiet] [-verbose]

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

GUI 制御

説明

グループ単位で実行を取り消し/やり直しできるコマンド シーケンスを終了します。コマンド シーケンスを開始するには、**startgroup** コマンドを使用します。

注記 : **undo** または **redo** で一度に取り消し/やり直しできるコマンド グループは複数作成できますが、コマンド グループをネストさせることはできません。**startgroup** を使用して新しいコマンド シーケンスを作成する前に、**endgroup** を使用してコマンド シーケンスを終了する必要があります。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、まず startgroup を実行し、関連するコマンドのシーケンスを実行して、endgroup を実行します。このコマンド シーケンスは、グループ単位で実行を取り消すことができます。

```
startgroup
create_pblock pblock_wbArbEngine
create_pblock pblock_usbEng
add_cells_to_pblock pblock_wbArbEngine [get_cells [list wbArbEngine]] -clear_locs
add_cells_to_pblock pblock_usbEng [get_cells [list usbEngine1/usbEngineSRAM]] -clear_locs
endgroup
```

関連項目

- [startgroup](#)
- [redo](#)
- [undo](#)

export_hardware

SDK 用にシステム ハードウェア プラットフォームをエクスポートします。

構文

```
export_hardware [-bitstream] [-dir arg] [-quiet] [-verbose] files [run]
```

戻り値

なし

使用法

名前	説明
[-bitstream]	ビットストリーム データを SDK のエクスポート ディレクトリにエクスポートします。
[-dir]	エクスポート ディレクトリを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
files	ハードウェア データをエクスポートするソース ファイルを指定します。
[run]	現在のインプリメンテーション run を指定します。

カテゴリ

XPS

説明

ソフトウェア開発に使用するハードウェアをエクスポートします。

エンベデッド プロセッサ システム ハードウェア プラットフォームを SDK にエクスポートします。SDK では、プロジェクトに含まれるエンベデッド プロセッサ ソースのソフトウェア設計がサポートされます。SDK にエクスポートするエンベデッド プロセッサ XPS プロジェクト ファイルを指定します。

デフォルトでは、指定したエンベデッド プロセッサのハードウェア仕様ファイル (.xml) が、*project_name.sdk/SDK/SDK_Export/hw* ディレクトリにデザインのエンベデッド プロセッサに基づくファイル名で保存されます。

これは、プロジェクト デザイン ソースのエンベデッド プロセッサ ディレクトリ *project_name.srcs/sources_1/edk/robot/_xps/* にある *system.xml* ファイルのコピーです。*system.xml* ファイルには、エンベデッド プロセッサの記述および XPS デザインのコンポーネントが含まれます。

export_hardware コマンドの出力を保存するディレクトリは、**-dir** オプションで指定できます。この出力を使用して、**launch_sdk** コマンドで SDK を起動できます。

このコマンドは、エクスポート プロセスのトランスクriptを返します。

```
export_hardware -bitstream [get_files *.xmp] # Exports hardware design for the XMP source
export_hardware -bitstream [get_files *.bd] # Exports hardware design for the BD source
```

引数

-bitstream (オプション) : ハードウェア仕様ファイルに加え、エンベデッド プロセッサのビットストリームと BMM モデル データをエクスポートします。

注記 : ビットストリーム ファイルが存在しない場合は、エラーが返されます。

-dir arg (オプション) : エクスポート ディレクトリを指定します。デフォルトでは、ハードウェア ファイルはローカル プロジェクト ディレクトリの *project_name.sdk/SDK/SDK_Export/hw* に保存されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

files (必須) : エクスポートする XPS プロジェクト ファイル (.xmp) のリストを含むファイル オブジェクトを指定します。

注記 : ファイル名を指定するのではなく、**get_files** を使用してファイル オブジェクトを指定します。

run (オプション) : エクスポートするインプリメンテーション run を指定します。

例

次の例では、ビットストリームおよび BMM モデル データを含むエンベデッド プロセッサ デザインを、デフォルト の SDK エクスポート ディレクトリにエクスポートしています。

```
export_hardware -bitstream [get_files *.xmp]
```

関連項目

- [create_xps](#)
- [generate_target](#)
- [get_files](#)
- [launch_sdk](#)

filter

リストにフィルターを適用し、新しいリストを作成します。

構文

```
filter [-regexp] [-nocase] [-quiet] [-verbose] [objects] [filter]
```

戻り値

新しいリスト

使用法

名前	説明
[-regexp]	=~ および !~ 演算子で正規表現を使用します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	フィルターを適用するオブジェクトのリストを指定します。
[filter]	式を使用してリストをフィルター処理します。

カテゴリ

オブジェクト、プロパティおよびパラメーター、XDC

説明

オブジェクトのリストから、指定したフィルター検索パターンに一致するオブジェクトのリストを返します。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイレンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

objects (オプション) : フィルターを適用するオブジェクトのリストを指定します。オブジェクトのリストを指定するには、`get_parts` などの `get_*` コマンドのいずれかを使用できます。

filter (オプション) : フィルター処理に使用する検索パターンを指定します。指定したパターンにより、オブジェクトのプロパティ値に基づいたオブジェクトのリストがフィルター処理されます。オブジェクトに設定されているプロパティは、`report_property` または `list_property` コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。part オブジェクトの場合、結果をフィルター処理できるプロパティには DEVICE、FAMILY、SPEED などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!=) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (bool) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

例

次の例では、指定したスピード グレードでパートのリストがフィルター処理されます。

```
filter [get_parts] {speed == -3}
```

次の例では、スピード グレード -3 または -2 でパートのリストがフィルター処理されます。どちらかのスピード グレードに一致するすべてのパートが返されます。

```
filter [get_parts] {speed == -3 || speed == -2}
```

次の例では正規表現を使用し、デザインに含まれる VStatus ポートで、0 文字以上の文字列と角かっこ内に 0 ~ 9 が 1 回以上現れるものを検索しています。

```
filter -regexp [get_ports] {NAME =~ VStatus.*\[0-9]+\]}
```

関連項目

- [get_parts](#)
- [get_ports](#)

find_bd_objs

指定のオブジェクトと指定の関係を持つピン、ポート、またはインターフェイスを検索します。

構文

```
find_bd_objs -relation arg [-boundary_type arg] [-quiet]  
[-verbose] objects...
```

戻り値

ピン、ポート、またはインターフェイス オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
-relation	指定のオブジェクトとの関連を指定します。有効な値は、connected_to、addressable_slave、addressing_master です。connected_to に設定すると、指定のソース オブジェクトに接続されているピン、ポート、またはインターフェイスが階層の境界を超えて検索されます。connected_to_hierip は connected_to に似ていますが、検索は IP にリンクされている階層の境界まで実行されます。
[-boundary_type]	ソース オブジェクトが階層ブロックのピンまたはインターフェイス ピン上にある場合に使用します。有効な値は、upper、lower、both です。lower に設定すると、下位階層から検索が実行されます。このオプションは、-relation が connected_to に設定されている場合のみ有効です。デフォルトは upper です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>objects</i>	検索の元となる 1 つまたは複数のソース オブジェクトを指定します。

カテゴリ

IP インテグレーター

説明

指定のオブジェクトと指定の関係を持つピン、ポート、またはインターフェイスを検索します。

このコマンドが正常に実行された場合はピン、ポート、またはインターフェイス オブジェクトのリストが返され、正常に実行されなかった場合はエラーが返されます。

引数

-relation (必須) : 指定のオブジェクトと検索するオブジェクトとの関係を指定します。有効な値は次のとおりです。

- connected_to : 指定のソース オブジェクトに接続されているピン、ポート、またはインターフェイスを階層の境界を超えて検索します。

注記 : connected_to_hierip : connected_to に似ていますが、IP にリンクされている階層の境界まで検索を実行します。

- addressable_slave : 指定の検索オブジェクトによりスレーブとしてアドレス指定可能なピン、ポート、またはインターフェイスを検索します。
- addressing_master : 指定の検索オブジェクトをマスターとしてアドレス指定するピン、ポート、またはインターフェイスを検索します。

-boundary_type [upper | lower | both] (オプション) : 指定のソース オブジェクトが階層モジュールのピンまたはインターフェイスピンを含む場合に使用します。デフォルトは upper で、関連のオブジェクトが階層モジュールと同じレベルで検索されます。lower に設定すると、検索は階層モジュール内で実行されます。both に設定すると、階層モジュールのレベルと階層モジュール内で関連オブジェクトが検索されます。

注記 : このオプションは、**-relation connected_to** オプションを使用している場合にのみ使用可能です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須) : 関連オブジェクトを検索するのに使用する 1 つまたは複数のオブジェクトを指定します。

例

次の例では、指定したオブジェクトに接続されているピン、ポート、およびインターフェイスのリストを取得しています。

```
find_bd_objs -relation connected_to \
[get_bd_pins /proc_sys_reset_1/peripheral_aresetn]
```

関連項目

- [create_bd_cell](#)
- [create_bd_intf_net](#)
- [create_bd_intf_pin](#)
- [create_bd_intf_port](#)
- [create_bd_net](#)
- [create_bd_pin](#)
- [create_bd_port](#)

find_top

供給されているファイル、ファイルセット、またはアクティブ ファイルセットから最上位モジュールの候補を検索します。ランク付けされた候補のリストを返します。

構文

```
find_top [-fileset arg] [-files args] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-fileset]	最上位モジュールの候補を検索するファイルセットを指定します。
[-files]	最上位モジュールの候補を検索するファイルを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

説明

現在のファイルセットで定義されているファイル、指定したファイルセットで定義されているファイル、または指定したファイルのリストから、最上位モジュールの候補を検索します。

このコマンドの戻り値は、デザインの最上位として最適な候補から順にモジュールをリストします。index コマンドを使用してインデックス 0 を指定すると、最適な候補を最上位モジュールとして選択できます。

引数

-fileset arg (オプション) : 最上位モジュールの候補を検索するシミュレーションまたはソース ファイルセットを指定します。デフォルトでは、現在のデザインの現在のファイルセットが検索されます。

-files arg (オプション) : 最上位モジュールの候補を検索するファイルを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、set_msg_config コマンドで定義できます。

例

次の例では、合成用に現在のデザインの最上位モジュールの最適候補を選択しています。

```
synth_design -top [lindex [find_top] 0]
```

注記: **find_top** では複数の候補が返されますが、インデックス 0 を指定すると合成に最適な候補が選択されます。

次の例では、指定のファイルのリストから最上位モジュールの最適な候補が返されます。

```
find_top -files [get_files -filter {NAME =~ *or1200*}]
```

次の例では、**find_top** コマンドの結果を **\$topVar** に設定し、**set_property** コマンドでこの変数を使用して現在のファイルセットの差上位モジュールを定義しています。

```
set topVar [ find_top -files [get_files -filter {NAME =~ *usbf*} ] ]
usbf_top
set_property top $topVar [current_fileset]
```

関連項目

- [set_property](#)
- [synth_design](#)

flush_vcd

VCD シミュレーション出力を VCD 出力ファイルに保存します (\$dumpflush Verilog システム タスクと同等)。

構文

flush_vcd [-quiet] [-verbose]

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

[シミュレーション](#)

説明

現在メモリにある HDL 信号情報を VCD (Value Change Dump) ファイルに記述します。

VCD は、ヘッダー情報、変数定義、HDL 信号値の変更の詳細を含む ASCII 形式のファイルです。VCD ファイルを使用すると、VCD ビューアーでシミュレーション結果を表示したり、デザインの消費電力予測を実行したりできます。

注記： **flush_vcd** コマンドを使用する前に、**open_vcd** コマンドを実行して VCD ファイルを開く必要があります。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、VCD バッファーの内容を現在の VCD ファイルに記述しています。

flush_vcd

関連項目

[open_vcd](#)

generate_target

指定したソースのターゲット データを生成します。

構文

```
generate_target [-force] [-quiet] [-verbose] name objects
```

戻り値

なし

使用法

名前	説明
[-force]	ターゲット データを強制的に再生成します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	生成するターゲットのリストを指定します。all を指定すると、サポートされるすべてのターゲットが生成されます。
<i>objects</i>	データを生成するオブジェクトを指定します。

カテゴリ

プロジェクト、XPS、IP フロー、IP インテグレーター

説明

IP コア (.xci および .xco)、DSP モジュール (.mdl)、またはエンベデッド プロセッサ サブデザイン (.xmp) の指定したソース ファイルのターゲット データを生成します。生成されるターゲット データは、FPGA デザイン フローで IP コア、DSP モジュール、またはエンベデッド プロセッサをサポートするのに必要なファイルです。

IP コアでは、instantiation_template (インスタンシエーション テンプレート)、synthesis (合成)、および simulation (シミュレーション) が標準的なターゲットです。IP によっては、testbench (テストベンチ)、example (サンプル デザイン)、miscellaneous (その他) などのターゲットをサポートするものもあります。

レガシ IP では、instantiation_template および synthesis ターゲットのみがサポートされます。Vivado に含まれる ネイティブ IP では、すべてのターゲットがサポートされない IP コアもありますが、通常すべてのターゲットがサポートされています。

DSP モジュールおよびエンベデッド プロセッサ サブデザインでは、synthesis (合成)、simulation (シミュレーション)、および implementation (インプリメンテーション) が標準的なターゲットです。

list_targets コマンドを使用すると、特定のソース ファイルでサポートされるターゲットを確認できます。

引数

-force (オプション) : ターゲット データの再生成を強制し、既存のターゲット データ ファイルを上書きします。
-force を指定しない場合、最新のターゲット データは再生成されません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、
TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが
非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 指定のソース用に作成するターゲット データのタイプの名前を指定します。IP コアでサポートされるターゲットは、オブジェクトの SUPPORTED_TARGETS プロパティにリストされています。このプロパティをクエリすると、コアでサポートされているターゲットを確認できます。標準的な値は次のとおりです。

- **all** : 指定のコアのすべてのターゲットを生成します。
- **instantiation_template** : IP コアの RTL モジュール定義を現在のデザインに追加するためのインスタンシエーション テンプレートを生成します。インスタンシエーション テンプレートは、デザイン階層の任意のレベルにコピーできます。
- **synthesis** : ネイティブ IP の合成中に使用される HDL ファイルまたは XST で生成された合成済みネットリスト ファイル (NGC) を生成します。
- **simulation** : シミュレーションで使用される HDL ファイルを生成します。
- **implementation** : 現在のデザインの IP コア、DSP モジュール、またはエンベデッド プロセッサをインプリメントするのに必要なデータを生成します。
- **example** : 一部のネイティブ IP コアでは、コアを含むサンプル プロジェクトを開くことができます。コアを開く前に、**open_example_project** コマンドを使用してサンプル ターゲット データを生成しておく必要があります。
- **testbench** : IP のシミュレーションに使用できるテストベンチを生成します。
- **miscellaneous** 一部の IP で、資料やスクリプトを提供するために使用します。

objects (必須) : ターゲットの生成元のオブジェクトを指定します。サポートされるオブジェクトは、IP コア オブジェクト、IP ソース ファイル (XCI または XCO)、System Generator からインポートされた DSP モジュール (MDL)、Xilinx Platform Studio (XPS) からインポートされたエンベデッド プロセッサ (XMP) などです。

注記 : ファイル名を指定するのではなく、**get_files** を使用してファイル オブジェクトを指定します。

例

次の例では、現在のプロジェクトに含まれるすべての IP コアに対してインプリメンテーション テンプレートを生成しています。ターゲット データが最新の場合でも強制的に再生成します。

```
generate_target instantiation_template [get_ips] -force
```

次の例では、指定の IP コア (XCI ファイル) の合成およびシミュレーションに必要なデータ ファイルを生成しています。

```
generate_target {Synthesis Simulation} \
[get_files C:/data/Projects/test_ip/test_addr_ip/test_addr_ip.xci \
-of_objects [get_filesets sources_1]]
```

次の例では、指定の IP オブジェクトの SUPPORTED_TARGETS プロパティをクエリし、example ターゲット データを生成します。

```
report_property -all [get_ips blk_mem*]
generate_target {example} [get_ips blk_mem*]
```

関連項目

- [add_files](#)
- [create_ip](#)
- [create_sysgen](#)
- [create_xps](#)
- [import_ip](#)
- [list_targets](#)
- [open_example_project](#)
- [read_ip](#)
- [report_property](#)
- [reset_target](#)

get_bd_addr_segs

セグメントのリストを取得します。

構文

```
get_bd_addr_segs [-regexp] [-hierarchical] [-filter arg] [-of_objects args]
[-addressed] [-addressables] [-quiet] [-verbose] [patterns]
```

戻り値

セグメント オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-hierarchical]	階層セルを含めます。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したピンまたはネットのセグメントを取得します。
[-addressed]	-of_objects で指定したオブジェクトのアドレス指定されたセグメントを取得します。
[-addressables]	-of_objects で指定したオブジェクトのアドレス指定可能なセグメントを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインのアドレス セグメントのリストを取得します。

注記 : メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-hierarchical (オプション) : IP インテグレーター サブシステム デザインのすべての階層レベルまたは現在のインスタンスのアドレス セグメントを取得します。このオプションを指定しない場合、サブシステム デザイン階層の最上位のアドレス セグメントのみが取得されます。**-hierarchical** を使用する場合、検索パターンは完全な階層名ではなく階層の各レベルに適用されるので、検索パターンには階層区切り文字を含めないでください。たとえば、検索パターンとして「U1/*」を指定すると、名前に「U1/」を含むオブジェクトが階層の各レベルで検索され、意図した結果が得られない可能性があります。

注記 : **-regexp** と共に使用する場合、検索パターンは完全な階層名と比較され、検索パターンとして「U1/*」を指定した場合に意図した結果が得られます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_bd_addr_segs** で返されるアドレス セグメントのリストに、オブジェクトのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。IP インテグレーター アドレス セグメント オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには OFFSET、RANGE、USAGE などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!=) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (bool) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : **get_bd_addr_spaces** コマンドで返される指定のアドレス空間または **get_bd_pins** および **get_bd_intf_pins** で返されるピンおよびインターフェイス ピンのアドレス空間に割り当てられているアドレス セグメントを取得します。

-addressed (オプション) : **-of_objects** オプションで指定されたアドレス空間のアドレス指定されたセグメントを取得します。使用されているアドレス空間のセグメントが返されます。

-addressables (オプション) : **-of_objects** オプションで指定されたオブジェクトのアドレス空間の、アドレス セグメントが割り当てられていないアドレス指定可能なセグメントを取得します。

注記 : **-addressables** と **-addressed** オプションを同時に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するアドレス セグメントを取得します。デフォルトのパターンはワイルドカード (*) で、現在の IP サブシステム デザインに含まれるすべてのアドレス セグメントのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてアドレス セグメントを検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、指定したアドレス空間のアドレス セグメントを取得しています。

```
get_bd_addr_segs -of_objects [get_bd_addr_spaces -of_objects \
[get_bd_cells /microblaze_1]]
/microblaze_1/Data/SEG1
/microblaze_1/Data/SEG3
/microblaze_1/Instruction/SEG2
```

注記 : パターンに一致するオブジェクトがない場合は、警告メッセージが表示されます。

関連項目

- [create_bd_addr_seg](#)
- [get_bd_addr_spaces](#)

get_bd_addr_spaces

アドレス空間のリストを取得します。

構文

```
get_bd_addr_spaces [-regexp] [-hierarchical] [-filter arg] [-of_objects args]
[-quiet] [-verbose] [patterns]
```

戻り値

アドレス空間オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-hierarchical]	階層セルを含めます。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したピンまたはネットのアドレス空間を取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインのアドレス空間のリストを取得します。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記：Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-hierarchical (オプション) : IP インテグレーター サブシステム デザインのすべての階層レベルまたは現在のインスタンスのアドレス空間を取得します。このオプションを指定しない場合、デザイン階層の最上位のアドレス空間のみが取得されます。**-hierarchical** を使用する場合、検索パターンは完全な階層名ではなく階層の各レベルに適用されるので、検索パターンには階層区切り文字を含めないでください。たとえば、検索パターンとして「U1/*」を指定すると、名前に「U1/」を含むオブジェクトが階層の各レベルで検索され、意図した結果が得られない可能性があります。

注記 : **-regexp** と共に使用する場合、検索パターンは完全な階層名と比較され、検索パターンとして「U1/*」を指定した場合に意図した結果が得られます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_bd_addr_spaces** で返されるオブジェクトのリストに、オブジェクトのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。IP インテグレーター アドレス空間オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、RANGE、OFFSET などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (! =~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : **get_bd_cells** で指定した IP インテグレーター セル オブジェクトのアドレス空間を取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するアドレス空間を取得します。デフォルトのパターンはワイルドカード (*) で、現在の IP サブシステム デザインに含まれるすべてのアドレス空間のリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてアドレス空間を検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、現在の IP インテグレーター サブシステム デザインのすべてのアドレス空間を 1 行に 1 つずつリストしています。

```
join [get_bd_addr_spaces] \n
/mdm_1/S_AXI
/microblaze_1/Data
/microblaze_1/Instruction
/microblaze_1_axi_intc/s_axi
/microblaze_1_local_memory/dlmb_bram_if_cntlr/SLMB
/microblaze_1_local_memory/dlmb_bram_if_cntlr/SLMB1
/microblaze_1_local_memory/dlmb_bram_if_cntlr/SLMB2
/microblaze_1_local_memory/dlmb_bram_if_cntlr/SLMB3
/microblaze_1_local_memory/dlmb_bram_if_cntlr/S_AXI_CTRL
/microblaze_1_local_memory/ilmb_bram_if_cntlr/SLMB
/microblaze_1_local_memory/ilmb_bram_if_cntlr/SLMB1
/microblaze_1_local_memory/ilmb_bram_if_cntlr/SLMB2
/microblaze_1_local_memory/ilmb_bram_if_cntlr/SLMB3
/microblaze_1_local_memory/ilmb_bram_if_cntlr/S_AXI_CTRL
/microblaze_1_local_memory/lmb_bram/S_1
```

注記： パターンに一致するオブジェクトがない場合は、警告メッセージが表示されます。

次の例では、現在のサブシステム デザインのすべてのアドレス空間を取得し、3 番目にリストされるアドレス空間に設定されているすべてのプロパティを取得しています。

```
report_property -all [lindex [get_bd_addr_spaces] 2 ]
Property  Type  Read-only  Visible  Value
CLASS      string  true      true      bd_addr_space
NAME       string  false     true      /microblaze_1/Instruction
OFFSET     string  false     true
PATH       string  true      true      /microblaze_1/Instruction
RANGE      string  false     true      -1
TYPE       string  false     true
```

関連項目

- [create_bd_addr_seg](#)
- [get_bd_addr_segs](#)
- [get_bd_cells](#)

get_bd_cells

セルのリストを取得します。

構文

```
get_bd_cells [-regexp] [-hierarchical] [-filter arg] [-of_objects args]
[-quiet] [-verbose] [patterns]
```

戻り値

セル オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-hierarchical]	階層セルを含めます。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したピンまたはネットのセルを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインまたは現在のサブシステム インスタンスのセルのリストを取得します。IP インテグレーター サブシステム セルは、IP インテグレーター カタログからの IP コア、または `create_bd_cell` コマンドでサブシステム デザインに作成された階層モジュールです。

注記：メモリおよびパフォーマンスを向上するため、`get_*` コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-hierarchical (オプション) : IP インテグレーター サブシステム デザインのすべての階層レベルまたは現在のインスタンスのセルを取得します。このオプションを指定しない場合、デザイン階層の最上位のセルのみが取得されます。**-hierarchical** を使用する場合、検索パターンは完全な階層セル名ではなく階層の各レベルに適用されるので、検索パターンには階層区切り文字を含めないでください。たとえば、検索パターンとして「U1/*」を指定すると、名前に「U1/」を含むインスタンスが階層の各レベルで検索され、意図した結果が得られない可能性があります。

注記 : **-regexp** と共に使用する場合、検索パターンは完全な階層名と比較され、検索パターンとして「U1/*」を指定した場合に意図した結果が得られます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_bd_cells** で返されたオブジェクトのリストに、セルのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。IP インテグレーター セル オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには VLVN、TYPE、LOCATION などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}  
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}
```

ブール型 (bool) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : **get_bd_nets** で指定したネット オブジェクト、あるいは **get_bd_pins** または **get_bd_intf_pins** コマンドで指定したピンに接続されているサブシステム セルを取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するサブシステム セルを検索します。デフォルトのパターンはワイルドカード (*) で、現在の IP サブシステム デザインに含まれるすべてのセルのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてセルを検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、指定の IP インテグレーター サブシステム ピンを含むセルのリストを取得し、そのリストから重複を削除しています。

```
lsort -unique [get_bd_cells -of_objects [get_bd_pins -hierarchical *aclk*]]
```

注記 : パターンに一致するセルがない場合は、警告メッセージが表示されます。

次の例では、サブシステム デザイン階層のすべてのレベルにあるすべてのセルのリストを取得し、そのリストをフィルターして指定のテキストを含むセルまたは階層を含むもののみを返しています。

```
get_bd_cells -hierarchical -filter {NAME=~"/newMod1/*"}
```

関連項目

- [create_bd_cell](#)
- [get_bd_intf_pins](#)
- [get_bd_nets](#)
- [get_bd_pins](#)
- [list_property](#)
- [report_property](#)

get_bd_designs

デザインのリストを取得します。

構文

```
get_bd_designs [-regexp] [-filter arg] [-quiet] [-verbose] [patterns...]
```

戻り値

デザイン オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在のプロジェクトで開いている IP サブシステム デザインで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトで開いている IP サブシステム デザインすべてのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get *** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記： Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_designs** で返されるオブジェクトのリストに、デザインのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。IP サブシステム デザイン オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、FILE_NAME があります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

プール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するデザインを検索します。デフォルトのパターンはワイルドカード (*) で、すべての IP サブシステム デザインが取得されます。複数のパターンを指定して、異なる検索条件に基づいてデザインを検索できます。

例

次の例では、現在のプロジェクトで開いている IP サブシステム デザインすべてのリストを取得しています。

```
get_bd_designs
```

関連項目

- [create_bd_design](#)
- [current_bd_design](#)
- [open_bd_design](#)
- [report_property](#)

get_bd_intf_nets

インターフェイス ネットのリストを取得します。

構文

```
get_bd_intf_nets [-regexp] [-hierarchical] [-filter arg] [-boundary_type arg]
[-of_objects args] [-quiet] [-verbose] [patterns]
```

戻り値

ピン オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-hierarchical]	階層セルを含めます。
[-filter]	式を使用してリストをフィルター処理します。
[-boundary_type]	ソース オブジェクトが階層ブロックのインターフェイス ピン上にある場合に使用します。有効な値は、upper、lower、both です。lower に設定すると、下位階層から検索が実行されます。デフォルトは upper です。
[-of_objects]	指定したオブジェクトのインターフェイス ネットを取得します。現在のところ、1 つの完全パス名のみです。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインに含まれるインターフェイス ネットで、検索パターンに一致するもののリストを取得します。デフォルトでは、サブシステム デザインに含まれるすべてのインターフェイス ネットのリストが返されます。

注記 : メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-hierarchical (オプション) : IP インテグレーター サブシステム デザインのすべての階層レベルのインターフェイス ネットを取得します。このオプションを指定しない場合、サブシステム デザインの最上位、または現在のサブシステム インスタンスのインターフェイス ネットのみが取得されます。**-hierarchical** を使用する場合、検索パターンは完全な階層名ではなく階層の各レベルに適用されるので、検索パターンには階層区切り文字を含めないでください。たとえば、検索パターンとして「/bridge_1/*」を指定すると、名前に「/bridge_1/」を含むネットが階層の各レベルで検索され、意図した結果が得られない可能性があります。

注記 : **-regexp** と共に使用する場合、検索パターンは完全な階層名と比較され、検索パターンとして「U1/*」を指定した場合に意図した結果が得られます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_bd_intf_nets** で返されるオブジェクトのリストに、ネットのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。IP インテグレーター サブシステム ネット オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、PATH があります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=^)、不一致 (!^) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-boundary_type [upper | lower | both] (オプション) : 指定の階層ピンのレベル (upper)、ピンまたはポートの下のレベル (lower)、またはその両方のネットを取得します。有効な値は、upper、lower、both です。デフォルト値は upper です。

注記 : このオプションを使用する場合、**-of_objects** オプションを使用して階層ピンを指定する必要があります。

-of_objects args (オプション) : 指定した IP インテグレーター サブシステム セル、ピン、またはポート オブジェクトに接続されているネットのリストを取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する IP サブシステム インターフェイス ネットを検索します。デフォルトのパターンはワイルドカード (*) で、現在の IP インテグレーター サブシステム デザインに含まれるすべてのインターフェイス ネットのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてネットを検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、IP インテグレーター階層モジュールの指定のピンに接続されているインターフェイス ネットを取得しています。階層モジュールのレベルのネットと階層モジュール内のネットの両方を返しています。

```
get_bd_intf_nets -boundary_type both -of_objects [get_bd_pins /newMod1/aclk]
```

注記 : パターンに一致するインターフェイス ネットがない場合は、警告メッセージが表示されます。

次の例では、IP インテグレーター サブシステム デザインのすべての階層レベルのインターフェイス ネットを取得しています。

```
get_bd_intf_nets -hierarchical
```

関連項目

- [connect_debug_port](#)
- [get_cells](#)
- [get_clocks](#)
- [get_pins](#)
- [get_ports](#)
- [list_property](#)
- [report_property](#)

get_bd_intf_pins

インターフェイス ピンのリストを取得します。

構文

```
get_bd_intf_pins [-regexp] [-hierarchical] [-filter arg] [-of_objects args]
[-quiet] [-verbose] [patterns]
```

戻り値

ピン オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-hierarchical]	階層セルを含めます。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したオブジェクトのインターフェイス ピンを取得します。現在のところ、1 つの完全パス名のみです。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在の IP サブシステム デザインに含まれるインターフェイス ピン オブジェクトで、検索パターンに一致するもののリストを取得します。デフォルトでは、サブシステム デザインに含まれるすべてのインターフェイス ピンのリストが返されます。

ピンおよびインターフェイス ピンは、IP インテグレーター セルまたは階層モジュールの外部接続です。ポートおよびインターフェイス ポートは、IP サブシステム デザインの外部接続です。ポート オブジェクトを選択するには、`get_bd_ports` および `get_bd_intf_port` コマンドを使用します。

注記 : メモリおよびパフォーマンスを向上するため、`get_*` コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-hierarchical (オプション) : IP インテグレーター サブシステム デザインのすべての階層レベルのインターフェイスピンを取得します。このオプションを指定しない場合、サブシステム デザインの最上位、または現在のサブシステム インスタンスのインターフェイスピンのみが取得されます。**-hierarchical** を使用する場合、検索パターンは完全な階層名ではなく階層の各レベルに適用されるので、検索パターンには階層区切り文字を含めないでください。たとえば、検索パターンとして「/bridge_1/*」を指定すると、名前に「/bridge_1/」を含むインターフェイスピンが階層の各レベルで検索され、意図した結果が得られない可能性があります。

注記 : **-regexp** と共に使用する場合、検索パターンは完全な階層名と比較され、検索パターンとして「U1/*」を指定した場合に意図した結果が得られます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_bd_intf_pins** で返されるオブジェクトのリストに、ピンのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。IP インテグレーター インターフェイスピン オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには DIR、TYPE などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"} 
```

ブール型 (bool) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED} 
```

-of_objects arg (オプション) : 指定した IP サブシステム セルまたはネットに接続されるピンを取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する IP インテグレーター サブシステム インターフェイス ピンを検索します。デフォルトのパターンはワイルドカード (*) で、現在の IP インテグレーター サブシステム デザインまたはインスタンスに含まれるすべてのインターフェイス ピンのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてピンを検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、指定したセルに接続されているインターフェイス ピンのリストが取得されます。

```
get_bd_intf_pins -of [get_bd_cells new_vidOut_1]
```

注記 : パターンに一致するインターフェイス ピンがない場合は、警告メッセージが表示されます。

次の例では、IP インテグレーター サブシステム デザインのすべての階層レベルのインターフェイス ピンで、指定の名前パターンに一致するものを取得しています。

```
get_bd_intf_pins -hierarchical m_apb*
```

次の例では、指定したサブシステム ネットに接続されているインターフェイス ピンのリストを取得しています。

```
get_bd_intf_pins -of [get_bd_intf_nets vidout_1_vtg_ce]
```

関連項目

- [create_bd_net](#)
- [create_bd_port](#)
- [get_bd_intf_pins](#)
- [get_bd_intf_ports](#)
- [get_bd_pins](#)
- [list_property](#)
- [report_property](#)

get_bd_intf_ports

インターフェイス ポートのリストを取得します。

構文

```
get_bd_intf_ports [-regexp] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

ポート オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したオブジェクトのインターフェイス ポートを取得します。現在のところ、1 つの完全パス名のみです。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインに含まれるインターフェイス ポート オブジェクトで、検索パターンに一致するもののリストを取得します。デフォルトでは、サブシステム デザインに含まれるすべてのインターフェイス ポートのリストが返されます。

ポートおよびインターフェイス ポートは、IP サブシステム デザインの外部接続です。ピンおよびインターフェイス ピンは、IP インテグレーター セルまたは階層モジュールの外部接続です。ピン オブジェクトを選択するには、`get_bd_pins` および `get_bd_intf_pins` コマンドを使用します。

注記：メモリおよびパフォーマンスを向上するため、`get_*` コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_bd_intf_ports** で返されるオブジェクトのリストに、インターフェイス ポートのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。IP サブシステム インターフェイス ポート オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには DIR、MODE、LOCATION などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

プール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : **get_bd_intf_nets** で返された指定の IP サブシステム インターフェイス ネットに接続されているインターフェイス ポートを取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するインターフェイス ポートを取得します。デフォルトのパターンはワイルドカード (*) で、サブシステム デザインに含まれるすべてのインターフェイス ポートのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてインターフェイス ポートを検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、サブシステム デザインでマスター モードで動作しているインターフェイス ポートを取得しています。

```
get_bd_intf_ports -filter {MODE=="master"}
```

注記：パターンに一致するインターフェイス ポートがない場合は、警告メッセージが表示されます。

関連項目

- [create_bd_intf_net](#)
- [create_bd_intf_port](#)
- [get_bd_intf_nets](#)
- [get_bd_intf_pins](#)
- [get_bd_intf_ports](#)
- [get_bd_nets](#)
- [get_bd_pins](#)
- [list_property](#)
- [report_property](#)

get_bd_nets

ネットのリストを取得します。

構文

```
get_bd_nets [-regexp] [-hierarchical] [-filter arg] [-boundary_type arg]
[-of_objects args] [-quiet] [-verbose] [patterns]
```

戻り値

ピン オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-hierarchical]	階層セルを含めます。
[-filter]	式を使用してリストをフィルター処理します。
[-boundary_type]	ソース オブジェクトが階層ブロックのピン上にある場合に使用します。有効な値は、upper、lower、both です。lower に設定すると、下位階層から検索が実行されます。デフォルトは upper です。
[-of_objects]	指定したオブジェクトのネットを取得します。現在のところ、1 つの完全パス名のみです。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインに含まれるネットで、検索パターンに一致するもののリストを取得します。デフォルトでは、サブシステム デザインに含まれるすべてのネットのリストが返されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-hierarchical (オプション) : IP インテグレーター サブシステム デザインのすべての階層レベルのネットを取得します。このオプションを指定しない場合、サブシステム デザインの最上位、または現在のサブシステム インスタンスのネットのみが取得されます。**-hierarchical** を使用する場合、検索パターンは完全な階層名ではなく階層の各レベルに適用されるので、検索パターンには階層区切り文字を含めないでください。たとえば、検索パターンとして「/bridge_1/*」を指定すると、名前に「/bridge_1/」を含むネットが階層の各レベルで検索され、意図した結果が得られない可能性があります。

注記 : **-regexp** と共に使用する場合、検索パターンは完全な階層名と比較され、検索パターンとして「U1/*」を指定した場合に意図した結果が得られます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_bd_nets** で返されるオブジェクトのリストに、ネットのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。IP インテグレーター サブシステム ネット オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、PATH があります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=^)、不一致 (!^) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-boundary_type [upper | lower | both] (オプション) : 指定の階層ピンのレベル (upper)、ピンまたはポートの下のレベル (lower)、またはその両方のネットを取得します。有効な値は、upper、lower、both です。デフォルト値は upper です。

注記 : このオプションを使用する場合、**-of_objects** オプションを使用して階層ピンを指定する必要があります。

-of_objects args (オプション) : 指定したセル、ピン、またはポート オブジェクトに接続されているネットのリストを取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する IP サブシステム ネットを検索します。デフォルトのパターンはワイルドカード (*) で、現在の IP インテグレーター サブシステム デザインに含まれるすべてのネットのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてネットを検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、IP インテグレーター 階層モジュールの指定のピンに接続されているネットを取得しています。階層モジュールのレベルのネットと階層モジュール内のネットの両方を返しています。

```
get_bd_nets -boundary_type both -of_objects [get_bd_pins /newMod1/aclk]
```

注記 : パターンに一致するネットがない場合は、警告メッセージが表示されます。

次の例では、IP インテグレーター サブシステム デザインのすべての階層レベルのネットを取得しています。

```
get_bd_nets -hierarchical
```

関連項目

- [connect_debug_port](#)
- [get_cells](#)
- [get_clocks](#)
- [get_pins](#)
- [get_ports](#)
- [list_property](#)
- [report_property](#)

get_bd_pins

ピンのリストを取得します。

構文

```
get_bd_pins [-regexp] [-hierarchical] [-filter arg] [-of_objects args]
[-quiet] [-verbose] [patterns]
```

戻り値

ピン オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-hierarchical]	階層セルを含めます。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したオブジェクトのピンを取得します。現在のところ、1 つの完全パス名のみです。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在の IP サブシステム デザインに含まれるピン オブジェクトで、検索パターンに一致するもののリストを取得します。デフォルトでは、サブシステム デザインに含まれるすべてのピンのリストが返されます。

ピンおよびインターフェイスピンは、IP インテグレーター セルまたは階層モジュールの外部接続です。ポートおよびインターフェイスポートは、IP サブシステム デザインの外部接続です。ポートオブジェクトを選択するには、`get_bd_ports` および `get_bd_intf_port` コマンドを使用します。

注記：メモリおよびパフォーマンスを向上するため、`get *` コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-hierarchical (オプション) : IP インテグレーター サブシステム デザインのすべての階層レベルのピンを取得します。このオプションを指定しない場合、サブシステム デザインの最上位、または現在のサブシステム インスタンスのピンのみが取得されます。**-hierarchical** を使用する場合、検索パターンは完全な階層名ではなく階層の各レベルに適用されるので、検索パターンには階層区切り文字を含めないでください。たとえば、検索パターンとして「/bridge_1/*」を指定すると、名前に「/bridge_1/」を含むピンが階層の各レベルで検索され、意図した結果が得られない可能性があります。

注記 : **-regexp** と共に使用する場合、検索パターンは完全な階層名と比較され、検索パターンとして「U1/*」を指定した場合に意図した結果が得られます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、`get_pins` で返されるオブジェクトのリストに、ピンのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。ピン オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには DIR、TYPE などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!=) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : 指定した IP サブシステム セルまたはネットに接続されるピンを取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する IP インテグレーター サブシステム ピンを検索します。デフォルトのパターンはワイルドカード (*) で、現在の IP インテグレーター サブシステム デザインまたはインスタンスに含まれるすべてのピンのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてピンを検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、指定したセルに接続されているピンのリストが取得されます。

```
get_bd_intf_pins -of [get_bd_cells new_vidOut_1]
```

注記 : パターンに一致するピンがない場合は、警告メッセージが表示されます。

次の例では、IP インテグレーター サブシステム デザインのすべての階層レベルのピンで、指定の名前パターンに一致するものを取得しています。

```
get_bd_pins -hierarchical m_apb*
```

次の例では、指定したサブシステム ネットに接続されているピンのリストを取得しています。

```
get_bd_pins -of [get_bd_nets vidout_1_vtg_ce]
```

関連項目

- [create_bd_net](#)
- [create_bd_port](#)
- [get_bd_intf_pins](#)
- [get_bd_intf_ports](#)
- [get_bd_pins](#)
- [list_property](#)
- [report_property](#)

get_bd_ports

ポートのリストを取得します。

構文

```
get_bd_ports [-regexp] [-filter arg] [-of_objects args] [-quiet] [-verbose]
[patterns]
```

戻り値

ポート オブジェクトのリスト、エラーが発生した場合は ""

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したオブジェクトのポートを取得します。現在のところ、1 つの完全パス名のみです。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインに含まれるポート オブジェクトで、検索パターンに一致するもののリストを取得します。デフォルトでは、サブシステム デザインに含まれるすべてのポートのリストが返されます。

ポートおよびインターフェイス ポートは、IP サブシステム デザインの外部接続です。ピンおよびインターフェイス ピンは、IP インテグレーター セルまたは階層モジュールの外部接続です。ピン オブジェクトを選択するには、`get_bd_pins` および `get_bd_intf_pins` コマンドを使用します。

注記：メモリおよびパフォーマンスを向上するため、`get_*` コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_bd_ports** で返されるオブジェクトのリストに、ポートのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。IP サブシステム ポート オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには DIR、MODE、LOCATION などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

プール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : **get_bd_nets** で返された指定の IP サブシステム ネットに接続されているポートを取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するポートを検索します。デフォルトのパターンはワイルドカード (*) で、サブシステム デザインに含まれるすべてのポートのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてポートを検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、指定した IP サブシステム ネットに接続されているポートを取得しています。

```
get_bd_ports -of_objects [get_bd_nets bridge_1_apb_m]
```

注記：パターンに一致するポートがない場合は、警告メッセージが表示されます。

関連項目

- [create_bd_net](#)
- [create_bd_port](#)
- [get_bd_intf_pins](#)
- [get_bd_intf_ports](#)
- [get_bd_pins](#)
- [list_property](#)
- [report_property](#)

get_bel_pins

BEL ピンのリストを取得します。デザインが読み込まれている場合、構築されたサイトタイプ BEL を取得します。

構文

```
get_bel_pins [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

bel_pindef

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定した BEL 、サイト、ピン、またはネットの bel_pindef を取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	bel_pindef を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、[XDC](#)

説明

指定した BEL のピンで、指定した検索パターンに一致するもののリストを返します。

デフォルトでは、デバイス上にあるすべての BEL のピンすべてのリストが取得されます。

注記 : メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_bel_pins** で返されるオブジェクトのリストに、ピンのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。PIN オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、IS_INVERTED があります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *args* (オプション) : このオプションは **get_sites** コマンドと共に使用でき、指定した BEL のピンを返します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する BEL ピンを検索します。デフォルトのパターンはワイルドカード (*) で、デバイス上の BEL ピンすべてのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてピンを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、デバイス上の指定したサイト範囲に関連する指定した BEL のピンが返されます。

```
get_bel_pins -of_objects [get_bels -of_objects [get_sites \
    -range {SLICE_X0Y0 SLICE_X1Y1}] ]
```

次の例では、デバイス上にあるすべての BEL のクロック イネーブル (CE) ピンが返されます。

```
get_bel_pins *CE
```

関連項目

- [get_bels](#)
- [get_sites](#)
- [list_property](#)
- [report_property](#)

get_bels

BEL のリストを取得します。デザインが読み込まれている場合、構築されたサイト タイプ BEL を取得します。

構文

```
get_bels [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

BEL

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したサイト、セル、クロック領域、ネットの BEL を取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	BEL を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、[xdc](#)

説明

基本エレメント (BEL) とは、フリップフロップ、LUT、キャリー ロジックなど、スライスを構成するロジックの構築ブロックです。このコマンドは、開いているデザインのターゲット デバイスの BEL で、指定した検索パターンに一致するものをリストします。

デフォルトでは、デバイスのすべての BEL のリストが取得されます。

注記 : メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_bels** で返されるオブジェクトのリストに、BEL のプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。BEL オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには **IS_OCCUPIED**、**TYPE** があります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が “” のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *args* (オプション) : このオプションは **get_sites** コマンドと共に使用でき、指定したサイト オブジェクトの BEL を返します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する BEL を検索します。デフォルトのパターンはワイルドカード (*) で、デバイスの BEL すべてのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいて BEL を検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、ターゲット パーツの BEL の総数が返されます。

```
llength [get_bels]
```

次の例では、指定のサイトに関連する BEL が返されます。

```
get_bels -of_objects [get_sites PHASER_IN_PHY_X0Y5]
```

関連項目

- [get_sites](#)
- [list_property](#)
- [report_property](#)

get_board_interfaces

VNV で指定されたバス定義をインプリメントする、ボードのインターフェイスのリストを取得します。

構文

```
get_board_interfaces [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose]
[patterns...]
```

戻り値

バスインターフェイスのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	バスインターフェイスを検索するパターンを指定します。デフォルトのパターンは、ワイルドカード (*) または -regexp を指定している場合は「.*」です。

カテゴリ

オブジェクト、プロジェクト、ボード

説明

現在のプロジェクトで使用されているボードで指定されているボードインターフェイスのリストを取得します。

current_board コマンドは、現在のプロジェクトで使用されているボードを返します。**get_boards** コマンドは、現在のプロジェクトで使用可能なボードのリストを返します。

Vivado Design Suite インストール エリアの `data/boards` フォルダーにあるボード ファイル `board.xml` には、ボード属性に関する情報が保存されています。ボードは、ザイリンクス デバイスを含むシステム全体を表しており、クロック制約、I/O ポート割り当て、サポートされるインターフェイスなど、FPGA デザインの主要な部分を定義するのに役立ちます。

ボードで定義されているインターフェイスは、Vivado Design Suite インストール エリアの `data/boards` フォルダーにある `board_rtl.xml` ファイルに保存されています。`board_rtl.xml` ファイルは、`board.xml` ファイルで参照されます。

現在のボードで使用可能なインターフェイスは、**create_bd_interface_port** または **create_bd_port** を使用して IP サブシステム デザインで必要なインターフェイスを定義するのに使用するか、**create_interface** および **create_port** を使用して FPGA デザイン全体で必要なインターフェイスを定義するのに使用できます。

引数

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_board_interfaces** で返されるオブジェクトのリストに、インターフェイスのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (bool) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のボードのインターフェイスすべてのリストを取得しています。

```
join [get_board_interfaces] \n
```

関連項目

- [create_interface](#)
- [current_board](#)
- [current_project](#)
- [get_board_pins](#)
- [get_boards](#)

get_board_pins

ボード ピン オブジェクトのリストを取得します。

構文

```
get_board_pins [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns ...]
```

戻り値

ボードのピンのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したボード インターフェイス オブジェクトのボード ピンを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ボード ピン名を検索するパターンを指定します。デフォルトのパターンは、ワイルドカード (*) または -regexp を指定している場合は「.*」です。

カテゴリ

オブジェクト、プロジェクト、ボード

説明

現在のプロジェクトで使用されているボードのボード ピン オブジェクトのリストを取得します。

current_board コマンドは、現在のプロジェクトで使用されているボードを返します。get_boards コマンドは、現在のプロジェクトで使用可能なボードのリストを返します。

Vivado Design Suite インストール エリアの data/boards フォルダーにあるボード ファイル board.xml には、ボード属性に関する情報が保存されています。ボードは、ザイリンクス デバイスを含むシステム全体を表しており、クロック制約、I/O ポート割り当て、サポートされるインターフェイスなど、FPGA デザインの主要な部分を定義するのに役立ちます。

ボードで使用されるピンは、Vivado Design Suite インストール エリアの data/boards フォルダーにある board_pinmap.xml ファイルに保存されています。board_pinmap.xml ファイルは、board.xml ファイルで参照されます。

ボードピンは、物理的なボード、システム、またはターゲットリファレンス デザイン (TRD) 上のピンを表しており、ザイリンクス FPGA パッケージピンが接続されています。LOC、IOSTANDARD、SLEW などのプロパティがあります。ボードピンはスカラーまたはベクターで、常にビットで表されます。

ボードピンは、**create_port** および **set_property PACKAGE_PIN** コマンドを使用して、FPGA デザインにポートを定義および配置するのに使用できます。

引数

-filter args (オプション)：結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_board_pins** で返されるオブジェクトのリストに、ピンのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記：フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が “” のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=^)、不一致 (!^) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects args (オプション)：指定したボードインターフェイス オブジェクトのボードピンを取得します。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、LED_8Bits ボード インターフェイスに割り当てられているボード ピンのリストを取得し、それらのピンを Tcl 変数に保存して、それらの各ピンの LOC プロパティを表示しています。

```
set boardPins [get_board_pins \
-of_objects [get_board_interfaces -filter {NAME == LED_8Bits}]]
foreach pin $boardPins {puts "The location of $pin is: [get_property LOC $pin]"}
The location of LEDs_8Bits_TRI_O[0] is: AB8
The location of LEDs_8Bits_TRI_O[1] is: AA8
The location of LEDs_8Bits_TRI_O[2] is: AC9
The location of LEDs_8Bits_TRI_O[3] is: AB9
The location of LEDs_8Bits_TRI_O[4] is: AE26
The location of LEDs_8Bits_TRI_O[5] is: G19
The location of LEDs_8Bits_TRI_O[6] is: E18
The location of LEDs_8Bits_TRI_O[7] is: F16
```

関連項目

- [create_interface](#)
- [current_board](#)
- [current_project](#)
- [get_board_interfaces](#)
- [get_boards](#)

get_boards

プロジェクトで使用可能なボードのリストを取得します。

構文

```
get_boards [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose] [patterns...]
```

戻り値

ボード オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ボード名を検索するパターンを指定します。デフォルトのパターンは、ワイルドカード (*) または -regexp を指定している場合は「.*」です。

カテゴリ

オブジェクト、プロジェクト、XPS

説明

現在のプロジェクトで使用可能な評価ボードのリストを取得します。

現在のプロジェクトで使用されているボードを取得するには、**current_board** コマンドを使用します。

Vivado Design Suite インストール エリアの `data/boards` フォルダーにあるボード ファイル `board.xml` には、ボード属性に関する情報が保存されています。ボードは、ザイリンクス デバイスを含むシステム全体を表しており、クロック制約、I/O ポート割り当て、サポートされるインターフェイスなど、FPGA デザインの主要な部分を定義するのに役立ちます。

ボードは、プロジェクトを定義する際にターゲット リファレンス デザインの一部として指定するか、現在のプロジェクトの BOARD プロパティを設定することにより指定するか、Vivado IDE の [Project Settings] ダイアログ ボックスで [Project device] を選択することにより指定します。プロジェクト設定の詳細は、『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。

重要： `set_property` コマンドでボードを指定すると、ターゲット デバイスも BOARD プロパティに合わせて変更されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_boards** で返されるオブジェクトのリストに、ボードのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。ボード オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、DEVICE、FAMILY などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定した検索パターンに一致するボードを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトで使用可能なすべてのボードのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてボードを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定の評価ボードのプロパティをレポートしています。

```
report_property [get_boards -filter {LIBRARY_NAME==artix7}]
```

次の例では、指定した検索パターンに一致するすべてのボードを取得しています。

```
get_boards {*ar* *kc*}
```

関連項目

- [current_board](#)
- [get_board_interfaces](#)
- [get_board_pins](#)
- [list_property](#)
- [report_property](#)
- [set_property](#)

get_cells

現在のデザインのセルのリストを取得します。

構文

```
get_cells [-hsc arg] [-hierarchical] [-regexp] [-nocase] [-filter arg]
[-of_objects args] [-match_style arg] [-quiet] [-verbose] [patterns]
```

戻り値

セル オブジェクトのリスト

使用法

名前	説明
[-hsc]	階層区切り文字を指定します。デフォルトはスラッシュ (/) です。
[-hierarchical]	すべての階層レベルで検索します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したピン、タイミング パス、ネット、BEL、またはサイトのセルを取得します。
[-match_style]	パターン一致のスタイルを指定します。有効な値は ucf、sdc で、デフォルトは sdc です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	セル名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

SDC、XDC、オブジェクト

説明

現在のデザインに含まれるセル オブジェクトで、検索パターンに一致するもののリストを取得します。デフォルトでは、デザインのすべてのセルのリストが返されます。

注記 : メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-hsc *arg* (オプション) : 階層区切り文字を指定します。デフォルトの階層区切り文字は / です。

-hierarchical (オプション) : デザイン階層のすべてのレベルからセルを取得します。このオプションを指定しない場合、デザイン階層の最上位のセルのみが取得されます。**-hierarchical** を使用する場合、検索パターンは完全な階層セル名ではなく階層の各レベルに適用されるので、検索パターンには階層区切り文字を含めないでください。たとえば、検索パターンとして「U1/*」を指定すると、名前に「U1/」を含むインスタンスが階層の各レベルで検索され、意図した結果が得られない可能性があります。これを下の例に示します。

重要 : **-regexp** と共に使用する場合、検索パターンは完全な階層名と比較され、検索パターンとして「U1/*」を指定した場合に意図した結果が得られます。

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_cells** で返されたオブジェクトのリストに、セルのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。cell オブジェクトの場合、結果をフィルター処理できるプロパティには IS_PARTITION、IS_PRIMITIVE、IS_LOC_FIXED などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=^)、不一致 (!^) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *arg* (オプション) : 指定したピンまたはネット オブジェクトに接続されているセルを取得します。

注記 : **-of_objects** を **-hierarchy** または検索パターンと共に使用することはできません。

-match_style (オプション) : 検索パターンが UCF 制約または SDC 制約に一致することを示します。デフォルトは SDC です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するセルを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトのすべてのセルのリストが返されます。複数のパターンを指定して、異なる検索条件に基づいてセルを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、すべての階層レベルで cpu または fft で開始するセルを検索し、各セルの後に改行文字を付けて 1 行ごとに表示しています。

```
join [get_cells -hier {cpu* fft*}] \n
```

次の例では、get_cells で返された 2 番目のオブジェクトに設定されているプロパティとその値を取得しています。

```
report_property [lindex [get_cells] 1]
```

注記 : パターンに一致するセルがない場合は、警告メッセージが表示されます。

次の例では、デザインのすべての階層レベルにインスタンシエートされているライブラリ セルのリストを表示しています。リストは名前順に並べられるので、それぞれのセルが表示されるのは 1 度だけです。

```
foreach cell [lsort -unique [get_property LIB_CELL [get_cells -hier \
-filter {IS_PRIMITIVE==1}]]][[SDLENTITYREF[nbsp]]]{puts[[SDLENTITYREF[nbsp]]]$cell}
```

次の例では、**-regexp** を使用した場合と使用しない場合の **-hierarchical** オプションの効果を示しています。

```
get_cells -hierarchical *mmcm*
mmcm_replicator_inst_1
mmcm_replicator_inst_1/mmcm_stage[0].mmcm_channel[0].mmcm
get_cells -hierarchical -regexp .*mmcm.*
mmcm_replicator_inst_1
mmcm_replicator_inst_1/mmcm_stage[0].mmcm_channel[0].mmcm
mmcm_replicator_inst_1/mmcm_stage[0].mmcm_channel[0].mmcm/GND
mmcm_replicator_inst_1/mmcm_stage[0].mmcm_channel[0].mmcm/MMCM_Base
```

注記 : 最初の例 (**-regexp** なし) では検索パターンが階層の各レベルに適用されるため、最後の 2 つのセル (GND および MMCM_Base) は一致せず、返されません。

関連項目

- [get_lib_cells](#)
- [get_nets](#)
- [get_pins](#)
- [list_property](#)
- [report_property](#)

get_clock_regions

現在のデバイスのクロック領域を取得します。

構文

```
get_clock_regions [-regexp] [-nocase] [-filter arg] [-of_objects args]
[-quiet] [-verbose] [patterns]
```

戻り値

クロック領域

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したサイト、SLR、またはセルのクロック領域を取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト

説明

ターゲットデバイスのクロック領域で、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、開いているデザインのデバイスのクロック領域すべてのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_clock_regions** で返されるオブジェクトのリストに、クロック領域のプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。クロック領域オブジェクトの場合、結果をフィルター処理できるプロパティには COLUMN_INDEX、HIGH_X、LOW_X などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *args* (オプション) : **get_slrs** コマンドで指定した SLR に含まれるクロック領域、あるいは **get_sites** または **get_cells** コマンドで指定したデバイス サイトまたはロジック セルに割り当てられているクロック領域を取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するクロック領域を検索します。デフォルトのパターンはワイルドカード (*) で、デバイスのクロック領域すべてのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてクロック領域を検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定の検索パターンに一致するクロック領域が返されます。

```
get_clock_regions X0*
```

次の例では、指定のプロパティが設定されているクロック領域が返されます。

```
get_clock_regions -filter {LOW_X==0}
```

注記 : これらの 2 つの例では、同じクロック領域のリストが返されます。

関連項目

- [get_sites](#)
- [list_property](#)
- [report_property](#)

get_clocks

現在のデザインのクロックのリストを取得します。

構文

```
get_clocks [-regexp] [-nocase] [-filter arg] [-of_objects args]
[-match_style arg] [-include_generated_clocks] [-quiet] [-verbose] [patterns]
```

戻り値

クロックのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したピンまたはネットのクロックを取得します。
[-match_style]	パターン一致のスタイルを指定します。有効な値は ucf、sdc で、デフォルトは sdc です。
[-include_generated_clocks]	自動推論されたクロックおよび生成されたクロックも含めます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	クロック名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

SDC、XDC、オブジェクト

説明

現在のデザインに含まれるクロックで、検索パターンに一致するもののリストを取得します。デフォルトでは、**all_clocks** コマンドと同様に、デザインのすべてのクロックのリストが返されます。

クロックを作成するには、**create_clock** または **create_generated_clock** コマンドを使用するか、MMCM の出力などで、ツールで自動的に生成することもできます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_clocks** で返されるオブジェクトのリストに、クロックのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。クロックオブジェクトの場合、結果をフィルター処理できるプロパティには PERIOD、WAVEFORM、IS_GENERATED などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *args* (オプション) : 指定したピンまたはネット オブジェクトに接続されているクロックを取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-match_style (オプション) : 検索パターンが UCF 制約または SDC 制約に一致することを示します。デフォルトは SDC です。

-include_generated_clocks (オプション) : 生成クロックも含めたすべてのクロックで、ソースまたはマスター クロックとして指定のパターンに一致するものを返します。このオプションは、*patterns* を指定して指定のマスター クロックから生成したクロックを取得する場合に使用してください。

注記 : **get_generated_clocks** コマンドを使用すると、生成したクロックのみを取得できます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するクロックを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトのすべてのクロックが返されます。複数のパターンを指定して、異なる検索条件に基づいてクロックを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、複数の検索パターンに一致するクロックのリストを取得しています。

```
get_clocks {*clock *ck *Clk}
```

注記 : パターンに一致するクロックがない場合は、警告メッセージが表示されます。

次の例では、マスター クロック オブジェクトと、そのクロックから生成したクロックを取得しています。

```
get_clocks -include_generated_clocks wbClk
```

次の例では、指定のクロックに設定されているプロパティとその値を取得しています。

```
report_property -all [get_clocks wbClk]
```

関連項目

- [all_clocks](#)
- [create_clock](#)
- [create_generated_clock](#)
- [get_generated_clocks](#)
- [list_property](#)
- [report_property](#)

get_debug_cores

現在のデザインのデバッグ コアのリストを取得します。

構文

```
get_debug_cores [-filter arg] [-of_objects args] [-regexp] [-nocase] [-quiet]
[-verbose] [patterns]
```

戻り値

デバッグ コア オブジェクトのリスト

使用法

名前	説明
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したデバッグ ポートまたはネットのコアを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	デバッグ コアを検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト

説明

現在のプロジェクトの Vivado ラボ ツール デバッグ コアで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトのデバッグ コアすべてのリストが取得されます。

デバッグ コアは、`create_debug_core` コマンドを使用してプロジェクトに追加できます。ILA デバッグ コア (labtools_ilav2) をプロジェクトに追加すると、デバッグ ハブ コア (labtools_xsdbmasterlib_v2) 内に含まれ、CLK ポートと PROBE ポートがデフォルトで含まれます。`create_debug_port` コマンドを使用して、デバッグ コアにポートを追加することもできます。

注記 : メモリおよびパフォーマンスを向上するため、`get_*` コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_debug_cores** で返されるオブジェクトのリストに、パートのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

プール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects args (オプション) : 指定したデバッグ ポートまたはネットに接続されているデバッグ コアを取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイレンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するデバッグ コアを取得します。デフォルトのパターンはワイルドカード (*) で、すべてのデバッグ コアが取得されます。複数のパターンを指定して、異なる検索条件に基づいてデバッグ コアを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、現在のプロジェクトの Vivado ラボ ツール デバッグ コアのリストを取得しています。

```
get_debug_cores
```

注記 : デバッグ ハブ コアがプロジェクトのデバッグ コアの 1 つとして返されます。このコアは直接作成できませんが、ILA コアをプロジェクトに追加すると自動的に追加されます。

次の例では、指定したデバッグ コアのプロパティを取得しています。

```
report_property [get_debug_cores myCore]
```

関連項目

- [create_debug_core](#)
- [create_debug_port](#)
- [get_debug_ports](#)
- [list_property](#)
- [report_property](#)
- [set_property](#)

get_debug_ports

現在のデザインのデバッグ ポートのリストを取得します。

構文

```
get_debug_ports [-filter arg] [-of_objects args] [-regexp] [-nocase] [-quiet]
[-verbose] [patterns]
```

戻り値

デバッグ ポート オブジェクトのリスト

使用法

名前	説明
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したデバッグ コアのポートを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	デバッグ ポートを検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト

説明

現在のプロジェクトの ILA デバッグ コアで定義されているポートで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトのデバッグ ポートすべてのリストが取得されます。

デバッグ ポートは、`create_debug_core` コマンドで ILA デバッグ コアを作成すると定義されます。`create_debug_port` コマンドを使用して、ポートを既存のデバッグ コアに追加することもできます。

注記 : メモリおよびパフォーマンスを向上するため、`get_*` コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_debug_ports** で返されるオブジェクトのリストに、ポートのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。デバッグ ポート オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには PORT_WIDTH、MATCH_TYPE などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}  
get_ports * -filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects args (オプション) : 指定したデバッグ コアのデバッグ ポートを取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイレンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するデバッグ ポートを取得します。デフォルトのパターンはワイルドカード (*) で、すべてのデバッグ ポートが取得されます。複数のパターンを指定して、異なる検索条件に基づいてデバッグ ポートを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、現在のプロジェクトの ILA デバッグ コアのポートで、PORT_WIDTH プロパティが 8 のもののリストを取得しています。

```
get_debug_ports -filter {PORT_WIDTH==8}
```

次の例では、指定したデバッグ ポートに設定されているプロパティを取得しています。

```
report_property [get_debug_ports myCore/PROBE1]
```

注記 : デバッグ ポートは、core_name/port_name の形式で指定します。

関連項目

- [create_debug_core](#)
- [create_debug_port](#)
- [list_property](#)
- [report_property](#)

get_delays

遅延オブジェクトを返します。

構文

```
get_delays [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose]
```

戻り値

なし

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したオブジェクトの遅延を取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

オブジェクト

get_designs

現在のプロジェクトに含まれるデザインのリストを取得します。

構文

```
get_designs [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose] [patterns]
```

戻り値

デザイン オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	デザイン名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト

説明

現在のプロジェクトで開いているデザインで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトで開いているデザインすべてのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_designs** で返されるオブジェクトのリストに、デザインのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。design オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには CONSTRSET、PART などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルターパターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するデザインを検索します。デフォルトのパターンはワイルドカード (*) で、すべてのデザインが取得されます。複数のパターンを指定して、異なる検索条件に基づいてデザインを検索できます。

例

次の例では、現在のプロジェクトで開いているデザインすべてのリストを取得しています。

```
get_designs
```

次の例では、検索パターンに一致するデザインに設定されているプロパティを取得しています。

```
report_property [get_designs r*]
```

注記： パターンに一致するデザインがない場合は、警告メッセージが表示されます。

関連項目

[report_property](#)

get_drc_checks

DRC ルール チェック オブジェクトのリストを取得します。

構文

```
get_drc_checks [-of_objects args] [-regexp] [-nocase] [-filter arg]
[-abbrev arg] [-quiet] [-verbose] [patterns]
```

戻り値

DRC チェック

使用法

名前	説明
[-of_objects]	指定した DRC ルール デックの DRC チェック オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-abbrev]	指定した略称のルールで ID が最大のものを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	DRC チェック オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

DRC、オブジェクト

説明

現在定義済みの DRC チェックのリストを取得します。リストには、デフォルトであらかじめ定義されているデザインルール チェックと、**create_drc_check** コマンドで作成したユーザー定義チェックの両方が含まれます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-of_objects args (オプション) : 指定したルール デック オブジェクトのデザイン ルール チェックを取得します。ルール デック オブジェクトは、**get_drc_ruledeck** コマンドを使用して指定する必要があります。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_drc_checks** で返されるルール チェックのリストに、プロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!=) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (bool) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-abbrev arg (オプション) : 指定したルール名または略称のデザイン ルール チェックを取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するデザイン ルール チェックを検索します。デフォルトのパターンはワイルドカード (*) で、すべてのルール チェックが取得されます。複数のパターンを指定して、異なる検索条件に基づいてルール チェックを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、すべての AVAL デザイン ルール チェックのリストを取得しています。

```
get_drc_checks AVAL*
```

次の例では、指定したルール デックのチェックを取得しています。

```
get_drc_checks -of_objects [get_drc_ruledeck placer_checks]
```

関連項目

- [create_drc_check](#)
- [get_drc_ruledecks](#)
- [list_property](#)
- [report_drc](#)
- [report_property](#)

get_drc_ruledecks

DRC ルール デック オブジェクトのリストを取得します。

構文

```
get_drc_ruledecks [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose]
[patterns]
```

戻り値

DRC ルール デック

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	DRC ルール デック オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

DRC、オブジェクト

説明

report_drc コマンドで使用するため、現在定義されているルール デックのリストを取得します。

ルール デックはデザイン ルール チェックのグループで、I/O 配置や配置などの FPGA デザインフローの異なる段階で **report_drc** コマンドにより実行されます。ツールには定義済みのルール デックが含まれていますが、**create_drc_ruledeck** コマンドを使用して新しいユーザー定義のルール デックを作成し、**add_drc_checks** コマンドを使用してチェックを追加できます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_drc_ruledecks** で返されるオブジェクトのリストに、プロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するルール デックを検索します。デフォルトのパターンはワイルドカード (*) で、すべてのルール デックが取得されます。複数のパターンを指定して、異なる検索条件に基づいてルール デックを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、現在のプロジェクトで定義されているすべてのルール チェックのリストを取得しています。

```
get_drc_ruledecks
```

次の例では、placer_checks ルール デックの各チェックを個別の行でリストしています。

```
foreach rule [get_drc_checks -of_objects [get_drc_ruledecks placer_checks]] \
  {puts $rule}
```

関連項目

- [add_drc_checks](#)
- [create_drc_ruledeck](#)
- [list_property](#)
- [report_drc](#)
- [report_property](#)

get_drc_vios

前回の report_drc コマンドからの DRC 違反のリストを取得します。

構文

```
get_drc_vios [-name arg] [-regexp] [-filter arg] [-nocase] [-quiet] [-verbose]
[patterns]
```

戻り値

違反オブジェクトのリスト

使用法

名前	説明
[-name]	指定した名前の結果を取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-filter]	式を使用してリストをフィルター処理します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	DRC 違反を検索するパターンを指定します。デフォルトのパターンは、ワイルドカード (*) または -regexp を指定している場合は「.*」です。

カテゴリ

DRC、オブジェクト

説明

report_drc コマンドで DRC チェックを実行した際に、create_drcViolation コマンドで作成された違反オブジェクトのリストを取得します。個々の違反オブジェクトの詳細は、report_property または list_property コマンドを使用して詳細を取得します。

注記：メモリおよびパフォーマンスを向上するため、get_* コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。append を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-name arg (オプション) : 指定した名前の DRC 結果セットの違反を取得します。このオプションを使用する場合、report_drc コマンドを -name オプションを使用して実行している必要があります。

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_drc_vios** で返されるオブジェクトのリストに、プロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}  
get_ports * -filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する違反を検索します。デフォルトのパターンはワイルドカード (*) で、すべての違反が取得されます。複数のパターンを指定して、異なる検索条件に基づいて違反を検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、現在のデザインの DRC 違反をレポートし、これらの違反のリストを返しています。

```
report_drc
get_drc_vios
```

次の例では、指定した違反のプロパティを取得しています。

```
report_property [lindex [get_drc_vios] 0]
```

次の例では、指定の DRC 結果セットで指定した検索パターンに一致する違反のリストが返されます。

```
get_drc_vios -name drc_1 {*BUF* *DPIP*}
```

関連項目

- [create_drc_check](#)
- [create_drcViolation](#)
- [report_drc](#)

get_files

ソース ファイルのリストを取得します。

構文

```
get_files [-regexp] [-nocase] [-filter arg] [-of_objects args] [-all] [-quiet]
[-verbose] [patterns]
```

戻り値

ファイル オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したファイルセットまたは複合ファイルのファイルを取得します。
[-all]	すべての内部ファイルを含めます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ファイル名を検索するパターンを指定します。デフォルトは*です。

カテゴリ

オブジェクト、プロジェクト

説明

現在のプロジェクトに含まれるファイルで、検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトのファイルすべてのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_files** で返されるオブジェクトのリストに、ファイルのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。file オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには FILE_TYPE、IS_ENABLED などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!=) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (bool) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *args* (オプション) : ファイルを検索するファイルセットを 1 つまたは複数指定します。デフォルトでは、すべてのファイルセットが検索されます。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-all (オプション) : IP およびその他のオブジェクトをサポートする内部ファイルも含め、デザインのすべてのファイルを返します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するファイルを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトのすべてのファイルが取得されます。複数のパターンを指定して、異なる検索条件に基づいてファイルを検索できます。

例

次の例では、デザインの Verilog ファイルが返されます。

```
get_files -filter {FILE_TYPE == Verilog}
```

次の例では、constrs_1 および sim_1 ファイルセットに含まれる Verilog ファイル (*.v) のリストを取得しています。

```
get_files -of_objects {constrs_1 sim_1} *.v
```

注記 : パターンに一致するファイルがない場合は、警告メッセージが表示されます。

関連項目

- [report_compile_order](#)
- [report_property](#)

get_filesets

現在のプロジェクトのファイルセットのリストを取得します。

構文

```
get_filesets [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose] [patterns]
```

戻り値

ファイルセット オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ファイルセット名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、プロジェクト

説明

現在のプロジェクトに含まれるファイルセットで、検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトのファイルセットすべてのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_filesets** で返されるオブジェクトのリストに、ファイルセットのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。ファイルセット オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには DESIGN_MODE、FILESET_TYPE などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するファイルセットを検索します。デフォルトのパターンはワイルドカード (*) で、すべてのファイルセットが取得されます。複数のパターンを指定して、異なる検索条件に基づいてファイルセットを検索できます。

例

次の例では、現在のプロジェクトのファイルセットすべてのリストを取得しています。

```
get_files -of_objects [get_filesets sources_1]
```

次の例では、project_2 をアクティブ プロジェクトに指定した後、s または r で始まるファイルセットのリストを取得します。

```
current_project project_2
get_filesets s* r* -quiet
```

注記：パターンに一致するファイルセットがない場合、通常は警告メッセージが表示されますが、上記の例では **-quiet** オプションが指定されているので、警告メッセージは表示されません。

次の例では、大文字/小文字を区別せずに、C で始まるファイルセットを取得しています。

```
get_filesets C.* -regexp -nocase
```

この例では、constrs_1 および constrs_2 制約セットが現在のプロジェクトで定義されていれば返されます。

関連項目

- [get_files](#)
- [report_property](#)

get_generated_clocks

現在のデザインに含まれる生成したクロックのリストを取得します。

構文

```
get_generated_clocks [-regexp] [-nocase] [-filter arg] [-of_objects args]
[-match_style arg] [-quiet] [-verbose] [patterns]
```

戻り値

クロックのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したピンまたはネットの生成クロックを取得します。
[-match_style]	パターン一致のスタイルを指定します。有効な値は ucf、sdc で、デフォルトは sdc です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	生成したクロック名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

xdc、オブジェクト

説明

現在のプロジェクトに含まれる生成したクロックで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトに含まれる生成したクロックすべてのリストが取得されます。

生成したクロックは、**create_generated_clock** を使用してデザインに追加するか、MMCM の出力でなど、ツールで自動的に生成することもできます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_generated_clocks** で返されるオブジェクトのリストに、クロックのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。生成したクロック オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには DUTY_CYCLE、MASTER_CLOCK などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *args* (オプション) : 生成したクロック オブジェクトが割り当てられているピンまたはネットを 1 つ以上指定します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-match_style (オプション) : 検索パターンが UCF 制約または SDC 制約に一致することを示します。デフォルトは SDC です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する生成クロックを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトのすべての生成したクロックが取得されます。

例

次の例では、現在のプロジェクトに含まれる生成したクロックすべてのリストを取得しています。

```
get_generated_clocks
```

関連項目

- [create_generated_clock](#)
- [get_clocks](#)
- [report_property](#)

get_gtbanks

GT バンクのリストを取得します。

構文

```
get_gtbanks [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

GT バンク

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したパッケージ ピンの GT バンクを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	GT バンクを検索するパターンを指定します。デフォルトは * です。

カテゴリ

XDC、オブジェクト

get_hierarchy_separator

階層区切り文字を取得します。

構文

```
get_hierarchy_separator [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

[SDC](#)、[XDC](#)

説明

デザインで現在設定されている階層区切り文字を取得します。階層区切り文字を設定するには、[set_hierarchy_separator](#) コマンドを使用します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、[set_msg_config](#) コマンドで定義できます。

例

次の例では、現在定義されている階層区切り文字を取得しています。

```
get_hierarchy_separator
```

関連項目

[set_hierarchy_separator](#)

get_hw_devices

ハードウェア デバイスのリストを取得します。

構文

```
get_hw_devices [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア デバイス

使用法

名前	説明
[-of_objects]	指定したハードウェア ターゲットのハードウェア デバイス オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア デバイス オブジェクトを検索するパターンを指定します。デフォルトは* です。

カテゴリ

ハードウェア、オブジェクト

get_hw_ilab_datas

ハードウェア ILA データ オブジェクトのリストを取得します。

構文

```
get_hw_ilab_datas [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア ILA データ

使用法

名前	説明
[-of_objects]	指定したハードウェア ILA のハードウェア ILA データ オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア ILA データ オブジェクトを検索するパターンを指定します。デフォルトは*です。

カテゴリ

ハードウェア、オブジェクト

get_hw_ilas

ハードウェア ILA のリストを取得します。

構文

```
get_hw_ilas [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア ILA

使用法

名前	説明
[-of_objects]	指定したハードウェア デバイスのハードウェア ILA オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア ILA オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_probes

ハードウェア プローブのリストを取得します。

構文

```
get_hw_probes [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア プローブ

使用法

名前	説明
[-of_objects]	指定したハードウェア ILA、ハードウェア VIO のハードウェア プローブ オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア プローブ オブジェクトを検索するパターンを指定します。デフォルトは* です。

カテゴリ

ハードウェア、オブジェクト

get_hw_servers

ハードウェア サーバーのリストを取得します。

構文

```
get_hw_servers [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose]
[patterns]
```

戻り値

ハードウェア サーバー

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア サーバー オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_sio_commons

ハードウェア SIO GT common のリストを取得します。

構文

```
get_hw_sio_commons [-of_objects args] [-regexp] [-nocase] [-filter arg]
[-quiet] [-verbose] [patterns]
```

戻り値

ハードウェア SIO GT common

使用法

名前	説明
[-of_objects]	指定したハードウェア サーバー、ハードウェアターゲット、ハードウェア デバイス、ハードウェア SIO IBERT、ハードウェア SIO PLL のハードウェア SIO common オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア SIO common オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_sio_gtgroups

ハードウェア SIO GT グループのリストを取得します。

構文

```
get_hw_sio_gtgroups [-of_objects args] [-regexp] [-nocase] [-filter arg]
[-quiet] [-verbose] [patterns]
```

戻り値

ハードウェア SIO GT グループ

使用法

名前	説明
[-of_objects]	指定したハードウェア サーバー、ハードウェア ターゲット、ハードウェア デバイス、ハードウェア SIO IBERT、ハードウェア SIO common、ハードウェア SIO PLL、ハードウェア SIO GT、ハードウェア SIO TX、ハードウェア SIO RX のハードウェア SIO GT グループ オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア SIO GT グループ オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_sio_gts

ハードウェア SIO GT のリストを取得します。

構文

```
get_hw_sio_gts [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア SIO GT

使用法

名前	説明
[-of_objects]	指定したハードウェア サーバー、ハードウェアターゲット、ハードウェア デバイス、ハードウェア SIO IBERT、ハードウェア SIO PLL、ハードウェア SIO TX、ハードウェア SIO RX、ハードウェア SIO リンクのハードウェア SIO GT オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア SIO GT オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_sio_iberts

ハードウェア SIO IBERT コアのリストを取得します。

構文

```
get_hw_sio_iberts [-of_objects args] [-regexp] [-nocase] [-filter arg]
[-quiet] [-verbose] [patterns]
```

戻り値

ハードウェア SIO IBERT コア

使用法

名前	説明
[-of_objects]	指定したハードウェア サーバー、ハードウェア ターゲット、ハードウェア デバイス、ハードウェア SIO GT、ハードウェア SIO common、ハードウェア SIO PLL、ハードウェア SIO TX、ハードウェア SIO RX、ハードウェア SIO リンクのハードウェア SIO IBERT オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア SIO IBERT オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_sio_linkgroups

ハードウェア SIO リンク グループのリストを取得します。

構文

```
get_hw_sio_linkgroups [-of_objects args] [-regexp] [-nocase] [-filter arg]
[-quiet] [-verbose] [patterns]
```

戻り値

ハードウェア SIO リンク グループ

使用法

名前	説明
[-of_objects]	指定したハードウェア SIO リンクのハードウェア SIO リンク グループ オブジェクトを追加します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア SIO リンク グループ オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_sio_links

ハードウェア SIO リンクのリストを取得します。

構文

```
get_hw_sio_links [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア SIO リンク

使用法

名前	説明
[-of_objects]	指定したハードウェア サーバー、ハードウェアターゲット、ハードウェア デバイス、ハードウェア SIO IBERT、ハードウェア SIO GT、ハードウェア SIO TX、ハードウェア SIO RX、ハードウェア SIO リンクのハードウェア SIO リンク オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア SIO リンク オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_sio_pll

ハードウェア SIO PLL のリストを取得します。

構文

```
get_hw_sio_pll [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア SIO PLL

使用法

名前	説明
[-of_objects]	指定したハードウェア サーバー、ハードウェアターゲット、ハードウェア デバイス、ハードウェア SIO IBERT、ハードウェア SIO GT、ハードウェア SIO common のハードウェア SIO PLL オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア SIO PLL オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_sio_rxes

ハードウェア SIO RX のリストを取得します。

構文

```
get_hw_sio_rxes [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア SIO RX

使用法

名前	説明
[-of_objects]	指定したハードウェア サーバー、ハードウェアターゲット、ハードウェア デバイス、ハードウェア SIO IBERT、ハードウェア SIO GT、ハードウェア SIO リンクのハードウェア SIO RX オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア SIO RX オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_sio_scans

ハードウェア SIO スキャンのリストを取得します。

構文

```
get_hw_sio_scans [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア SIO スキャン

使用法

名前	説明
[-of_objects]	指定したハードウェア SIO RX のハードウェア SIO スキャン オブジェクトを追加します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア SIO スキャン オブジェクトを検索するパターンを指定します。デフォルトは*です。

カテゴリ

ハードウェア、オブジェクト

get_hw_sio_txs

ハードウェア SIO TX のリストを取得します。

構文

```
get_hw_sio_txs [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア SIO TX

使用法

名前	説明
[-of_objects]	指定したハードウェア サーバー、ハードウェアターゲット、ハードウェア デバイス、ハードウェア SIO IBERT、ハードウェア SIO GT、ハードウェア SIO リンクのハードウェア SIO TX オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア SIO TX オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_hw_targets

ハードウェア ターゲットのリストを取得します。

構文

```
get_hw_targets [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア ターゲット

使用法

名前	説明
[-of_objects]	指定したハードウェア サーバーのハードウェア ターゲット オブジェクトを追加します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア ターゲット オブジェクトを検索するパターンを指定します。デフォルトは* です。

カテゴリ

ハードウェア、オブジェクト

get_hw_vios

ハードウェア VIO のリストを取得します。

構文

```
get_hw_vios [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

ハードウェア VIO

使用法

名前	説明
[-of_objects]	指定したハードウェア デバイスのハードウェア VIO オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ハードウェア VIO オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

ハードウェア、オブジェクト

get_interfaces

現在のデザインの I/O ポートインターフェイスのリストを取得します。

構文

```
get_interfaces [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

インターフェイス オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したピンまたはネットのインターフェイスを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	I/O ポートインターフェイスを検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト

説明

現在のプロジェクトに含まれる I/O インターフェイスで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトに含まれる I/O インターフェイスすべてのリストが取得されます。

注記: メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_interfaces** で返されるオブジェクトのリストに、インターフェイスのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルターパターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!=) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects args (オプション) : インターフェイスが割り当てられているピンまたはネットを 1 つ以上指定します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns : 指定したパターンに一致するインターフェイスを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトに含まれるすべてのインターフェイスのリストが取得されます。

例

次の例では、プロジェクトに含まれるインターフェイスすべてのリストを取得しています。

```
get_interfaces
```

関連項目

- [create_interface](#)
- [delete_interface](#)

get_io_standards

I/O 規格のリストを表示します。

構文

```
get_io_standards [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

I/O 規格

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定した BEL、サイト、パッケージ ピン、I/O バンク、ポートの I/O 規格を取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	I/O 規格を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト

説明

ターゲット デバイスで使用可能な I/O 規格のリストを取得します。

注記 : メモリおよびパフォーマンスを向上するため、`get_*` コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_io_standards** で返されるオブジェクトのリストに、オブジェクトのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。I/O 規格の場合、結果をフィルター処理するのに使用できるプロパティには NAME, IS_DCI, IS_DIFFERENTIAL などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *arg* (オプション) : 指定したパッケージピン、サイト、BEL、I/O バンク、またはポート オブジェクトの I/O 規格を取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する I/O 規格を検索します。デフォルトのパターンはワイルドカード (*) で、ターゲット パーツで使用可能なすべての I/O 規格が取得されます。複数のパターンを指定して、異なる検索条件に基づいて I/O 規格を検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、ターゲット デバイスで使用可能な差動 I/O 規格のリストを取得しています。

```
get_io_standards -filter {IS_DIFFERENTIAL}
```

注記 : パターンに一致するオブジェクトがない場合は、警告メッセージが表示されます。

関連項目

- [list_property](#)
- [report_property](#)

get_iobanks

I/O バンクのリストを取得します。

構文

```
get_iobanks [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

I/O バンクのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したパッケージ ピンの I/O バンクを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	I/O バンクを検索するパターンを指定します。デフォルトは* です。

カテゴリ

XDC、オブジェクト

説明

現在のプロジェクトのターゲット デバイスの I/O バンクで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、ターゲット デバイスの I/O バンクすべてのリストが取得されます。

注記: メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_ipbanks** で返されるオブジェクトのリストに、I/O バンクのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。I/O バンク オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには **DCI CASCADE**、**INTERNAL_VREF** などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が “” のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : 指定したオブジェクトに接続されている I/O バンクのリストを取得します。指定可能なオブジェクト タイプは、パッケージ ピン、ポート、およびサイトです。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する I/O バンクを検索します。デフォルトのパターンはワイルドカード (*) で、デザインの I/O バンクすべてのリストが取得されます。

例

次の例では、指定したパッケージ ピンの I/O バンクが返されます。

```
get_iobanks -of_objects [get_package_pins H4]
```

関連項目

- [get_package_pins](#)
- [get_ports](#)
- [get_sites](#)
- [place_ports](#)
- [report_property](#)

get_ipdefs

現在の IP カタログから IP のリストを取得します。

構文

```
get_ipdefs [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose] [patterns...]
```

戻り値

カタログ IP オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	カタログ IP を検索するパターンを指定します。デフォルトは * です。デフォルトのパターンは、ワイルドカード (*) または -regexp を指定している場合は「.*」です。

カテゴリ

オブジェクト、IP フロー

説明

現在のプロジェクトの IP カタログで定義されている IP コアで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、カタログで定義された IP コアすべてが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_ipdefs** で返されるオブジェクトのリストに、オブジェクトのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。ipdefs オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには VLNV、NAME、IS_AXI などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=^)、不一致 (!^) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : IP カタログで指定したパターンに一致する IP コアを検索します。デフォルトのパターンはワイルドカード (*) で、カタログに含まれるすべての IP コアのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてコア定義を検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、NAME プロパティが指定のパターンに一致する IP コアのリストが返されます。

```
get_ipdefs -filter {NAME=~*agilent*}
```

注記 : フィルター演算子「=^」により、指定のパターンにおおよそ一致する IP コアが検索されます。

次の例では、AXI 準拠の IP コアのリストが返されます。

```
get_ipdefs -filter {IS_AXI==1}
```

関連項目

- [create_ip](#)
- [generate_target](#)
- [get_ips](#)
- [import_ip](#)
- [update_ip_catalog](#)

get_ips

現在のデザインの IP のリストを取得します。

構文

```
get_ips [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose] [patterns ...]
```

戻り値

IP オブジェクトのリスト

使用法

名前	説明
<code>[-regexp]</code>	検索パターンを正規表現で指定します。
<code>[-nocase]</code>	パターンの大文字/小文字を区別せずに検索します。
<code>[-filter]</code>	式を使用してリストをフィルター処理します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>[patterns]</code>	IP 名を検索するパターンを指定します。デフォルトのパターンは、ワイルドカード (*) または <code>-regexp</code> を指定している場合は「.*」です。

カテゴリ

オブジェクト、プロジェクト、IP フロー

説明

現在のプロジェクトに含まれる IP コアで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトの IP すべてのリストが返されます。

注記：メモリおよびパフォーマンスを向上するため、`get_*` コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

`-regexp` (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび `-filter` オプションの式は正規表現で記述する必要があります。ザイレンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記： Tcl ビルトイン コマンド `regexp` はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_ip** で返されるオブジェクトのリストに、オブジェクトのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。IP オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、DELIVERED_TARGETS などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (bool) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するデザイン内の IP コアを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトに含まれるすべての IP コアのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてコアを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、名前が「EDK」という文字列で始まる IP コアのリストが返されます。

```
get_ip EDK*
```

関連項目

- [create_ip](#)
- [generate_target](#)
- [get_ipdefs](#)
- [import_ip](#)
- [update_ip_catalog](#)

get_lib_cells

ライブラリ セルのリストを取得します。

構文

```
get_lib_cells [-regexp] [-filter arg] [-nocase] [-include_unsupported]
[-of_objects args] [-quiet] [-verbose] patterns
```

戻り値

ライブラリ セルのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-filter]	式を使用してリストをフィルター処理します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します。このオプションは、-regexp を使用した場合にのみ適用されます。
[-include_unsupported]	テストのみのライブラリ セルを含めます。
[-of_objects]	指定したオブジェクトのライブラリ セルを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
patterns	ライブラリ セル名を検索するパターンを指定します。

カテゴリ

オブジェクト

説明

現在のデザインのターゲット パーツのライブラリに含まれるセルのリストを取得します。このコマンドを使用すると、特定のライブラリ セルやセル タイプを検索して、そのセルのプロパティを取得できます。

このコマンドを実行するには、ライブラリ名とセル名 (lib_name/cell_name) を含む階層名が必要です。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_lib_cells** で返されたオブジェクトのリストに、セルのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

プール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : 指定のインスタンス (cells/insts) またはライブラリ ピン (get_lib_pins) のライブラリ セルのリストを取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (必須) : 指定したパターンに一致するライブラリ セルを検索します。パターンには、ライブラリ名とセル名の両方を指定する必要があります。

例

次の例では、現在のデザインのターゲット パーツのライブラリのセル数を取得し、そのライブラリの AND タイプのセル数を取得しています。

```
llength [get_lib_cells [get_libs]/*]  
795  
llength [get_lib_cells [get_libs]/AND*]  
18
```

次の例では、指定したセル オブジェクトのライブラリ セルを取得しています。

```
get_lib_cells -of_objects [lindex [get_cells] 1]
```

関連項目

- [get_cells](#)
- [get_libs](#)
- [get_lib_pins](#)
- [list_property](#)
- [report_property](#)

get_lib_pins

ライブラリ セル ピンのリストを取得します。

構文

```
get_lib_pins [-regexp] [-filter arg] [-nocase] [-of_objects args] [-quiet]
[-verbose] patterns
```

戻り値

ライブラリ セル ピンのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-filter]	式を使用してリストをフィルター処理します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します。このオプションは、-regexp を使用した場合にのみ適用されます。
[-of_objects]	指定したオブジェクトのライブラリ セル ピンを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
patterns	ライブラリ セル ピン名を検索するパターンを、<ライブラリ セル パターン>/<ライブラリ セル ピン パターン> の形式で指定します。

カテゴリ

オブジェクト

説明

現在のデザインのターゲット パーツのセル ライブラリから、指定したセルのピンのリストを取得します。

注記 : このコマンドを実行するには、ライブラリ名、セル名、ピンを含む階層名が必要です (lib_name/cell_name/pins)。

注記 : メモリおよびパフォーマンスを向上するため、get_* コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。append を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_lib_pins** で返されるオブジェクトのリストに、ピンのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

プール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : 指定したピン オブジェクトまたはライブラリ セル (**get_lib_cells**) のライブラリ セル ピンのリストを取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (必須) : 指定したパターンに一致するピンを検索します。パターンには、ライブラリ名、セル名、およびピンを指定する必要があります。

例

次の例では、ライブラリ セル ピンすべてのリストを取得しています。

```
get_lib_pins xt_vortex6/AND2/*
```

次の例では、ターゲット デバイスのセル ライブラリに含まれるすべてのセルの、ピンすべてのリストを取得しています。

```
get_lib_pins [get_libs]/*/*
```

関連項目

- [get_libs](#)
- [get_lib_cells](#)
- [list_property](#)
- [report_property](#)

get_libs

ライブラリのリストを取得します。

構文

```
get_libs [-regexp] [-filter arg] [-nocase] [-quiet] [-verbose] [patterns]
```

戻り値

ライブラリのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-filter]	式を使用してリストをフィルター処理します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します。このオプションは、-regexp を使用した場合にのみ適用されます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ライブラリ名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト

説明

現在のデザインのターゲット デバイスのセル ライブラリを取得します。デバイス ファミリによって使用できるプリミティブは異なるので、デバイス ファミリごとに 1 つのライブラリがあります。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、`get_libs` で返されるオブジェクトのリストに、ライブラリのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルターパターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!=) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : 指定したオブジェクトのライブラリのリストを取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するライブラリを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトに含まれるすべてのライブラリのリストが取得されます。

例

次の例では、ターゲット パーツのセル ライブラリを取得しています。

```
get_libs
```

関連項目

- [get_lib_cells](#)
- [get_lib_pins](#)
- [list_property](#)
- [report_property](#)

get_macros

現在のデザインに含まれるマクロのリストを取得します。

構文

```
get_macros [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

マクロ オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したセルのマクロを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	マクロ名を検索するパターンを指定します。デフォルトは*です。

カテゴリ

XDC、オブジェクト

説明

現在のデザインに含まれるマクロで、検索パターンに一致するもののリストを取得します。デフォルトでは、デザインのすべてのマクロのリストが返されます。

注記: メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_macros** で返されるオブジェクトのリストに、マクロのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。マクロ オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、ABSOLUTE_GRID、RLOCS などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *arg* (オプション) : 指定したピンまたはネット オブジェクトに接続されているマクロを取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するマクロを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトに含まれるすべてのマクロのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてマクロを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定の検索パターンに一致するマクロに現在割り当てられているプロパティを返しています。

```
report_property [get_macro *Macro1]
```

注記 : パターンに一致するマクロがない場合は、警告メッセージが表示されます。

関連項目

- [create_macro](#)
- [list_property](#)
- [report_property](#)
- [update_macro](#)

get_msg_config

現在のメッセージ カウント、最大表示回数、または set_msg_config コマンドで定義されたメッセージ制御ルールを返します。

構文

```
get_msg_config [-id arg] [-severity arg] [-rules] [-limit] [-count] [-quiet]
[-verbose]
```

戻り値

なし

使用法

名前	説明
[-id]	メッセージ ID を指定します。-limit または -count と共に使用する必要があります。デフォルトは空です。
[-severity]	メッセージの重要度を指定します。-limit または -count と共に使用する必要があります。デフォルトは空です。
[-rules]	現在のプロジェクトに設定されているメッセージ制御ルールをすべて示す表を表示します。
[-limit]	-id または -severity に一致するメッセージの最大表示回数を表示します。
[-count]	-id または -severity に一致するメッセージのこれまでに生成された回数を表示します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

指定のメッセージ ID または重要度の現在のメッセージ カウントまたは最大表示回数、あるいは現在のプロジェクトで定義されているすべてのメッセージ制御ルールを返します。メッセージ制御ルールは、set_msg_config コマンドで定義します。

-count を使用すると、指定の ID または重要度のメッセージがこれまでに生成された回数が表示されます。

-limit を使用すると、指定の ID または重要度のメッセージに現在設定されている最大表示回数が表示されます。

-rules を使用すると、現在定義されているすべてのメッセージ制御ルールが表で示されます。

注記：1 つの `get_msg_config` コマンドでは、最大表示回数、現在のカウント、ルールのいずれかのみを返すことができます。複数指定すると、エラーが返されます。

引数

-id arg (オプション)：メッセージ ID を指定します。各メッセージには、独自の ID が含まれます。たとえば、「Common 17-54」、「Netlist 29-28」です。

-severity value (オプション)：メッセージの重要度を指定します。次の 5 種類の重要度があります。

- **ERROR**：デザインの結果が使用不可となったり、ユーザーの操作がないと回避できないような問題が発生していることを示すエラー メッセージです。
- **{CRITICAL WARNING}**：入力や制約に適用されないものがあったり、FPGA ファミリには適していないものがあることを示すクリティカル警告メッセージです。修正することが強く推奨されます。

注記：これは 2 単語の値なので、{} で囲む必要があります。

- **WARNING**：制約または仕様が指定どおりに適用されず、デザインが最適な結果にならない可能性を示す警告メッセージです。修正するかどうかは、ユーザーが判断します。
- **INFO**：STATUS メッセージと同じですが、重要度とメッセージ ID タグが含まれる点が異なります。INFO メッセージには、必要に応じてアンサー データベースで確認できるようにメッセージ ID が含まれます。
- **STATUS**：デザインの処理に関するプロセスおよびフィードバックのステータスを示すステータス メッセージです。STATUS メッセージには、メッセージ ID は含まれません。

-rules (オプション)：現在のプロジェクトで `set_msg_config` コマンドを使用して定義されているメッセージ制御ルールを返します。

注記：-rule を指定すると、-id または -severity の指定にかかわらず、現在のプロジェクトの制御ルールがすべて返されます。

-limit (オプション)：指定したメッセージ ID または重要度のメッセージの現在の最大表示回数を返します。

-count (オプション)：指定したメッセージ ID または重要度のメッセージの現在のカウントを返します。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、`set_msg_config` コマンドで定義できます。

例

次の例では、指定した情報メッセージの現在のカウントを取得しています。

```
get_msg_config -id "Common 17-81" -count
```

次の例では、現在のプロジェクトで定義されているメッセージ制御ルールを返しています。

```
get_msg_config -rules
```

関連項目

- [reset_msg_config](#)
- [set_msg_config](#)

get_msg_count

メッセージ カウントを取得します。

構文

```
get_msg_count [-severity arg] [-id arg] [-quiet] [-verbose]
```

戻り値

メッセージ数

使用法

名前	説明
[-severity]	メッセージの重要度 (「ERROR」、「CRITICAL WARNING」など) を指定します。-id と共に使用することはできません。デフォルトは ALL です。
[-id]	「Common 17-99」など、メッセージの ID を指定します。-severity と共に使用することはできません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

指定した重要度またはメッセージ ID のメッセージが、ツールが起動されてから表示された回数を取得します。ツールで表示される各メッセージには、アプリケーション サブシステム コードとメッセージ識別子を含む固有のメッセージ ID が付けられています。次に、メッセージ ID の例を示します。

```
"Common 17-54"
"Netlist 29-28"
"Synth 8-3295"
```

このコマンドを実行すると、メッセージのこれまでに表示された回数がメッセージの最大表示回数にどれくらい近いかを確認できます。メッセージの現在の最大表示回数は、**get_msg_limit** コマンドで確認できます。メッセージの最大表示回数は、**set_msg_limit** コマンドで変更できます。

デフォルトでは、すべてのメッセージの合計カウントが返されます。指定した重要度のメッセージや、指定した ID のメッセージ カウントのみを取得することもできます。

引数

-severity value (オプション) : メッセージの重要度を指定します。次の 5 種類の重要度があります。

- **ERROR** : デザインの結果が使用不可となったり、ユーザーの操作がないと回避できないような問題が発生していることを示すエラー メッセージです。
- **{CRITICAL WARNING}** : 入力や制約に適用されないものがあったり、FPGA ファミリには適していないものがあることを示すクリティカル警告メッセージです。修正することが強く推奨されます。

注記 : これは 2 単語の値なので、{} または "" で囲む必要があります。

- **WARNING** : 制約または仕様が指定どおりに適用されず、デザインが最適な結果にならない可能性を示す警告メッセージです。修正するかどうかは、ユーザーが判断します。
- **INFO** : STATUS メッセージと同じですが、重要度とメッセージ ID タグが含まれる点が異なります。INFO メッセージには、必要に応じてアンサー データベースで確認できるようにメッセージ ID が含まれます。
- **STATUS** : デザインの処理に関するプロセスおよびフィードバックのステータスを示すステータス メッセージです。STATUS メッセージには、メッセージ ID は含まれません。

-id value (オプション) : ツールのメッセージ ビューやほかのレポートに表示されるメッセージ ID を指定します。特定のメッセージ ID を指定すると、そのメッセージのカウントを取得できます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、すべてのメッセージのカウントを取得しています。

```
get_msg_count -severity ALL
get_msg_count
```

注記 : -severity または -id を指定しない場合、デフォルトですべてのメッセージのカウントが返されるので、上記のどちらの行も同じ結果になります。

次の例では、指定した ID のメッセージ カウントを取得しています。

```
get_msg_count -id "Netlist 29-28"
```

関連項目

- [get_msg_limit](#)
- [set_msg_limit](#)

get_msg_limit

メッセージの最大表示回数を取得します。

構文

```
get_msg_limit [-severity arg] [-id arg] [-quiet] [-verbose]
```

戻り値

メッセージの制限数

使用法

名前	説明
[-severity]	メッセージの重要度 (「ERROR」、「CRITICAL WARNING」など) を指定します。-id と共に使用することはできません。デフォルトは ALL です。
[-id]	「Common 17-99」など、メッセージの ID を指定します。-severity と共に使用することはできません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

ツールの起動中に表示されるメッセージの最大表示回数をレポートします。定義した最大表示回数に到達すると、メッセージは表示されなくなります。デフォルト値は 4,294,967,295 です。このデフォルト値は、**set_msg_limit** コマンドで変更できます。

ツールで表示される各メッセージには、アプリケーション サブシステム コードとメッセージ識別子を含む固有のメッセージ ID が付けられています。次に、メッセージ ID の例を示します。

```
"Common 17-54"
"Netlist 29-28"
"Synth 8-3295"
```

引数

-id arg (オプション) : 最大表示回数を返すメッセージ ID を指定します。たとえば、「Common 17-54」、「Netlist 29-28」です。

-severity value (オプション) : メッセージの重要度を指定します。次の 5 種類の重要度があります。

- **ERROR** : デザインの結果が使用不可となったり、ユーザーの操作がないと回避できないような問題が発生していることを示すエラー メッセージです。
 - **{CRITICAL WARNING}** : 入力や制約に適用されないものがあったり、FPGA ファミリには適していないものがあることを示すクリティカル警告メッセージです。修正することが強く推奨されます。
- 注記** : これは 2 単語の値なので、{} で囲む必要があります。
- **WARNING** : 制約または仕様が指定どおりに適用されず、デザインが最適な結果にならない可能性を示す警告メッセージです。修正するかどうかは、ユーザーが判断します。
 - **INFO** : STATUS メッセージと同じですが、重要度とメッセージ ID タグが含まれる点が異なります。INFO メッセージには、必要に応じてアンサー データベースで確認できるようにメッセージ ID が含まれます。
 - **STATUS** : デザインの処理に関するプロセスおよびフィードバックのステータスを示すステータス メッセージです。STATUS メッセージには、メッセージ ID は含まれません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、CRITICAL WARNING メッセージの最大表示回数を返しています。

```
get_msg_limit -severity {CRITICAL WARNING}
```

-severity または -id を使用しない場合、デフォルトですべてのメッセージの最大表示回数が返されます。

次の例では、指定したメッセージ ID の最大表示回数を返しています。

```
get_msg_limit -id "Netlist 29-28"
```

関連項目

- [set_msg_limit](#)
- [set_msg_severity](#)

get_net_delays

ドライバーから各ロード ピンまでのネットの配線遅延または予測遅延を取得します。

構文

```
get_net_delays [-regexp] [-nocase] [-filter arg] [-of_objects args] [-through]
[-to args] [-quiet] [-verbose] [patterns]
```

戻り値

ネット遅延

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したネットのネット遅延を取得します。
[-through]	指定の PIP を介する遅延を取得します。デフォルトでは、PIP 遅延は含まれません。
[-to]	指定のロードまでのネット遅延を取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ネット遅延を検索するパターンを指定します。デフォルトは * です。

カテゴリ

タイミング、ネットリスト、オブジェクト

get_nets

現在のデザインのネットのリストを取得します。

構文

```
get_nets [-hsc arg] [-hierarchical] [-regexp] [-nocase] [-filter arg]
[-of_objects args] [-match_style arg] [-top_net_of_hierarchical_group]
[-segments] [-boundary_type arg] [-quiet] [-verbose] [patterns]
```

戻り値

ネット オブジェクトのリスト

使用法

名前	説明
[-hsc]	階層区切り文字を指定します。デフォルトはスラッシュ (/) です。
[-hierarchical]	すべての階層レベルで検索します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したピン/ポート、セル、タイミング パス、またはクロックのネットを取得します。
[-match_style]	パターン一致のスタイルを指定します。有効な値は ucf、sdc で、デフォルトは sdc です。
[-top_net_of_hierarchical_group]	階層ネットの上位に属するネット セグメントを返します。
[-segments]	階層全体のネットのセグメントをすべて返します。
[-boundary_type]	ピンと同じレベル (upper)、下のレベル (lower)、またはその両方 (both) にある階層ピンに接続されているネット セグメントを返します。有効な値は upper、lower、both で、デフォルトは upper です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ネット名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

SDC、XDC、オブジェクト

説明

現在のデザインに含まれるネットで、検索パターンに一致するもののリストを取得します。デフォルトでは、デザインに含まれるすべてのネットのリストが返されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト（セル、ネット、ピン、ポートなど）のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-hsc arg (オプション)：デフォルトの階層区切り文字はスラッシュ (/) です。それ以外の階層区切り文字を指定する場合は、このオプションを使用します。

-hierarchical (オプション) : 現在のインスタンスから開始し、デザイン階層のすべてのレベルからネットを取得します。このオプションを指定しない場合、**current_instance** コマンドで設定された現在のインスタンスのネットのみが取得されます。**-hierarchical** を使用する場合、検索パターンは完全な階層ネット名ではなく階層の各レベルに適用されるので、検索パターンには階層区切り文字を含めないでください。たとえば、検索パターンとして「U1/*」を指定すると、名前に「U1/」を含むネットが階層の各レベルで検索され、意図した結果が得られない可能性があります。**-hierarchical** を使用した検索の例は、「[get_cells](#)」を参照してください。

注記: **-regexp** と共に使用する場合、検索パターンは完全な階層名と比較され、検索パターンとして「U1/*」を指定した場合に意図した結果が得られます。

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイレンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記: Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション)：結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、`get_nets` で返されるオブジェクトのリストに、ネットのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、`report_property` または `list_property` コマンドで確認できます。ネットオブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには PARENT、TYPE、MARK_DEBUG などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記： フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「REFSET」が含まれない入力ポートを取得しています。

```
get ports * -filter {DIRECTION == IN && NAME =~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *arg* (オプション) : 指定したセル、ピン、ポート、またはクロックに接続されているネットのリストを取得します。

注記 : **-of_objects** を **-hierarchy** または検索パターンと共に使用することはできません。

-match_style [sdc | ucf] (オプション) : 検索パターンが UCF 制約または SDC 制約に一致することを示します。デフォルトは SDC です。

-top_net_of_hierarchical_group (オプション) : 階層ネットの最上位ネットセグメントまたはすべてのネットの最上位ネットセグメントを取得します。このオプションを **-segment** と共に使用すると、階層ネットのすべてのセグメントから最上位ネットセグメントが返されます。

-segments (オプション) : 階層ネットのすべての階層レベルにあるすべてのセグメントを取得します。このオプションは、指定のネットだけではなく、指定のネットのすべてのセグメントを返す点が **-hierarchical** オプションとは異なります。

-boundary_type [upper | lower | both] (オプション) : 指定の階層ピンのレベル (upper)、ピンまたはポートの下のレベル (lower)、またはその両方のネットセグメントを取得します。有効な値は、upper、lower、both です。デフォルト値は upper です。

注記 : このオプションを使用する場合、**-of_objects** オプションを使用して階層ピンを指定する必要があります。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンと一致するネットを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトに含まれるすべてのネットのリストが返されます。複数のパターンを指定して、異なる検索条件に基づいてネットを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定したセルに接続されているネットのリストを取得しています。

```
get_nets -of_objects [lindex [get_cells] 1]
```

注記 : パターンに一致するネットがない場合は、警告メッセージが表示されます。

次の例では、connect_debug_port コマンドでデバッグ用にマークされたネットのリストが返されます。

```
get_nets -hier -filter {MARK_DEBUG==1}
```

次の例では、指定の階層ピン オブジェクトに接続されているネットを返し、ピン オブジェクトに接続されているネット セグメントを返して、最後にピン オブジェクトに接続されている最上位ネット セグメントを返しています。

```
get_nets -of [get_pins cpuEngine/or1200_cpu/or1200_sprs/esr_reg[9]_i_3/I0]
cpuEngine/or1200_cpu/or1200_sprs/flagforw
get_nets -segments -of [get_pins cpuEngine/or1200_cpu/or1200_sprs/esr_reg[9]_i_3/I0]
cpuEngine/or1200_cpu/or1200_alu/flagforw cpuEngine/or1200_cpu/flagforw cpuEngine/or1200_cpu/or1200_sprs/flagforw
get_nets -top -segments -of \
[get_pins cpuEngine/or1200_cpu/or1200_sprs/esr_reg[9]_i_3/I0]
cpuEngine/or1200_cpu/flagforw
```

関連項目

- [connect_debug_port](#)
- [get_cells](#)
- [get_clocks](#)
- [get_pins](#)
- [get_ports](#)
- [list_property](#)
- [report_property](#)

get_nodes

デバイスのノードのリストを取得します。

構文

```
get_nodes [-of_objects args] [-regexp] [-nocase] [-filter arg] [-uphill]
[-downhill] [-flyover] [-from args] [-to args] [-quiet] [-verbose] [patterns]
```

戻り値

ノード

使用法

名前	説明
[-of_objects]	指定したネット、タイル、ノード、BEL ピン、サイトピン、ワイヤ、PIP のノード オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-uphill]	-of_objects で指定したサイトピン、PIP、ノード、またはタイルからドライバー側のノードを取得します。
[-downhill]	-of_objects で指定したサイトピン、PIP、ノード、またはタイルからロード側のノードを取得します。
[-flyover]	指定したタイルを飛び越すノードを取得します。
[-from]	指定した PIP またはサイトピンから開始するノードを取得します。-uphill と共に使用できます。デフォルトでは -downhill が使用されます。-all が自動的に使用されます。
[-to]	指定したワイヤまたはサイトピンで終了するノードを取得します。-uphill と共に使用できます。デフォルトでは -downhill が使用されます。-all が自動的に使用されます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ノード オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、[XDC](#)

説明

開いているデザインのデバイスのノードで、指定した検索パターンに一致するものをリストします。

デフォルトでは、デバイスのノードすべてのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト（セル、ネット、ピン、ポートなど）のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション)：検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記：Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション)：パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args：結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_nodes** で返されるオブジェクトのリストに、ノードのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。ノード オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには **IS_INPUT_PIN**、**IS_BEL_PIN**、**NUM_WIRES** などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記：フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が “” のプロパティに一致します。

文字列比較では、フィルターパターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!=) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (bool) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects args (オプション)：指定したサイトピン、ノード、タイル、またはワイヤのノードを取得します。

注記：**-of_objects** を検索パターンと共に使用することはできません。

-uphill (オプション)：**-of_objects** オプションで指定したオブジェクトのドライバー側（ロジック ネットワークの前方）にあるノードを返します。

-downhill (オプション)：**-of_objects** オプションで指定したオブジェクトのロード側（ロジック ネットワークの後方）にあるノードを返します。

-flyover (オプション) : 指定のタイルを通過 (フライオーバー) するノードを返します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンと一致するノードを検索します。デフォルトのパターンはワイルドカード (*) で、デバイスのノードすべてのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてノードを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定のタイルに関連するノードが返されます。

```
get_nodes -of_objects [get_tiles CLBLM_R_X11Y158]
```

次の例では、指定のノードのロード側にあるノードが返されます。

```
get_nodes -downhill -of_objects [get_nodes LIOB33_SING_X0Y199/IOB_PADOUT0]
```

関連項目

- [get_nodes](#)
- [get_site_pins](#)
- [get_tiles](#)
- [get_wires](#)
- [list_property](#)
- [report_property](#)

get_objects

1つまたは複数の HDL スコープで、指定したパターンに一致する HDL オブジェクトのリストを取得します。

構文

```
get_objects [-filter arg] [-recursive] [-r] [-regexp] [-nocase] [-quiet]
[-verbose] [patterns...]
```

戻り値

指定したパターンに一致するすべてのオブジェクト

使用法

名前	説明
[-filter]	結果のリストに指定した式のフィルターを適用します。
[-recursive]	サブスコープでも検索を実行します。
[-r]	サブスコープでも検索を実行します。
[-regexp]	検索パターンが正規表現で記述されていることを指定します。デザイン オブジェクトを供給するアプリケーションで、検索の実行方法が決定されます。文字列を指定する必要があります。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	検索パターンを指定します。デフォルトは * で、すべての HDL オブジェクトが返されます。

カテゴリ

シミュレーション

説明

1つまたは複数の HDL スコープで、指定したパターンに一致する HDL オブジェクトのリストを取得します。

HDL オブジェクトには、Verilog または VHDL テストベンチおよびソース ファイルで定義されている HDL 信号、変数、または定数が含まれます。HDL 信号には、Verilog の wire または reg エンティティ、および VHDL 信号が含まれます。HDL 変数には、Verilog の real、realtime、time、event などがあります。HDL 定数には、Verilog のパラメーターおよび localparam、VHDL ジェネリックおよび定数が含まれます。

HDL スコープは、Verilog のモジュール、関数、タスク、プロセス、begin-end ブロックなど、HDL コードの宣言部分で定義されます。VHDL スコープには、エンティティ/アーキテクチャ定義、関数、プロシージャ、およびプロセス ブロックが含まれます。

引数

-recursive | -r (オプション) : 現在のスコープおよびそのサブスコープすべてにコマンドを適用します。

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_cells** で返されたオブジェクトのリストに、セルのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。HDL オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、SCOPE、TYPE などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が “” のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=^)、不一致 (!^) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : HDL オブジェクトを検索するパターンを指定します。デフォルトのパターンはワイルドカード (*) で、現在のスコープのすべての HDL オブジェクトが返されます。検索パターンは、次の 2 つの方法で定義できます。

- *patterns* : 取得するオブジェクトの検索パターンのみを指定します。この方法では、現在のスコープおよび **-recursive** を使用している場合はそのサブスコープに含まれるオブジェクトが返されます。
- *scope/pattern* : スコープ (現在のスコープを基準に指定) と検索パターンを指定します。この場合、現在のスコープから指定の *scope* および **-recursive** を使用している場合はそのサブスコープが特定され、検索 *pattern* に一致するオブジェクトが返されます。

例

次の例では、現在のスコープを指定し、そのスコープに含まれるすべての HDL オブジェクトを取得しています。

```
current_scope ./cpuEngine
get_objects
```

次の例では、現在のスコープに含まれるオブジェクト数を返し、現在のスコープとそのサブスコープに含まれるすべてのオブジェクトの数を返しています。

```
llength [get_objects]
182
llength [get_objects -recursive ]
2182
```

次の例では、*scope/pattern* 検索パターンを指定しています。cpuEngine スコープおよびそのサブスコープが特定され、検索パターン **cl*** に一致するオブジェクトが返されます。

```
get_objects -filter {type == internal_signal} cpuEngine/cl* -recursive
  /top/cpuEngine/clk_i
  /top/cpuEngine/iwb_biu/clk
  /top/cpuEngine/iwb_biu/clmode
  /top/cpuEngine/or1200_cpu/clk
  ...
  /top/cpuEngine/or1200_immu_top/or1200_immu_tlb/itlb_mr_ram/clk
```

次の例では、現在のスコープとそのサブスコープで **cl** または **ma** で開始する内部信号を返しています。

```
get_objects -filter {type == internal_signal} ma* cl* -recursive
```

関連項目

- [current_scope](#)
- [list_property](#)
- [report_objects](#)
- [report_property](#)

get_package_pins

パッケージ ピンのリストを取得します。

構文

```
get_package_pins [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

パッケージ ピン オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したサイト、I/O バンク、ポートのパッケージ ピン オブジェクトのリストを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	パッケージ ピンを検索するパターンを指定します。デフォルトは * です。

カテゴリ

XDC、オブジェクト

説明

ターゲット デバイスに対して選択したパッケージのピンのリストを取得します。デフォルトでは、パッケージのすべてのピンのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_package_pins** で返されるオブジェクトのリストに、ピンのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。ピンオブジェクトの場合、結果をフィルターできるプロパティには IS_CLK_CAPABLE、IS_VREF、IS_GLOBAL_CLK などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *arg* (オプション) : 指定したオブジェクトに接続されているパッケージ ピンを取得します。指定可能なオブジェクトはサイト、I/O バンク、またはポートです。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するピンを検索します。デフォルトのパターンはワイルドカード (*) で、パッケージのすべてのピンが返されます。複数のパターンを指定して、異なる検索条件に基づいてピンを検索できます。

例

次の例では、ターゲット デバイスのピンすべてのリストを取得しています。

```
get_package_pins
```

次の例では、パッケージのクロック兼用 (CC) ピンの数を取得しています。

```
llength [get_package_pins -filter {IS_CLK_CAPABLE==1}]
```

注記 : パターンに一致するピンがない場合は、警告メッセージが表示されます。

関連項目

- [get_jobanks](#)
- [get_sites](#)
- [list_property](#)
- [report_property](#)

get_param

パラメーター値を取得します。

構文

```
get_param [-quiet] [-verbose] name
```

戻り値

パラメーター値

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	パラメータ名を指定します。

カテゴリ

プロパティおよびパラメーター

説明

指定したツール パラメーターに現在定義されている値を取得します。これらのパラメーターは、ツールのさまざまな動作を制御するためのユーザー定義可能なコンフィギュレーション設定です。各パラメーターが何をコンフィギュレーションまたは制御するかは、**report_param** を参照してください。

引数

name (必須) : 値を取得するパラメーターの名前を指定します。ユーザー定義可能なパラメーターのリストは、**list_param** を実行すると確認できます。このコマンドでは、パラメーターの完全な名前を指定する必要があります。パターン一致は実行されず、1 つのパラメーターのみ指定可能です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、マルチスレッド プロセスで使用される MaxThreads パラメーターの現在の値が返されます。

```
get_param general.MaxThreads
```

関連項目

- [list_param](#)
- [report_param](#)
- [reset_param](#)
- [set_param](#)

get_parts

ツールで使用可能なパートのリストを取得します。

構文

```
get_parts [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose] [patterns]
```

戻り値

パート オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	パート名を検索するパターンを指定します。デフォルトは * です。デフォルトのパターンは、ワイルドカード (*) または -regexp を指定している場合は「.*」です。

カテゴリ

オブジェクト

説明

現在のプロジェクトのパートで、検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトのパートすべてのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_parts** で返されるオブジェクトのリストに、ペーツのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。ペーツ オブジェクトの場合、結果をフィルター処理できるプロパティには DEVICE、FAMILY、SPEED などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するパーツを検索します。デフォルトのパターンはワイルドカード (*) で、すべてのパーツのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてパーツを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、7vx485t パーツでスピード グレード -1 のもののリストを取得しています。

```
get_parts -filter {DEVICE =~ xc7vx485t && speed == -1}
```

次の例では、7 シリーズおよび Virtex-6 パーツの数を取得しています。

```
llength [get_parts -regexp {xc7v.* xc6v.*} -nocase]
```

注記 : パターンに一致するパーツがない場合は、警告メッセージが表示されます。

関連項目

- [list_property](#)
- [report_property](#)

get_path_groups

現在のデザインのパス グループのリストを取得します。

構文

```
get_path_groups [-regexp] [-nocase] [-quiet] [-verbose] [patterns]
```

戻り値

パス グループのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	パス グループ名と一致するパターンを指定します。デフォルトは * です。

カテゴリ

XDC、オブジェクト

説明

現在のプロジェクトのタイミング パス グループで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、デザインのすべてのパス グループのリストが取得されます。

デザインで新しいクロックを作成すると、そのクロックドメイン内のすべてのパスを含むパス グループが自動的に作成されます。パス グループは、**group_path** コマンドを使用して手動で作成することもできます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するパス グループを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトのすべてのパス グループが取得されます。

例

次の例では、デザインのすべてのパス グループのリストを取得しています。

```
get_path_groups
```

次の例では、名前に Clk という文字列を含むパス グループを取得しています。

```
get_path_groups *Clk*
```

注記 : パターンに一致するパス グループがない場合は、警告メッセージが表示されます。

関連項目

[group_path](#)

get_pblocks

現在のデザインの Pblock のリストを取得します。

構文

```
get_pblocks [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

Pblock オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したセルのライブラリを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	Pblock 名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、フロアプラン、XDC

説明

現在のプロジェクトで定義されている Pblock で、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトの Pblock すべてのリストが取得されます。

注記 : メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_pblocks** で返されるオブジェクトのリストに、Pblock のプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。Pblock オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、GRID_RANGES などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects arg (オプション) : 指定のセルが割り当てられている Pblocks を取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する Pblock を検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトのすべての Pblock が返されます。

例

次の例では、現在のプロジェクトの Pblock すべてのリストを取得しています。

```
get_pblocks
```

次の例では、スライス範囲が指定されていない Pblock のリストを取得しています。

```
get_pblocks -filter {GRIDTYPES !~ SLICE}
```

次の例では、指定したセルの Pblock の割り当てを取得しています。

```
get_pblocks -of [get_cells CORE/BR_TOP/RLD67_MUX/REG_PMBIST_C1]
```

関連項目

- [create_pblock](#)
- [get_cells](#)

get_pins

現在のデザインのピンのリストを取得します。

構文

```
get_pins [-hsc arg] [-hierarchical] [-regexp] [-nocase] [-leaf] [-filter arg]
[-of_objects args] [-match_style arg] [-quiet] [-verbose] [patterns]
```

戻り値

ピン オブジェクトのリスト

使用法

名前	説明
[-hsc]	階層区切り文字を指定します。デフォルトはスラッシュ (/) です。
[-hierarchical]	すべての階層レベルで検索します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-leaf]	-of_objects で指定したネットの下位/グローバルピンを取得します。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したセル、ネット、タイミング パス、またはクロックのピンを取得します。
[-match_style]	パターン一致のスタイルを指定します。有効な値は ucf、sdc で、デフォルトは sdc です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ピン名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

SDC、XDC、オブジェクト

説明

現在のデザインに含まれるピン オブジェクトで、検索パターンに一致するもののリストを取得します。デフォルトでは、デザインに含まれるすべてのピンのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-hsc arg (オプション) : デフォルトの階層区切り文字はスラッシュ (/) です。それ以外の階層区切り文字を指定する場合は、このオプションを使用します。

-hierarchical (オプション) : デザイン階層のすべてのレベルからピンを取得します。このオプションを指定しない場合、デザイン階層の最上位のピンのみが取得されます。**-hierarchical** を使用する場合、検索パターンは完全な階層セル名ではなく階層の各レベルに適用されるので、検索パターンには階層区切り文字を含めないでください。たとえば、検索パターンとして「U1/*」を指定すると、名前に「U1/」を含むインスタンスが階層の各レベルで検索され、意図した結果が得られない可能性があります。**-hierarchical** を使用した検索の例は、「[get_cells](#)」を参照してください。

注記 : **-regexp** と共に使用する場合、検索パターンは完全な階層名と比較され、検索パターンとして「U1/*」を指定した場合に意図した結果が得られます。

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイレンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-leaf (オプション) : **-of_object** オプションで指定したオブジェクトに対して、プリミティブまたはブラック ボックスセルからの下位ピンを含めます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、[get_pins](#) で返されるオブジェクトのリストに、ピンのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。ピン オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには PARENT、TYPE などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が “” のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=^)、不一致 (!^) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}
```

-of_objects arg (オプション) : 指定したセル、ポート、またはクロックに接続されるピンを取得します。

注記 : **-of_objects** を **-hierarchy** または検索パターンと共に使用することはできません。

-match_style [sdc | ucf] (オプション) : 検索パターンが UCF 制約または SDC 制約に一致することを示します。デフォルトは SDC です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するピンを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトに含まれるすべてのピンのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてピンを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定したセルに接続されているピンのリストが取得されます。

```
get_pins -of_objects [lindex [get_cells] 1]
```

注記 : パターンに一致するピンがない場合は、警告メッセージが表示されます。

関連項目

- [list_property](#)
- [report_property](#)

get_pips

現在のデバイスのプログラマブル インターコネクト ポイント (PIP) のリストを取得します。

構文

```
get_pips [-regexp] [-nocase] [-filter arg] [-of_objects args] [-uphill]
[-downhill] [-from args] [-to args] [-quiet] [-verbose] [patterns]
```

戻り値

PIP

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したサイト、タイル、ワイヤ、ノード、PIP、またはネットの PIP を取得します。
[-uphill]	指定したワイヤまたは PIP からドライバー側の PIP を取得します。
[-downhill]	指定したワイヤまたは PIP からロード側の PIP を取得します。
[-from]	指定した PIP またはサイトピンから開始する PIP の順序付きリストを取得します。-uphill と共に使用できます。デフォルトでは -downhill が使用されます。-all が自動的に使用されます。
[-to]	指定したワイヤまたはサイトピンで終了する PIP の順序付きリストを取得します。-uphill と共に使用できます。デフォルトでは -downhill が使用されます。-all が自動的に使用されます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	PIP を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、[XDC](#)

説明

プログラマブル インターコネクト ポイント (PIP) は、ロジック ネットワークを接続するのに使用する物理配線パスです。このコマンドは、デバイス上の PIP で、指定した検索パターンに一致するものをリストします。このコマンドを実行するには、デザインを開いておく必要があります。

デフォルトでは、デバイス上の PIP すべてのリストが取得されます。ただし、デバイス上には PIP が多数あるので、このコマンドをデフォルトで実行することはお勧めしません。**-of_objects** オプションを使用して返される PIP の数を制限してください。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト（セル、ネット、ピン、ポートなど）のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション)：検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイレンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記：Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション)：パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション)：結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_pips** で返されるオブジェクトのリストに、PIP のプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。PIP オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには **IS_DIRECTIONAL**、**FROM_PIN** があります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記：フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が “” のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!=) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects args (オプション)：指定したサイト、タイル、またはワイヤ オブジェクトの PIP を取得します。

注記：このオプションを常に使用し、**get_pips** コマンドのランタイムとメモリ使用量を制限することをお勧めします。そうしないと、返される PIP の数が極端に大きくなる可能性があります。**-of_objects** オプションを検索パターンと共に使用することはできません。

-uphill (オプション) : 指定したワイヤまたは PIP のドライバー側 (ロジック ネットワークの前方) にある PIP を返します。

-downhill (オプション) : 指定したワイヤまたは PIP のロード側 (ロジック ネットワークの後方) にある PIP を返します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する PIP を検索します。デフォルトのパターンはワイルドカード (*) で、デバイス上の PIP すべてのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいて PIP を検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定のタイルに関連する PIP が返されます。

```
get_pips -of_object [get_tiles DSP_R_X9Y75]
```

関連項目

- [get_sites](#)
- [get_tiles](#)
- [get_wires](#)
- [list_property](#)
- [report_property](#)

get_ports

現在のデザインに含まれるポートのリストを取得します。

構文

```
get_ports [-regexp] [-nocase] [-filter arg] [-of_objects args]
[-match_style arg] [-quiet] [-verbose] [patterns]
```

戻り値

ポート オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したネット、インスタンス、サイト、クロック、タイミング パス、I/O 規格、I/O バンク、パッケージ ピンのポートを取得します。
[-match_style]	パターン一致のスタイルを指定します。有効な値は ucf、sdc で、デフォルトは sdc です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ポート名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

SDC、XDC、オブジェクト

説明

現在のデザインに含まれるポート オブジェクトで、検索パターンに一致するもののリストを取得します。デフォルトでは、デザインに含まれるすべてのポートのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、`get_ports` で返されるオブジェクトのリストに、ポートのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。ports オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには PARENT、TYPE などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が “” のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}  
ポート名
```

プール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}  
ポート名
```

-of_objects arg (オプション) : 指定したセル、ネット、クロック、またはタイミング パス オブジェクトに接続されているポートを取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-match_style [sdc | ucf] (オプション) : 検索パターンが UCF 制約または SDC 制約に一致することを示します。デフォルトは SDC です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するポートを検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトに含まれるすべてのポートのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてポートを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定したセルに接続されているピンのリストが取得されます。

```
get_ports -of_objects [lindex [get_cells] 1]
```

注記 : パターンに一致するポートがない場合は、警告メッセージが表示されます。

関連項目

- [list_property](#)
- [report_property](#)

get_projects

プロジェクトのリストを取得します。

構文

```
get_projects [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose] [patterns]
```

戻り値

プロジェクト オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	プロジェクト名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、プロジェクト

説明

現在開いているプロジェクトで、検索パターンに一致するもののリストを取得します。デフォルトでは、開いているプロジェクトすべてのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_projects** で返されるオブジェクトのリストに、プロジェクトのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。プロジェクト オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、DIRECTORY、TARGET_LANGUAGE などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=^)、不一致 (!^) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するプロジェクトを返します。デフォルトのパターンはワイルドカード (*) で、すべてのパートのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてプロジェクトを検索できます。

例

次の例では、開いているプロジェクトすべてのリストを取得しています。

```
get_projects
```

次の例では、project_found という変数が get_projects コマンドで返されるプロジェクトリストの長さに設定され、そのプロジェクトが検出されたか、検出されなかったかが表示されます。

```
set project_found [llength [get_projects ISC*] ]  
if {$project_found > 0} {puts "Project Found."} else {puts "No Projects Found."}
```

注記： パターンに一致するプロジェクトがない場合は、警告メッセージが表示されます。

関連項目

- [create_project](#)
- [current_project](#)
- [open_project](#)

get_property

オブジェクトのプロパティを取得します。

構文

```
get_property [-quiet] [-verbose] name object
```

戻り値

プロパティ値

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	値を取得するプロパティの名前を指定します。
<i>object</i>	プロパティを取得するオブジェクトを指定します。

カテゴリ

オブジェクト、プロパティおよびパラメーター、XDC

説明

指定したオブジェクトの指定したプロパティの値を取得します。指定のオブジェクトに指定のプロパティが設定されていない場合、または値が設定されていない場合は、何も返されません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 値を取得するプロパティの名前を指定します。大文字/小文字は区別されません。

object (必須) : オブジェクトを指定します。

例

次の例では、指定したセルの NAME プロパティを取得しています。

```
get_property NAME [lindex [get_cells] 3]
```

関連項目

- [create_property](#)
- [get_cells](#)
- [list_property](#)
- [list_property_value](#)
- [report_property](#)
- [reset_property](#)
- [set_property](#)

get_runs

run (実行パターン) のリストを取得します。

構文

```
get_runs [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose] [patterns]
```

戻り値

run オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	run 名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、プロジェクト

説明

現在のプロジェクトの合成 run およびインプリメンテーション run で、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、プロジェクトで定義されている run すべてのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get *** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイレンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記： Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、`get_runs` で返されるオブジェクトのリストに、`run` のプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。`run` オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには `CONSTSET`、`IS_IMPLEMENTATION`、`IS_SYNTHESIS`、`FLOW` などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が “” のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=^)、不一致 (!^) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する `run` を検索します。デフォルトのパターンはワイルドカード (*) で、プロジェクトで定義されているすべての `run` のリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいて `run` を検索できます。

例

次の例では、現在のプロジェクトで完了していない `run` すべてのリストを取得しています。

```
get_runs -filter {PROGRESS < 100}
```

次の例では、指定のパターンに一致する `run` のリストを取得しています。

```
get_runs imp*
```

注記 : パターンに一致する `run` がない場合は、警告メッセージが表示されます。

関連項目

- [create_run](#)
- [current_run](#)
- [report_property](#)

get_scopes

スコープの子 HDL スコープのリストを取得します。

構文

```
get_scopes [-filter arg] [-regexp] [-nocase] [-recursive] [-r] [-quiet]
[-verbose] [patterns...]
```

戻り値

HDL スコープ オブジェクト

使用法

名前	説明
[-filter]	結果のリストに指定した式のフィルターを適用します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-recursive]	子スコープでも検索を実行します (glob または -regexp を指定した場合のみ有効)。
[-r]	子スコープでも検索を実行します (glob または -regexp を指定した場合のみ有効)。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	スコープを検索するパターン文字列を指定します。デフォルトは * で、すべての子スコープが取得されます。

カテゴリ

[シミュレーション](#)

get_selected_objects

選択したオブジェクトを取得します。

構文

```
get_selected_objects [-primary] [-quiet] [-verbose]
```

戻り値

選択したオブジェクトのリスト

使用法

名前	説明
[-primary]	選択規則に従って選択されたオブジェクトは含みません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

オブジェクト、GUI 制御

説明

現在 Vivado IDE で **select_objects** コマンドにより選択されているオブジェクトを取得します。プライマリ選択オブジェクトとセカンダリ選択オブジェクトの両方を取得できます。

注記：この Tcl コマンドは、Vivado を GUI モードで実行している場合にのみ機能します。

プライマリ オブジェクトとは直接選択されたオブジェクトで、セカンダリ オブジェクトとは [Tools] → [Options] → [Selection Rules] で現在定義されている選択規則に従って選択されたオブジェクトを指します。選択規則の設定方法は、『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。

引数

-primary (オプション)：プライマリ選択オブジェクトのみを返します。セカンダリオブジェクトは返しません。デフォルトでは、現在選択されているオブジェクトがすべて返されます。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在選択されているすべてのオブジェクト (プライマリとセカンダリの両方) のプロパティがレポートされます。

```
report_property [get_selected_objects]
```

関連項目

[select_objects](#)

get_site_pins

サイト ピンのリストを取得します。

構文

```
get_site_pins [-of_objects args] [-regexp] [-nocase] [-filter arg] [-quiet]
[-verbose] [patterns]
```

戻り値

サイト ピン

使用法

名前	説明
[-of_objects]	指定したサイト、XDEF サイト、ノード、ピン、ネットのサイト ピン オブジェクトを取得します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	サイト ピン オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、XDC

説明

開いているデザインに含まれる指定のサイト、ノード、論理セル ピン、またはネット オブジェクトのサイト ピンのリストを返します。

このコマンドでは、実行時間を手短縮し、計算リソースを削減するため、**-of_objects** オプションを使用することをお勧めします。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-of_objects args (オプション) : 指定したサイト、ノード、ピン、またはネット オブジェクトのサイトピンを取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_site_pins** で返されるオブジェクトのリストに、サイトピンのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。サイトピン オブジェクトの場合、結果をフィルターできるプロパティには IS_CLOCK、IS_DATA、IS_PART_OF_BUS などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するサイトピンを検索します。デフォルトのパターンはワイルドカード (*) で、指定したオブジェクトのサイトピンすべてのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてサイトピンを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定したネットのサイトピンが返されます。

```
get_site_pins -of_objects [get_nets *Clk]
IOB_X1Y24/I
```

次の例では、サイト SLICE_X21Y92 に関連する出力サイトピンが返されます。

```
get_site_pins -of_objects [get_sites SLICE_X21Y92] -filter {DIRECTION==OUT}
SLICE_X21Y92/A SLICE_X21Y92/AMUX SLICE_X21Y92/AQ SLICE_X21Y92/B SLICE_X21Y92/BMUX SLICE_X21Y92/
BQ SLICE_X21Y92/C SLICE_X21Y92/CMUX SLICE_X21Y92/COUT SLICE_X21Y92/CQ SLICE_X21Y92/D SLICE_X21Y92/
DMUX SLICE_X21Y92/DQ
```

関連項目

- [get_nets](#)
- [get_nodes](#)
- [get_pins](#)
- [get_sites](#)
- [list_property](#)
- [report_property](#)

get_site_pips

指定したオブジェクトのサイト PIP のリストを取得します。

構文

```
get_site_pips [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

サイト PIP

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したサイトのサイト PIP を取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	サイト PIP を検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、[xdc](#)

説明

プログラマブルインターフェクト ポイント (PIP) は、ロジック ネットワークを接続するのに使用する物理配線パスです。このコマンドは、指定のサイトの PIP で、指定した検索パターンに一致するものをリストします。このコマンドを実行するには、デザインを開いておく必要があります。

このコマンドでは、**-of_objects** オプションを使用して PIP を取得するサイトを指定する必要があります。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_site_pips** で返されるオブジェクトのリストに、プロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。PIP オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには **IS_DIRECTIONAL**、**FROM_PIN** があります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *args* (オプション) : このオプションは **get_sites** コマンドと共に使用でき、指定したサイトの PIP を返します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する PIP を検索します。デフォルトのパターンはワイルドカード (*) で、**-of_objects** で指定したサイトの PIP すべてのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてピンを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、デバイス上の指定したサイト範囲に関連する指定した BEL のピンが返されます。

```
get_site_pips -of_objects [get_sites SLICE_X21Y92]
```

関連項目

- [get_sites](#)
- [list_property](#)
- [report_property](#)

get_sites

サイトのリストを取得します。

構文

```
get_sites [-regexp] [-filter arg] [-nocase] [-range args] [-of_objects args]
[-quiet] [-verbose] [patterns]
```

戻り値

サイト オブジェクトのリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-filter]	式を使用してリストをフィルター処理します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-range]	サイトを検索する範囲を指定します。範囲は、2 つのサイト名で定義します。
[-of_objects]	指定したオブジェクトのサイトを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	サイトを検索するパターンを指定します。ボンディングされたサイトも、パッケージ ピン名に一致します。デフォルトは * です。

カテゴリ

XDC、オブジェクト

説明

ターゲット デバイスのサイトで、指定した検索パターンに一致するもののリストを取得します。デフォルトでは、ターゲット デバイスのサイトすべてのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、`get_*` コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、`get_sites` で返されるオブジェクトのリストに、サイトのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。サイト オブジェクトの場合、結果をフィルター処理できるプロパティには SITE_TYPE、IS_OCCUPIED、NUM_INPUTS、NUM_OUTPUTS などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (bool) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-range arg (オプション) : 指定の範囲内のサイトを取得します。サイトの範囲は、同じ SITE_TYPE の 2 つのサイト値 (例 : {SLICE_X2Y12 SLICE_X3Y15}) で指定します。サイトの SITE_TYPE は、**report_property** コマンドで確認できます。

注記 : タイプの異なる 2 つのサイトで範囲を指定すると、エラーが発生します。

-of_objects arg (オプション) : 指定したオブジェクトのサイトを取得します。指定可能なオブジェクトは、タイル、BEL、ピン、パッケージピン、ポート、Pblock、I/O バンク、セル、およびクロック領域です。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するサイトを検索します。デフォルトのパターンはワイルドカード (*) で、ターゲット デバイスのサイトすべてのリストが取得されます。

例

次の例では、ターゲット デバイスのサイトすべてのリストを取得しています。

```
get_sites
```

次の例では、ターゲット デバイスで使用されていないサイト数を取得しています。

```
llength [get_sites -filter {IS_OCCUPIED==0}]
```

注記 : パターンに一致するサイトがない場合は、警告メッセージが表示されます。

次の例では、デバイス上のすべてのサイトが取得され、固有の SITE_TYPE が返されます。

```
set sites [get_sites]
set type {}
foreach x $sites {
    set prop [get_property SITE_TYPE $x]
    if { [lsearch -exact $type $prop] == -1 } {
        lappend type $prop
    }
}
foreach y $type {
    puts "SITE_TYPE: $y"
}
```

次の例では、3 つのサイト範囲の指定方法を示しています。

```
get_sites -range {SLICE_X0Y0 SLICE_X1Y1}
SLICE_X0Y0 SLICE_X0Y1 SLICE_X1Y0 SLICE_X1Y1
get_sites -range SLICE_X0Y0 -range SLICE_X1Y1
SLICE_X0Y0 SLICE_X0Y1 SLICE_X1Y0 SLICE_X1Y1
get_sites -range {SLICE_X0Y0:SLICE_X1Y1}
SLICE_X0Y0 SLICE_X0Y1 SLICE_X1Y0 SLICE_X1Y1
```

関連項目

- [get_cells](#)
- [get_property](#)
- [list_property](#)
- [report_property](#)

get_slrs

SLR (Super Logic Region) のリストを取得します。

構文

```
get_slrs [-regexp] [-filter arg] [-nocase] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

SLR のリスト

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-filter]	式を使用してリストをフィルター処理します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-of_objects]	指定したオブジェクトの SLR を取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	SLR 名を検索するパターンを指定します。デフォルトは* です。

カテゴリ

XDC、オブジェクト

説明

ターゲットデバイスの SLR (Super Logic Region) のリストを取得します。複数の SLR が含まれないデバイスでは、SLR0 が返されます。

注記 : メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_slrs** で返されるオブジェクトのリストに、SLR のプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。SLR オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NUM_CHANNELS、NUM_SLLS、NUM_TILES などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *arg* (オプション) : **get_cells** コマンドで指定したセル オブジェクトが割り当てられている SLR を取得します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致する SLR を検索します。デフォルトのパターンはワイルドカード (*) で、現在のデザインに含まれるすべての SLR のリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいて SLR を検索できます。

注記 : 複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、現在のデザインのターゲット デバイスに含まれるすべての SLR が返されます。

```
get_slrs
```

注記 : パターンに一致するセルがない場合は、警告メッセージが表示されます。

関連項目

- [list_property](#)
- [report_property](#)

get_tiles

タイルのリストを取得します。

構文

```
get_tiles [-regexp] [-nocase] [-filter arg] [-of_objects args] [-quiet]
[-verbose] [patterns]
```

戻り値

タイル

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したサイト、BEL、サイトピン、BEL ピン、ノード、ワイヤ、PIP、ネット、クロック領域のタイルを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	タイルを検索するパターンを指定します。デフォルトは*です。

カテゴリ

オブジェクト、[XDC](#)

説明

開いているデザインのデバイス上にあるタイルのリストを返します。デフォルトでは、デバイス上のタイルすべてのリストが取得されます。

注記: メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトインコマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter *args* (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_tiles** で返されるオブジェクトのリストに、タイル オブジェクトのプロパティに基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。タイル オブジェクトの場合、結果をフィルター処理できるプロパティには NUM_ARCS、NUM_SITES、IS_GT_SITE_TILE などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!=?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects *args* (オプション) : 指定したサイト、BEL、サイトピン、ノード、ワイヤ、または PIP のタイルを取得します。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するタイルを検索します。デフォルトのパターンはワイルドカード (*) で、デバイス上のタイルすべてのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてタイルを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、タイミング アークの数がそれぞれ 100 および 150 を超えるタイルの数が返されます。

```
llength [get_tiles -filter {NUM_ARCS>100} ]  
13468  
llength [get_tiles -filter {NUM_ARCS>150} ]  
11691
```

関連項目

- [get_bels](#)
- [get_nodes](#)
- [get_pips](#)
- [get_site_pins](#)
- [get_sites](#)
- [get_wires](#)
- [list_property](#)
- [report_property](#)

get_timing_arcs

タイミング アークのリストを取得します。

構文

```
get_timing_arcs [-from args] [-to args] [-filter arg] [-of_objects args]
[-quiet] [-verbose]
```

戻り値

タイミング アーク オブジェクトのリスト

使用法

名前	説明
[-from]	タイミング アークの開始点を指定します。
[-to]	タイミング アークの終点を指定します。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したのセルのタイミング アークを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、XDC、オブジェクト、タイミング

説明

指定したオブジェクトのタイミング アークのリストを取得します。指定のプロパティに基づいて、タイミング アークをフィルター処理できます。

タイミング アークはタイミング パスの一部で、2 つのピン間のワイヤであったり、入力ピンと出力ピンの間にあるロジック インスタンスの内部パスであったりします。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-from args (オプション) : タイミング アークの開始点を指定します。ポート、ピン、またはセルを開始点として指定できます。また、クロック オブジェクトを指定すると、そのクロックが供給されるオブジェクトが開始点となるタイミング アークが返されます。

-to args (オプション) : タイミング アークの終点を指定します。ポート、ピン、またはセル オブジェクトを終点として指定できます。また、クロック オブジェクトを指定すると、そのクロックが供給されるオブジェクトが終点となるタイミング アークが返されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_timing_arcs** で返されるオブジェクトのリストに、オブジェクトのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。タイミング アーク オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには FROM_PIN、TO_PIN、LIB_CELL などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=)、不一致 (!=) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

プール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects args (オプション) : 指定したセル オブジェクトからのタイミング アークを取得します。セルを指定すると、そのセルのすべてのセル アークが返されます。

注記 : **-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、指定したバッファーの出力ピンからのタイミング アークが返されます。

```
report_property -all [get_timing_arcs -of_objects [get_cells go_IBUF_inst]]
```

次の例では、指定したセルのタイミング アークが返されます。

```
get_timing_arcs -of_objects [get_cells count_reg[6]]
{count_reg[6]/C --> count_reg[6]/Q [Reg Clk to Q] }
{count_reg[6]/C --> count_reg[6]/D [setup] }
{count_reg[6]/C --> count_reg[6]/D [hold] }
{count_reg[6]/C --> count_reg[6]/CLR [recovery] }
{count_reg[6]/C --> count_reg[6]/CE [hold] }
{count_reg[6]/C --> count_reg[6]/CLR [removal] }
{count_reg[6]/C --> count_reg[6]/CE [setup] }
{count_reg[6]/CLR --> count_reg[6]/Q [Reg Set/Clr] }
```

関連項目

- [report_timing](#)
- [set_msg_limit](#)

get_timing_paths

タイミング パスを取得します。

構文

```
get_timing_paths [-from args] [-rise_from args] [-fall_from args]
[-to args] [-rise_to args] [-fall_to args] [-through args]
[-rise_through args] [-fall_through args] [-delay_type arg] [-setup] [-hold]
[-max_paths arg] [-nworst arg] [-unique_pins] [-slack_lesser_than arg]
[-slack_greater_than arg] [-group args] [-no_report_unconstrained]
[-user_ignored] [-sort_by arg] [-filter arg] [-regexp] [-nocase]
[-match_style arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-from]	タイミング パスの開始点 (ピン、ポート、セル、またはクロック) を指定します。
[-rise_from]	ピン、ポート、セル、またはクロックの立ち上がりエッジを開始点として指定します。
[-fall_from]	ピン、ポート、セル、またはクロックの立ち下がりエッジを開始点として指定します。
[-to]	タイミング パスの終点 (ピン、ポート、セル、またはクロック) を指定します。
[-rise_to]	ピン、ポート、セル、またはクロックの立ち上がりエッジを終点として指定します。
[-fall_to]	ピン、ポート、セル、またはクロックの立ち下がりエッジを終点として指定します。
[-through]	タイミング パスの通過点 (ピン、ポート、セル、またはネット) を指定します。
[-rise_through]	ピン、ポート、セル、またはネットの立ち上がりエッジを通過点として指定します。
[-fall_through]	ピン、ポート、セル、またはネットの立ち下がりエッジを通過点として指定します。
[-delay_type]	パス遅延のタイプを指定します。有効な値は max、min、min_max、max_rise、max_fall、min_rise、min_fall で、デフォルトは max です。
[-setup]	最大遅延タイミング パスを取得します (-delay_type max と同じ)。
[-hold]	最小遅延タイミング パスを取得します (-delay_type min と同じ)。
[-max_paths]	出力するパスの最大数を指定します。1 以上の値を指定します。デフォルト値は 1 です。

名前	説明
<code>[-nworst]</code>	エンドポイントまでのワーストパスを N 値をリストします。1 以上の値を指定します。デフォルト値は 1 です。
<code>[-unique_pins]</code>	ピンの各固有のセットに対して、パス グループに 1 つのパスを表示します。
<code>[-slack_lesser_than]</code>	この値よりも小さいスラックのパスを含めます。デフォルトは $1e+30$ です。
<code>[-slack_greater_than]</code>	この値よりも大きいスラックのパスを含めます。デフォルトは $-1e+30$ です。
<code>[-group]</code>	指定のグループのパスのみを返します。
<code>[-no_report_unconstrained]</code>	制約が適用されていないパスは取得しません。
<code>[-user_ignored]</code>	<code>set_false_path</code> または <code>set_clock_groups</code> タイミング制約のためにスラックが無限のパスのみをレポートします。
<code>[-sort_by]</code>	パスの並べ替え順を指定します。有効な値は <code>group</code> 、 <code>slack</code> で、デフォルトは <code>slack</code> です。
<code>[-filter]</code>	式を使用してリストをフィルター処理します。
<code>[-regexp]</code>	検索パターンを正規表現で指定します。
<code>[-nocase]</code>	パターンの大文字/小文字を区別せずに検索します (<code>-regexp</code> を指定した場合のみ有効)。
<code>[-match_style]</code>	パターン一致のスタイルを指定します。有効な値は <code>ucf</code> 、 <code>sdc</code> で、デフォルトは <code>ucf</code> です。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。

カテゴリ

[SDC](#)、[オブジェクト](#)、[タイミング](#)

説明

指定の条件を満たすタイミング パス オブジェクトを取得します。`report_timing` に渡すタイミング パスを定義するためなどに使用できます。また、このコマンドを使用してカスタム レポートおよび解析を作成できます。

`get_timing_paths` コマンドは `report_timing` コマンドと似ていますが、`report_timing` でがファイルまたは文字列を返すのに対し、`get_timing_paths` はタイミング パス オブジェクトを返すので、そのプロパティを取得したり、ほかの Tcl コマンドに渡したりできます。

注記：メモリおよびパフォーマンスを向上するため、`get_*` コマンドでは 1 つのタイプのオブジェクト（セル、ネット、ピン、ポートなど）のコンテナーリストが返されます。`lappend` を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-from args (オプション) : 解析するタイミング パスの開始点を指定します。ポート、ピン、またはセルをタイミング パスの開始点として指定できます。また、クロック オブジェクトを指定すると、そのクロックが供給されるオブジェクトが開始点として指定されます。

-rise_from args (オプション) : **-from** オプションと同様ですが、指定した開始点からの信号の立ち上がりエッジのみをタイミング 解析で考慮します。クロック オブジェクトを指定すると、そのクロックの立ち上がりエッジで駆動されるパスのみが開始点として考慮されます。

-fall_from args (オプション) : **-from** オプションと同様ですが、指定した開始点からの信号の立ち下がりエッジのみをタイミング 解析で考慮します。クロック オブジェクトを指定すると、そのクロックの立ち下がりエッジで駆動されるパスのみが開始点として考慮されます。

-to args (オプション) : 解析するタイミング パスの終点 (デスティネーション オブジェクト) を指定します。ポート、ピン、またはセル オブジェクトを終点として指定できます。また、クロック オブジェクトを指定すると、そのクロックが供給されるオブジェクトが終点として指定されます。

-rise_to args (オプション) : **-to** オプションと同様ですが、指定した終点に到達する信号の立ち上がりエッジのみをタイミング 解析で考慮します。クロック オブジェクトを指定すると、そのクロックの立ち上がりエッジでキャプチャされるパスのみが終点として考慮されます。

-fall_to args (オプション) : **-to** オプションと同様ですが、指定した終点に到達する信号の立ち下がりエッジのみをタイミング 解析で考慮します。クロック オブジェクトを指定すると、そのクロックの立ち下がりエッジでキャプチャされるパスのみが終点として考慮されます。

-through args (オプション) : 指定したピン、セル インスタンス、またはネットを通過するパスのみをタイミング 解析で考慮します。個別の **-through** (または **-rise_through** および **-fall_through**) 点を順次指定し、デザインを通過する特定のパスを定義できます。特定のパスを定義するには、通過点の指定順序が重要です。通過点を複数のオブジェクトで指定することもできます。この場合、指定の通過オブジェクトのいずれかを通過するタイミング パスが考慮されます。

-rise_through args (オプション) : **-through** オプションと同様ですが、指定したオブジェクトで立ち上がるパスのみでタイミング 解析を実行します。

-fall_through args (オプション) : **-through** オプションと同様ですが、指定したオブジェクトで立ち下がるパスのみでタイミング 解析を実行します。

-delay_type arg (オプション) : タイミング レポートを実行する際に解析に使用する遅延のタイプを指定します。有効な値は min、max、min_max、max_rise、max_fall、min_rise、min_fall です。デフォルトは max です。

-setup (オプション) : セットアップ違反がないかどうかをチェックします。これは **-delay_type max** を指定するのと同じです。

-hold (オプション) : ホールド違反がないかどうかをチェックします。これは **-delay_type min** を指定するのと同じです。

注記 : **-setup** と **-hold** の両方を指定すると、**-delay_type min_max** を指定するのと同じになります。

-max_paths arg (オプション) : **-sort_by** での指定の応じて、スラック順に並べた場合に出力するパスの最大数、またはグループごとに並べた場合にパス グループごとに出力するパスの最大数を指定します。1 以上の値を指定します。デフォルトは 1 で、ワースト タイミング パス 1 つ、またはグループごとにワースト パス 1 つがレポートされます。

-nworst arg (オプション) : 各エンドポイントに対して表示するタイミング パス数を指定します。タイミング レポートには、指定した数のワースト パスが表示されます。有効な値は 1 以上で、デフォルト値は 1 です。

-slack_greater_than arg (オプション) : 算出されたスラック値が指定した値より大きいパスのタイミングをレポートします。**-slack_lesser_than** と共に使用すると、スラック値の範囲を指定できます。

-slack_lesser_than arg (オプション) : 算出されたスラック値が指定した値より小さいパスのタイミングをレポートします。**-slack_greater_than** と共に使用すると、スラック値の範囲を指定できます。

-group arg (オプション) : 指定したパス グループのパスのタイミングをレポートします。

-no_report_unconstrained (オプション) : 制約が適用されていないパスのタイミングはレポートしません。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_timing_paths** で返されるオブジェクトのリストに、オブジェクトのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。タイミング パス オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには **DATAPATH_DELAY**、**ENDPOINT_PIN**、**ENDPOINT_CLOCK** などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

次の例では、100 個のクリティカル パス オブジェクトを取得し、パス グループ **clk_tx_clk_core_1** のもののみを取得しています。

```
get_timing_paths -max_paths 100 -filter {GROUP == clk_tx_clk_core_1}
```

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=?)、不一致 (!?) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

ブール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-regexp (オプション) : 検索パターンが正規表現で記述していることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイレンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-match_style [sdc | ucf] (オプション) : 検索パターンが UCF 制約または SDC 制約に一致することを示します。デフォルト値は UCF です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、指定した終点からのワースト タイミング パスを 5 つを取得し、リストの 4 つ目のタイミング パスのプロパティをすべてレポートしています。

```
report_property -all [lindex [get_timing_paths -to [get_ports led_pins[*]]\n-nworst 5] 3]
```

次の例では、**custom_report** というプロシージャを定義し、そのプロシージャを使用して **clk_tx_clk_core_1** パス グループの 100 個のワースト パスをレポートしています。

```
proc custom_report { listOfPaths } {\n    puts [format {%-40s %-40s %-20s %-20s %7s} "Startpoint" "Endpoint" "Launch Clock" "Capture Clock" "Slack"]\n    puts [string repeat "-" 140]\n    foreach path $listOfPaths {\n        set startpoint [get_property STARTPOINT_PIN $path]\n        set startclock [get_property STARTPOINT_CLOCK $path]\n        set endpoint [get_property ENDPOINT_PIN $path]\n        set endclock [get_property ENDPOINT_CLOCK $path]\n        set slack [get_property SLACK $path]\n        puts [format {%-40s %-40s %-20s %-20s %7s} $startpoint $endpoint $startclock $endclock $slack]\n    }\n}\nset paths [get_timing_paths -group clk_tx_clk_core_1 -max_paths 100]\ncustom_report $paths
```

次の例では、タイミング パス オブジェクトを **report_timing** コマンドで使用する方法を示しています。

```
set paths [get_timing_paths -group clk_tx_clk_core_1 -max_paths 100]\nreport_timing -of_objects $paths
```

これは、次のコマンドと同等です。

```
report_timing -group clk_tx_clk_core_1 -max_paths 100
```

関連項目

- [report_property](#)
- [report_timing](#)

get_value

選択した HDL オブジェクト (変数、信号、ワイヤ、レジスタ) の現在の値を取得します。

構文

```
get_value [-radix arg] [-quiet] [-verbose] hdl_object
```

戻り値

HDL オブジェクトの値

使用法

名前	説明
[-radix]	HDL オブジェクトの値の表示に使用する基底を指定します。有効な値は、default、dec、bin、oct、hex、unsigned、ascii です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
hdl_object	現在の値を取得する HDL オブジェクトを指定します。

カテゴリ

シミュレーション

説明

現在のシミュレーション時間における 1 つの HDL オブジェクトの値を取得します。

注記：複数の HDL オブジェクトの値を取得する場合は、**report_values** コマンドを使用してください。

HDL オブジェクトには、Verilog または VHDL テストベンチおよびソース ファイルで定義されている HDL 信号、変数、または定数が含まれます。HDL 信号には、Verilog の wire または reg エンティティ、および VHDL 信号が含まれます。HDL 変数には、Verilog の real、realtime、time、event などがあります。

HDL 定数には、Verilog のパラメーターおよび localparam、VHDL ジェネリックおよび定数が含まれます。HDL スコープは、Verilog のモジュール、ファンクション、タスク、プロセス、begin-end ブロックなど、HDL コードの宣言部分で定義されます。VHDL スコープには、エンティティ/アーキテクチャ定義、ファンクション、プロシージャ、およびプロセス ブロックが含まれます。

引数

-radix arg (オプション)：指定したオブジェクトの値を表示するのに使用する基底を指定します。有効な値は、default、dec、bin、oct、hex、unsigned、ascii です。

注記：dec は、符号付き 10 進数を示します。符号なしデータの場合は、unsigned を指定してください。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

hdl_object (必須) : 値を取得する 1 つの HDL オブジェクトの名前を指定します。オブジェクトは、名前または **get_objects** コマンドで指定します。

例

次の例では、sysClk 信号の値を取得しています。

```
get_value sysClk
    Z
```

次の例では、指定したバスから返された値の基数 **bin**、**dec**、および **unsigned** による違いを示しています。

```
get_value -radix bin /test/bench_VStatus_pad_0_i[7:0]
    10100101
get_value -radix unsigned /test/bench_VStatus_pad_0_i[7:0]
    165
get_value -radix dec /test/bench_VStatus_pad_0_i[7:0]
    -91
```

関連項目

- [current_time](#)
- [get_objects](#)
- [set_value](#)
- [report_values](#)

get_wave_configs

指定のオプションに一致する波形コンフィギュレーションを取得します。

構文

```
get_wave_configs [-regexp] [-nocase] [-filter arg] [-quiet] [-verbose]
[patterns ...]
```

戻り値

指定のオプションに一致する波形コンフィギュレーション

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	結果のリストに指定した式のフィルターを適用します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	波形コンフィギュレーション名を検索するパターン文字列を指定します。

カテゴリ

波形

get_wires

ワイヤのリストを取得します。

構文

```
get_wires [-regexp] [-nocase] [-filter arg] [-of_objects args] [-uphill]
[-downhill] [-from args] [-to args] [-quiet] [-verbose] [patterns]
```

戻り値

ワイヤ

使用法

名前	説明
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
[-of_objects]	指定したタイル、ノード、PIP、またはネットのワイヤを取得します。
[-uphill]	指定した PIP からドライバー側のワイヤを取得します。
[-downhill]	指定した PIP からロード側のワイヤを取得します。
[-from]	指定した PIP またはサイトピンから開始するワイヤの順序付きリストを取得します。-uphill と共に使用できます。デフォルトでは -downhill が使用されます。-all が自動的に使用されます。
[-to]	指定したワイヤまたはサイトピンで終了するワイヤの順序付きリストを取得します。-uphill と共に使用できます。デフォルトでは -downhill が使用されます。-all が自動的に使用されます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	ワイヤを検索するパターンを指定します。デフォルトは * です。

カテゴリ

オブジェクト、XDC

説明

開いているデザインに含まれるワイヤで、指定した検索パターンに一致するものをリストします。

デフォルトでは、デザインに含まれるすべてのワイヤのリストが取得されます。

注記：メモリおよびパフォーマンスを向上するため、**get_*** コマンドでは 1 つのタイプのオブジェクト (セル、ネット、ピン、ポートなど) のコンテナーリストが返されます。**lappend** を使用するなどしてリストにオブジェクトを追加できますが、現在リストに含まれるオブジェクトと同じタイプのオブジェクトしか追加できません。リストに異なるタイプのオブジェクトや文字列を追加しようとすると、Tcl エラーが返されます。

引数

-regexp (オプション)：検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記：Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション)：パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション)：結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、**get_wires** で返されるオブジェクトのリストに、ワイヤのプロパティ値に基づいてフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。フィルターには、任意のプロパティと値の組み合わせを使用できます。ワイヤ オブジェクトの場合、結果をフィルター処理するのに使用できるプロパティには NAME、NUM_DOWNHILL_PIPS、NUM_UPHILL_PIPS などがあります。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記：フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が “” のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

プール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-of_objects args (オプション)：指定したノード、PIP、またはタイルのワイヤを取得します。

注記：**-of_objects** を検索パターンと共に使用することはできません。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns (オプション) : 指定したパターンに一致するワイヤを検索します。デフォルトのパターンはワイルドカード (*) で、デザインに含まれるすべてのワイヤのリストが取得されます。複数のパターンを指定して、異なる検索条件に基づいてワイヤを検索できます。

注記 : 複数の検索パターンは中かっこ ({}) またはダブルクオーテーション (") で囲み、1 つのエレメントとして指定します。

例

次の例では、指定のタイルに関連するワイヤが返されます。

```
get_wires -of_objects [get_tiles IO_INT_INTERFACE_L_X0Y198]
```

関連項目

- [get_nodes](#)
- [get_pips](#)
- [get_tiles](#)
- [list_property](#)
- [report_property](#)

group_bd_cells

階層セルを作成し、セルのグループをその階層セルに移動します。セル間の接続は保持されます。これらのセルとその他のセルの接続は、階層セルをまたぐことにより保持されます。

構文

```
group_bd_cells [-prefix arg] [-quiet] [-verbose] [target_cell_name]
[cells...]
```

戻り値

正しく処理された場合は 0

使用法

名前	説明
[-prefix]	セルに追加する接頭辞を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[target_cell_name]	ターゲット セルを指定します。
[cells]	セル名を検索するパターンを指定します。デフォルトは * です。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインに新しい階層モジュールを作成し、指定したセルをそのモジュール内に移動します。

指定したセルのグループを階層モジュールに移動することもできます。指定したセル間の接続は保持されます。移動するセル間の接続は保持されます。移動するセルとその他の移動されないセルの間の接続は、IP インテグレーターにより階層の境界をまたぐためにピンおよびポートを追加することにより自動的に保持されます。

階層モジュールを **create_bd_cells** コマンドを使用して作成し、**move_bd_cells** コマンドを使用してその階層モジュールにセルを移動することも可能です。

このコマンドが正常に実行された場合は作成された階層モジュールの名前が返され、正常に実行されなかった場合はエラー メッセージが返されます。

引数

-prefix arg (オプション) : 階層モジュールに移動するセルに適用する接頭辞を指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

target_cell_name (必須) : 作成する階層モジュールの名前を指定します。

cells (必須) : 現在の IP サブシステム デザインから階層モジュールに移動するセルのリストを指定します。セルの指定には、**get_bd_cells** コマンドを使用します。

例

次の例では、現在の IP インテグレーター サブシステム デザインに新しい階層ブロックを作成し、指定した 3 つのセルに接頭辞を指定してブロック内に移動しています。

```
group_bd_cells -prefix M1_ module1 [get_bd_cells /microblaze_1_xlconcat] \
[get_bd_cells /microblaze_1_axi_intc] [get_bd_cells /proc_sys_reset_1]
```

関連項目

- [get_bd_cells](#)
- [move_bd_cells](#)

group_path

コスト ファンクション 計算用にパスをグループ化します。

構文

```
group_path [-name arg] [-from args] [-to args] [-through args] [-quiet]
[-verbose]
```

戻り値

なし

使用法

名前	説明
[-name]	グループ名を指定します。
[-from]	指定した開始点から開始するパスを含めます。
[-to]	指定した終点で終了するパスを含めます。
[-through]	ピン、セル、またはネットを通過するパスを含めます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、XDC

説明

主にタイミング解析で使用するコスト ファンクション 計算用にパスをグループ化します。タイミング パスは、開始点、終点、または通過点で指定できます。パス グループを作成すると、report_timing コマンドを使用してそのパス グループに対してタイミング解析を実行できます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

デザインで現在定義されているパス グループは、get_path_groups コマンドを使用して確認できます。

引数

-name arg (オプション)：パス グループの名前を指定します。指定したパス グループ名が既に存在する場合は、指定したパスが既存のグループに追加されます。

-from args (オプション)：指定した開始点から開始するパスを含めます。開始点はピン、ポート、またはクロックで指定できます。

-to path_names (オプション)：指定した終点で終了するパスを含めます。終点はピン、ポート、またはクロックで指定できます。

-through *element_names* (オプション) : 指定したピン、セル、またはネットを通過するパスを含めます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、「*signal*reg/D」に一致するレジスタが終点となるパスを含む signal_grp という名前のパス グループを作成し、そのグループのタイミングをレポートしています。

```
group_path -to [get_pins *signal*reg/D -hierarchical] -name signal_grp
report_timing -group signal_grp
```

パス グループ signal_grp は、**get_path_groups** コマンドでも返されます。

```
get_path_groups
signal_grp
```

関連項目

- [get_path_groups](#)
- [report_timing](#)

help

1つまたは複数の項目に対するヘルプを表示します。

構文

```
help [-category arg] [-args] [-syntax] [-long] [-prop arg] [-class arg]
[-quiet] [-verbose] [pattern_or_object]
```

戻り値

なし

使用法

名前	説明
[-category]	コマンドを検索するカテゴリを指定します。
[-args]	引数の説明を表示します。
[-syntax]	構文の説明を表示します。
[-long]	詳細な説明を表示します。
[-prop]	指定したプロパティのヘルプを表示します。デフォルトは*です。
[-class]	指定したオブジェクトのクラスのプロパティに関する情報を表示します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[pattern_or_object]	指定したパターンに一致するトピックのヘルプを表示します。デフォルトは*です。

カテゴリ

[プロジェクト](#)

説明

指定した Tcl コマンドの説明、ザイリンクス Tcl コマンドのカテゴリのリスト、または指定したパターンに一致するコマンドのリストを返します。

引数を指定せずに **help** を使用すると、Tcl コマンドのカテゴリのリストが返されます。コマンド カテゴリは、File I/Oなどの特定のファンクションを実行するコマンド グループです。

help コマンドで使用するオプションによって、コマンド構文のみ、コマンド構文と各引数の簡単な説明、またはコマンドの詳細な説明と使用例を表示できます。

Vivado Design Suite のメモリ使用量を制限するため、ツールの一部の機能はその機能が使用されるときにのみメモリに読み込まれます。ある機能に関連するすべての Tcl コマンドのリストとヘルプ テキストを表示するには、**load_features** コマンドを使用してその機能をメモリに読み込む必要があります。

help コマンドでは、デザイン オブジェクトに設定可能なさまざまなプロパティに関する情報も表示できます。プロパティに関する情報を表示するには、**-prop** および **-class** オプションを使用してください。

このコマンドでは、指定のヘルプ テキストまたはエラーが返されます。

引数

-category arg (オプション) : 指定したカテゴリのコマンドのリストを取得します。

-syntax (オプション) : コマンドの構文のみを返します。

-args (オプション) : 指定したコマンドの簡単な説明を取得します。デフォルトでは、指定したコマンドの詳細な説明が返されます。簡単な説明を表示する場合にこのオプションを使用します。

-long (オプション) : コマンドの構文、引数の簡単な説明、およびコマンドの詳細な説明と使用例を返します。これがデフォルトです。

-prop arg (オプション) : デザイン オブジェクトまたはデザイン オブジェクトのクラスの特定のプロパティに関する情報を返します。このオプションを使用する場合は、**-class** も使用するか、1 つのデザイン オブジェクトを指定する必要があります。

-class arg (オプション) : 指定したオブジェクトのクラスのプロパティに関する情報を返します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

pattern (オプション) : 指定したコマンドに関する情報、または指定したパターンに一致するコマンドのリストを返します。

注記 : プロパティに関する情報を返すには、**-prop** または **-class** を使用してデザイン オブジェクトを指定する必要があります。

例

次の例では、ザイリンクス Tcl コマンドのカテゴリのリストが返されます。

```
help
```

次の例では、Vivado Design Suite のシミュレータ機能を読み込み、Simulation および Waveform カテゴリの Tcl コマンドのリストを返しています。

```
load_features simulator
help -category simulation
help -category waveform
```

次の例では、指定した検索パターンに一致するコマンドのリストを返しています。

```
help *file*
```

このリストは、**remove_files** や **delete_files** などの特定の機能のコマンドを検索するのに使用できます。

次の例では、**remove_files** コマンドとその引数の詳細な説明が返されます。

```
help remove_files
```

注記 : **-args** オプションを使用すると、そのコマンドの簡単な説明を表示できます。

次の例では、**short** というプロシージャを定義し、指定したコマンドの簡単な説明を取得しています。

```
proc short cmdName {help -args $cmdName}
```

注記 : このプロシージャを **init.tcl** ファイルに追加すると、ツールを起動するたびにこのコマンドを読み込むことができます。**init.tcl** ファイルの詳細は、第 1 章「概要」を参照してください。

次の例では、デザイン オブジェクトまたはデザイン オブジェクトのクラスのプロパティに関する情報を表示する方法を示しています。

```
help -prop NAME -class cell
help -prop NAME [get_cell cpuEngine]
```

注記 : この例では、最初のコマンドで NAME プロパティに関する一般的な情報を返し、2 番目のコマンドで指定したデザイン オブジェクトの NAME プロパティの値を返しています。

関連項目

- [list_features](#)
- [list_property](#)
- [load_features](#)
- [report_property](#)

highlight_objects

オブジェクトを指定の色でハイライトします。

構文

```
highlight_objects [-color_index arg] [-rgb args] [-color arg] [-quiet]
[-verbose] objects
```

戻り値

なし

使用法

名前	説明
[-color_index]	色を色インデックスで指定します。
[-rgb]	色を RGB で指定します。
[-color]	有効な値は、red、green、blue、magenta、yellow、cyan、および orange です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	ハイライトするオブジェクトを指定します。

カテゴリ

GUI 制御

説明

色オプションのいずれかで指定した色で指定のオブジェクトをハイライトします。オブジェクトのハイライトを解除するには、**unhighlight_objects** コマンドを使用します。

注記：ハイライト色を指定する色オプションは、1 つのみ指定します。複数の色オプションを指定すると、-rgb、-color_index、-color の優先順位で色が決定されます。

引数

-color_index arg (オプション)：有効な値は 1 ~ 19 の整数で、オブジェクトのハイライト色を色インデックスで指定します。色インデックスは、[Tools] → [Options] → [Themes] の [Highlight] の下に定義されています。テーマの設定方法の詳細は、『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。

-rgb args (オプション)：オブジェクトのハイライト色を、RGB コードを使用して {R G B} の形式で指定します。たとえば、{255 255 0} は黄色を指定します。

-color *arg* (オプション) : オブジェクトのハイライト色を色の名前で指定します。指定可能な値は、red、green、blue、magenta、yellow、cyan、および orange です。

注記：白は **select_objects** コマンドで指定したオブジェクトを表示するのに使用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須) : ハイライトするオブジェクトを指定します。

例

次の例では、現在選択されているオブジェクトを赤でハイライトします。

```
highlight_objects -color red [get_selected_objects]
```

関連項目

- [get_selected_objects](#)
- [unhighlight_objects](#)

implement_debug_core

デバッグ コアをインプリメントします。

構文

```
implement_debug_core [-quiet] [-verbose] [cores...]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[cores]	デバッグ コアを指定します。

カテゴリ

説明

現在のプロジェクトに含まれる Vivado ラボ ツール デバッグ コアをインプリメントします。ツールは指定した ILA デバッグ コアに対して 1 回実行され、すべてのコアが指定されると、デバッグ ハブ コアに対してもう 1 回実行されます。現在 **create_debug_core** コマンドでサポートされるコアタイプは、ILA コア (labtools_ilav2) のみです。プロジェクトに ILA コアを含めてコンフィギュレーションするため、ツールによりデバッグ ハブ コア (labtools_xsdbmasterlib_v2) が自動的に追加されます。

デバッグ ハブ コアおよび ILA コアは、最初ブラック ボックスとして作成されます。これらのコアは、配置配線を実行する前にインプリメントしておく必要があります。**create_debug_core** でコアを作成した後、**create_debug_port** および **connect_debug_port** でポートを追加および接続しますが、デバッグ コアの内容はデザインでは定義されていません。

デバッグ コアは、**launch_runs** コマンドを使用してインプリメンテーション run を起動すると自動的にインプリメントされますが、**implement_debug_core** コマンドを使用すると、デザイン全体をインプリメントせずに、CORE Generator ツールで 1 つ以上のコアをインプリメントできます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

cores (オプション) : インプリメントするデバッグ コアを 1 つ以上指定します。コアを指定しない場合、すべてのデバッグ コアがインプリメントされます。

例

次の例では、現在のプロジェクトに含まれるすべてのデバッグ コアをインプリメントします。

```
implement_debug_core [get_debug_cores]
```

関連項目

- [connect_debug_port](#)
- [create_debug_core](#)
- [create_debug_port](#)
- [get_debug_cores](#)
- [launch_runs](#)

import_files

ファイルまたはディレクトリをアクティブなファイルセットにインポートします。

構文

```
import_files [-fileset arg] [-force] [-norecurse] [-flat] [-relative_to arg]
[-quiet] [-verbose] [files...]
```

戻り値

インポートされたファイル オブジェクトのリスト

使用法

名前	説明
[-fileset]	ファイルセット名を指定します。
[-force]	プロジェクト ディレクトリの同じ名前のファイルを上書きします。
[-norecurse]	下位ディレクトリの検索をディスエーブルにします。
[-flat]	ファイルをフラットなディレクトリ構造でインポートします。
[-relative_to]	指定したディレクトリに相対するパスでファイルをインポートします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[files]	ファイルセットにインポートするファイルの名前を指定します。

カテゴリ

プロジェクト、シミュレーション

説明

1 つ以上のファイルまたは 1 つ以上のディレクトリ内のソース ファイルを指定したファイルセットにインポートします。

このコマンドは、指定したファイルセットを参照することによりファイルを追加する **add_files** コマンドとは異なり、**project.srcs\<fileset>\imports** のローカル プロジェクト フォルダーにファイルをインポートしてから、そのファイルを指定したファイルセットに追加します。

引数

-fileset name (オプション) : 指定したソース ファイルを追加するファイルセットを指定します。指定したファイルセットが存在しない場合は、エラー メッセージが表示されます。ファイルセットを指定しない場合は、デフォルトでソース ファイルセットに追加されます。

-force (オプション) : ローカル プロジェクト ディレクトリおよびファイルセットの同じ名前のファイルを上書きします。

-norecurse (オプション)：指定したディレクトリの下位ディレクトリでコマンドを実行しないよう指定します。このオプションを指定しない場合、下位ディレクトリでもプロジェクトに追加可能なソース ファイルが検索されます。

-flat (オプション)：相対パスを保持せずに、すべてのファイルをインポート フォルダーにインポートします。デフォルトでは、ファイルをデザインにインポートする際にディレクトリ構造が保持されます。

-relative_to arg (オプション)：指定したディレクトリに相対するパスでファイルをインポートします。これにより、ローカル プロジェクトのディレクトリ構造でインポート ファイルへのパスを保持できます。ファイルは、指定したディレクトリに相対的なパスを使用してインポート フォルダーにインポートされます。

注記： **-relative_to** オプションは、**-flat** オプションを指定すると無視されます。**-flat** オプションを使用すると、インポート ファイルのディレクトリ構造が削除されます。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

files (オプション)：指定したファイルセットに追加する 1 つまたは複数のファイルまたはディレクトリ名を指定します。ディレクトリ名を指定した場合は、そのディレクトリとそれに含まれる下位ディレクトリの有効なソース ファイルすべてが追加されます。ファイルを指定しない場合は、現在のプロジェクトのソース セットのファイルがインポートされます。

注記：パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、top.ucf ファイルを constrs_1 制約ファイルセットにインポートしています。

```
import_files -fileset constrs_1 top.ucf
```

次の例では、**-fileset** オプションが指定されていないので、有効なソース ファイルがソース ファイルセット (**sources_1**) にデフォルトでインポートされます。また、**-norecurse** オプションが指定されているので、指定した \level1 ディレクトリのみが検索され、下位ディレクトリは検索されません。**-flat** オプションは指定されていないので、すべての有効なソース ファイルがプロジェクトの \imports フォルダーにインポートされます。

```
import_files C:/Data/FPGA_Design/level1 -norecurse -flat
```

注記： **-flat** オプションが指定されていないので、プロジェクトの \imports フォルダー内に \level1 ディレクトリが作成されます。

次の例では、**-fileset** オプションが指定されていないので、ファイルがソース ファイルセット (**sources_1**) にインポートされます。有効なソース ファイルは \level1 ディレクトリおよびすべての下位ディレクトリからインポートされ、**-relative_to** オプションが使用されているので、プロジェクトの \imports フォルダーに \Data ディレクトリから書き込まれます。

```
import_files C:/Data/FPGA_Design/level1 -relative_to C:/Data
```

関連項目

[add_files](#)

import_ip

IP ファイルをインポートしてファイルセットに追加します。

構文

```
import_ip [-srcset arg] [-name arg] [-quiet] [-verbose] [files]
```

戻り値

追加されたファイル オブジェクトのリスト

使用法

名前	説明
[-srcset]	アップグレードするオブジェクトを含むソース ファイル セットを指定します。デフォルトは、現在のソース ファイルセットです。
[-name]	インポートされた IP の新しい名前を指定します。複数のファイルを指定する場合は使用できません。デフォルトは、インポートされた IP の現在の名前です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[files]	インポートする IP ファイルの名前を指定します。XCI または XCO ファイル名のリストを指定します。

カテゴリ

プロジェクト、IP フロー

説明

既存の XCI または XCO ファイルを IP ソースとして現在のプロジェクトにインポートします。

このコマンドを使用すると、既存の IP ファイル直接読み込み、ローカル プロジェクト フォルダーにインポートできます。IP ファイルを参照することにより現在のプロジェクトに追加するには、**read_ip** または **add_files** コマンドを使用します。

現在の IP カタログから新しい IP ファイルを作成するには、**create_ip** を使用します。

引数

-srcset arg (オプション) : IP ファイルをインポートするソース ファイルセットを指定します。このオプションを指定しない場合のデフォルトのソース ファイルセットは **sources_1** です。

-name arg (オプション) : 現在のソース ファイルセットに追加する IP オブジェクトに割り当てる名前を指定します。このオプションは、*files* で 1 つのファイルを指定している場合にのみ使用できます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

files (オプション) : 現在のプロジェクトにインポートする IP ファイルの名前を指定します。既存の XCI または XCO ファイルを指定する必要があります。XCI ファイルは、IP のパラメーター情報を含む IP-XACT 形式のファイルです。XCO ファイルは、IP コアを生成する際に使用されるカスタマイズ パラメーターとプロジェクト オプションがすべて記述された CORE Generator ファイルです。XCI または XCO ファイルは、現在のプロジェクトでコアを作成し直すために使用されます。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次の例では、10 ギガビット イーサネットコアを現在のプロジェクトに追加し、IP_block1 という名前を割り当てています。

```
import_ip C:/Data/FPGA_Design/10gig_eth.xci -name IP_block1
```

関連項目

- [add_files](#)
- [create_ip](#)
- [generate_target](#)
- [read_ip](#)

import_synplify

指定した Synplify プロジェクト ファイルをインポートします。

構文

```
import_synplify [-copy_sources] [-quiet] [-verbose] file
```

戻り値

Synplify ファイルからインポートされたファイル オブジェクトのリスト

使用法

名前	説明
[-copy_sources]	作成したプロジェクトに Synplify プロジェクト ファイルのソース をすべてコピーします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	インポートする Synplify プロジェクト ファイルの名前を指定します。

カテゴリ

プロジェクト

説明

Synplify 合成プロジェクト ファイル (.prj) を、合成で使用されるさまざまなソース ファイルも含めて現在のプロジェクトに追加します。

引数

-copy_sources (オプション) : Synplify プロジェクト ソース ファイルを、現在のディレクトリから参照するのではなく、ローカル プロジェクト ディレクトリ構造にコピーします。デフォルトでは、現在のディレクトリからソース ファイルが参照されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : ソース ファイルをインポートする Synplify プロジェクト ファイルの名前を指定します。

例

次の例では、新しいプロジェクトを作成し、指定の Synplify プロジェクトファイルにインポートしています。Synplify プロジェクトからのソース ファイルは、ローカル プロジェクト ディレクトリにコピーされます。

```
create_project syn_test C:/Data/FPGA_Design/syn_test
import_synplify -copy_sources C:/Data/syn_data.prj
```

関連項目

[create_project](#)

import_xise

作成したプロジェクトに XISE プロジェクトファイルの設定をインポートします。

構文

```
import_xise [-copy_sources] [-quiet] [-verbose] file
```

戻り値

True

使用法

名前	説明
[-copy_sources]	作成したプロジェクトに ISE ソースすべてをコピーします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	インポートする XISE プロジェクトファイルの名前を指定します。

カテゴリ

プロジェクト

説明

現在のプロジェクトに ISE プロジェクトファイル (XISE) をインポートします。これにより、ISE プロジェクトを Vivado Design Suite にすばやく移行して、合成、シミュレーション、インプリメンテーションを実行できます。すべてのプロジェクトソースファイル、制約ファイル、シミュレーションファイルおよび run 設定が ISE プロジェクトからインポートされ、現在のプロジェクトで作成され直されます。

このコマンドは、新しい空のプロジェクトで実行する必要があります。ソースファイル、制約、および run 設定は ISE プロジェクトからインポートされるので、既存のソースファイルや制約ファイルがある場合はすべて上書きされます。

引数

-copy_sources (オプション) : ISE プロジェクトのソースファイルを、現在のディレクトリから参照するのではなく、ローカル プロジェクトディレクトリ構造にコピーします。デフォルトでは、現在のディレクトリからソースファイルが参照されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記: メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 現在のプロジェクトにインポートする ISE プロジェクトファイル (.xise) の名前を指定します。

例

次の例では、importISE という新規プロジェクトを作成し、ISE プロジェクトファイル (first_use.xise) をインポートしています。

```
create_project importISE C:/Data/importISE import_xise \
C:/Data/FPGA_design/ise_designs/drp_des/first_use.xise
```

注記: この例では **-copy_sources** オプションが指定されていないので、ISE プロジェクトのすべてのソースファイルは現在の場所から参照されてプロジェクトに追加されます。

関連項目

[create_project](#)

import_xst

指定した XST プロジェクト ファイルをインポートします。

構文

```
import_xst [-copy_sources] [-quiet] [-verbose] file
```

戻り値

XST ファイルからインポートされたファイル オブジェクトのリスト

使用法

名前	説明
[-copy_sources]	作成したプロジェクトに XST プロジェクト ファイルのソースをすべてコピーします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	インポートする XST プロジェクト ファイルの名前を指定します。

カテゴリ

プロジェクト

説明

XST 合成プロジェクト ファイル (XST の実行に使用されるさまざまなソース ファイル) を現在のプロジェクトに追加します。

引数

-copy_sources (オプション) : XST プロジェクト ソース ファイルを、現在のディレクトリから参照するのではなく、ローカル プロジェクト ディレクトリ構造にコピーします。デフォルトでは、現在のディレクトリからソース ファイルが参照されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : ソース ファイルをインポートする XST プロジェクト ファイルの名前を指定します。

例

次の例では、`xst_test` という新規プロジェクトを作成し、`drp_des.xst` ファイルをインポートしています。

```
create_project xst_test C:/Data/FPGA_Design/xst_test
import_xst C:/Data/ise_designs/drp_des.xst
```

関連項目

[create_project](#)

infer_diff_pairs

CSV または XDC ファイルからインポートされたポートに対して差動ペアを推論します。

構文

```
infer_diff_pairs [-file_type arg] [-quiet] [-verbose] [file...]
```

戻り値

なし

使用法

名前	説明
[-file_type]	入力ファイル タイプを指定します。有効な値は csv または xdc です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[file]	ピン配置 CSV ファイルまたは XDC ファイルを指定します。

カテゴリ

ファイル入力および出力

説明

infer_diff_pairs コマンドは、I/O ピン配置プロジェクトで **read_csv** または **read_xdc** コマンドを使用して I/O ピン情報をインポートした後に使用できます。

信号名、差動ペア信号、差動ペア タイプ、I/O 規格など、ファイルで差動ペアを定義するために使用される属性は複数あります。

ツールでは、次の条件で差動ペアが特定されます。

- 差動ペアの定義が一致：2 つのポートエントリで、[DiffPair Signal] にお互いの [Signal Name] の値が指定され、[DiffPair Type] で一方に N、もう一方に P が指定されている。ツールで差動ペアを作成する際に、[I/O Standard] などのほかの属性に互換性があるかどうかがチェックされます。
- 差動ペアの定義が片方のみ：2 つのポートエントリの [DiffPair Type] で一方に N、もう一方に P が指定されているが、1 つのポートでのみ [DiffPair Signal] にもう一方の [Signal Name] の値が指定されている。この場合、すべての属性に互換性があれば、差動ペアが作成されます。
- 1 つのポートのみでの差動ペア定義：1 つのポートエントリで、[I/O Standard] に差動規格、[DiffPair Type] に値が指定されているが、[DiffPair Signal] に指定されている信号が CSV ファイルに含まれない。この場合、このポートのプロパティと一致するプロパティで、もう一方の差動ペア (N 側または P 側) が作成されます。
- 名前から差動ペアを推論：2 つのポートエントリで、N 側と P 側が推論されるような名前が指定されている。この場合、すべての属性に互換性があれば、差動ペアが推論されます。

CSV または XDC ファイルからポート定義を読み込むと、データから差動ペアが推論されたことがレポートされます。**infer_diff_pairs** コマンドを使用すると、必要に応じてこれらの差動ペアを推論できます。

引数

-file_type [csv | xdc] (オプション) : 差動ペアを推論する際に読み込むファイルのタイプを指定します。有効なファイル タイプは CSV および XDC です。デフォルトはありません。**-file_type** の指定は必須です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (オプション) : 以前にインポートしたファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次の例では、指定した XDC ファイルをインポートし、差動ペアを推論しています。

```
read_xdc C:/Vivado_Install/io_1.xdc
infer_diff_pairs C:/Vivado_Install/io_1.xdc -file_type xdc
```

関連項目

- [read_csv](#)
- [read_xdc](#)

launch_chipscope_analyzer

run に対して ChipScope Analyzer ツールを起動します。

構文

```
launch_chipscope_analyzer [-run arg] [-csproject arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-run]	ChipScope Analyzer で開くインプリメント済み run を指定します。
[-csproject]	ChipScope プロジェクトを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール起動、ChipScope

説明

アクティブな run または指定したインプリメント済みデザイン run に対して ChipScope™ Pro Analyzer ツールを起動します。インプリメンテーションの前に、ネットリスト デザインを ChipScope で使用するよう設定するには、`create_debug_core`、`create_debug_port`、および `connect_debug_port` コマンドを使用します。

インプリメント済みデザインに対して `launch_chipscope_analyzer` を実行するには、BitGen でビットストリーム ファイルを生成しておく必要があります。BitGen が実行されていない場合、エラー メッセージが表示されます。

注記：ビットストリーム ファイルを作成するには、`write_bitstream` コマンドを使用するだけでは不十分です。2 番目の例に示されている手順に従う必要があります。

引数

`-run arg` : ChipScope Pro Analyzer を起動するのに使用する run の名前を指定します。run は、インプリメントが完了しており、ビットストリーム ファイル (.bit) が生成されている必要があります。ChipScope では、指定の run のビットストリーム ファイルと `debug_nets.cdc` ファイルが使用されます。

`-csproject arg` : ChipScope Pro Analyzer で開くプロジェクトの名前を指定します。プロジェクト名を指定しない場合、デフォルトのプロジェクト名 `csdefaultproj.cpj` が使用されます。プロジェクト名を指定する場合は、拡張子 `.cpj` も含める必要があります。

注記：プロジェクトは `project/project.data/sources_1/cs` フォルダーに作成されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、使用するインプリメンテーション run を指定し、作成する ChipScope プロジェクトの名前を指定して、ChipScope Pro Analyzer を起動します。

```
launch_chipscope_analyzer -run impl_3 -csproject impl_3_cs_project
```

次の例では、impl_4 という run に **add_step Bitgen** プロパティを設定し、impl_4 run を起動して、その run に対して ChipScope Pro Analyzer を起動します。

```
set_property add_step Bitgen [get_runs impl_4]
launch_runs impl_4 -jobs 2
launch_chipscope_analyzer -run impl_4
```

注記 : この例では、ChipScope プロジェクトの名前は `csdefaultproj.cpj` となります。

関連項目

- [connect_debug_port](#)
- [create_debug_core](#)
- [create_debug_port](#)
- [launch_runs](#)
- [set_property](#)
- [write_bitstream](#)

launch_impact

run に対して iMPACT コンフィギュレーション ツールを起動します。

構文

```
launch_impact [-run arg] [-ipf arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-run]	iMPACT で開くインプリメント済み run を指定します。
[-ipf]	iMPACT のプロジェクトを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール起動、ChipScope

説明

iMPACT を起動し、デバイスをコンフィギュレーションしてプログラム ファイルを生成します。また、デザイン コンフィギュレーション データをリードバックして検証、コンフィギュレーション問題をデバッグ、XSVF ファイルを実行できます。

iMPACT を使用する前に、**write_bitstream** コマンドを使用してビットストリーム ファイルを生成する必要があります。このコマンドを実行すると、読み込まれたファイルのリストが返されます。

引数

-run (オプション) : 指定した run で iMPACT を起動します。run を指定しない場合、アクティブ インプリメンテーション run で iMPACT が起動します。

-ipf arg (オプション) : 結果を保存する iMPACT プロジェクト ファイルを指定します。iMPACT プロジェクト ファイル (IPF) には、iMPACT の前回のセッションからの情報が含まれます。ターゲット デバイスは、指定した IPF ファイルの設定に基づいてコンフィギュレーションされます。**-ipf** オプションを指定しない場合、ターゲット デバイスはデフォルト設定でコンフィギュレーションされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、指定したインプリメンテーション run で iMPACT を起動しています。

```
launch_impact -run impl_3
```

関連項目

[write_bitstream](#)

launch_modelsim

ModelSim シミュレータを使用してシミュレーションを実行します。

構文

```
launch_modelsim [-simset arg] [-mode arg] [-type arg] [-noclean_dir]
[-scripts_only] [-install_path arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-simset]	シミュレーション ファイルセットの名前を指定します。
[-mode]	シミュレーション モードを指定します。有効な値は behavioral、post-synthesis、post-implementation で、デフォルトは behavioral です。
[-type]	ネットリストのタイプを指定します。有効な値は functional、timing です。このオプションは、-mode を post-synthesis または post-implementation に設定している場合にのみ有効です。
[-noclean_dir]	シミュレーション実行ディレクトリ ファイルを削除しません。
[-scripts_only]	スクリプトのみを生成します。
[-install_path]	ModelSim のインストール ディレクトリ パスを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール起動、シミュレーション

説明

Mentor Graphics 社の ModelSim または Questa Advanced Simulator ツールを起動します。シミュレーション実行時にこれらのシミュレータが正しく起動するようにするには、指定したシミュレータをインストールし、\$PATH で指定しておく必要があります。

ModelSim または Questa を起動するには、プロジェクトのターゲット シミュレータ プロパティを設定しておく必要があります。

```
set_property target_simulator ModelSim [current_project]
```

また、**compile_simlib** コマンドを使用してターゲットシミュレータ用にザイリンクス シミュレーション ライブラリをコンパイルする必要があります。ライブラリがコンパイルされると、シミュレータで **modelsim.ini** ファイルを使用してこれらのコンパイル済みライブラリが参照されます。**modelsim.ini** ファイルはデフォルトの初期化ファイルで、参照ライブラリパス、最適化、コンパイラ、シミュレータの設定を指定する制御変数が含まれます。**modelsim.ini** は、次のディレクトリに含まれます。

- **compile_simlib** の実行時に **-directory** オプションで指定したパス
- MODELSIM 環境変数で定義されたパス
- MGC_WD 環境変数で定義されたパス
- プロジェクトのシミュレーション実行ディレクトリ

注記：これらのディレクトリに **modelsim.ini** がない場合は、警告メッセージが表示されます。

このコマンドを実行すると、シミュレータのトランскriptが返されます。

引数

-simset arg (オプション)：シミュレーションで使用するシミュレーション テストベンチおよびソースを含むシミュレーション ファイルセットを指定します。指定しない場合、現在のシミュレーション ファイルセットが使用されます。

-mode [behavioral | post_synthesis | post_implementation] (オプション)：シミュレーション モードを指定します。構文を検証してデザインが意図したとおりに動作することを確認する HDL デザインソースのビヘイビア シミュレーション、合成後のネットリストの論理またはタイミング シミュレーション、配置配線後の回路の動作を検証するインプリメンテーション後の論理またはタイミング シミュレーションを指定します。デフォルト モードは **behavioral** です。

-type [functional | timing] (オプション)：**-mode behavioral** を指定している場合は使用できません。ネットリストのみの論理シミュレーションを指定するか、ネットリストと SDF ファイルを使用したタイミング シミュレーションを指定します。デフォルトは **functional** です。合成後のタイミング シミュレーションでは、**synth_design** コマンドからの SDF コンポーネント遅延が使用されます。インプリメンテーション後のタイミング シミュレーションでは、**place_design** および **route_design** コマンドからの SDF 遅延が使用されます。

注記：**-type** オプションを **-mode behavioral** と共に使用すると、エラーが返されます。

-noclean_dir (オプション)：シミュレータを起動する前にシミュレーション実行ディレクトリのファイルを削除しません。ただし、シミュレータで生成された一部のファイルは、シミュレータを再起動すると上書きされるかアップデートされます。デフォルトでは、シミュレータを起動する前にシミュレーション実行ディレクトリのファイルが削除されます。

-scripts_only (オプション)：ModelSim を起動するコマンド スクリプトのみを作成します。このスクリプトは、後でシミュレータを起動するのに使用できます。

-install_path arg (オプション)：ModelSim 実行ファイル (vlog.exe、vcom.exe、および vsim.exe) を含むディレクトリを指定します。このオプションを指定しない場合、現在 PATH で定義されているディレクトリが検索されます。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、ModelSim シミュレータを post-implementation タイミング モードで起動しています。

```
launch_modelsim -mode post-implementation -type timing
```

関連項目

- [compile_simlib](#)
- [launch_xsim](#)
- [set_property](#)

launch_runs

run のセットを実行します。

構文

```
launch_runs [-jobs arg] [-scripts_only] [-all_placement] [-dir arg]
[-to_step arg] [-next_step] [-host args] [-remote_cmd arg] [-email_to args]
[-email_all] [-pre_launch_script arg] [-post_launch_script arg] [-force]
[-quiet] [-verbose] runs...
```

戻り値

なし

使用法

名前	説明
[-jobs]	並列実行するジョブ数を指定します。デフォルトは 1 です。
[-scripts_only]	スクリプトのみを生成します。
[-all_placement]	固定配置と未固定配置すべてを ISE にエクスポートします。デフォルトでは、固定配置のみがエクスポートされます。
[-dir]	実行ディレクトリを指定します。
[-to_step]	実行する最後のステップを指定します。複数の run を実行する際は無視されます。-next_step と共に使用することはできません。
[-next_step]	次のステップを実行します。複数の run を実行する際は無視されます。-to_step と共に使用することはできません。
[-host]	指定したリモートホストで指定したジョブ数を実行します。例 : -host {machine1 2} -host {machine2 4}
[-remote_cmd]	リモートホストにログインするコマンドを指定します。デフォルトは ssh -q -o BatchMode=yes です。
[-email_to]	ジョブが完了したときに通知する電子メール アドレスのリストを指定します。
[-email_all]	各ジョブの完了時に電子メールを送信します。
[-pre_launch_script]	各ジョブの前に実行するスクリプトを指定します。
[-post_launch_script]	各ジョブの完了後に実行するスクリプトを指定します。
[-force]	未決定の制約変更があつても、コマンドを実行します。変更は失われます (パーシャル リコンフィギュレーション デザインの場合)。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
runs	実行する run を指定します。

カテゴリ

プロジェクト

説明

Vivado ツールをプロジェクト モードで実行している場合に合成 run およびインプリメンテーション run を実行します。プロジェクト モードおよび非プロジェクト モードの詳細は、『Vivado Design Suite ユーザー ガイド：デザインフローの概要』(UG892) を参照してください。

run は、**create_run** コマンドを使用して設定し、その属性を **set_property** コマンドを使用して設定しておく必要があります。同じ **launch_runs** コマンドで合成 run とインプリメンテーション run の両方を指定できますが、インプリメンテーション run を実行するには、親の合成 run が完了している必要があります。

非プロジェクト モードでは、Vivado 合成は **synth_design** コマンドを使用して直接実行できるので、run を定義しておく必要はありません。

非プロジェクト モードでは、**opt_design**、**power_opt_design**、**place_design**、**route_design**、**phys_opt_design**、および **write_bitstream** コマンドを使用して、Vivado インプリメンテーションの各段階を実行できます。

引数

-jobs arg (オプション)：ローカル ホストで並列実行するジョブ数を指定します。リモート ホストのジョブ数は、**-host** オプションの一部として指定されるので、**-jobs** と **-host** の両方を指定する必要はありません。

-scripts_only (オプション)：指定した run ごとに **runme.bat** というスクリプトを生成し、後で実行できるようにします。

-all_placement (オプション)：ユーザーが割り当てた (固定) 配置と自動的に割り当てられた (未固定) 配置をインプリメンテーション用にエクスポートします。デフォルトでは、ユーザーが割り当てた (固定) 配置のみがインプリメンテーション用にエクスポートされます。

-dir arg (オプション)：run 結果を書き込むディレクトリを指定します。指定したディレクトリに、各 run ごとに個別のフォルダーが作成されます。デフォルトでは、各 run の結果が **project.runs** ディレクトリの下に個別のフォルダーで書き込まれます。

-to_step arg (オプション) : run をインプリメンテーション プロセスの指定した段階まで実行し、停止します。たとえば、インプリメンテーションを place_design の段階まで実行し、停止できます。これにより、run 全体を完了せずに特定の段階の run 結果を確認できます。インプリメンテーション run での有効な段階は、次のとおりです。

- opt_design : ターゲット デバイスのリソースがより効率的に使用されるように、論理デザインを最適化します (オプション)。これはオプションですが、通常はデフォルトでイネーブルになっています。
- power_opt_design : インプリメント済み FPGA の消費電力を削減するため、ロジック デザインのエレメントを最適化します (オプション)。
- place_design : ロジック セルをターゲット デバイスに配置します。この段階は必須です。
- power_opt_design (Post-Place) : 配置済みロジック エレメントの消費電力を最適化します (オプション)。複数の単語が含まれているので、ダブルクオーテーションまたは中かっこで囲んで指定する必要があります (例 : **-to_step "power_opt_design (Post-Place)"**)。
- phys_opt_design : ファンアウトの大きいネットのドライバーを複製してロードを分散することにより、デザインのタイミングを最適化します (オプション)。
- route_design : デザインの接続をターゲット FPGA 上に配線します。この段階は必須です。
- write_bitstream : ザイリンクス デバイス コンフィギュレーションのビットストリーム ファイルを生成します。この段階は必須です。

注記 : **-to_step** で指定するには、**set_property** コマンドを使用して、インプリメンテーション run でその段階をイネーブルにしておく必要があります。そうでないと、エラーが返されます。

-next_step (オプション) : 前回の run の実行を停止した段階から再開します。このオプションは、**-to_step** オプションを使用して実行した前回の run を完了するために使用します。

注記 : **-to_step** と **-next_step** オプションを同時に指定することはできません。また、複数の run を実行する場合は無視されます。

-host args (オプション) : 指定したリモート ホストで指定したジョブ数を実行します。オプションは、「-host {machine1 2}」のように {hostname jobs} という形式で使用します。**-host** オプションを指定しない場合、run はローカル ホストから実行されます。

注記 : このオプションは、Linux でのみサポートされます。

-remote_cmd arg (オプション) : ジョブを実行するため、リモート ホストにログインするのに使用するコマンドを指定します。デフォルトのリモート コマンドは「ssh -q -o BatchMode=yes」です。

-email_to args (オプション) : run が完了したことを示す通知を送付する電子メール アドレスを指定します。

-email_all (オプション) : 各 run が完了するごとに電子メールが送付されます。

-pre_launch_script arg (オプション) : 各ジョブの前に実行する Tcl スクリプトを指定します。

-post_launch_script arg (オプション) : すべてのジョブの完了後に実行する Tcl スクリプトを指定します。

-force (オプション) : パーシャル リコンフィギュレーション デザインの制約の変更が未決定かどうかにかかわらず、run を実行します。

注記 : このオプションは、パーシャル リコンフィギュレーション プロジェクトでのみ使用できます。未決定の制約の変更は、指定した run では失われます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

runs (必須) : 実行する合成 run および インプリメンテーション run を指定します。1 つまたは複数の run を指定できます。

例

次の例では、並列実行するジョブ数を 2 に設定し、3 つの異なる合成 run を実行します。

```
launch_runs synth_1 synth_2 synth_4 -jobs 2
```

注記 : 各 run の結果は、*project.runs* ディレクトリの *synth_1*、*synth_2*、および *synth_4* フォルダーにそれぞれ書き込まれます。

次の例では、run 結果を書き込むディレクトリが作成されます。この場合、指定したディレクトリに *impl_3*、*impl_4*、および *synth_3* が書き込まれます。また、**-scripts_only** オプションにより、各フォルダーに *runme.bat* スクリプトが書き込まれますが、この段階では run は実行されません。

```
launch_runs impl_3 impl_4 synth_3 -dir C:/Data/FPGA_Design/results -scripts_only
```

次の例では、*impl_1* run に対して Vivado Implementation 2013 フローを設定し、オプションの最適化のいくつかをイネーブルにした後、**place_design** 段階まで実行しています。

```
set_property flow {Vivado Implementation 2013} [get_runs impl_1]
set_property STEPS.POWER_OPT_DESIGN.IS_ENABLED true [get_runs impl_1]
set_property STEPS.POST_PLACE_POWER_OPT_DESIGN.IS_ENABLED true [get_runs impl_1]
set_property STEPS.PHYS_OPT_DESIGN.IS_ENABLED true [get_runs impl_1]
launch_runs -to_step place_design impl_1
```

関連項目

- [create_run](#)
- [get_runs](#)
- [opt_design](#)
- [phys_opt_design](#)
- [place_design](#)
- [power_opt_design](#)
- [reset_run](#)
- [route_design](#)
- [set_property](#)
- [synth_design](#)
- [write_bitstream](#)

launch_sdk

ザイリンクス ソフトウェア開発キット (SDK) を起動します。

構文

```
launch_sdk [-bit arg] [-bmm arg] [-workspace arg] [-lp arg] [-hwspec arg]
[-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-bit]	FPGA プログラム用のビットストリームを指定します。
[-bmm]	BRAM 初期化用の BMM ファイルを指定します。
[-workspace]	SDK プロジェクト用のワークスペース ディレクトリを指定します。
[-lp]	ドライバー、OS、ライブラリ検索ディレクトリのリストにリポジトリ パスを追加します。
[-hwspec]	ハードウェア プラットフォーム仕様ファイル (<system>.xml) を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール起動、XPS

説明

プロジェクトのエンベデッド プロセッサ システムのソフトウェアを設計するソフトウェア開発キット (SDK) を起動します。

このコマンドは、エンベデッド プロセッサのハードウェア仕様ファイル (system.xml) を SDK で使用するためにエクスポートする **export_hardware** コマンドの後に実行します。export_hardware コマンドでは、デフォルトで指定したエンベデッド プロセッサのハードウェア仕様ファイル (.xml) が *project_name.sdk/SDK/SDK_Export/hw* ディレクトリに、デザインのエンベデッド プロセッサに基づくファイル名で保存されます。

このコマンドを実行すると、SDK ツール起動のトランскriptが返されます。

引数

-bit arg (オプション) : FPGA プログラム用のビットストリームを指定します。

-bmm arg (オプション) : BRAM 初期化用の BMM ファイルを指定します。

-workspace arg (オプション) : SDK プロジェクト用のワークスペース ディレクトリを指定します。このフォルダーにソフトウェア プロジェクトが保存されます。

-lp arg (オプション) : ドライバー、OS、ライブラリ検索ディレクトリのライブラリまたはリポジトリ パスを指定します。これは、アプリケーション ソフトウェア スタックの最下位層を構成するライブラリとドライバーのコレクションです。

-hwspec arg (オプション) : エンベデッド プロセッサ デザインのハードウェア プラットフォーム仕様ファイル (.xml) を指定します。これは export_hardware コマンドでエクスポートされたファイルか、プロジェクトの sources/edk ディレクトリにある system.xml ファイルです。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、プロジェクトのハードウェア仕様ファイルとワークスペースを指定して、SDK を起動しています。

```
launch_sdk -hwspec C:/Data/export_sdk/hw/robot.xml -workspace C:/Data/sdk_work/
```

関連項目

[export_hardware](#)

launch_xsim

Vivado シミュレータを使用してシミュレーションを実行します。

構文

```
launch_xsim [-simset arg] [-noclean_dir] [-scripts_only] [-mode arg]
[-type arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-simset]	シミュレーション ファイルセットの名前を指定します。
[-noclean_dir]	シミュレーション実行ディレクトリ ファイルを削除しません。
[-scripts_only]	スクリプトのみを生成します。
[-mode]	シミュレーション モードを指定します。有効な値は behavioral、post-synthesis、post-implementation で、デフォルトは behavioral です。
[-type]	ネットリストのタイプを指定します。有効な値は functional、timing です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール起動、シミュレーション

説明

Vivado シミュレータを起動し、ビヘイビア シミュレーション、合成後またはインプリメンテーション後のネットリストの論理シミュレーションまたはタイミング シミュレーションを実行します。

Vivado シミュレータを起動すると、まず xelab、RTL エラボレーター、コンパイラ、およびリンクが実行され、シミュレータで使用されるシミュレーション スナップショットが作成されます。

その後、xelab で作成されたスナップショットを使用して Vivado シミュレータが起動します。

このコマンドを実行すると、xelab および xsim のトランスクriptが返されます。

引数

-simset arg (オプション) : シミュレーションで使用するシミュレーション テストベンチおよびソースを含むシミュレーション ファイルセットを指定します。指定しない場合、現在のシミュレーション ファイルセットが使用されます。

-noclean_dir (オプション) : シミュレータを起動する前にシミュレーション実行ディレクトリのファイルを削除しません。ただし、シミュレータで生成された一部のファイルは、シミュレータを再起動すると上書きされるかアップデートされます。デフォルトでは、シミュレータを起動する前にシミュレーション実行ディレクトリのファイルが削除されます。

-scripts_only (オプション) : Vivado シミュレータを起動するためのスクリプトのみを生成します。シミュレータは起動しません。このスクリプトを後でシミュレータを起動するのに使用できます。

-mode [behavioral | post_synthesis | post_implementation] (オプション) : シミュレーション モードを指定します。構文を検証してデザインが意図したとおりに動作することを確認する HDL デザインソースのビヘイビアーシミュレーション、合成後のネットリストの論理またはタイミングシミュレーション、配置配線後の回路の動作を検証するインプリメンテーション後の論理またはタイミングシミュレーションを指定します。デフォルト モードは **behavioral** です。

-type [functional | timing] (オプション) : **-mode behavioral** を指定している場合は使用できません。ネットリストのみの論理シミュレーションを指定するか、ネットリストと SDF ファイルを使用したタイミングシミュレーションを指定します。デフォルトは **functional** です。合成後のタイミングシミュレーションでは、**synth_design** コマンドからの SDF コンポーネント遅延が使用されます。インプリメンテーション後のタイミングシミュレーションでは、**place_design** および **route_design** コマンドからの SDF 遅延が使用されます。

注記: **-type** オプションを **-mode behavioral** と共に使用すると、エラーが返されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記: コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記: メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、Vivado シミュレータを post_implementation モードで起動しています。

```
launch_xsim -mode post_implementation
```

関連項目

- [close_sim](#)
- [current_sim](#)

limit_vcd

ディスク上の VCD ファイルの最大サイズをバイト数で指定します (\$dumplimit Verilog タスクと同等)。

構文

```
limit_vcd [-quiet] [-verbose] filesize
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
filesize	VCD ファイルの最大サイズをバイト数で指定します。

カテゴリ

シミュレーション

説明

VCD (Value Change Dump) ファイルの最大サイズをバイト数で指定します。このコマンドは、Verilog の **\$dumplimit** シミュレータ指示子と同様の操作を実行します。

ファイルのサイズが指定の最大値に達した場合、ダンプ プロセスは停止し、VCD ファイルにファイル サイズの最大値に達したことを示すコメントが記述されます。

注記 : **limit_vcd** コマンドを実行する前に、**open_vcd** コマンドを実行する必要があります。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

filesize (必須) : 開いている VCD ファイルの最大サイズをバイト数で指定します。

例

次の例では、現在の VCD オブジェクトの最大サイズを設定しています。

```
limit_vcd 1000
```

関連項目

- [checkpoint_vcd](#)
- [flush_vcd](#)
- [log_vcd](#)
- [open_vcd](#)

link_design

ネットリスト デザインを開きます。

構文

```
link_design [-name arg] [-part arg] [-constrset arg] [-top arg] [-mode arg]
[-quiet] [-verbose]
```

戻り値

デザイン オブジェクト

使用法

名前	説明
[-name]	デザイン名を指定します。
[-part]	ターゲット パーツを指定します。
[-constrset]	使用する制約ファイルセットを指定します。
[-top]	構造ネットリストが Verilog の場合、最上位モジュール名を指定します。
[-mode]	デザイン モードを指定します。有効な値は default、out_of_context で、デフォルトは default です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール

説明

新規または既存のネットリスト デザインを開き、ネットリストと制約をターゲット パーツとリンクしてデザインを作成します。この操作は、**open_run** コマンドでも実行できます。

ネットリスト デザインを開くには、現在のソースファイルセットの `design_mode` を `GateLvl` に設定しておく必要があります。GateLvl に設定していないと、次のようなエラー メッセージが表示されます。

```
ERROR: The design mode of 'sources_1' must be GateLvl.
```

引数

-name arg (オプション) : 新規または既存のネットリスト デザインの名前を指定します。

-part arg (オプション) : 新規デザインを作成する際に使用するザイリンクス デバイスを指定します。-part オプションを指定しない場合、デフォルトのパーツが使用されます。

-constrset arg (オプション) : デザインを開く際に使用する制約ファイルセットの名前を指定します。

注記 : **-constrset** オプションには、既存の制約ファイルセットを指定する必要があります。新しいファイルセットを作成することはできません。新しいファイルセットを作成する場合は、**create_fileset** を使用します。

-top arg (オプション) : ネットリストのデザイン階層の最上位モジュールを指定します。

-mode [default | out_of_context] (オプション) : ブロックが合成済みで、I/O バッファーの挿入がディスエーブルであった場合、生成された EDIF を **-mode out_of_context** を使用して Vivado Design Suite に読み込むことができます。これにより、I/O バッファーなしのモジュールをインプリメントでき、入力または出力が未接続のために最適化で削除されるのを回避でき、デザイン用に DRC ルールを変更できます。詳細は、『Vivado Design Suite ユーザー ガイド：階層デザイン』(UG905) を参照してください。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、Net1 というネットリスト デザインが新規に作成されます。

```
link_design -name Net1
```

注記 : この例では、デフォルトのソース セット、制約セット、パートが使用されます。

次の例では、Net1 というネットリスト デザインを開いて、使用する制約セットを指定しています。

```
link_design -name Net1 -constrset con1
```

関連項目

[open_run](#)

list_features

使用可能な機能をリストします。

構文

```
list_features [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール

説明

Vivado Design Suite のメモリ使用量を削減するため、Tcl コマンドの機能グループがあり、それらの機能グループのコマンドは、その機能グループからのコマンドを使用するか、**load_features** コマンドを使用して機能を読み込むまで使用できません。

このコマンドは、Vivado Design Suite で使用可能な、**load_features** コマンドを使用して読み込むことができる機能をリストします。

注記：機能が既に読み込まれている場合は、読み込み可能な機能としてリストされません。

このコマンドを実行すると、機能のリストまたはエラー メッセージが返されます。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、Vivado Design Suite に読み込むことができる機能をリストしています。

```
list_features
```

関連項目

- [help](#)
- [load_features](#)

list_param

すべてのパラメーター名を取得します。

構文

```
list_param [-quiet] [-verbose]
```

戻り値

リスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロパティおよびパラメーター

説明

ユーザー定義可能なコンフィギュレーション パラメーターのリストを取得します。これらのパラメーターでは、さまざまなツールの設定と動作が指定されます。特定のパラメーターの詳細を確認するには、**report_param** コマンドを実行し、パラメーターの説明とそのパラメーターの現在の値を返します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、ユーザー定義可能なパラメーターすべてのリストが表示されます。

```
list_param
```

関連項目

- [get_param](#)
- [report_param](#)
- [reset_param](#)
- [set_param](#)

list_property

オブジェクトのプロパティをリストします。

構文

```
list_property [-class arg] [-regexp] [-quiet] [-verbose] [object] [pattern]
```

戻り値

プロパティ名のリスト

使用法

名前	説明
[-class]	プロパティを取得するオブジェクトタイプを指定します。オブジェクトを指定した場合は無視されます。
[-regexp]	検索パターンを正規表現として処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[object]	プロパティを取得するオブジェクトを指定します。
[pattern]	プロパティを検索するパターンを指定します。デフォルトは*です。

カテゴリ

オブジェクト、プロパティおよびパラメーター

説明

指定したオブジェクトまたはクラスのプロパティすべてのリストを取得します。

注記： `report_property` でもオブジェクトのプロパティリストが返されますが、プロパティのタイプと値も含まれます。

引数

`-class arg` (オプション)：プロパティのリストを取得するオブジェクトのクラスを指定します。

注記： `-class` と `object` の両方を指定した場合、指定のオブジェクトのプロパティが返されます。

`-quiet` (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、`TCL_OK` が返されます。

注記： コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

`-verbose` (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記： メッセージの制限は、`set_msg_config` コマンドで定義できます。

object (オプション) : プロパティを取得するオブジェクトを 1 つ指定します。

注記: 複数のオブジェクトを指定すると、エラー メッセージが表示されます。

例

次の例では、指定したオブジェクトのプロパティすべてが返されます。

```
list_property [get_cells cpuEngine]
```

関連項目

- [create_property](#)
- [get_cells](#)
- [get_property](#)
- [list_property_value](#)
- [report_property](#)
- [reset_property](#)
- [set_property](#)

list_property_value

オブジェクトの有効なプロパティ値をリストします。

構文

```
list_property_value [-default] [-class arg] [-quiet] [-verbose] name [object]
```

戻り値

プロパティ値のリスト

使用法

名前	説明
[-default]	デフォルト値のみを表示します。
[-class]	有効なプロパティ値を取得するオブジェクト タイプを指定します。オブジェクトを指定した場合は無視されます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
name	有効な値を取得するプロパティの名前を指定します。
[object]	有効なプロパティ値を取得するオブジェクトを指定します。

カテゴリ

オブジェクト、プロパティおよびパラメーター

説明

指定のオブジェクトのクラスまたは指定のオブジェクトの列挙型プロパティに有効な値をリストします。

注記： このコマンドでは、列挙型プロパティ以外のプロパティの有効な値はリストされません。**report_property** コマンドを実行すると、プロパティ タイプが返されるので、列挙型プロパティを特定できます。

引数

-default (オプション)： 指定したオブジェクトのクラスのデフォルト値を返します。

-class *arg* (オプション)： プロパティ値を取得するオブジェクトのクラスを指定します。オブジェクトのクラスは、実際のオブジェクトを指定する代わりに使用できます。

-quiet (オプション)： コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記： コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 値を取得するプロパティの名前を指定します。このコマンドでは、列挙型の値、または定義済みの値のセットを持つプロパティのみを指定できます。指定したプロパティの有効な値すべてが返されます。

object (オプション) : オブジェクトを指定します。**-class** オプションでオブジェクトのタイプを指定する代わりに、実際のオブジェクトを指定できます。

例

次の例では、セル オブジェクトから KEEP_HIERARCHY プロパティの有効な値のリストが返されます。

```
list_property_value KEEP_HIERARCHY -class cell
```

次の例でも同じリストが返されますが、セル クラスに代わりに実際のセル オブジェクトが指定されています。

```
list_property_value KEEP_HIERARCHY [get_cells cpuEngine]
```

次の例では、現在のデザインをデザイン クラスとして使用し、指定のプロパティのデフォルト値を取得しています。

```
list_property_value -default BITSTREAM.GENERAL.COMPRESS [current_design]
```

関連項目

- [create_property](#)
- [current_design](#)
- [get_cells](#)
- [get_property](#)
- [list_property](#)
- [report_property](#)
- [reset_property](#)
- [set_property](#)

list_targets

指定したソースのターゲットをリストします。

構文

```
list_targets [-quiet] [-verbose] files
```

戻り値

ターゲットのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
files	ターゲットをリストするソース ファイルを指定します。

カテゴリ

プロジェクト、XPS

説明

指定の IP コア、DSP モジュール、エンベデッド プロセッサ ソース、または IP サブシステムに使用可能なターゲットをリストします。指定可能なファイル タイプは .xci、.xco、.mdl、.xmp、.bd です。

ターゲットを生成するには、**generate_targets** コマンドを使用します。

このコマンドを実行すると、使用可能なターゲットのリストが返されます。指定のファイル オブジェクトに使用可能なターゲットがない場合は、戻り値はありません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

files (必須) : ソース ファイルのリストを含むファイル オブジェクトを指定します。

注記 : ファイル名を指定するのではなく、**get_files** を使用してファイル オブジェクトを指定します。

例

次の例では、デザインに含まれる DSP モジュールに使用可能なターゲットをリストしています。

```
list_targets [get_files *.mdl]
```

関連項目

- [create_bd_design](#)
- [create_sysgen](#)
- [create_xps](#)
- [generate_target](#)
- [get_files](#)
- [import_ip](#)
- [read_ip](#)

load_features

指定の機能の Tcl コマンドを読み込みます。

構文

```
load_features [-quiet] [-verbose] [features...]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[features]	読み込む機能を指定します。使用可能な機能のリストは、list_features コマンドを使用すると取得できます。

カテゴリ

ツール

説明

Vivado Design Suite の指定の機能をメモリに読み込みます。

Vivado Design Suite のメモリ使用量を削減するため、Tcl コマンドの機能グループがあり、それらの機能グループのコマンドは、その機能グループからのコマンドを使用するか、load_features コマンドを使用して機能を読み込むまで使用できません。

たとえば、load_features simulator コマンドを使用するか、launch_xsim を使用して Vivado シミュレータを起動すると、Vivado シミュレータ用のコマンドが読み込まれます。

load_features コマンドを使用すると、ツールの機能を実際に実行せずに、Vivado Design Suite のある機能に関連するすべての Tcl コマンドをリストし、それらのコマンドのヘルプ テキストを表示できます。

list_features コマンドを使用すると、読み込むことができる機能のリストを取得できます。機能のリストは、リリースによって異なります。

このコマンドが正常に実行された場合は何も返されず、正常に実行できない場合はエラー メッセージが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

features : 読み込む機能のリストを指定します。

例

次の例では、Vivado シミュレータ機能を読み込んでいます。

```
load_features simulator
```

次の例では、Vivado Design Suite で読み込み可能な機能をすべて読み込んでいます。

```
load_features [list_features]
```

関連項目

- [help](#)
- [list_features](#)

lock_design

デザインのネットリスト、配置、または配線をロックまたはロック解除します。

構文

```
lock_design [-level arg] [-unlock] [-export] [-quiet] [-verbose] [cell]
```

戻り値

なし

使用法

名前	説明
[-level]	ロック レベルを指定します。有効な値は logical、placement、および routing で、デフォルトは placement です。
[-unlock]	セルのロックを解除します。セルを指定しない場合は、デザイン全体のロックが解除されます。
[-export]	制約がエクスポート可能であることをマークします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[cell]	ロックするセルを指定します。セルを指定しない場合は、デザイン全体がロックされます。デフォルトは * です。

カテゴリ

プロジェクト

log_saif

指定したワイヤ、信号、またはレジスタのスイッチング アクティビティを SAIF (Switching Activity Interchange Format) ファイルに記録します。

構文

```
log_saif [-quiet] [-verbose] hdl_objects ...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
hdl_objects	記録する HDL オブジェクトを指定します。

カテゴリ

シミュレーション

説明

現在のシミュレーション中に指定の HDL 信号のスイッチング アクティビティ レートを記述します。

SAIF (Switching Activity Interchange format) ファイルは、ヘッダー情報とデザインの指定の信号のトグル回数を含む ASCII ファイルです。信号が 0、1、X、または Z となる期間を指定するタイミング属性も含まれます。

`log_saif` コマンドを使用するには、`open_saif` コマンドを使用して現在のシミュレーションのスイッチング アクティビティ レートを記録する SAIF ファイルを開いておく必要があります。

引数

`-quiet` (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

`-verbose` (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

`hdl_objects` : スイッチング アクティビティを記録する HDL 信号の名前を指定します。

例

次の例では、current_scope のすべての信号のスイッチング アクティビティを記録しています。

```
log_saif [ get_objects ]
```

次の例では、スコープ /tb/UUT 内の c で始まる内部信号のみを SAIF に記録しています。

```
log_saif [get_objects -filter { type == internal_signal }/tb/UUT/c*]
```

関連項目

- [close_saif](#)
- [get_objects](#)
- [open_saif](#)

log_vcd

指定したワイヤ、信号、またはレジスタのシミュレーション出力を VCD (Value Change Dump) ファイルに記録します。

構文

```
log_vcd [-level arg] [-quiet] [-verbose] [hdl_objects ...]
```

戻り値

なし

使用法

名前	説明
[-level]	記録する HDL スコープのレベル数を指定します。デフォルトは 0 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hdl_objects]	記録する HDL オブジェクトを指定します。

カテゴリ

シミュレーション

説明

VCD (Value Change Dump) ファイルに記録する HDL オブジェクトを指定します。デザインによっては、シミュレーション結果が非常に大きくなります。log_vcd コマンドを使用すると、記録する内容を指定できます。このコマンドは、Verilog の \$dumpvars システム タスクと同様の操作を実行します。

HDL オブジェクトには、Verilog または VHDL テストベンチおよびソース ファイルで定義されている HDL 信号、変数、または定数が含まれます。HDL 信号には、Verilog の wire または reg エンティティ、および VHDL 信号が含まれます。HDL 変数には、Verilog の real、realtime、time、event などがあります。

このコマンドでは、VCD ファイルに記録する HDL オブジェクトとデザイン階層のレベル数を指定します。実際の オブジェクトの値は、シミュレーションの特定の時間に checkpoint_vcd または flush_vcd コマンドを実行したときに VCD ファイルに記述されます。

注記： *_vcd コマンドを実行する前に、open_vcd コマンドを実行する必要があります。

戻り値はありません。

引数

-level arg (オプション) : VCD ファイルに記述する HDL オブジェクトを検索するデザイン階層レベル数を指定します。デフォルト値は 0 で、*hdl_objects* で定義された階層レベルとその下のレベルすべてにある指定の HDL オブジェクトの値が記述されます。1 に設定すると、*hdl_objects* で指定した階層レベルの HDL オブジェクトの値が VCD ファイルに記述されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

hdl_objects (オプション) : VCD ファイルに値の変化を記録する HDL オブジェクトを指定します。階層レベルも指定します。たとえば `/tb/UUT/*` を指定した場合は、デザインの `/tb/UUT` レベル内のすべての HDL オブジェクトが指定されます。

例

次の例では、スコープ `/tb/UUT` に含まれるすべてのポートの値の変化を記録しています。

```
log_vcd [get_objects -filter { type == port } /tb/UUT/* ]
```

注記 : `-levels` が指定されていないので、指定したレベルの下のすべてのレベルで指定したパターンに一致するポートが検索されます。

次の例では、現在のスコープのすべてのオブジェクトを VCD ファイルに記録しています。

```
log_vcd *
log_vcd [ get_objects * ]
```

次の例では、ルートスコープ `/tb/UUT` に含まれる内部信号で、名前が C で開始するものの値の変化を記録しています。

```
log_vcd [get_objects -filter { type == internal_signal }./C*]
```

関連項目

- [checkpoint_vcd](#)
- [flush_vcd](#)
- [open_vcd](#)

log_wave

指定したワイヤ、信号、またはレジスタシミュレーション出力を Vivado シミュレーション波形ビューアーに記録します。add_wave とは異なり、波形ビューアー (波形コンフィギュレーション) に波形オブジェクトは追加しません。出力の波形データベース (wdb) への記録をイネーブルにするだけです。

構文

```
log_wave [-recursive] [-r] [-quiet] [-verbose] hdl_objects ...
```

戻り値

なし

使用法

名前	説明
[-recursive]	サブスコープでも検索を実行します。
[-r]	サブスコープでも検索を実行します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
hdl_objects	トレースする HDL オブジェクトを指定します。

カテゴリ

シミュレーション

ltrace

シミュレーションする HDL 文のファイル名と行番号の表示のオン/オフを切り替えます。

構文

```
ltrace [-quiet] [-verbose] value
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>value</i>	有効な値は、on、true、yes、off、false、no です。

カテゴリ

シミュレーション

make_diff_pair_ports

2 つのポートから差動ペアを作成します。

構文

```
make_diff_pair_ports [-quiet] [-verbose] ports...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>ports</i>	差動ペアにするポートを指定します。

カテゴリ

ピン配置

説明

既存の 2 つのポートから差動ペアを作成します。

差動ペアを構成するためには、2 つのポートの方向、インターフェイス、およびその他のプロパティが一致している必要があります。そうでない場合は、エラーが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

***ports* (必須)** : 差動ペアを作成する 2 つのポート オブジェクトを指定します。差動入力の正の側のポートを最初に指定する必要があります。

例

次の例では、ポートを 2 つ指定して差動ペアを作成しています。

```
make_diff_pair_ports port_Pos1 port_Neg1
```

関連項目

- [create_interface](#)
- [create_port](#)

make_wrapper

指定したソースの HDL ラッパーを生成します。

構文

```
make_wrapper [-top] [-testbench] [-inst_template] [-fileset arg] [-import]
[-force] [-quiet] [-verbose] files
```

戻り値

なし

使用法

名前	説明
[-top]	指定したソースの最上位ラッパーを作成します。
[-testbench]	指定したソースのテストベンチを作成します。
[-inst_template]	指定したソースのインスタンシエーション テンプレートを作成します。テンプレートはプロジェクトには追加されず、参照用にのみ作成されます。
[-fileset]	ファイルセット名を指定します。
[-import]	生成されたラッパーをプロジェクトにインポートします。
[-force]	既存のソースを上書きします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
files	ラッパーを生成するソース ファイルを指定します。

カテゴリ

プロジェクト、XPS、System Generator

説明

IP コア、DSP モジュール、またはエンベデッド プロセッサ サブデザインをプロジェクトにインスタンシエートするための Verilog または VHDL ラッパーを作成します。

注記： ラッパーは、プロジェクトの TARGET_LANGUAGE プロパティの設定に応じて Verilog または VHDL で生成されます。

このコマンドを実行すると、ラッパーの作成に関する情報が返されます。

引数

-top (オプション) : 指定したソースの最上位 Verilog または VHDL ラッパーを作成します。ラッパーにより、DSP モジュールまたはエンベデッド プロセッサ サブデザインがデザイン階層の最上位としてインスタンシエートされます。

-testbench (オプション) : 指定の DSP モジュールまたはエンベデッド プロセッサ サブデザインのシミュレーション テストベンチを作成します。これには DUT モジュール インスタンシエーションが含まれます。

-inst_template (オプション) : 指定したソースのインスタンシエーション テンプレートを作成します。テンプレートはプロジェクトには追加されず、参照用にのみ作成されます。インスタンシエーション テンプレートは別の RTL ファイルにコピーして貼り付けると、その階層でモジュールのインスタンスを作成できます。

-fileset (オプション) : プロジェクトにインポートする際にラッパーを追加するファイルセットを指定します。

-import (オプション) : ラッパー ファイルをプロジェクトにインポートし、適切なファイルセットに追加します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

files (必須) : ラッパーを生成するファイルを指定します。

例

次の例では、指定したエンベデッド プロセッサ ソースをデザイン階層に統合するためのインスタンシエーション テンプレートを作成しています。

```
make_wrapper -inst_template -fileset [get_filesets sources_1] \
  [get_files C:/Data/edk/xpsTest1/xpsTest1.xmp]
```

関連項目

- [add_files](#)
- [create_sysgen](#)
- [create_xps](#)
- [generate_target](#)
- [import_ip](#)
- [list_targets](#)
- [read_ip](#)

mark_objects

GUI でオブジェクトをマークします。

構文

```
mark_objects [-rgb args] [-color arg] [-quiet] [-verbose] objects
```

戻り値

なし

使用法

名前	説明
<code>[-rgb]</code>	色を RGB で指定します。
<code>[-color]</code>	有効な値は、red、green、blue、magenta、yellow、cyan、および orange です。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>objects</code>	マークするオブジェクト

カテゴリ

GUI 制御

説明

GUI で指定のオブジェクトをマークします。指定したオブジェクトの位置を見つけやすくするため、アイコンマークが配置されます。マークは、色オプションのいずれかで指定した色で表示されます。

オブジェクトのマークを解除するには、`unmark_objects` コマンドを使用します。

注記：1 つの色オプションのみを使用してください。両方の色オプションを指定すると、`-rgb` の方が `-color` より優先されます。

引数

`-rgb args` (オプション)：オブジェクトをマークする色を、RGB コードを使用して {R G B} の形式で指定します。たとえば、{255 255 0} は黄色を指定し、{0 255 0} は緑を指定します。

`-color arg` (オプション)：オブジェクトをマークする色を色の名前で指定します。指定可能な値は、red、green、blue、magenta、yellow、cyan、および orange です。

`-quiet` (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須) : マークするオブジェクトを 1 つ以上指定します。

例

次の例では、現在選択されているオブジェクトをマークする赤いアイコンを追加します。

```
mark_objects -color red [get_selected_objects]
```

関連項目

- [get_selected_objects](#)
- [unmark_objects](#)

move_bd_cells

セルを階層セルに移動します。セル間の接続は保持されます。これらのセルとその他のセルの接続は、階層セルをまたいで保持されます。

構文

```
move_bd_cells [-prefix arg] [-quiet] [-verbose] [parent_cell] [cells...]
```

戻り値

正しく処理された場合は 0

使用法

名前	説明
[-prefix]	セルに追加する接頭辞を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[parent_cell]	親セルを指定します。
[cells]	セル名を検索するパターンを指定します。デフォルトは*です。

カテゴリ

IP インテグレーター

説明

IP インテグレーター セルを、現在のサブシステム デザイン内の指定の階層モジュールに移動します。移動するセル間の接続は保持されます。移動するセルとその他の移動されないセルの間の接続は、IP インテグレーターにより、階層の境界をまたぐためにピンおよびポートを追加して自動的に保持されます。

IP サブシステム デザインのセルは **copy_bd_objs** を使用して階層モジュールにコピーし、**group_bd_objs** を使用してグループ化したり階層モジュールに追加したりできます。

このコマンドが正常に実行された場合は親セルの名前が返され、正常に実行されなかった場合はエラー メッセージが返されます。

引数

-prefix arg (オプション) : 階層モジュールに移動するセルに適用する接頭辞を指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

parent_cell : セルの移動先となる階層モジュールの名前を指定します。

cells (オプション) : 現在の IP サブシステム デザインから階層モジュールに移動するセルのリストを指定します。セルの指定には、**get_bd_cells** コマンドを使用します。

例

次に例を示します。

```
move_bd_cells -prefix mod1_ /myModule1 [get_bd_cells /myAxiFifo_1]  
/myModule1
```

関連項目

[copy_bd_objs](#)

move_files

ファイルを、元のプロパティを保持したまま、1 つのファイルセットから別のファイルセットに移動します。

構文

```
move_files [-fileset arg] [-quiet] [-verbose] [files...]
```

戻り値

移動されたファイルのリスト

使用法

名前	説明
[-fileset]	移動先のファイルセットの名前を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[files]	移動するファイルの名前を指定します。

カテゴリ

プロジェクト、シミュレーション

open_bd_design

ディスク ファイルから既存の IP サブシステム デザインを開きます。

構文

open_bd_design [-quiet] [-verbose] name

戻り値

デザイン オブジェクト、エラーが発生した場合はなし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	開く IP サブシステム デザインの名前を指定します。

カテゴリ

IP インテグレーター

説明

Vivado IDE の IP インテグレーター機能の IP サブシステムを開きます。IP サブシステムは、**create_bd_design** コマンドを使用して作成されている必要があります。

このコマンドでは、開いた IP サブシステム デザインの名前が返されるか、コマンドが正しく実行されなかった場合はエラーが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name : Vivado Design Suite の IP インテグレーター機能で開く IP サブシステム デザインのパスとファイル名を指定します。ファイル名にはファイル拡張子を含める必要があります。

例

次の例では、現在のプロジェクトの指定した IP サブシステム デザインを開いています。

```
open_bd_design C:/Data/project1/project1.src/sources_1/bd/design_1/design_1.bd
```

関連項目

- [close_bd_design](#)
- [create_bd_design](#)
- [current_bd_design](#)
- [save_bd_design](#)

open_example_project

指定した IP のサンプル プロジェクトを開きます。

構文

```
open_example_project [-dir arg] [-force] [-in_process] [-quiet]
[-verbose] objects...
```

戻り値

開いたプロジェクト

使用法

名前	説明
[-dir]	サンプル プロジェクトを作成するディレクトリへのパスを指定します。
[-force]	サンプル プロジェクトが既に存在する場合に上書きします。
[-in_process]	同じプロセス内にサンプル プロジェクトを開きます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>objects</i>	サンプル プロジェクトを開くオブジェクトを指定します。

カテゴリ

プロジェクト、IP フロー

説明

指定した IP コアのサンプル プロジェクトを開きます。サンプル プロジェクトを使用すると、現在のプロジェクトに統合せずに、スタンドアロン プロジェクトで IP コアの機能を試すことができます。

引数

-dir arg (オプション) : サンプル プロジェクトを保存するディレクトリへのパスを指定します。

-force (オプション) : サンプル プロジェクトを新しく開き、指定のパスに存在しているサンプル プロジェクトを上書きします。

-in_process (オプション) : 現在のプロジェクトと同じツール プロセス内にサンプル プロジェクトを開きます。このオプションを指定しない場合 (デフォルト)、サンプル プロジェクトの新しいプロセス インスタンスが開きます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須) : サンプル プロジェクトを開く IP コアを指定します。

例

次の例では、サンプル プロジェクトのターゲット データを生成し、指定の IP コアのサンプル プロジェクトを開いています。

```
generate_target {example} [get_ips blk_mem*]
open_example_project -force [get_ips blk_mem*]
```

注記 : **open_example_project** コマンドを使用する前に、example ターゲット データを生成しておく必要があります。これにより、指定の IP コアを開いてコンフィギュレーションする Tcl スクリプトが作成されます。

関連項目

- [create_ip](#)
- [generate_target](#)
- [get_ips](#)
- [import_ip](#)

open_hw

ハードウェア ツールを開きます。

構文

`open_hw [-quiet] [-verbose]`

戻り値

なし

使用法

名前	説明
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。

カテゴリ

ハードウェア

open_hw_target

ハードウェア サーバー上のハードウェア ターゲットへの接続を開きます。

構文

```
open_hw_target [-quiet] [-verbose] [hw_target]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_target]	ハードウェア ターゲットを指定します。デフォルトは、現在のハードウェア ターゲットです。

カテゴリ

ハードウェア

open_io_design

I/O ピン配置デザインを開きます。

構文

```
open_io_design [-name arg] [-part arg] [-constrset arg] [-quiet] [-verbose]
```

戻り値

デザイン オブジェクト

使用法

名前	説明
[-name]	デザイン名を指定します。
[-part]	ターゲット パーツを指定します。
[-constrset]	使用する制約ファイルセットを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

説明

新規または既存の I/O ピン配置デザインを開きます。

注記 : I/O ピン配置デザインを開くには、現在のソース ファイルセットの design_mode プロパティを PinPlanning に設定しておく必要があります。GateLvl に設定していないと、次のようなエラー メッセージが表示されます。

```
ERROR: The design mode of 'sources_1' must be PinPlanning
```

引数

-name arg (オプション) : 新規または既存の I/O ピン配置デザインの名前を指定します。

-part arg (オプション) : 新規デザインを作成する際に使用するザイリンクス デバイスを指定します。-part オプションを指定しない場合、デフォルトのパーツが使用されます。

-constrset arg (オプション) : I/O ピン配置デザインを開く際に使用する制約ファイルセットの名前を指定します。

注記 : -constrset オプションには、既存の制約ファイルセットを指定する必要があります。新しいファイルセットを作成することはできません。新しいファイルセットを作成する場合は、**create_fileset** を使用します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、myIO という I/O ピン配置デザインが新規に作成されます。

```
open_io_design -name myIO
```

注記 : この場合、デフォルトのソース セット、制約セット、パートが使用されます。

次の例では、myIO という既存の I/O デザインを開いて、使用する制約セットを指定しています。

```
open_io_design -name myIO -constrset topCon
```

関連項目

[create_project](#)

open_project

Vivado プロジェクトファイル (.xpr) を開きます。

構文

```
open_project [-read_only] [-quiet] [-verbose] file
```

戻り値

開いたプロジェクト オブジェクト

使用法

名前	説明
<code>[-read_only]</code>	プロジェクトを読み出し専用モードで開きます。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>file</code>	開くプロジェクト ファイルを指定します。

カテゴリ

プロジェクト

説明

プロジェクトファイル (.xpr) を開き、デザインソースファイルおよび階層を編集、I/O ピン配置とフロアプランを実行、デバイスを合成およびインプリメントできるようにします。

引数

-read_only (オプション) : プロジェクトを読み出し専用モードで開きます。プロジェクトへの変更は保存できません。変更を保存する必要がある場合は、**save_project_as** コマンドを使用して新しい編集可能なプロジェクトとして保存します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 開くプロジェクト ファイルを指定します。ファイルへのパスと .ppr ファイル拡張子を含める必要があります。

例

次の例では、Designs ディレクトリの my_project1 という名前のプロジェクトを開きます。

```
open_project C:/Designs/project1.xpr
```

注記：プロジェクトに拡張子 .xpr を付けて指定しないと、ツールでファイルがプロジェクト ファイルとして認識されません。プロジェクト ファイル名にファイルへのパスも含めておかないと、指定したファイルが見つからないことを示すエラー メッセージが表示されます。

関連項目

- [create_project](#)
- [current_project](#)

open_run

ネットリストまたはインプリメンテーション デザインで run を開きます。

構文

```
open_run [-name arg] [-quiet] [-verbose] run
```

戻り値

デザイン オブジェクト

使用法

名前	説明
[-name]	デザイン名を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>run</i>	デザインで開く run を指定します。

カテゴリ

プロジェクト

説明

指定の合成 run をネットリスト デザインで開くか、インプリメンテーション run をインプリメンテーション デザインで開きます。ターゲット パーツと制約セットを定義する run プロパティを合成結果またはインプリメンテーション結果と組み合わせて、ツールにデザイン ビューが作成されます。

引数

-name (オプション) : 開くデザインの名前を指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

run (必須) : 開く合成 run またはインプリメンテーション run の名前を指定します。run をデザインとして開く前に、合成またはインプリメンテーションを完了している必要があります。

注記 : 実行されていない run を開こうとすると、エラー メッセージが表示されます。

例

次の例では、指定の合成 run をネットリスト デザインとして synthPass1 という名前で開きます。

```
open_run -name synthPass1 synth_1
```

次の例では、impl_1 というインプリメンテーション run を開いています。

```
open_run impl_1
```

関連項目

[launch_runs](#)

open_saif

消費電力予測用に信号のスイッチング レートを保存するファイルを開きます。スイッチング レートは、SAIF (Switching Activity Interchange Format) 形式で記述されます。1 つのシミュレーション run に対して開くことができるのは 1 つの SAIF ファイルのみです。

構文

```
open_saif [-quiet] [-verbose] file_name
```

戻り値

開いた SAIF オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file_name	情報を保存する SAIF ファイルの名前を指定します。

カテゴリ

シミュレーション

説明

現在のシミュレーションにおいて、信号のスイッチング アクティビティを保存する SAIF (Switching Activity Interchange Format) ファイルを作成するか開きます。このファイルは、後で **report_power** コマンドで使用できます。

SAIF (Switching Activity Interchange format) ファイルは、ヘッダー情報とデザインの指定の信号のトグル回数を含む ASCII ファイルです。信号が 0、1、X、または Z となる期間を指定するタイミング属性も含まれます。

SAIF ファイルは VCD ファイルより小さいので、消費電力解析に推奨されます。

SAIF ファイルを開くと、**log_saif** コマンドを使用して、シミュレーションからのスイッチング アクティビティを SAIF ファイルに記述できます。

シミュレーション中、一度に開くことのできる SAIF ファイルは 1 つのみです。SAIF ファイルを閉じるには、**close_saif** コマンドを使用します。

このコマンドでは、開いた SAIF ファイルのオブジェクト ID が返されるか、コマンドが正しく実行されなかった場合はエラーが返されます。

引数

-quiet : コマンドをメッセージを表示せずに実行します。コマンド ライン エラーは無視され、コマンドでエラーが発生した場合でもエラー メッセージは表示されません。

-verbose : 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。

file_name : 開く SAIF ファイルの名前を指定します。

例

次の例では、指定の SAIF ファイルを開いています。

```
open_saif myData.saif
```

open_vcd

シミュレーション出力を保存する VCD (Value Change Dump) ファイルを開きます。\$dumpfile Verilog システム タスクと同様の操作を実行します。

構文

```
open_vcd [-quiet] [-verbose] [file_name]
```

戻り値

VCD オブジェクト (このオブジェクトが現在の VCD)

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[file_name]	ファイル名を指定します。デフォルトは dump.vcd です (LRM 規格)。

カテゴリ

[シミュレーション](#)

説明

シミュレーション出力を保存する VCD (Value Change Dump) ファイルを作成するか開きます。このコマンドは、Verilog の **\$dumpfile** シミュレータ指示子と同様の操作を実行します。

VCD は、ヘッダー情報、変数定義、HDL 信号値の変更の詳細を含む ASCII 形式のファイルです。VCD ファイルを使用すると、VCD ビューアーでシミュレーション結果を表示したり、デザインの消費電力予測を実行したりできます。

VCD ファイルを開くと、**checkpoint_vcd**、**flush_vcd**、または **log_vcd** コマンドを使用してシミュレーションでの値の変化を VCD ファイルに記述できます。また、**stop_vcd** および **start_vcd** コマンドを使用して、データの記述を一時停止したり再開したりできます。

VCD ファイルのサイズを制限するには、**limit_vcd** コマンドを使用します。

VCD ファイルを閉じるには、**close_vcd** コマンドを使用します。

注記 : ***_vcd** コマンドを実行する前に、**open_vcd** コマンドを実行する必要があります。一度に開くことのできる VCD ファイルは 1 つのみです。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file_name (オプション) : VCD 情報を保存するファイルの名前を指定します。ファイル名を指定しない場合、デフォルトのファイル名 `dump.vcd` が使用されます。指定の VCD ファイルが既に存在する場合は、その VCD ファイルがリセットされ、内容が上書きされます。

例

次の例では、指定の VCD ファイル (`design1.vcd`) を開き、値の変化を記述できるようにしています。**log_vcd** コマンドは /tb/UUT スコープのデザイン階層レベルのみにあるすべてのポートを特定し、VCD ファイルに記述します。シミュレーションは指定の時間実行され、**flush_vcd** コマンドで HDL オブジェクトの現在の値を VCD ファイルに記述します。その後、**close_vcd** コマンドで開いているファイルを閉じています。

```
open_vcd design1.vcd
log_vcd -level 1 [get_objects filter { type == port } /tb/UUT/* ]
run 1000
flush_vcd
close_vcd
```

関連項目

- [checkpoint_vcd](#)
- [close_vcd](#)
- [flush_vcd](#)
- [limit_vcd](#)
- [log_vcd](#)
- [read_vcd](#)
- [start_vcd](#)
- [stop_vcd](#)

open_wave_config

波形コンフィギュレーションを開きます。

構文

```
open_wave_config [-data_source arg] [-quiet] [-verbose] [filename]
```

戻り値

開いた波形コンフィギュレーション

使用法

名前	説明
[-data_source]	WCFG ファイルで参照されているデータ ソースを開くか (open)、既存のデータ ソースに接続するか (connect) を指定します。有効な値は、open および connect です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[filename]	新しい波形コンフィギュレーションおよび対応する波形ウィンドウを作成する WCFG ファイルの名前を指定します。

カテゴリ

波形

open_wave_database

以前のシミュレーションで生成された波形データベース (WDB) ファイルを開き、シミュレーション オブジェクトを返します。

構文

```
open_wave_database [-quiet] [-verbose] wdb
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
wdb	ファイル名を指定します。

カテゴリ

[シミュレーション](#)

opt_design

現在のネットリストを最適化します。デフォルトでは、リターゲット、定数伝搬、スイープ、およびブロック RAM の消費電力最適化が実行されます。

構文

```
opt_design [-retarget] [-propconst] [-sweep] [-bram_power_opt] [-remap]
[-resynth_area] [-resynth_seq_area] [-directive arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-retarget]	デバイス ファミリを移行する際に、ブロックを移行します。
[-propconst]	制約を最下位インスタンスに伝搬します。
[-sweep]	未接続の最下位インスタンスを削除します。
[-bram_power_opt]	ブロック RAM の消費電力最適化を実行します。
[-remap]	ロジックを LUT に最適に再マップします。
[-resynth_area]	再合成を実行します。
[-resynth_seq_area]	順序ロジックの最適化を含む再合成を実行します。
[-directive]	コマンドのモードを指定します。有効な値は、「引数」セクションを参照してください。デフォルトは Default です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール

説明

デザイン ネットリストをターゲット パーツ用に最適化します。最適化では、サードパーティツールからの合成済みネットリストや合成中に最適化されなかったネットリストを向上できます。

このコマンドをインプリメンテーションでデザインを配置配線する前に実行し、デザインを最適化してネットリストを簡略化します。

opt_design コマンドは、デフォルトで次の最適化を実行します。

- ・ 再ターゲット
- ・ 定数伝搬
- ・ スイープ
- ・ ブロック RAM 消費電力最適化

注記：特定の最適化のコマンドライン オプションを使用すると、その最適化のみが実行され、ほかの最適化は通常デフォルトで実行される最適化であってもディスエーブルになります。

LUT 再マップを実行するには、**-remap** を指定する必要があります。

エリア ベースの再合成を実行するには、**-resynth_area** または **-directive ExploreArea** を指定する必要があります。

順序ロジック エリア ベースの再合成を実行するには、**-resynth_seq_area** または **-directive ExploreSequentialArea** を指定する必要があります。

引数

-retarget (オプション)：デザインをあるデバイス ファミリから別のファミリに移行する際に、あるタイプのブロックを別のタイプのブロックに移行します。たとえば、インスタンシエートされた MUXCY または XORCY コンポーネントを CARRY4 ブロックに、DCM を MMCM に移行します。リターゲット最適化では、可能な場合に、インバーターのダウンストリーム ロジックへの吸収も実行されます。

注記：**-retarget** の指定はオプションですが、ほかの最適化のコマンドライン オプションを指定していなければ、この最適化はデフォルトで実行されます。

-propconst (オプション)：定数入力を回路で伝搬し、ネットリストを簡略化します。定数の伝搬により、ネットリストから余分な組み合わせロジックを削除できる可能性があります。

-sweep (オプション)：ロードのないセルおよびネットを削除することにより、不要なロジックを削除します。

-bram_power_opt (オプション)：ブロック RAM セルで消費電力の最適化をイネーブルにします。完全なデュアル ポート RAM の読み出されないポートの WRITE_MODE を NO_CHANGE に変更し、ブロック RAM の出力にクロック ゲーティングを適用します。

-remap (オプション)：デザインを再マップし、複数の LUT を 1 つの LUT に結合してロジックの段数を削減します。

-resynth_area (オプション)：エリア モードで合成を再実行し、LUT の数を削減します。

-resynth_seq_area (オプション)：組み合わせロジックおよび順序ロジックを削減するよう再合成を実行します。**-resynth_area** の最適化のスーパーセットを実行します。

-directive arg (オプション) : 最適化モード (デザインの目標) を指定します。1 つの **opt_design** コマンドに対して 1 つのモードのみを指定できます。値では大文字/小文字が区別されます。有効な値は次のとおりです。

- **Explore** : 結果を向上するために最適化を複数回実行します。
- **ExploreArea** : エリアを縮小することを優先して最適化を複数回実行します。
- **ExploreSequentialArea** : レジスタおよび関連の組み合わせロジックを縮小することを優先して最適化を複数回実行します。
- **AddRemap** : デフォルトの最適化を実行し、LUT 再マップを含めてロジック レベルを削減します。
- **RuntimeOptimized** : 反復回数を少なくし、最適化の結果よりも実行時間を短縮することを優先します。
- **Default** : デフォルトの最適化を実行します。

各モードの効果は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

注記 : **-directive** オプションは全体的な最適化ストラテジを制御するもので、特定の最適化オプションとは互換性がありません。**-quiet** および **-verbose** とのみ使用できます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。各最適化により変更されたロジックに関する詳細な情報が表示されます。

例

次の例では、3 つのデフォルトの最適化 (リターゲット、定数伝搬、およびスイープ) を実行し、詳細な結果を取得しています。

```
opt_design -verbose
```

次の例では、スイープおよび再マップ最適化を実行しています。

```
opt_design -sweep -retarget
```

注記 : この例では **-sweep** と **-retarget** が指定されているため、**-propconst** および **-bram_power_opt** 最適化はディスエーブルになります。

次の例では、よりよい結果が得られるように、**opt_design** コマンドでさまざまなアルゴリズムを使用しています。

```
opt_design -directive Explore
```

次の例では、エリアを縮小することを優先しながら、よりよい結果が得られるように **opt_design** コマンドでさまざまなアルゴリズムを使用しています。

```
opt_design -directive ExploreArea
```

関連項目

- [phys_opt_design](#)
- [place_design](#)
- [power_opt_design](#)
- [route_design](#)
- [synth_design](#)

phys_opt_design

現在の配置済みネットリストを最適化します。

構文

```
phys_opt_design [-fanout_opt] [-placement_opt] [-rewire] [-critical_cell_opt]
[-dsp_register_opt] [-bram_register_opt] [-hold_fix] [-retime]
[-force_replication_on_nets args] [-directive arg] [-pinswap] [-quiet]
[-verbose]
```

戻り値

なし

使用法

名前	説明
[-fanout_opt]	ファンアウトの大きいタイミング クリティカル ネットをセルを複製して最適化します。
[-placement_opt]	タイミング クリティカル ネットに対して配置に基づく最適化を実行します。
[-rewire]	配線を再実行します。
[-critical_cell_opt]	タイミング クリティカル ネットをセルを複製して最適化します。
[-dsp_register_opt]	DSP レジスタの最適化を実行します。
[-bram_register_opt]	ブロック RAM レジスタの最適化を実行します。
[-hold_fix]	ホールド違反のスラックの向上を試みます。
[-retime]	リタイミング最適化を実行します。
[-force_replication_on_nets]	ネットに対して複製最適化を強制的に実行します。
[-directive]	コマンドのモードを指定します。有効な値は、「引数」セクションを参照してください。デフォルトは Default です。
[-pinswap]	ピン スwap最適化を実行します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール

説明

デザインの負のスラック パスに対してタイミング ドリブンの最適化を実行します。最適化が実行されるためには、パスの負のスラックがワースト ネガティブ スラック (WNS) に近いことが必要です。このコマンドは、**place_design** コマンドの後、**route_design** コマンドの前に実行してください。

デフォルトでは、ファンアウトの大きいネットの最適化 (**-fanout_opt**)、クリティカル パスの配置に基づく最適化 (**-placement_opt**)、配線の再実行 (**-rewire**)、クリティカル セルの最適化 (**-critical_cell_opt**)、DSP レジスタの最適化 (**-dsp_register_opt**)、ブロック RAM レジスタの最適化 (**-bram_register_opt**)、および最終的なファンアウト最適化が実行されます。

注記：特定の最適化を明示的に指定すると、ほかの最適化はディスエーブルになります。

物理最適化には、複製、リタイミング、ホールド違反の修正、および配置の向上が関連します。**phys_opt_design** コマンドでは、ネットリストおよび配置の必要な変更が自動的に実行されます。

ホールド違反の修正を実行するには、**-hold_fix** オプションまたは **-directive Explore** オプションを指定する必要があります。

このコマンドを複数回実行すると、次の実行では前回の結果が最適化されます。

このコマンドを実行すると、処理された各ネット、実行された最適化のサマリ、および最適化前後の WNS がレポートされます。複製されたオブジェクトの名前は、元のオブジェクト名に **_replica** と複製されたオブジェクトカウントが付いたものになります。

引数

-fanout_opt (オプション)：ファンアウトの大きいタイミング クリティカル ネットに対し、ドライバーを複製して遅延を削減することにより、遅延ドリブンの最適化を実行します。

-placement_opt (オプション)：タイミング クリティカル ネットの遅延を削減するようセルを移動します。

-rewire (オプション)：ロジック コーンを再構築してロジック レベルを削減し、クリティカル信号の遅延を削減します。

-critical_cell_opt (オプション)：タイミング クリティカル ネットのセルを複製して遅延を削減します。

-dsp_register_opt (オプション)：レジスタをスライスから DSP ブロックに移動したり、DSP ブロックからスライスに移動したりして、クリティカル パス遅延を削減します。

-bram_register_opt (オプション)：レジスタをスライスからブロック RAM に移動したり、ブロック RAM からスライスに移動したりして、クリティカル パス遅延を削減します。

-hold_fix (オプション)：データ パス遅延を挿入し、ホールド タイム違反を修正します。

-retime (オプション)：レジスタを組み合わせロジックの前後に移動してリタイミングし、パス遅延のバランスを取ります。

-directive arg (オプション) : 物理最適化モード (デザインの目標) を指定します。1 つの **phys_opt_design** コマンドに対して 1 つのモードのみを指定できます。値では大文字/小文字が区別されます。有効な値は次のとおりです。

- **Explore** : 最適化を複数回実行し、ホールド違反の修正およびファンアウトの大きいネットの複製を含め、異なるアルゴリズムを実行します。
- **AggressiveExplore** : Explore と似ていますが、異なる最適化アルゴリズムが使用され、より厳しい目標が設定されます。
- **AlternateReplication** : クリティカル セルの複製に異なるアルゴリズムを使用します。
- **AggressiveFanoutOpt** : ファンアウトに関連する最適化に異なるアルゴリズムを使用し、より厳しい目標を設定します。
- **AlternateDelayModeling** : ネット遅延の予測に代替アルゴリズムを使用して、すべての最適化を実行します。
- **AddRetime** : デフォルトの **phys_opt_design** フローを実行し、レジスタのリタイミングを追加します。
- **Default** : **phys_opt_design** をデフォルト設定で実行します。

各モードの効果は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

注記 : **-directive** オプションは全体的な最適化ストラテジを制御するもので、特定の最適化オプションとは互換性がありません。**-quiet** および **-verbose** とのみ使用できます。

-force_replication_on_nets args (オプション) : タイミング スラックにかかわらず、指定のネットのドライバーを強制的に複製します。複製はロードの配置に基づいて実行され、複製が十分かどうかを解析する必要があります。さらに複製が必要な場合は、コマンドを複数回実行することによりネットを繰り返し複製できます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、開いているデザインに対して物理最適化を実行しています。

```
phys_opt_design
```

次の例では、レジスタのリタイミングと、DSP ブロックおよびブロック RAM でのレジスタの最適化を実行しています。

```
phys_opt_design -retime -dsp_register_opt -bram_register_opt
```

次の例では、よりよい結果を得るため、**phys_opt_design** の実行回数を増やしています。

```
phys_opt_design -directive Explore
```

次の例では、ネットの複製がより考慮されるよう指定しています。

```
phys_opt_design -directive AggressiveFanoutOpt
```

関連項目

- [opt_design](#)
- [place_design](#)
- [power_opt_design](#)
- [route_design](#)

place_cell

1 つまたは複数のインスタンスを新しい場所に移動または配置します。サイトおよびセルは正しい順序でリストし、サイトの数とセルの数は同じにする必要があります。

構文

```
place_cell [-quiet] [-verbose] cell_site_list ...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
cell_site_list	セルおよびサイトを交互にリストします。

カテゴリ

フロアプラン

説明

ターゲット パーツのデバイスリソースにセルを配置します。セルは、特定の BEL サイト (**SLICE_X49Y60/A6LUT** など) または使用可能なスライスリソース (**SLICE_X49Y60** など) に配置できます。スライスを指定して BEL を指定しない場合、ツールにより指定されたスライス内の適切な BEL が検索されます。

指定のサイトにセルを配置する際、そのサイトに既に別のオブジェクトが配置されている場合、次のようなエラーが返されます。

```
"Cannot set site and bel property of instances. Site SLICE_X49Y61 is already occupied."
```

サイトに既にオブジェクトが配置されているかどうかは、BEL サイトの **IS_OCCUPIED** プロパティをクエリすると確認できます。

```
get_property IS_OCCUPIED [get_bels SLICE_X48Y60/D6LUT]
```

注記：スライスの **IS_OCCUPIED** プロパティをクエリしても、スライス内に既にオブジェクトが配置されている BEL があるかどうかがわかるだけで、スライスが完全に占有されているかどうかはわかりません。

このコマンドは、セルを配置する場合、およびセルをデバイス上のあるサイトから別のサイトに移動する場合に使用できます。配置されていないセルを配置する構文も、配置されている移動する構文も同じです。

配置されているセルを移動する際、サイトのスライスのみを指定した場合は、セルが現在配置されているのと同様に、新しいスライスの同じ BEL サイトに配置するよう試みられます。たとえば、B6LUT からセルを移動する場合に新しいスライスを指定すると、新しいスライスの B6LUT にセルを配置するよう試みられます。この BEL サイトに既にオブジェクトが配置されている場合は、エラーが返されます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

cell_site_list (必須) : {*cell_name site*} という形式でセルおよびサイトのリストを指定します。まずセル名を指定し、その後にセルを配置する BEL サイトまたはスライスを指定します。スライスを指定すると、そのスライス内の使用可能な BEL が選択されます。次のようにセル/サイトのペアを複数リストして、複数のセルを複数のサイトに配置できます。

```
{cell_name1 site1 cell_name2 site2 cell_name3 site3 ...cell_nameN siteN}
```

例

次の例では、指定したセルを指定した BEL サイトに配置しています。

```
place_cell div_cntr_reg_inferredi_4810_15889 SLICE_X49Y60/D6LUT
```

次の例では、指定したセルを指定したスライスに配置しています。

```
place_cell div_cntr_reg_inferredi_4810_15889 SLICE_X49Y61
```

注記：ツールにより、適切な BEL サイトが選択されます。使用可能な BEL が存在しない場合は、エラーが返されます。

次の例では、複数のセルを複数のサイトに配置しています。

```
place_cell { \
cpuEngine/cpu_iwb_adr_o/buffer_fifo/i_4810_17734 SLICE_X49Y60/A6LUT \
cpuEngine/or1200_cpu/or1200_mult_mac/i_4775_15857 SLICE_X49Y60/B6LUT \
cpuEngine/cpu_iwb_adr_o/buffer_fifo/xlnx_opt_LUT_i_4810_18807_2 SLICE_X49Y60/C6LUT }
```

関連項目

- [create_cell](#)
- [remove_cell](#)
- [unplace_cell](#)

place_design

ポートと最下位インスタンスを自動配置します。

構文

```
place_design [-directive arg] [-no_timing_driven] [-unplace] [-cells args]
[-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-directive]	コマンドのモードを指定します。有効な値は、「引数」セクションを参照してください。デフォルトは Default です。
[-no_timing_driven]	タイミング ドリブン モードで実行しません。
[-unplace]	制約で固定されていないすべてのインスタンスの配置を解除します。
[-cells]	指定したセルに対して操作を実行します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール

説明

現在のデザインに含まれる指定のポートおよびロジック インスタンスまたはすべてのポートおよびロジック セルを、ターゲット パーツのデバイスリソースに配置します。負のタイミング スラックを最小限に抑え、ワイヤの長さをなるべく短くし、同時に配置を分散させて配線が密集しないように、ツールにより配置が最適化されます。

配置はインプリメンテーションの 1 つの段階です。Vivado ツールをプロジェクト モードで実行している場合は、インプリメンテーションは **launch_runs** コマンドを使用してすべてを自動的に実行できます。

非プロジェクト モードでは、インプリメンテーション プロセスを **opt_design**、**place_design**、**phys_opt_design**、**power_opt_design**、および **route_design** の各コマンドを使用して手動で実行する必要があります。プロジェクト モードおよび非プロジェクト モードの詳細は、『Vivado Design Suite ユーザー ガイド：デザインフローの概要』(UG892) を参照してください。

配置および配線はどちらも、インクリメンタル コンパイル フローを使用して、デザイン チェックポイント ファイル (DCP) に保存されている結果に基づいて、インクリメンタルに実行できます。インクリメンタル配置および配線の詳細は、**read_checkpoint** コマンドまたは『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

また、**place_ports** を使用するか、またはセルに LOC プロパティを設定して、デザインの一部のエレメントを手動で配置しておいてから、残りの部分を **place_design** を使用して自動的に配置することも可能です。

このコマンドを実行するには、合成済みデザインを開いている必要があります。最適でないネットリストを配置するのを避けるため、**opt_design** コマンドを実行しておくことをお勧めします。

引数

-directive arg (オプション) : 特定のデザイン目標を達成するよう配置します。1 つの **place_design** コマンドに対して 1 つのモードのみを指定できます。値では大文字/小文字が区別されます。有効な値は次のとおりです。

- **Explore** : 詳細配置および配置後の最適化のエフォートを増加します。
- **WLDrivenBlockPlacement** : RAM および DSP ブロックをワイヤ長に基づいて配置します。タイミング ドリブンの配置を無効にし、ブロックとの接続距離を最短にするよう配置します。
- **LateBlockPlacement** : RAMB および DSP ブロックの詳細配置を、配置の最終段階で実行します。通常、ブロックは配置プロセスの初期段階で有効なサイトに割り当てられます。その代わり、まず大まかなブロック配置を実行し (適切な列に揃っていない可能性もあり)、詳細配置中にブロックを有効なサイトに配置します。
- **ExtraNetDelay_high** : ファンアウトが大きく距離の長いネットの予測遅延を増加します。high、medium、low の 3 つのレベルがサポートされます。**ExtraNetDelay_high** では増加量が最も大きくなります。
- **ExtraNetDelay_medium** : ファンアウトが大きく距離の長いネットの予測遅延を増加します。high、medium、low の 3 つのレベルがサポートされます。**ExtraNetDelay_medium** はデフォルト レベルの増加量を適用します。
- **ExtraNetDelay_low** : ファンアウトが大きく距離の長いネットの予測遅延を増加します。high、medium、low の 3 つのレベルがサポートされます。**ExtraNetDelay_low** では増加量が最も小さくなります。
- **SpreadLogic_high** : ロジックをデバイス全体に分散します。high、medium、low の 3 つのレベルがサポートされます。**SpreadLogic_high** では分散度が最も高くなります。
- **SpreadLogic_medium** : ロジックをデバイス全体に分散します。high、medium、low の 3 つのレベルがサポートされます。**SpreadLogic_medium** では分散度が中程度になります。
- **SpreadLogic_low** : ロジックをデバイス全体に分散します。high、medium、low の 3 つのレベルがサポートされます。**SpreadLogic_low** では分散度が最も低くなります。
- **ExtraPostPlacementOpt** : 配置後の最適化のエフォートを増加します。
- **SSI_ExtraTimingOpt** : SLR 間でのタイミング ドリブンの分割に代替アルゴリズムを使用します。
- **SSI_SpreadSLLs** : SLR 間で分割を実行し、接続の多い領域に追加のエリアを割り当てます。
- **SSI_BalanceSLLs** : SLR 間で SLL のバランスが取られるように SLR 間で分割を実行します。
- **SSI_BalanceSLRs** : SLR 間でセルの数のバランスが取られるように SLR 間で分割を実行します。
- **SSI_HighUtilSLRs** : 各 SRL でロジックを近くに配置するよう指定します。
- **RuntimeOptimized** : 反復回数を少なくし、デザイン パフォーマンスよりも実行時間を短縮することを優先します。
- **Quick** : 最も高速な、タイミング ドリブンでない、有効なデザインを得るために最低限の配置を実行します。
- **Default** : **place_design** をデフォルト設定で実行します。

配置ストラテジおよび **-directive** オプションの詳細は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

注記 : **-directive** オプションは全体的な配置ストラテジを制御するもので、特定の配置オプションとは互換性がありません。**-quiet** および **-verbose** とのみ使用できます。また、**read_checkpoint -incremental** を使用してインクリメンタル コンパイル フローが指定されている場合は、**-directive** オプションは無視されます。

-no_timing_driven (オプション) : デフォルトのタイミング ドリブン配置アルゴリズムをディスエーブルにします。このオプションを使用するとワイヤの長さに基づいて高速な配置が実行されますが、タイミング制約は無視されます。

-unplace (オプション) : 制約で固定されていないすべてのインスタンスの配置を解除します。配置が固定されているセル (IS_LOC_FIXED が true) は変更されません。

注記 : 固定されているセルの配置を解除するには、**set_property** を使用して IS_LOC_FIXED を False に設定しておく必要があります。

-cells args (オプション) : 指定したセル オブジェクトを配置または配置解除します。このオプションを指定しない場合、**place_design** コマンドまたは **place_design -unplace** コマンドはすべてのセルに対して実行されます。セルは **get_cells** コマンドで指定する必要があります。名前で指定することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインを配置しています。

```
place_design
place_design -unplace -cells \
[get_cells {usbEngine0/usbEngineSRAM/snoopyRam_reg_6 \
usbEngine0/dma_out/buffer_fifo/infer_fifo.block_ram_performance fifo_ram_reg \
usbEngine0/usbEngineSRAM/snoopyRam_reg_25}]
```

次の例では、よりよい配置結果を得るために、異なる配置アルゴリズムを試しています。

```
place_design -directive Explore
```

次の例では、指定したセル オブジェクトの配置を解除しています。

```
place_design
place_design -unplace -cells \
[get_cells {usbEngine0/usbEngineSRAM/snoopyRam_reg_6 \
usbEngine0/dma_out/buffer_fifo/infer_fifo.block_ram_performance fifo_ram_reg \
usbEngine0/usbEngineSRAM/snoopyRam_reg_25}]
```

関連項目

- [launch_runs](#)
- [opt_design](#)
- [place_ports](#)
- [phys_opt_design](#)
- [power_opt_design](#)
- [read_checkpoint](#)
- [route_design](#)
- [set_property](#)

place_pblocks

Pblock をスライスの容量に基づいてサイズ変更し、接続に基づいて再配置します。

構文

```
place_pblocks [-effort arg] [-utilization arg] [-quiet] [-verbose] pblocks...
```

戻り値

なし

使用法

名前	説明
[-effort]	Pblock ごとの配置ツールのエフォート レベルを指定します。有効な値は LOW、MEDIUM、HIGH で、デフォルトは HIGH です。
[-utilization]	Pblock ごとの使用率を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>Pblock</i>	配置する Pblock のリストを指定します。

カテゴリ

フロアプラン

説明

Pblock をFPGA のファブリックに配置します。Pblock は **create_pblock** コマンドを使用して作成し、**add_cells_to_pblock** コマンドを使用してロジックを割り当てておく必要があります。

注記：空の Pblock も指定どおり配置されますが、1 つの CLB タイル (2 つのスライス) が使用されます。

引数

-effort arg (オプション)：各 Pblock をファブリックに配置するのに Pblock 配置ツールで使用するエフォート レベルを指定します。有効な値は LOW、MEDIUM、HIGH で、デフォルトは HIGH です。

-utilization arg (オプション)：Pblock が FPGA に配置されたときに、Pblock に割り当てられたロジック エレメントで消費されるデバイスリソースの割合を指定します。たとえば 50% に指定した場合、半分のリソースを Pblock のロジックに割り当て、残りの半分のリソースはその他のデザイン エレメントを組み込むために残すように指定されます。使用率を大きくすると Pblock は小さくなり配置が困難になりますが、使用率を小さくすると Pblock が大きくなります。

注記：Pblock の使用率は合成後の予測です。実際の結果は異なる可能性があり、**resize_pblock** コマンドを使用して Pblock のサイズ調整が必要な場合があります。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

pblocks (必須) : FPGA のファブリックに配置する Pblock を 1 つ以上指定します。

例

次の例では、指定した Pblock を使用率を 75% に設定して配置しています。

```
place_pblocks -effort LOW -utilization 75 block1 block2 block3 block4 block5
```

関連項目

- [add_cells_to_pblock](#)
- [create_pblock](#)
- [resize_pblock](#)

place_ports

ポートのセットを自動的に配置します。

構文

```
place_ports [-skip_unconnected_ports] [-check_only] [-iobank args] [-quiet]
[-verbose] [ports...]
```

戻り値

なし

使用法

名前	説明
[-skip_unconnected_ports]	接続のないポートは配置しません。
[-check_only]	I/O クロック配置 DRC のみをチェックします。
[-iobank]	配置を指定したバンクに制限します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[ports]	配置するポートを指定します。指定しない場合、すべてのポートが配置されます。

カテゴリ

ピン配置

説明

ポートを使用可能な I/O またはクロック サイト、あるいは指定の I/O バンクに自動配置します。

place_ports コマンドでは、ユーザーにより既に配置されているポート、配置および固定されているポートは変更されません。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-skip_unconnected_ports (オプション)：接続されていないポートは配置しません。

-check_only (オプション)：クロック配置 DRC を実行します。これは、report_drc コマンドで使用可能な PLCK チェックでもあります。このオプションを使用すると、ポートは配置されず、有効な配置がチェックされるのみです。

-iobank args (オプション) : ポートを指定の I/O バンク オブジェクトに配置します。I/O バンク オブジェクトは、**get_iobanks** コマンドで返されます。

注記 : ポートの配置を特定の I/O バンクに制限した場合、指定のポート数を配置するのに十分なサイトがないと、配置エラーが発生します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

ports (オプション) : 配置するポートの名前を指定します。ポートを指定しない場合は、すべてのポートが配置されます。

注記 : 既に配置されているポートを指定した場合、これらのポートは置き換えられたり移動されたりしません。

例

次の例では、**get_ports** コマンドで返されたポート オブジェクトを、**get_iobanks** で返されたデバイスの I/O バンク 13 に配置しています。

```
place_ports -iobank [get_iobanks 13] [get_ports DataOut_pad_1_o]
```

次の例では、すべての入力ポートをデバイスの I/O バンク 12、13、14、および 15 に配置しています。

```
place_ports -iobank [get_iobanks {12 13 14 15}] [all_inputs]
```

関連項目

- [create_port](#)
- [get_iobanks](#)
- [make_diff_pair_ports](#)
- [remove_port](#)

power_opt_design

高度なクロック ゲーティングを使用して、ダイナミック消費電力を最適化します。

構文

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

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

電力

説明

フリップフロップのクロック イネーブルを利用してクロック ゲーティングを変更することにより、デザインのダイナミック消費電力を最適化します。クロック ゲーティング最適化はデザイン全体に対して自動的に実行され、デザインの動作が変化する可能性のある既存のロジックやクロックを変更せずに消費電力を削減します。

power_opt_design の前に **read_saif** または **read_vcd** コマンドを使用して、デザインの最適化でアクティビティデータが考慮されるようにすることもできます。

set_power_opt コマンドを使用して、消費電力最適化に特定のセルを含めるか除外するかを指定できます。

消費電力最適化は、合成後または配置後に実行してください。このコマンドを配置前に実行すると、デザインの消費電力を削減するために最適化が実行されます。配置後に実行すると、タイミングを保持しながらデザインの消費電力を削減するために最適化が実行されます。配置後に実行すると、**power_opt_design** コマンドで実行可能な最適化は制限されます。最適な結果を得るには、このコマンドを配置前に実行してください。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、開いているデザインの消費電力最適化を実行しています。

```
power_opt_design
```

関連項目

- [opt_design](#)
- [phys_opt_design](#)
- [read_saif](#)
- [read_vcd](#)
- [report_power](#)
- [report_power_opt](#)
- [set_power_opt](#)

pr_verify

2 つのデザイン チェックポイントがボード上で置換可能であるかどうかを検証します。

構文

```
pr_verify [-full_check] [-quiet] [-verbose] file1 file2
```

戻り値

なし

使用法

名前	説明
[-full_check]	デフォルトでは、最初の差異のみがレポートされます。このオプションを true に設定すると、配置または配線の差異がすべてレポートされます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file1	デザイン チェックポイント (.dcp) ファイル 1 を指定します。
file2	デザイン チェックポイント (.dcp) ファイル 2 を指定します。

カテゴリ

ファイル入力および出力

program_hw_devices

ハードウェア デバイスをプログラムします。

構文

```
program_hw_devices [-quiet] [-verbose] [hw_device ...]
```

戻り値

ハードウェア デバイスのリスト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_device]	ハードウェア デバイスのリストを指定します。デフォルトは現在のハードウェア デバイスです。

カテゴリ

ハードウェア

ptrace

シミュレーションする HDL プロセス名の表示のオン/オフを切り替えます。

構文

```
ptrace [-quiet] [-verbose] value
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
value	有効な値は、on、true、yes、off、false、no です。

カテゴリ

シミュレーション

read_checkpoint

デザイン チェックポイントを読み込みます。

構文

```
read_checkpoint [-incremental] [-part arg] [-quiet] [-verbose] file
```

戻り値

なし

使用法

名前	説明
[-incremental]	インプリメンテーションを再利用するデザイン チェックポイント ファイルを指定します。
[-part]	チェックポイント パーツを変更します。チェックポイントに XDEF が含まれている場合、エラーが発生することがあります。-cell を使用している場合は無視されます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>file</i>	デザイン チェックポイント ファイルを指定します。

カテゴリ

ファイル入力および出力

説明

ネットリスト、制約、およびオプションでインプリメント済みデザインの配置配線情報を含むデザイン チェックポイント ファイル (DCP) を読み込みます。デザインのどの段階でもデザイン チェックポイントを保存し、復元できます。

チェックポイントを読み込む場合、あらかじめプロジェクトを作成する必要はありません。**read_checkpoint** コマンドは、デザイン データをメモリに読み込み、デザインを非プロジェクトモードで開きます。プロジェクトモードおよび非プロジェクトモードの詳細は、『Vivado Design Suite ユーザー ガイド：デザイン フローの概要』(UG892) を参照してください。

また、デザイン チェックポイントをインポートして、新しいターゲット パーツを指定して新しいザイリンクス デバイスに移行することもできます。

注記： Vivado ツールで複数のデザイン チェックポイントを開いている場合、**current_project** コマンドを使用してデザインを切り替える必要があります。**current_design** を使用して、どのチェックポイントがアクティブ デザインかを確認できます。

引数

-incremental arg (オプション) : 既に開いているデザインにチェックポイント ファイルを読み込み、インクリメンタル コンパイル デザイン フローをイネーブルにします。arg には、インクリメンタル デザイン チェックポイント (DCP) ファイルのパスとファイル名を指定します。インクリメンタル コンパイル フローでは、インクリメンタル デザインからのセル配置およびネット配線が既存のデザイン オブジェクトと比較され、メモリ内のデータベースに適用されます。インクリメンタル デザイン チェックポイントを読み込んだら、**report_incremental_reuse** コマンドを使用してインクリメンタル チェックポイントから現在のデザインに再利用される物理データの割合を確認できます。**place_design** および **route_design** コマンドでインクリメンタル配置配線が実行され、再利用された配置配線情報が保持されて、デザイン ソリューションに組み込まれます。

注記 : **-incremental** オプションでデザイン チェックポイントを読み込むと、物理データが現在のメモリ内のデザインに読み込まれます。インクリメンタル デザイン データをクリアするには、**open_run** を使用して合成済みデザインを再読み込みするか、新しいインクリメンタル デザインを読み込んで既に読み込まれているものを上書きします。インクリメンタル配置および配線の詳細は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

-part arg (オプション) : インポートしたチェックポイント デザインのターゲット パーツを指定します。このオプションは、ネットリストおよび制約を含むチェックポイント ファイルをインポートする際に有益です。また、インプリメント済みデザインの配置配線情報を含むチェックポイント ファイルに対してターゲット パーツを変更する際にも使用できます。これは通常、ネットリストおよび制約を読み込んだ後にエラーとなります。

次に **read_checkpoint -incremental** コマンドを実行すると、前のインクリメンタル チェックポイントから再利用された物理データが置き換えられます。インクリメンタル チェックポイントを読み込んだ後にインクリメンタル配置配線 フローを実行するつもりでない場合は、元のチェックポイントを再度読み込む必要があります。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : チェックポイント ファイルのパスとファイル名を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次の例では、指定したチェックポイント ファイルをインポートし、デザインのターゲット パーツを指定しています。

```
read_checkpoint C:/Data/state1/checkpoint.dcp -part xc7k325tffg900-2
```

関連項目

- [current_design](#)
- [current_project](#)
- [write_checkpoint](#)

read_csv

パッケージ ピンとポート配置情報をインポートします。

構文

```
read_csv [-quiet_diff_pairs] [-quiet] [-verbose] file
```

戻り値

なし

使用法

名前	説明
[-quiet_diff_pairs]	I/O ポートをインポートする際に、差動ペアの推論に関する警告メッセージを表示しないようにします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	ピン配置 CSV ファイルを指定します。

カテゴリ

ファイル入力および出力

説明

CSV (Comma Separated Value) ファイルから、ポート定義およびパッケージ ピン配置情報をインポートします。

CSV ファイルのポート定義は、I/O ピン配置プロジェクトにインポートできます。ピン配置プロジェクトでは、CSV ファイルをインポートすると、現在のポート定義が置き換えられます。インポートした CSV ファイルに存在しないポートは削除されます。

その他のプロジェクトでは、ポート定義はソース デザイン データに定義されていますが、パッケージ ピンの割り当ておよびポート属性は指定の CSV ファイルから読み込むことができます。

CSV ファイルのポート名にスペースを含めることはできません。CSV ファイルから読み込まれたポート名にスペースが含まれていると、エラー メッセージが表示されます。CSV ファイルの形式および要件は、『Vivado Design Suite ユーザー ガイド：I/O およびクロックの配置』(UG899) を参照してください。

引数

-quiet_diff_pairs (オプション)：CSV ファイルをインポートする際、差動ペアと認識されたピンに関するメッセージが表示されます。このオプションを指定すると、差動ペアの推論に関するメッセージは表示されません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

file (必須) : インポートする CSV ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次の例では、開いているプロジェクトに CSV ファイルをインポートしています。

```
read_csv C:/Data/pinList.csv
```

次の例では、新しい I/O ピン配置プロジェクトを設定してから、指定した CSV ファイルをインポートし、CSV ファイルの差動ペアを推論しています。

```
create_project myPinPlan C:/Data/myPinPlan -part xc7v285tffg1157-1
set_property design_mode PinPlanning [current_fileset]
open_io_design -name io_1
read_csv C:/Data/import.csv
infer_diff_pairs -filetype csv C:/Data/import.csv
```

注記 : プロジェクトの特性は、ソースファイルセットの `design_mode` プロパティで指定します。

関連項目

- [create_project](#)
- [infer_diff_pairs](#)
- [open_io_design](#)
- [set_property](#)
- [write_csv](#)

read_edif

1 つまたは複数の EDIF または NGC ファイルを読み込みます。

構文

`read_edif [-quiet] [-verbose] files`

戻り値

追加されたファイル オブジェクトのリスト

使用法

名前	説明
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>files</code>	EDIF または NGC ファイルの名前を指定します。

カテゴリ

ファイル入力および出力

説明

EDIF または NGC ネットリスト ファイルを現在のプロジェクトのデザイン ソース ファイルセットにインポートします。

引数

`-quiet` (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

`-verbose` (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

`files` (必須) : インポートする EDIF または NGC ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次の例では、開いているプロジェクトに EDIF ファイルをインポートしています。

```
read_edif C/Data/bft_top.edf
```

関連項目

[write_edif](#)

read_hw_ilab_data

ファイルからハードウェア ILA データを読み出します。

構文

```
read_hw_ilab_data [-quiet] [-verbose] file
```

戻り値

ハードウェア ILA データ オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	ハードウェア ILA データ ファイル名を指定します。

カテゴリ

ハードウェア

read_hw_sio_scan

ファイルからハードウェア SIO スキャン データを読み出します。ハードウェア SIO スキャン データを指定しない場合は、作成されます。

構文

```
read_hw_sio_scan [-quiet] [-verbose] file [hw_sio_scan]
```

戻り値

ハードウェア SIO スキャン オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	ハードウェア SIO スキャン ファイル名を指定します。
[hw_sio_scan]	ハードウェア SIO スキャン データ オブジェクトを指定します。 デフォルトはなしです。

カテゴリ

ハードウェア

read_ip

1 つまたは複数の IP ファイルを読み込みます。

構文

`read_ip [-quiet] [-verbose] files`

戻り値

追加された IP ファイル オブジェクトのリスト

使用法

名前	説明
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>files</code>	IP ファイル名を指定します。

カテゴリ

ファイル入力および出力、IP フロー

説明

指定した IP ファイルのリストを読み込み、デザインおよび現在のファイルセットに追加します。

`add_files` コマンドと同様に、ファイルは参照で現在のプロジェクトに追加されます。

Vivado ツールを非プロジェクトモードで実行しており、プロジェクトソースファイルを管理するプロジェクトファイルがない場合に、このコマンドを使用してソースファイルの内容をメモリのデザインに読み込むことができます。非プロジェクトモードの詳細は、『Vivado Design Suite ユーザー ガイド：デザイン フローの概要』(UG892) を参照してください。

IP コアを追加し、ファイルをローカル プロジェクト ディレクトリにインポートするには、`import_ip` コマンドを使用します。

このコマンドを実行すると、読み込まれたファイルのリストが返されます。

引数

`-quiet` (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

`-verbose` (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、`set_msg_config` コマンドで定義できます。

files (必須) : 現在のプロジェクトに読み込む IP ファイルのリストを指定します。XCI および XCO ファイル フォーマットの両方がサポートされています。XCI ファイルは、IP のパラメーター情報を含む IP-XACT 形式のファイルです。XCO ファイルは、IP コアを生成する際に使用されるカスタマイズ パラメーターとプロジェクトオプションがすべて記述された CORE Generator ファイルです。

例

次の例では、指定した IP ファイルを読み込んでいます。

```
read_ip C:/test_ip/char_fifo.xci
```

関連項目

- [add_files](#)
- [import_ip](#)

read_saif

SAIF フォーマットのシミュレーション データをインポートします。

構文

```
read_saif [-strip_path arg] [-no_strip] [-out_file arg] [-quiet]
[-verbose] file
```

戻り値

なし

使用法

名前	説明
[-strip_path]	現在のデザインのインスタンス名を SAIF ファイルに現れるものと同じに指定します。
[-no_strip]	SAIF ファイルから階層の最初の 2 つのレベルを削除しません。
[-out_file]	不一致のネットを含む出力ファイルの名前を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	読み込む SAIF ファイルの名前を指定します。

カテゴリ

ファイル入力および出力、電力、シミュレーション

説明

report_power コマンドでの消費電力解析または **power_opt_design** コマンドでの消費電力最適化に使用する SAIF (Switching Activity Interchange Format) ファイルを読み込みます。**read_saif** コマンドを実行すると、SAIF ファイルからのアクティビティがデザイン ノードにアノテートされ、消費電力が正しく予測されます。

SAIF ファイルを読み込んだ後に **report_power** または **power_opt_design** コマンドを実行すると、そのファイルからのアクティビティ レートが解析および最適化に使用されます。

スイッチング アクティビティは、**read_vcd** コマンドを使用して VCD ファイルのフォーマットで読み込むこともできます。

引数

-strip_path arg (オプション) : SAIF ファイルのエレメントから指定のインスタンス パス接頭辞を削除し、エレメントが現在のデザインのインスタンスに適切にマップされるようにします。

-no_strip (オプション) : SAIF ファイルから階層の最初の 2 つのレベルを削除しません。

-out_file arg (オプション) : 不一致のネットおよびその他のメッセージがレポートされる出力ファイルの名前を指定します。このファイルは、SAIF ファイルのインポート中に作成されます。このオプションを指定しない場合、これらの情報はファイルに保存されません。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 読み込む SAIF ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次に例を示します。

```
read_saif -strip_path /design//top/F1 C:/Data/design1.saif
```

関連項目

- [power_opt_design](#)
- [read_vcd](#)
- [report_power](#)

read_schematic

回路図をインポートします。

構文

```
read_schematic [-name arg] [-quiet] [-verbose] file
```

戻り値

以前にエクスポートされたファイルの名前

使用法

名前	説明
[-name]	[Schematic] ビューのタイトルを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	入力ファイルを指定します。

カテゴリ

ファイル入力および出力

説明

Vivado Design Suite から **write_schematic** コマンドを使用して以前にエクスポートされたネイティブ回路図ファイルをインポートします。

引数

-name arg (オプション) : 回路図ファイルを読み込んだときに開く [Schematic] ビューの名前を指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

files (必須) : 読み込む回路図ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次の例では、指定した回路図ファイルを読み込んで、Vivado IDE で Sheet_1 という [Schematic] ビューを開いています。

```
read_schematic C:/Data/mySchematic.txt -name Sheet_1
```

関連項目

[write_schematic](#)

read_twx

TRACE スタティック タイミング解析ツールからタイミング結果を読み込みます。

構文

```
read_twx [-cell arg] [-pblock arg] [-quiet] [-verbose] name file
```

戻り値

なし

使用法

名前	説明
[-cell]	指定したセルに対してレポートファイルの名前を解釈します。
[-pblock]	指定した Pblock に対してレポートファイルの名前を解釈します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	結果のセット名を指定します。
<i>file</i>	TRACE インポートファイルの名前を指定します。

カテゴリ

ファイル入力および出力

説明

ザイリンクス TRACE (Timing Reporter And Circuit Evaluator) ツールで生成された TWX 形式のタイミング レポートをインポートします。TWX は最上位にインポートするか (デフォルト)、特定のセル レベルまたは特定の Pblock に相対的にインポートできます。

TWX ファイルをインポートすると、タイミング結果が GUI の [Timing Results] ビューに表示されます。

引数

-cell arg (オプション) : TWX ファイルをインポートする現在のデザインの階層セルの名前を指定します。タイミング パスは、指定したセルに適用されます。

-pblock arg (オプション) : 現在のデザインの Pblock の名前を指定します。タイミング パスは、指定したブロックに対してインポートされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : TWX ファイルのタイミング パスをインポートする際に作成する [Timing Results] ビューの名前を指定します。

注記 : *name* と *file* の両方を指定し、*name* を先に指定する必要があります。

file (必須) : インポートする TWX ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次の例では、指定した TWX ファイルをデザインの最上位に読み込んでいます。

```
read_twx C:/Data/timing_files/bft.twx
```

関連項目

[report_timing](#)

read_vcd

シミュレーション データを VCF フォーマットでインポートします。

構文

```
read_vcd [-strip_path arg] [-no_strip] [-out_file arg] [-quiet]
[-verbose] file
```

戻り値

なし

使用法

名前	説明
[-strip_path]	現在のデザインのインスタンス名を VCD ファイルに現れるものと同じに指定します。
[-no_strip]	VCD ファイルから階層の最初の 2 つのレベルを削除しません。
[-out_file]	不一致のネットを含む出力ファイルの名前を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	読み込む VCD ファイルの名前を指定します。

カテゴリ

ファイル入力および出力、電力、シミュレーション

説明

report_power コマンドでの消費電力解析または **power_opt_design** コマンドでの消費電力最適化に使用する VCD (Value Change Dump) ファイルを読み込みます。**read_vcd** コマンドを実行すると、デザイン ノードに VCD ファイルからのアクティビティがアノテートされます。

VCD ファイルを読み込んだ後に **report_power** または **power_opt_design** コマンドを実行すると、そのファイルからのアクティビティ レートが使用されます。

注記 : VCD ファイルは、**open_vcd**、**log_vcd**、および **flush_vcd** コマンドを使用して Vivado シミュレータで書き込むことができますが、**read_vcd** コマンドを使用して Vivado シミュレータに読み込むことはできません。

スイッチング アクティビティは、**read_saif** コマンドを使用して SAIF ファイルのフォーマットで読み込むこともできます。

引数

-strip_path arg (オプション) : VCD ファイルのエレメントから指定のインスタンス パス接頭辞を削除し、エレメントが現在のデザインのインスタンスに適切にマップされるようにします。

-no_strip (オプション) : VCD ファイルから階層の最初の 2 つのレベルを削除しません。

-out_file arg (オプション) : 不一致のネットおよびその他のメッセージがレポートされる出力ファイルの名前を指定します。このファイルは、VCD ファイルのインポート中に作成されます。このオプションを指定しない場合、これらの情報はファイルに保存されません。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 読み込む VCD ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次の例では、design1 いう VDC ファイルを読み込んでいます。

```
read_vcd C:/Data/design1.vcd
```

関連項目

- [flush_vcd](#)
- [log_vcd](#)
- [open_vcd](#)
- [read_saif](#)
- [report_power](#)

read_verilog

1 つまたは複数の Verilog ファイルを読み込みます。

構文

```
read_verilog [-library arg] [-sv] [-quiet] [-verbose] files...
```

戻り値

追加されたファイル オブジェクトのリスト

使用法

名前	説明
[-library]	ライブラリ名を指定します。デフォルトは work です。
[-sv]	SystemVerilog コンパイルをイネーブルにします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
files	Verilog ファイル名を指定します。

カテゴリ

ファイル入力および出力

説明

Verilog または SystemVerilog ソース ファイルを読み込みます。このコマンドは **add_files** コマンドと似ており、Verilog ファイルを読み込むと、ソース ファイルセットに追加されます。-library オプションを指定すると、ファイルはライブラリ プロパティを設定して追加されます。

Vivado ツールを非プロジェクト モードで実行しており、プロジェクト ソース ファイルを管理するプロジェクト ファイルがない場合に、このコマンドを使用してソース ファイルの内容をメモリのデザインに読み込むことができます。非プロジェクト モードの詳細は、『Vivado Design Suite ユーザー ガイド：デザイン フローの概要』(UG892) を参照してください。

SystemVerilog には Verilog が含まれるので、**read_verilog** コマンドで両方のファイル タイプを読み込むことができますが、SystemVerilog ファイルでは SystemVerilog モードのコンパイルをイネーブルにするため **-sv** オプションを指定する必要があります。このモードでは、ツールで SystemVerilog のキーワードおよび構文が認識されます。

1 つのプロジェクトに、Verilog ファイル (.v) と SystemVerilog ファイル (.sv) の両方、および VHDL ファイル (**read_vhdl** を使用) を含めることができます。ツールで合成用にこれらのファイルをコンパイルする際に、各ファイル タイプにそれぞれコンパイル ユニットが作成され、同じ タイプのファイルは一緒にコンパイルされます。

引数

-library *arg* (オプション) : Verilog ファイルが参照するライブラリを指定します。デフォルトの Verilog ライブラリは work です。

-sv (オプション) : ファイルを SystemVerilog コンパイル グループとして読み込みます。

注記 : SystemVerilog には Verilog が含まれるので、Verilog ソース ファイルにユーザー定義名として SystemVerilog の予約キーワードが使用されていなければ、-sv オプションを使用して Verilog ファイルを読み込んでも SystemVerilog コンパイル モードがイネーブルになります。ただし、SystemVerilog ファイルを -sv を使用せずに Verilog コンパイル ユニットとして追加することはできません。

files (必須) : 読み込む Verilog ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例は、指定した Verilog ファイルを読み込んで、ソース ファイルセットに追加しています。

```
read_verilog C:/Data/FPGA_Design/new_module.v
```

次の例では、SystemVerilog ファイルと Verilog ファイルに 1 つずつコンパイル ユニットを作成しています。

```
read_verilog -sv { file1.sv file2.sv file3.sv }
read_verilog { file1.v file2.v file3.v}
```

関連項目

- [add_files](#)
- [read_vhdl](#)

read_vhdl

1 つまたは複数の VHDL ファイルを読み込みます。

構文

```
read_vhdl [-library arg] [-quiet] [-verbose] files
```

戻り値

追加されたファイル オブジェクトのリスト

使用法

名前	説明
[-library]	VHDL ライブラリ名を指定します。デフォルトは work です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
files	VHDL ファイル名を指定します。

カテゴリ

ファイル入力および出力

説明

VHDL ソース ファイルを読み込みます。このコマンドは **add_files** コマンドと似ており、VHDL ファイルを読み込むと、ソース ファイルセットに追加されます。-library オプションを指定すると、ファイルはライブラリ プロパティを設定して追加されます。

Vivado ツールを非プロジェクト モードで実行しており、プロジェクト ソース ファイルを管理するプロジェクト ファイルがない場合に、このコマンドを使用してソース ファイルの内容をメモリのデザインに読み込むことができます。非プロジェクト モードの詳細は、『Vivado Design Suite ユーザー ガイド：デザイン フローの概要』(UG892) を参照してください。

引数

-library arg (オプション) : VHDL ファイルが参照するライブラリを指定します。デフォルトの VHDL ライブラリは work です。

file (必須) : 読み込む VHDL ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、指定した VHDL ファイルを読み込んで、ソース ファイルセットに追加しています。

```
read_vhdl C:/Data/FPGA_Design/new_module.vhdl
```

関連項目

[add_files](#)

read_xdc

1 つまたは複数のファイルから物理制約とタイミング制約を読み込みます。

構文

```
read_xdc [-cells args] [-ref arg] [-quiet_diff_pairs] [-quiet]
[-verbose] files
```

戻り値

ファイルのリスト

使用法

名前	説明
[-cells]	制約をインポートするセルを指定します。
[-ref]	制約をインポートする参照を指定します。
[-quiet_diff_pairs]	I/O ポートをインポートする際に、差動ペアの推論に関する警告メッセージを表示しないようにします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
files	読み込む入力ファイルを指定します。

カテゴリ

ファイル入力および出力

説明

ザイリンクス デザイン制約 (XDC) から物理制約をインポートします。XDC は最上位にインポートするか (デフォルト)、特定のセル、または特定のセルのインスタンスにインポートできます。最上位にインポートすると、指定した XDC ファイルがアクティブな制約ファイルセットに追加されます。

注記： XDC からの制約は、同じ名前の現在の制約を上書きします。このため、XDC を読み込む際には重要な制約が上書きされないよう注意が必要です。

このコマンドは **add_files** コマンドと似ており、XDC はローカルのプロジェクト ディレクトリにインポートされるのではなく、参照されて追加されます。

Vivado ツールを非プロジェクトモードで実行しており、プロジェクトソース ファイルを管理するプロジェクトファイルがない場合に、このコマンドを使用してソース ファイルの内容をメモリのデザインに読み込むことができます。非プロジェクトモードの詳細は、『Vivado Design Suite ユーザー ガイド：デザイン フローの概要』(UG892) を参照してください。

引数

-ref arg (オプション) : XDC ファイルから制約を読み込み、現在のデザインのどこにインスタンシエートされているかにかかわらず、このオプションで指定したモジュールのすべてのインスタンスに適用します。

-cells args (オプション) : XDC ファイルの制約を指定のインスタンスに適用します。制約は指定したセル インスタンスにのみ適用され、XDC ファイルはアクティブな制約ファイルセットには追加されません。

注記 : -cells オプションを指定する場合は、デザインを開いておく必要があります。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : インポートする XDC ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、まず現在の作業ディレクトリ、次にツールを起動したディレクトリで指定ファイルが検索されます。

例

次の例では、XDC ファイルを読み込み、現在のデザインに適用しています。

```
read_xdc file_1.xdc
```

次の例では、XDC ファイルを読み込み、現在のデザインに含まれる指定したモジュールのすべてのインスタンスに適用しています。

```
read_xdc -ref hex2led file_2.xdc
```

次の例では、XDC ファイルを読み込み、指定したモジュール内の指定したインスタンスのみに適用しています。

```
read_xdc -ref sixty -cells lsbcount file_3.xdc
```

次の例では、XDC ファイルを読み込み、異なるモジュールのインスタンスであっても、指定したインスタンスに適用しています。

```
read_xdc -cells one_decode sixty/msbcount file_4.xdc
```

関連項目

- [add_files](#)
- [infer_diff_pairs](#)
- [write_xdc](#)

redo

取り消されたコマンドをやり直します。

構文

```
redo [-list] [-quiet] [-verbose]
```

戻り値

やり直し可能なタスクのリスト (-list を使用した場合)

使用法

名前	説明
[-list]	やり直し可能なタスクのリストを返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

GUI 制御

説明

取り消されたコマンドをやり直します。このコマンドを繰り返し使用し、一連のコマンドをやり直すことができます。

startgroup および **endgroup** コマンドを使用してコマンド グループを作成した場合、**redo** コマンドでコマンド グループがシーケンスとしてやり直されます。

引数

-list (オプション) : やり直すことができるコマンドのリストを取得します。**undo** コマンドを使用すると、コマンドのリストを順にさかのぼってコマンドが取り消されます。**redo** コマンドを使用すると、これらのコマンドをやり直すことができます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、やり直し可能なコマンドのリストが返されます。

```
redo -list
```

関連項目

- [undo](#)
- [startgroup](#)
- [endgroup](#)

refresh_design

現在のデザインを最新情報に更新します。

構文

```
refresh_design [-part arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-part]	ターゲット パーツを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

説明

ハード ドライブのプロジェクト データから現在のデザインを読み込み直します。これにより、メモリのデザインが上書きされ、保存されていない変更は失われます。

引数

-part arg (オプション) : デザインを読み込み直したときに使用する新しいターゲット パーツを指定します。これにより、ハード ドライブのプロジェクト データで指定されている制約ファイルの パーツが上書きされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、ハード ディスクのプロジェクト データから現在のデザインを読み込み直します。これにより、メモリにある未保存の変更が上書きされます。

```
refresh_design
```

注記： デザインに対する一連の変更を取り消して、デザインを保存されている状態に戻します。

次の例では、指定した Virtex-6 パーツをターゲット デバイスとして使用して現在のデザインを更新しています。2 つ目のコマンドは、選択したパーツをアクティブなインプリメンテーション run のターゲット デバイスにするために必要です。

```
refresh_design -part xc6vcx75tff784-1
set_property part xc6vcx75tff784-1 [get_runs impl_6]
```

注記： ターゲット パーツを変更しない場合は、2 つ目のコマンドは必要ありません。

関連項目

[set_property](#)

refresh_hw_device

ハードウェア デバイスを更新します。デバイスからデバイスおよびコア情報を読み込みます。

構文

```
refresh_hw_device [-update_hw_probes arg] [-quiet] [-verbose] [hw_device]
```

戻り値

なし

使用法

名前	説明
[-update_hw_probes]	プローブ ファイルから読み込んでハードウェア プローブ情報をアップデートします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_device]	ハードウェア デバイスを指定します。デフォルトは、現在のハードウェア デバイスです。

カテゴリ

ハードウェア

refresh_hw_server

ハードウェア サーバーへの接続を更新します。

構文

```
refresh_hw_server [-quiet] [-verbose] [hw_server]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_server]	ハードウェア サーバーを指定します。

カテゴリ

ハードウェア

refresh_hw_sio

指定のハードウェア オブジェクトのステータスを更新します。任意のハードウェア オブジェクトを入力できます。少なくとも 1 つのオブジェクトが必要です。オブジェクトに存在しないプロパティを指定した場合、そのプロパティは更新されません。

構文

```
refresh_hw_sio [-regexp] [-properties args] [-quiet] [-verbose] hw_objects
```

戻り値

なし

使用法

名前	説明
<code>[-regexp]</code>	プロパティのリストに正規表現が含まれることを示します。
<code>[-properties]</code>	更新するプロパティのリストを指定します。デフォルトは、オブジェクトのすべてのプロパティです。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>hw_objects</code>	ハードウェア オブジェクトを指定します。

カテゴリ

ハードウェア

refresh_hw_target

ハードウェア ターゲットを更新します。

構文

```
refresh_hw_target [-quiet] [-verbose] [hw_target]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_target]	ハードウェア ターゲットを指定します。

カテゴリ

ハードウェア

refresh_hw_vio

ハードウェア プローブの INPUT_VALUE および ACTIVITY_VALUE プロパティをハードウェア VIO コアから読み出した値でアップデートします。

構文

```
refresh_hw_vio [-update_output_values arg] [-quiet] [-verbose] [hw_vios...]
```

戻り値

なし

使用法

名前	説明
[-update_output_values]	ハードウェア プローブの OUTPUT_VALUE プロパティを VIO コアから読み出した値でアップデートします。デフォルトは false です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_vios]	ハードウェア VIO オブジェクトのリストを指定します。

カテゴリ

ハードウェア

regenerate_bd_layout

レイアウトを再生成します。

構文

```
regenerate_bd_layout [-routing] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-routing]	ブロックの配置を保持して配線を再生成します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

IP インテグレーター

説明

現在の IP サブシステム デザインのレイアウトを開いているキャンバスに再生成します。このコマンドは、Vivado IDE でサブシステム デザインのグラフィカル エレメントをアップデートおよび再描画します。

引数

-routing (オプション) : IP インテグレーター キャンバスの配線を更新しますが、オブジェクトの配置は更新しません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、Vivado IDE の IP インテグレーター キャンバスを更新しています。

```
regenerate_bd_layout
```

関連項目

- [current_bd_design](#)
- [open_bd_design](#)
- [start_gui](#)

reimport_files

最新でないファイルをインポートし直します。

構文

```
reimport_files [-force] [-quiet] [-verbose] [files...]
```

戻り値

インポートされたファイル オブジェクトのリスト

使用法

名前	説明
[-force]	ローカル ファイルの方が新しい場合でもインポートし直します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[files]	インポートし直すファイルのリストを指定します。ファイルを指定しない場合は、プロジェクトの最新でないファイルがすべてインポートし直されます。

カテゴリ

プロジェクト

説明

プロジェクト ファイルをインポートし直します。これにより、参照先のソース ファイルからローカルのプロジェクト ファイルがアップデートされます。

引数

-force (オプション) : ローカルのプロジェクト ファイルの方が参照先のソース ファイルよりも新しい場合でも、ファイルを再インポートします。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

files (オプション) : 再インポートするファイルのリストを指定します。ファイルを指定しない場合は、プロジェクトの最新でないファイルがすべてインポートし直されます。**-force** を使用してファイルを指定しない場合は、プロジェクトのすべてのファイルがインポートし直されます。

例

次の例では、ファイルが最新でないか、ローカル ファイルが参照先ソース ファイルよりも新しいかどうかにかかわらず、すべてのプロジェクト ファイルが再インポートされます。

```
reimport_files -force
```

注記：上書きされるローカル ファイルの方が新しくても、警告メッセージは表示されません。

次の例では、元のソース ファイルがローカル プロジェクト ファイルよりも新しい場合にのみ、指定のファイルをプロジェクトにインポートし直します。

```
reimport_files C:/Data/FPGA_Design/source1.v C:/Data/FPGA_Design/source2.vhd1
```

関連項目

- [add_files](#)
- [import_files](#)

remove_bps

シミュレーションからブレークポイントを削除します。

構文

```
remove_bps [-all] [-file arg] [-line arg] [-quiet] [-verbose]
[BreakPointObjsOrIds ...]
```

戻り値

なし

使用法

名前	説明
[-all]	すべてのブレークポイントを削除します。
[-file]	ブレークポイントを削除するファイルを指定します。
[-line]	ブレークポイントを削除する行番号を指定します。デフォルトは -1 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[BreakPointObjsOrIds]	削除するブレークポイント オブジェクトまたはブレークポイント オブジェクト ID のリストを指定します。

カテゴリ

シミュレーション

remove_cell

現在のデザインからセルを削除します。

構文

```
remove_cell [-quiet] [-verbose] cells...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
cells	削除するセルの名前を指定します。

カテゴリ

[ネットリスト](#)

説明

開いている合成済みデザインまたはインプリメント済みデザインの現在のネットリストからセルを削除します。

注記：セルをライブラリ マクロ (マクロ プリミティブとも呼ばれる) から削除することはできません。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソースファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザイン チェックポイントとして保存するか、**write_*** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリスト ファイルにエクスポートできます。

注記：エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

cells (必須) : 削除するセルのリストを指定します。インスタンス名は、デザインの最上位からの階層名で指定できます。この場合、階層インスタンス名に階層区切り文字を使用する必要があります。現在の階層区切り文字を確認するには、**get_hierarchy_separator** コマンドを使用します。

例

次の例では、現在のデザインのメモリにあるネットリストから fftEngine を削除します。

```
remove_cell fftEngine
remove_cell usbEngine0/usb_out
```

関連項目

- [create_cell](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

remove_cells_from_pblock

Pblock からセルを削除します。

構文

```
remove_cells_from_pblock [-quiet] [-verbose] pblock cells...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>pblock</i>	セルを削除する Pblock を指定します。
<i>cells</i>	削除するセルを指定します。

カテゴリ

フロアプラン、XDC

説明

指定したロジック インスタンスを Pblock から削除します。Pblock にセルを追加するには、**add_cells_to_pblock** コマンドを使用します。

注記： Pblock からセルを削除しても、配置が解除されるわけではありません。LOC 制約はそのまま残ります。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

pblock (必須)：指定のインスタンスを削除する Pblock の名前を指定します。

cells (必須)：指定の Pblock から削除する 1 つまたは複数のセル オブジェクトを指定します。

例

次の例では、pb_cpuEngine Pblock から指定したセルを削除しています。

```
remove_cells_from_pblock pb_cpuEngine [get_cells cpuEngine/cpu_dwb_dat_o/*]
```

関連項目

[add_cells_to_pblock](#)

remove_conditions

シミュレーションから条件を削除します。名前は Tcl glob パターンを使用して指定できます。

構文

```
remove_conditions [-all] [-quiet] [-verbose] [ConditionObjs]
```

戻り値

なし

使用法

名前	説明
[-all]	すべての条件を削除します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[ConditionObjs]	条件オブジェクトを ID または名前で指定します。

カテゴリ

シミュレーション

remove_drc_checks

ユーザー ルール デックから DRC ルール チェック オブジェクトを削除します。

構文

```
remove_drc_checks [-of_objects args] [-regexp] [-nocase] [-filter arg]
-ruledesk arg [-quiet] [-verbose] [patterns]
```

戻り値

DRC チェック

使用法

名前	説明
[-of_objects]	指定した DRC ルール デックの DRC チェック オブジェクトを削除します。
[-regexp]	検索パターンを正規表現で指定します。
[-nocase]	パターンの大文字/小文字を区別せずに検索します (-regexp を指定した場合のみ有効)。
[-filter]	式を使用してリストをフィルター処理します。
-ruledesk	変更する DRC ルール デックを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[patterns]	DRC チェック オブジェクトを検索するパターンを指定します。デフォルトは * です。

カテゴリ

DRC、オブジェクト

説明

DRC ルール デック オブジェクトから指定のデザイン ルール チェックを削除します。

ルール デックとは、デザイン ルール チェックのグループで、I/O 配置などの FPGA デザイン フローの異なる段階で **report_drc** コマンドにより実行されます。ツールには定義済みのルール デックが含まれていますが、**create_drc_ruledesk** コマンドを使用して新しいユーザー定義のルール デックを作成できます。

ルール デックにチェックを追加するには、**add_drc_checks** コマンドを使用します。

DRC ルール チェック オブジェクトには **is_enabled** プロパティがあり、**set_property** コマンドを使用して TRUE または FALSE に設定できます。新しいルール チェックを作成すると、**is_enabled** プロパティはデフォルトで TRUE に設定されます。**is_enabled** プロパティを FALSE に設定すると、ルール デックからルールを削除せずに、**report_drc** を実行したときにルール チェックが使用されないようにすることができます。

このコマンドを実行すると、指定のルール デックから削除されたデザイン ルール チェックのリストが返されます。

引数

-of_objects arg: 指定した DRC ルール デック オブジェクトのルール チェックを追加します。このオプションを使用すると、あるルール デックのルール チェックで検索条件に一致し、フィルター条件を満たすものを、別のルール デックにコピーできます。

-regexp (オプション) : 検索パターンが正規表現で記述されていることを指定します。このオプションを使用する場合、検索パターンおよび **-filter** オプションの式は正規表現で記述する必要があります。ザイリンクスの正規表現では、Tcl コマンドは常に検索文字列の先頭にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、部分文字列を含めるよう検索を拡張できます。正規表現構文については、http://www.tcl.tk/man/tcl8.5/TclCmd/re_syntax.htm を参照してください。

注記 : Tcl ビルトイン コマンド **regexp** はアンカーされておらず、標準 Tcl コマンドと同様に機能します。詳細は、<http://www.tcl.tk/man/tcl8.5/TclCmd/regexp.htm> を参照してください。

-nocase (オプション) : パターンの大文字/小文字を区別せずに検索します。このオプションは、**-regexp** を使用した場合にのみ適用されます。

-filter args (オプション) : 結果のリストに指定した式のフィルターを適用します。このオプションを使用すると、検索パターンにより返されるオブジェクトのリストに、指定したプロパティ値に基づくフィルターを適用できます。オブジェクトに設定されているプロパティは、**report_property** または **list_property** コマンドで確認できます。

フィルター検索パターンにはクオーテーションを使用し、ネット、ピン、セル名、またはほかのプロパティに含まれる特殊文字のエスケープ処理を回避してください。文字列比較では大文字と小文字が区別され、常に検索文字列の冒頭および末尾にアンカーされています。検索文字列の先頭または末尾に「.*」を追加して、プロパティ値の部分文字列を含めるよう検索を拡張できます。

注記 : フィルターでは、オブジェクトに指定のプロパティが存在し、指定のパターンがオブジェクトのプロパティ値に一致すると、そのオブジェクトが返されます。ワイルドカード (*) を使用すると、定義値が "" のプロパティに一致します。

文字列比較では、フィルター パターンに使用できる演算子は等価 (==)、不等価 (!=)、一致 (=~)、不一致 (!~) です。数値比較演算子 <、>、<=、および >= も使用できます。複数のフィルター式を AND (&&) および OR (||) で組み合わせることもできます。次の例では、名前に文字列「RESET」が含まれない入力ポートを取得しています。

```
get_ports * -filter {DIRECTION == IN && NAME !~ "*RESET*"}
```

プール型 (**bool**) プロパティでは、フィルター式が True か False かを直接評価できます。

```
-filter {IS_PRIMITIVE && !IS_LOC_FIXED}
```

-ruledeck arg: 指定のデザイン ルール チェックを追加する ルール デックの名前を指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

patterns : 指定したパターンに一致するデザイン ルール チェックをルール デックから削除します。デフォルトのパターンはワイルドカード (*) で、すべてのルール チェックが指定のルール デックから削除されます。複数のパターンを指定して、異なる検索条件に基づいてルール チェックを削除できます。

注記：複数の検索パターンは {} で囲み、1 つのエレメントとして指定します。

例

次の例では、指定した検索パターンに一致するルール チェックを my_rules ルール デックから削除しています。

```
remove_drc_checks -filter {GROUP == AVAL} -ruledesk my_rules
```

次の例では、指定の DRC チェックが実行されないように指定しています。

```
set_property IS_ENABLED FALSE [get_drc_checks RAMW-1]
```

次の例では、指定したルール デックからすべてのルール チェックを削除しています。

```
remove_drc_checks -ruledesk my_rules
```

関連項目

- [add_drc_checks](#)
- [create_drc_check](#)
- [create_drc_ruledesk](#)
- [get_drc_checks](#)
- [get_drc_ruledecks](#)
- [list_property](#)
- [report_drc](#)
- [report_property](#)

remove_files

ファイルセットからファイルまたはディレクトリを削除します。

構文

```
remove_files [-fileset arg] [-quiet] [-verbose] files...
```

戻り値

削除されたファイルのリスト

使用法

名前	説明
[-fileset]	ファイルセット名を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
files	削除するファイルの名前を指定します。

カテゴリ

プロジェクト、シミュレーション

説明

現在のファイルセットまたは指定のファイルセットから指定のファイル オブジェクトを削除します。ファイルは現在のプロジェクトから削除されますが、ディスクからは削除されません。

ファイルはファイル名の文字列で指定するか、**get_files** コマンドでファイル オブジェクトを返すことにより指定します。文字列として指定すると、現在のファイルセットまたはしてのファイルセットでファイルが検索されます。**get_files** でファイル オブジェクトを指定すると、ファイルセットはそのオブジェクトにより定義され、**-fileset** は無視されます。

このコマンドが正しく実行された場合、何も返されません。指定のファイルが見つからない場合はエラーが返されます。

引数

-fileset arg (オプション) : 指定のファイルを検索するファイルセットの名前を指定します。デフォルトでは、**current_fileset** コマンドで定義された現在のファイルセットからファイルが削除されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

files (必須) : プロジェクトから削除するファイルの名前を指定します。

注記 : ファイルを指定しない場合、ファイルは削除されません。

例

次の例では、C:/Design/top.xdc というファイルを制約セット constrs_1 から削除しています。

```
remove_files -fileset constrs_1 C:/Design/top.xdc
```

複数のファイルを削除する場合は、次のように指定します。

```
remove_files -fileset sim_1 top_tb1.vhdl top_tb2.vhdl
```

次の例では、現在のプロジェクトのファイル オブジェクトをすべて取得し、削除しています。

```
remove_files [get_files]
```

注意 : これにより、デザインからすべてのファイルが削除されます。

関連項目

- [add_files](#)
- [current_fileset](#)
- [get_files](#)

remove_forces

add_force コマンドを使用して信号、ワイヤ、またはレジスタに適用している強制的な値の設定を解除します。

構文

```
remove_forces [-all] [-quiet] [-verbose] [ForceObj...]
```

戻り値

なし

使用法

名前	説明
[-all]	すべての force オブジェクトを削除します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[ForceObj]	force オブジェクトまたは ID を指定します。

カテゴリ

シミュレーション

説明

現在のシミュレーションの指定の force オブジェクトまたは force ID を削除します。

指定の HDL オブジェクトへの強制的な値の設定は、add_forces コマンドを使用して適用します。このコマンドは、現在のシミュレーションからこれらの値の強制を解除します。

重要： テストベンチまたはモジュールの HDL オブジェクトに **force/release** 文がある場合、それらの文より add_force コマンドが優先されます。remove_force コマンドを使用してこれらのオブジェクトを解除すると、通常の操作が再開し、Verilog **force/release** 文も有効になります。

このコマンドが正常に実行された場合は何も返されず、正常に実行できない場合はエラーが返されます。

引数

-all (オプション) : 現在のシミュレーションのすべての値の強制を解除します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記： コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

ForceObj(オプション) : 指定の force オブジェクトのみを削除します。add_force コマンドで値の強制を設定すると、force ID が返されます。

例

次の例では、add_force コマンドを使用して force オブジェクトを作成し、force ID を Tcl 変数に保存して、その force オブジェクトを削除しています。

```
set f10 [ add_force reset 1 300 ]
remove_forces $f10
```

次の例では、現在のシミュレーションからすべての force オブジェクトを削除しています。

```
remove_forces -all
```

関連項目

- [get_objects](#)
- [add_force](#)

remove_hw_sio_link

既存のハードウェア SIO リンクを削除します。

構文

```
remove_hw_sio_link [-quiet] [-verbose] hw_sio_links
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hw_sio_links</i>	ハードウェア SIO リンクを指定します。

カテゴリ

ハードウェア

remove_hw_sio_linkgroup

既存のハードウェア SIO リンク グループを削除します。

構文

```
remove_hw_sio_linkgroup [-quiet] [-verbose] hw_sio_linkgroups
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hw_sio_linkgroups</i>	ハードウェア SIO リンク グループを指定します。

カテゴリ

ハードウェア

remove_hw_sio_scan

既存のハードウェア SIO スキャンを削除します。

構文

```
remove_hw_sio_scan [-quiet] [-verbose] hw_sio_scans
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hw_sio_scans</i>	ハードウェア SIO スキャンを指定します。

カテゴリ

ハードウェア

remove_net

現在のデザインからネットを削除します。

構文

```
remove_net [-prune] [-quiet] [-verbose] nets...
```

戻り値

なし

使用法

名前	説明
[-prune]	ネットの削除を実行する際に、disconnect_net の実行により未接続となったピンおよびポートを削除します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
nets	削除するネットのリストを指定します。

カテゴリ

ネットリスト

説明

開いている合成済みデザインまたはインプリメント済みデザインのネットリストから指定のネットを削除します。

注記：ネットをライブラリ マクロ (マクロ プリミティブとも呼ばれる) から削除することはできません。

バスを削除する場合は、バスインデックスではなくバス名を指定する必要があります。これにより、バスに関連付けられている一部のビットだけではなく、バス全体を確実に削除できます。バスのビットを削除してサイズを変更するには、**resize_net_bus** コマンドを使用します。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソース ファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザイン チェックポイントとして保存するか、**write.*** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリスト ファイルにエクスポートできます。

注記：エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

引数

-prune (オプション)：ネットの削除により未接続になったピンまたはポートを削除します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

nets (必須) : 現在のデザインのネットリストから削除するネットのリストを指定します。

例

次の例では、バス ネットの 1 ビットを削除しようとした場合に表示される警告メッセージを示し、その後にバスのルート名を指定してバス全体を削除しています。

```
remove_net DataIn_pad_1_i[0]
WARNING: [Coretcl-82] No nets matched 'DataIn_pad_1_i[0]'.
remove_net DataIn_pad_1_i
```

関連項目

- [create_net](#)
- [disconnect_net](#)
- [resize_net_bus](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

remove_pin

現在のデザインからピンを削除します。

構文

```
remove_pin [-quiet] [-verbose] pins...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>pins</i>	削除するピンのリストを指定します。

カテゴリ

ネットリスト

説明

開いている合成済みデザインまたはインプリメント済みデザインの現在のネットリストからピンを削除します。

注記：ピンをライブラリ マクロまたはマクロ プリミティブから削除することはできません。

バスピンを削除する場合は、バスインデックスではなくプライマリピン名を指定する必要があります。これにより、バスに関連付けられている一部のビットだけではなく、バスピン全体を確実に削除できます。バスピンのビットを削除してサイズを変更するには、**resize_pin_bus** コマンドを使用します。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソースファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザインチェックポイントとして保存するか、**write *** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリスト ファイルにエクスポートできます。

注記：エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

pins (必須) : ネットリストから削除するピンのリストを指定します。ピンは、ピンが存在するセル インスタンスを基準に階層で指定する必要があります。

例

次の例では、現在のデザインのメモリにあるネットリストから fftEngine を削除します。

```
remove_cell fftEngine
```

関連項目

- [create_cell](#)
- [create_net](#)
- [create_pin](#)
- [create_port](#)
- [resize_pin_bus](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

remove_port

ネットリストから指定した最上位ポートのリストを削除します。

構文

```
remove_port [-quiet] [-verbose] ports...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
ports	削除するポートまたはバス ポートのリストを指定します。

カテゴリ

ピン配置

説明

指定したポートまたはバスを削除します。

バス ポートを削除する場合は、バスインデックスではなくプライマリ ポート名を指定する必要があります。これにより、バスに関連付けられている一部のビットだけではなく、バス ポート全体を確実に削除できます。バス ポートのビットを削除してサイズを変更するには、**resize_port_bus** コマンドを使用します。

remove_port コマンドでは **create_port** コマンドで追加されたポートは削除可能ですが、RTL またはネットリスト デザインで定義されているポートは削除できません。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソースファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザイン チェックポイントとして保存するか、**write *** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリスト ファイルにエクスポートできます。

注記： エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

引数

-quiet (オプション)： コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記： コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

ports : 削除するポートの名前を 1 つ以上指定します。

例

次の例では、指定したポートが削除されます。

```
remove_port PORT0
```

次の例では、バスの指定した 2 つのポートが削除されます。

```
remove_port BUS[1] BUS[2]
```

次の例では、差動ペア ポートの N 側と P 側の両方が削除されます。

```
remove_port D_BUS_P[0]
```

注記 : 差動ペアの N 側または P 側のみを削除した場合、もう一方も削除されます。

関連項目

- [create_cell](#)
- [create_net](#)
- [create_pin](#)
- [create_port](#)
- [create_interface](#)
- [place_ports](#)
- [resize_port_bus](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

rename_ref

セル参照の名前を変更します。

構文

```
rename_ref [-ref arg] [-to arg] [-prefix_all arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-ref]	名前を変更するセル参照を指定します。
[-to]	新しい名前を指定します。
[-prefix_all]	現在のデザインで該当する階層セル参照すべての名前を変更します。新しい名前は、元の名前に指定の接頭辞を付けたものになります。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

[ネットリスト](#)

説明

1 つのプリミティブでないセルの参照名を変更するか、現在の合成済みデザインまたはインプリメント済みデザインのプリミティブでないすべてのセルに参照接頭辞を付けます。

このコマンドを使用すると、現在のデザインのプリミティブでない参照名を変更し、別のデザインの参照名と競合しないようにすることができます。これにより、2 つのモジュールまたはデザインを、名前の競合を回避して、一緒に合成またはシミュレーションできます。

このコマンドを実行すると、1 つのセルの参照名が変更された場合は何も返されず、**-prefix_all** を使用した場合は名前が変更されたセルの数が返されます。コマンドが正常に実行できなかった場合は、エラーが返されます。

引数

-ref arg (オプション) : プリミティブでないセルの現在の参照名を指定します。

-to arg (オプション) : 参照名を指定した値に名前します。

-prefix_all arg (オプション) : 最上位モジュールを除く現在のデザインのプリミティブでないセルすべての参照名に、指定の接頭辞を付けます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、指定の参照名を指定の値に変更しています。

```
rename_ref -ref usbf_top -to MOD1_usbf_top
```

次の例では、現在のデザインのプリミティブでないセルすべてに指定の参照名接頭辞を付けています。

```
rename_ref -prefix_all MOD1_
```

関連項目

- [launch_xsim](#)
- [synth_design](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

reorder_files

アクティブなファイルセットでソース ファイルの順序を変更します。

構文

```
reorder_files [-fileset arg] [-before arg] [-after arg] [-front] [-back]
[-auto] [-disable_unused] [-quiet] [-verbose] files...
```

戻り値

なし

使用法

名前	説明
[-fileset]	ファイルの順序を並べ替えるファイルセットを指定します。
[-before]	このファイルの前にリストされたファイルを移動します。
[-after]	このファイルの後にリストされたファイルを移動します。
[-front]	リストされたファイルを一番前に移動します (デフォルト)。
[-back]	リストされたファイルを一番後に移動します。
[-auto]	指定したファイルセットを自動的に並べ替えます。
[-disable_unused]	TOP デザイン ユニットに関連していないファイルをすべてディスエーブルにします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
files	移動するファイルを指定します。

カテゴリ

プロジェクト

説明

指定したファイルセットのソース ファイルを並べ替えます。指定したファイルを、ファイルセット内で一番前または一番後に移動したり、ほかのファイルの前後に移動したりできます。このコマンドには自動並べ替え機能もあり、現在の最上位モジュールの要件に基づいてファイルを並べえることもできます。

引数

-fileset arg (オプション) : ファイルを並べ替えるファイルセットを指定します。デフォルトでは、sources_1 ソース ファイルセットが指定されます。

-before arg (オプション) : 指定したファイルをファイルセットのこのファイルの前に配置します。ファイルは、そのファイルセットへの完全パスで指定する必要があります。

-after *arg* (オプション) : 指定したファイルをファイルセットのこのファイルの後に配置します。ファイルは、そのファイルセットへの完全パスで指定する必要があります。

-front (オプション) : 指定したファイルをファイルセットのファイルリストの一番前に配置します。

-back (オプション) : 指定したファイルをファイルセットのファイルリストの一番後に配置します。

-auto (オプション) : プロジェクトの現在の最上位モジュールの階層要件に基づいて、ファイルを自動的に並べ替えます。このオプションは、最上位モジュールを **set_property top** コマンドで変更した後によく使用されます。

-disable_unused (オプション) : 最上位モジュールに基づく階層で現在使用されないファイルをすべてディスエーブルにします。このオプションは、最上位モジュールを **set_property top** コマンドで変更した後によく使用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

files (必須) : ファイルセットで位置を変更するファイルを 1 つまたは複数指定します。ファイルは、ファイルセットの完全パス名で指定する必要があり、指定した順に並べ替えられます。

例

次の例では、指定したファイルをソース ファイルセットの一番前に移動しています。

```
reorder_files -front {C:/Data/FPGA/file1.vhdl C:/Data/FPGA/file2.vhdl}
```

注記 : **-fileset** オプションが指定されていないので、デフォルトのソース ファイルセットが使用されます。

次の例では、デザインで最上位モジュールを設定し、この最上位モジュールの階層に基づいてファイルを自動的に並べ替え、未使用のファイルをディスエーブルにしています。

```
set_property top block1 [current_fileset]
reorder_files -auto -disable_unused
```

関連項目

- [add_files](#)
- [create_fileset](#)
- [current_fileset](#)
- [remove_files](#)

replace_bd_cell

cell1 への接続を解除して cell2 に置き換え、解除した接続を cell2 に接続します。

構文

```
replace_bd_cell [-quiet] [-verbose] [cell1] [cell2...]
```

戻り値

正しく処理された場合は 0

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[cell1]	接続を解除するセルを指定します。
[cell2]	cell1 から解除した接続を接続するセルを指定します。

カテゴリ

IP インテグレーター

説明

1 つの IP インテグレーターに現在割り当てられている接続を、現在のデザインの別の IP インテグレーター セルに移動します。このコマンドは、ソース セルからターゲット セルに接続を移動することにより、セルを別のセルにすばやく置き換えるためのものです。

セル上のピンおよびインターフェイス ピンへの接続は可能な限り保持され、接続が削除された場合はクリティカル警告が表示されます。

このコマンドが正常に実行された場合は 0 が返され、正常に実行されなかった場合はエラーが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

cell1 (必須) : 接続を移動する IP インテグレーター セルを指定します。

注記: セルは IP インテグレーター サブシステム デザインからは削除されません。

cell2 : 接続の移動先となる IP インテグレーター セルを指定します。

例

次の例では、セル lmb_v10_1 からの接続を、セル lmb_bram_cntlr_1 上の同じ名前のピンおよびインターフェイスピンに移動します。

```
replace_bd_cell [get_bd_cells /lmb_v10_1] [get_bd_cells /lmb_bram_if_cntlr_1]
CRITICAL WARNING: [BD 41-1164] The interface pin 'LMB_Sl_0' with bus definition
'xilinx.com:interface:lmb:1.0' is not found on the cell '/lmb_bram_if_cntlr_1'.
Its connection to the interface net 'Conn' has been removed.
CRITICAL WARNING: [BD 41-1166] The pin 'SYS_Rst' is not found on the cell '/lmb_bram_if_cntlr_1'.
Its connection to the net 'sys_rst_1' has been removed.
0
```

注記: ピンの不一致により既存の接続が削除された場合は、クリティカル警告が返されます。

関連項目

- [create_bd_cell](#)
- [get_bd_cells](#)

report_bps

指定したブレークポイント オブジェクトの詳細を表示します。

構文

```
report_bps [-quiet] [-verbose] [BreakPointObjs ...]
```

戻り値

ブレークポイント ID、ファイル名、行番号

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[BreakPointObjs]	レポートするブレークポイント オブジェクトのリストを指定します。

カテゴリ

シミュレーション

report_carry_chains

キャリー チェーンをレポートします。

構文

```
report_carry_chains [-file arg] [-append] [-return_string] [-max_chains arg]
[-quiet] [-verbose]
```

戻り値

レポート

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	既存のファイルの最後に追加します。
[-return_string]	レポートを文字列として返します。
[-max_chains]	レポートするチェーンの最大数を指定します。デフォルトは 1 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

現在開いているデザインで使用されているキャリー チェーンの詳細をレポートします。レポートには、すべてのキャリー チェーンの平均深さと、各キャリー チェーンの深さがレポートされます。

デフォルトでは最長のキャリー チェーンがレポートされますが、レポートするチェーン数は指定できます。

このコマンドを実行すると、キャリー チェーン レポートが返されます。

引数

-file arg (オプション) : レポートを保存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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-max_chains *arg* (オプション) : レポートするキャリー チェーン数を指定します。デフォルトでは、最長のキャリー チェーンがレポートされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、デザインのキャリー チェーンを長い方から 10 個レポートしています。

```
report_carry_chains -max_chains 10
```

report_clock_interaction

クロック間のタイミング パスとクロックが供給されないレジスタをレポートします。

構文

```
report_clock_interaction [-delay_type arg] [-setup] [-hold]
[-significant_digits arg] [-no_header] [-file arg] [-append] [-name arg]
[-return_string] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-delay_type]	パス遅延のタイプを指定します。有効な値は max、min、min_max で、デフォルトは max です。
[-setup]	最大遅延タイミング パスを考慮します (-delay_type max と同じ)。
[-hold]	最小遅延タイミング パスを考慮します (-delay_type min と同じ)。
[-significant_digits]	有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 2 です。
[-no_header]	レポート ヘッダーを生成しません。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-name]	GUI パネルに表示する結果の名前を指定します。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

[レポート](#)

説明

クロックの関連性と複数のクロック ドメインを通過する信号をレポートし、メタステーブル状態、データ損失、非干渉性などの問題を特定します。このコマンドを実行するには、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必要があります。

注記: デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

-delay_type *value* (オプション) : クロック関連性レポートを実行する際に解析に使用する遅延のタイプを指定します。有効な値は min、max、および min_max です。デフォルトは max です。

-setup (オプション) : セットアップ違反がないかどうかをチェックします。これは **-delay_type max** を指定するのと同じです。

-hold (オプション) : ホールド違反がないかどうかをチェックします。これは **-delay_type min** を指定するのと同じです。

注記 : **-setup** と **-hold** の両方を指定すると、**-delay_type min_max** を指定するのと同じになります。

-significant_digits *arg* (オプション) : 出力結果の有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 2 です。

-no_header (オプション) : レポートにヘッダーを含めません。

-file *arg* (オプション) : レポートを保存するファイルを指定します。**-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-name *arg* (オプション) : ツールの GUI モードで表示する [Clock Interaction] ビューの名前を指定します。指定した名前が開いているレポートビューで既に使用されている場合は、そのビューが閉じられ、新しいレポートが開きます。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、インターフェクト遅延モデルを設定し、デバイスのスピード グレードを選択して、**report_clock_interaction** を実行しています。

```
set_delay_model -interconnect none
set_speed_grade -3
report_clock_interaction -delay_type min_max -significant_digits 3 -name "results_1"
```

次の例では、クロック関連性レポートを GUI および指定のファイルに出力し、返された文字列を指定の変数に割り当てています。

```
set clk_int [report_clock_interaction -file clk_int.txt -name clk_int1 \
-return_string]
```

関連項目

- [create_clock](#)
- [create_generated_clock](#)
- [report_clocks](#)
- [set_delay_model](#)
- [set_speed_grade](#)

report_clock_networks

クロック ネットワークをレポートします。

構文

```
report_clock_networks [-file arg] [-append] [-name arg] [-return_string]
[-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-name]	GUI パネルに表示する結果の名前を指定します。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

デザインに含まれる各クロック ネットのネットワーク ファンアウトをレポートします。-name オプションを指定すると、クロック ネットワークが階層ツリーでグラフィカルに表示されます。

標準出力、ファイル、または文字列に返されるデフォルトのレポートでは、クロック ネット名とクロックの開始点であるインスタンス ピンが示されます。

レポートは、-file、-return_string、または -name を指定した場合を除き、デフォルトでは標準出力に返されます。

引数

-file arg (オプション) : クロック ネットワーク レポートを保存するファイルを指定します。-append オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-name arg (オプション) : GUI に出力する結果の名前を指定します。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、クロック ネットワーク名と開始点を指定したファイルにレポートしています。

```
report_clock_networks -file C:/Data/ClkNets.txt
```

report_clock_utilization

デザインのクロック ネットに関する情報をレポートします。

構文

```
report_clock_utilization [-file arg] [-append] [-write_xdc arg]
[-return_string] [-quiet] [-verbose]
```

戻り値

レポート

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	既存のファイルの最後に追加します。
[-write_xdc]	クロック制約を出力するファイルの名前を指定します。指定しない場合、クロック制約はクロック レポートに追加されます。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

デザインのクロック ネットに関する情報と、ターゲット デバイスでのクロック リソースの使用率を返します。

注記： デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

-file arg (オプション) : レポートを保存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。

注記： パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-write_xdc arg (オプション) : さまざまなクロックリソースの XDC ロケーション制約を出力するファイルの名前を指定します。パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

注記 : クロック使用率レポートの XDC 制約は、「Location of...」で開始します。**-write_xdc** オプションを指定すると、それらの行が指定のファイルに出力されます。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、デザインのクロック ネットに関する情報とターゲット デバイスでのクロックリソースの使用率が返され、指定のファイルに保存されます。

```
report_clock_utilization -file C:/Data/FPGA_Design/clock_util.txt
```

次の例では、クロック ネットとクロックリソースの使用率を標準出力に表示し、XDC ロケーション制約は指定のファイルに記述しています。

```
report_clock_utilization -write_xdc clock_util_xdc.txt
```

注記 : XDC ファイル名の一部としてパスが指定されていないので、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

関連項目

- [create_clock](#)
- [create_generated_clock](#)

report_clocks

クロックをレポートします。

構文

```
report_clocks [-file arg] [-append] [-return_string] [-quiet] [-verbose]
[clocks]
```

戻り値

なし

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[clocks]	クロックのリストを指定します。デフォルトは * です。

カテゴリ

レポート

説明

デザインに含まれるすべてのクロックを含む表を返します。返されるクロックには、現在の合成済みデザインまたはインプリメント済みデザインに含まれる伝搬されたクロック、生成されたクロック、自動生成されたクロック、仮想クロック、および反転クロックが含まれます。各クロック ネットに関するより詳細な情報は、**report_clock_utilization** コマンドを使用して取得できます。

注記： デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

-file arg (オプション)： レポートを保存するファイルを指定します。 **-append** オプションが指定されていなければ、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。

注記： パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

clocks (オプション) : 指定したパターンに一致するクロックを検索します。デフォルトのパターンはワイルドカード (*) で、デザインのすべてのクロックが返されます。複数のパターンを指定して、異なる検索条件に基づいてクロックを検索できます。

例

次の例では、現在のデザインに含まれるクロックの名前、周期、波形、およびソースが返されます。

```
report_clocks -file C:/Data/FPGA_Design/clock_out.txt
```

次の例では、デザインに含まれるクロックで、名前に「Clock」が含まれるもののが返されます。

```
report_clocks *Clock*
```

関連項目

- [create_clock](#)
- [create_generated_clock](#)
- [report_clock_utilization](#)

report_compile_order

ファイルを解析し、階層を構築して、コンパイル順をレポートします。

構文

```
report_compile_order [-fileset arg] [-missing_instances] [-constraints]
[-file arg] [-append] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-fileset]	コンパイル順をレポートするファイルセットを指定します。
[-missing_instances]	デザイン階層で不足しているインスタンスをレポートします。
[-constraints]	制約のコンパイル順をレポートします。
[-file]	結果を保存するファイルの名前を指定します。
[-append]	結果を既存のファイルに追加します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

説明

さまざまなアクティブ ファイルセット (制約、デザインソース、およびシミュレーションソース) のファイルのコンパイル順をレポートします。

合成、インプリメンテーション、シミュレーションのファイル処理順が返されます。-fileset を使用してファイルセットを使用したり、-constraints オプションを使用してアクティブ制約セットでの制約の処理順をレポートできます。

get_files -compile_order コマンドを使用すると、ファイル オブジェクトを単にレポートするだけでなく、コンパイル順に並べられたファイル オブジェクトのリストを取得できます。

デフォルトではレポートは Tcl コンソールまたは標準出力に表示されますが、ファイルに記述することもできます。

引数

-fileset arg (オプション) : レポートするファイルセットを指定します。

-missing_instances (オプション) : 現在のファイルセットまたは指定のファイルセットで、ソース ファイルのないセルのリストを返します。

-constraints (オプション) : 現在のデザインの制約ファイルのコンパイル順を、デザインの IP の制約も含めてレポートします。

-file arg (オプション) : レポートを保存するファイルを指定します。-append オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : -append コマンドは、-file オプションを使用している場合にのみ使用可能です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、set_msg_config コマンドで定義できます。

例

次の例では、現在のデザインのアクティブ ファイルセットのコンパイル順をレポートしています。

```
report_compile_order
```

次の例では、現在のデザインでソース ファイルのないセルのリストを返し、指定のファイルに保存しています。

```
report_compile_order -missing_instances -file C:/Data/report1.txt -append
```

次の例では、アクティブ制約セットのファイルのコンパイル順をリストしています。

```
report_compile_order -constraints
```

関連項目

- [current_fileset](#)
- [get_files](#)
- [update_compile_order](#)

report_conditions

指定した条件オブジェクトの詳細を表示します。

構文

```
report_conditions [-quiet] [-verbose] [ConditionObjs ...]
```

戻り値

各条件オブジェクトの名前、ID、条件式、およびコマンド

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[ConditionObjs]	条件オブジェクトを ID または名前で指定します。

カテゴリ

シミュレーション

report_config_timing

タイミング解析に影響する設定をレポートします。

構文

```
report_config_timing [-file arg] [-append] [-name arg] [-return_string] [-all]
[-no_header] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-file]	結果をファイルに出力します。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-name]	GUI パネルに表示する結果の名前を指定します。
[-return_string]	レポートを文字列として返します。
[-all]	すべてのコンフィギュレーション設定をレポートします。デフォルトでは、一般的に重要な設定のみがレポートされます。
[-no_header]	レポート ヘッダーを生成しません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート、タイミング

説明

現在のデザインのタイミング制約のコンフィギュレーションをレポートします。

デフォルトでは、主要なタイミング制約のみを含む簡略されたレポートが示されます。タイミングに関するすべてのコンフィギュレーションを返すには、**-all** オプションを使用します。

引数

-file arg (オプション) : タイミング制約コンフィギュレーション レポートを保存するファイルを指定します。 **-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-name arg (オプション) : GUI に出力する結果の名前を指定します。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-all (オプション) : デザインの属性および制約に関連するすべてのタイミングのステートをレポートします。デフォルトでは、主要なタイミング属性に関連するもののみがレポートされます。

-no_header (オプション) : レポートにヘッダーを含めません。デフォルトでは、レポートに次の情報を含むヘッダーが含まれます。

- レポート タイプ : timer_configuration
- デザイン : デザインの最上位モジュール
- パーツ : ターゲット パーツのデバイス、パッケージ、おおびスピード グレード
- バージョン : レポートの作成に使用されたツールのバージョン
- 日付 : レポートの日付
- コマンド : レポートの作成に使用されたコマンド オプション

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のタイミング コンフィギュレーションをレポートし、情報を文字列として返し、その文字列を指定の Tcl 変数に設定しています。

```
set timeConfig [report_config_timing -all -no_header -return_string]
puts $timeConfig
```

関連項目

[delete_timing_results](#)

report_control_sets

デザイン特有の制御セットについてレポートします。

構文

```
report_control_sets [-file arg] [-append] [-sort_by args] [-cells args]
[-return_string] [-quiet] [-verbose]
```

戻り値

レポート

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	既存のファイルの最後に追加します。
[-sort_by]	並べ替え条件を指定します。-verbose オプションを使用している場合にのみ使用できます。有効な値は clk、clkEn、set です。 例：report_control_sets -verbose -sort_by {clk clkEn}
[-cells]	制御セットをレポートするセル/BEL インスタンスを指定します。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

現在のデザインの制御セットをレポートします。

制御セットとは、スライスレジスタおよび LUT の制御信号 (クロック、CE、SR) をまとめたものです。レジスタを同じデバイスリソースにパックするには、同じ制御セットに含める必要があります。制御信号を持たないレジスタを制御信号を持つレジスタと共にデバイスのリソースにパックすることはできません。制御セット数が多いと、デバイスにフィットさせるのが困難になり、配線の密集やタイミング問題を引き起こす可能性があります。

デフォルトでは、制御セット数のみを示すレポートが返されます。-verbose オプションを使用すると、デザイン全体または指定のセルの制御セットすべての詳細が返されます。

引数

-file *arg* (オプション) : レポートを保存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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-sort_by *args* (オプション) : **-verbose** オプションで生成された詳細レポートを指定の条件で並べ替えます。有効な並べ替え条件は、**clk**、**clkEn**、および **set** です。

注記 : **-sort_by** オプションは、**-verbose** オプションと共に使用します。

-cells *args* (オプション) : 指定したセル オブジェクトの制御セットをレポートします。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインの制御セットを、**clk** および **clkEn** 信号で並べ替えてレポートします。

```
report_control_sets -verbose -sort_by {clk clkEn}
```

次の例では、指定したセルの制御セットを **clk** および **set** で並べ替えてレポートします。

```
report_control_sets -verbose -sort_by {clk set} -cells [get_cells usb*]
```

report_datasheet

データシートをレポートします。

構文

```
report_datasheet [-significant_digits arg] [-file arg] [-append]
[-return_string] [-sort_by arg] [-name arg] [-show_all_corners] [-group args]
[-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-significant_digits]	有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 3 です。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-return_string]	レポートを文字列として返します。
[-sort_by]	並べ替え順を指定します。有効な値は clock、port で、デフォルトは clock です。
[-name]	GUI パネルに表示する結果の名前を指定します。
[-show_all_corners]	すべてのコーナーを表示します。
[-group]	スキーを算出する出力ポートのグループを定義します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

現在のデザインのデータシートレポートを生成します。データシートレポートには、.twr ファイルにレポートされるのと同様の、デザインの I/O パッドでのタイミング特性が含まれます。

たとえば、クロックに対する入力 I/O のセットアップおよびホールド タイム、クロックから出力パッドへの最大/最小遅延、入力/出力バスのスキーなどです。

引数

-significant_digits arg (オプション) : 表示する桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 3 です。

-file arg (オプション) : レポートを保存するファイルを指定します。-append オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : -append コマンドは、-file オプションを使用している場合にのみ使用可能です。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを -file オプションと共に使用することはできません。

-sort_by [port | clock] (オプション) : 並べ替え順を指定します。有効な値は clock または port です。デフォルトは clock で、レポートがクロック順に並べられます。

-show_all_corners (オプション) : すべてのプロセスコーナーをレポートします。

-group [get_ports {xxx1 xxx2 ... xxxN}] (オプション) : 解析するポートのグループを定義します。ポートは、get_ports コマンドで返されるポートオブジェクトのリストとして指定する必要があります。リストの最初のポートが、スキー算出の基準として使用されます。ほとんどの場合、これはソース同期出力インターフェイスのクロックポートです。それぞれ基準クロックポートを持つグループを複数指定できます。-group を指定しない場合は、ソースクロックに基づいて出力ポートのグループが自動的に検出され、そのクロックに基づくスキーがレポートされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、set_msg_config コマンドで定義できます。

例

次の例では、ポート順に並べたすべてのプロセスコーナーのデータシートが返されます。

```
report_datasheet -sort_by port -show_all_corners
```

次の例では、2 つのポートグループに対してスキー算出のデータシートをレポートしています。各グループの最初のポートが、そのグループのスキー算出の基準として使用されます。この例では、CLK0OUT が DATA0 ~ 4 のクロック、CLK1OUT が DATA4 ~ 7 のクロックです。

```
report_datasheet -file ds.txt -group [get_ports {CLK0OUT DATA0 DATA1 DATA2 DATA3}] \
-group [get_ports {CLK1OUT DATA4 DATA5 DATA6 DATA7}]
```

関連項目

[get_ports](#)

report_debug_core

デバッグ コアの詳細をレポートします。

構文

```
report_debug_core [-file arg] [-append] [-return_string] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

現在のプロジェクトのさまざまな Vivado デバイス ツールのデバッグ コアおよびこれらのコアのパラメーターをレポートします。デバッグ コアをプロジェクトに追加するには、**create_debug_core** コマンドを使用します。

注記： デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

-file arg (オプション) : レポートを保存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。

注記： パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記： **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-return_string (オプション) : レポートを文字列として返します。このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、デバッグ コアのレポートが指定したディレクトリの指定したファイルに書き込まれます。

```
report_debug_core -file C:/Data/FPGA_Design/project_1_cores.txt
```

関連項目

[create_debug_core](#)

report_default_switching_activity

指定したデフォルト タイプのスイッチング アクティビティを取得します。

構文

```
report_default_switching_activity [-static_probability] [-toggle_rate]
-type args [-file arg] [-return_string] [-append] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-static_probability]	スタティック確率をレポートします。
[-toggle_rate]	トグル レートをレポートします。
-type	ベクターなしの伝搬エンジンで指定したタイプのデフォルトのシード値をレポートします。有効なデフォルト タイプの値は、input、input_set、input_reset、input_enable、register、dsp、bram_read_enable、bram_write_enable、output_enable、clock、all です。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-return_string]	デフォルトのスイッチング アクティビティを文字列として返します。
[-append]	デフォルトのスイッチング アクティビティをファイルの最後に追加します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

指定のエレメント タイプに対して現在設定されているデフォルトのスイッチング アクティビティを表示します。

レポートされる値は、`set_default_switching_activity` コマンドで定義します。

注記： デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

`-static_probability` (オプション) : レポートにスタティック確率を含めます。トグル レートは含めません。

-toggle_rate (オプション) : レポートにトグル レートを含めます。スタティック確率は含めません。

注記 : **-toggle_rate** または **-static_probability** を指定して結果を制限しない限り、トグル レートとスタティック確率の両方がレポートされます。

-type types (必須) : レポートするコンポーネントのタイプを指定します。有効な値は、input、input_set、input_reset、input_enable、register、dsp、bram_read_enable、bram_write_enable、output_enable、clock、all です。

-file arg (オプション) : レポートを保存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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-return_string (オプション) : レポートをデータ セットとしてではなく、文字列として返します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、DSP のデフォルトのスイッチング アクティビティ属性をレポートしています。

```
report_default_switching_activity -type dsp
Default Dsp Probability = 0.50
Default Dsp Toggle Rate (%) = 12.50
```

次の例では、すべてのタイプに対してデフォルトのスイッチング アクティビティ属性をレポートし、Tcl 変数 swa1 に保存しています。

```
set swa1 [report_default_switching_activity -type all -return_string]
```

注記 : **-return_string** オプションを使用しなくてもコマンドは正しく実行されますが、\$swa 変数にレポートされた情報は割り当てられません。

関連項目

- [power_opt_design](#)
- [report_power](#)
- [report_switching_activity](#)
- [reset_default_switching_activity](#)
- [reset_switching_activity](#)
- [set_default_switching_activity](#)
- [set_switching_activity](#)

report_disable_timing

ディスエーブルにしたタイミング アークをレポートします。

構文

```
report_disable_timing [-file arg] [-append] [-return_string] [-quiet]
[-verbose]
```

戻り値

なし

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート、タイミング

説明

現在の合成済みデザインまたはインプリメント済みデザインのタイミング解析から除外されるタイミング パスのレポートを表示します。

レポートは、タイミング パスに関連付けられているオブジェクトを示す [Cell or Port]、タイミング パスを示す [From] と [To]、条件、およびタイミングから除外されている理由を示す列で構成されます。除外されている理由には次のものがあります。

- constraint : **set_disable_timing** 制約が指定されている
- constant : ロジック定数
- loop : ロジック ループを断絶する
- bidirect instance path : フィードバック パスが双方向インスタンスを通過する
- bidirect net path : フィードバック パスが双方向ピン持つネット上にある

注記： デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

-file *arg* (オプション) : レポートを保存するファイルを指定します。-append オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : -append コマンドは、-file オプションを使用している場合にのみ使用可能です。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを -file オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、タイミング解析から除外されるすべてのタイミング パスがレポートされます。

```
report_disable_timing
```

次の例では、タイミング解析から除外されるタイミング パスのレポートを文字列として出力し、変数に保存してから表示しています。

```
set bad_time [report_disable_timing -return_string]
puts $bad_time
```

関連項目

- [report_timing](#)
- [set_disable_timing](#)

report_drc

DRC を実行します。

構文

```
report_drc [-name arg] [-rules args] [-ruledesk arg] [-file arg]
[-format arg] [-append] [-return_string] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-name]	GUI パネルに表示する結果の名前を指定します。
[-rules]	DRC ルールを指定します。指定可能なルールは、get_drc_checks コマンドを使用すると取得できます。
[-ruledesk]	DRC ルール チェックのコンテナーであるルール デックを指定します。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-format]	レポートのフォーマットを指定します。有効な値は text および xml です。-file を使用した場合にのみ適用されます。xml を指定した場合、-append は使用できません。デフォルトは text です。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

DRC、レポート

説明

デザインに対して指定のデザインルール チェックまたはルール デックを実行し、検出されたエラーまたは違反をレポートします。

ツールには、**report_drc** コマンドで使用可能な定義済みのデザインルール チェックが多数含まれています。**get_drc_checks** コマンドを使用すると、定義済みのデザインルール チェックを取得できます。**create_drc_check** コマンドを使用すると、カスタム デザインルール チェックを作成できます。

ルール デックはデザイン ルール チェックのグループで、I/O 配置や配置などの FPGA デザイン フローの異なる段階で `report_drc` コマンドにより実行されます。ツールには定義済みのルール デックが含まれていますが、`create_drc_ruledeck` コマンドを使用して新しいユーザー定義のルール デックを作成できます。`get_drc_ruledecks` コマンドを使用すると、`report_drc` コマンドで使用可能な定義済みのルール デックを取得できます。

-rules または **-ruledeck** オプションを指定しない場合、デフォルトのルール デックが実行されます。ユーザー定義 DRC を作成すると、デフォルトのルール デックに自動的に追加されます。

DRC ツールは、ルール チェック オブジェクトの **IS_ENABLED** プロパティを使用してイネーブルまたはディスエーブルにできます。ルールの **IS_ENABLED** が `false` の場合、**-rules** または **-ruledeck** で間接的に指定されていても、`report_drc` コマンドでは実行されません。

注記： `reset_drc_check` コマンドを使用すると、DRC ルールのプロパティをデフォルト設定に戻すことができます。

このコマンドを実行するには、デザイン ルール チェックを実行するデザインを開いている必要があります。このコマンドを実行すると、デザイン ルール チェックで検出された違反の結果がレポートされます。違反は、`get_drc_vios` コマンドを使用してリストできます。

`report_drc` コマンドの現在の結果をリセットして検出された違反をクリアするには、`reset_drc` コマンドを使用します。

引数

-name *arg* (オプション) : GUI モードで実行した場合の結果の名前を指定します。

-ruledeck *arg* (オプション) : DRC ルール デックの名前を指定します。ルール デックは、DRC ルール チェックのリストの名前です。ツールであらかじめ定義されているルール デックまたはユーザー定義ルール デックを指定できます。指定のルール デックに追加されているデザイン ルール チェックが、デザインに対して実行されます。カスタム ルール デックを定義するには、`create_drc_ruledeck` コマンドを使用します。定義済みのルール デックを取得するには、`get_drc_ruledecks` コマンドを使用します。

-rules *args* (オプション) : DRC レポートを実行するルールをリストします。指定したルール チェックが現在のデザインに対してチェックされます。ルールは、グループ名またはフル キーでリストされます。**-rules** オプションを使用すると、指定したデザイン ルール チェックで一時的なユーザー定義ルール デックが作成され、この一時的なルール デックが実行されます。

注記： **-ruledeck** と **-rules** を同時に使用することはできません。

-file *arg* (オプション) : DRC レポートを保存するファイルを指定します。**-append** オプションが指定されていなければ、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。

注記： パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記： **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記： このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、使用可能なルール デックをリストしています。結果には、ツールであらかじめ定義されているルール デックと、ユーザー定義ルール デックがすべて含まれます。

```
get_drc_ruledecks
```

次の例では、指定のルール デックに含まれる DRC ルールのリストが返されます。

```
get_drc_checks -of_objects [get_drc_ruledecks placer_checks]
```

次の例では、現在のデザインに対して指定した DRC ルール デックおよびチェックを実行し、その出力を指定したファイルに書き込んでいます。

```
report_drc -ruledeck placer_checks -file C:/Data/DRC_Rpt1.txt
report_drc -rules {IOCNT-1 IOPCPR-1 IOPCMGT-1 IOCTMGT-1 IODIR-1} \
    -file C:/Data/DRC_Rpt1.txt -append
```

注記 : 2 番目の **report_drc** コマンドには **-append** が指定されているので、コマンドの結果は指定のファイルの最後に追加されます。

関連項目

- [create_drc_check](#)
- [create_drc_ruledeck](#)
- [create_drcViolation](#)
- [get_drc_checks](#)
- [get_drc_ruledecks](#)
- [get_drc_vios](#)
- [reset_drc](#)
- [reset_drc_check](#)

report_drivers

HDL ワイヤまたは信号オブジェクトのドライバーと現在の駆動値を表示します。

構文

```
report_drivers [-quiet] [-verbose] hdl_object
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hdl_object</i>	レポートする HDL オブジェクトを指定します。

カテゴリ

シミュレーション

report_environment

システム情報をレポートします。

構文

```
report_environment [-file arg] [-format arg] [-append] [-return_string]
[-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-file]	システム情報を指定のファイルに書き込みます。
[-format]	レポートのフォーマットを指定します。有効な値は text および xml です。-file を使用した場合にのみ適用されます。xml を指定した場合、-append は使用できません。デフォルトは text です。
[-append]	レポートを既存のファイルの最後に追加します。
[-return_string]	レポートの内容を文字列値として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

ツールを実行しているシステム環境の詳細をレポートします。オペレーティング システムのバージョン、CPU、メモリ、使用可能なディスク容量、さまざまな環境変数の設定が含まれます。

デフォルトでは出力は標準出力に表示されますが、ファイルに書き込むこともできます。

引数

-file arg (オプション) : レポートを保存するファイルを指定します。 **-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在の環境を指定したファイルに保存しています。

```
report_environment -file C:/Data/toolEnv.txt
```

report_exceptions

タイミング例外をレポートします。

構文

```
report_exceptions [-from args] [-rise_from args] [-fall_from args] [-to args]
[-rise_to args] [-fall_to args] [-through args] [-rise_through args]
[-fall_through args] [-no_header] [-file arg] [-append] [-return_string]
[-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-from]	タイミング パスの開始点 (ピン、ポート、セル、またはクロック) を指定します。
[-rise_from]	ピン、ポート、セル、またはクロックの立ち上がりエッジを開始点として指定します。
[-fall_from]	ピン、ポート、セル、またはクロックの立ち下がりエッジを開始点として指定します。
[-to]	タイミング パスの終点 (ピン、ポート、セル、またはクロック) を指定します。
[-rise_to]	ピン、ポート、セル、またはクロックの立ち上がりエッジを終点として指定します。
[-fall_to]	ピン、ポート、セル、またはクロックの立ち下がりエッジを終点として指定します。
[-through]	タイミング パスの通過点 (ピン、ポート、セル、またはネット) を指定します。
[-rise_through]	ピン、ポート、セル、またはネットの立ち上がりエッジを通過点として指定します。
[-fall_through]	ピン、ポート、セル、またはネットの立ち下がりエッジを通過点として指定します。
[-no_header]	レポート ヘッダーを生成しません。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート、タイミング

説明

現在のデザインのタイミング制約で定義されているセットアップおよびホールド チェックに適用されるタイミング例外をレポートするか、指定したタイミング パスの例外をレポートします。タイミング例外は、`set_false_path` や `set_multicycle_path` などのデザインのタイミング パスのデフォルトを変更するタイミング制約で定義できます。

デフォルトでは例外レポートは標準出力に出力されますが、その後の処理用にファイルまたは Tcl 文字列変数に出力することもできます。

引数

-from *args* (オプション) : 例外をレポートするタイミング パスの開始点のリストを指定します。

-rise_from *args* (オプション) : 立ち上がりエッジの例外をレポートするタイミング パスの開始点のリストを指定します。

-fall_from *args* (オプション) : 立ち下がりエッジの例外をレポートするタイミング パスの開始点のリストを指定します。

-to *args* (オプション) : 例外をレポートするタイミング パスの終点のリストを指定します。

-rise_to *args* (オプション) : 立ち上がりエッジの例外をレポートするタイミング パスの終点のリストを指定します。

-fall_to *args* (オプション) : 立ち下がりエッジの例外をレポートするタイミング パスの終点のリストを指定します。

-through *args* (オプション) : タイミング パスが通過するピン、セル、またはネットのリストを指定します。

-rise_through *args* (オプション) : 立ち上がりタイミング パスが通過するピン、セル、またはネットのリストを指定します。

-fall_through *args* (オプション) : 立ち下がりタイミング パスが通過するピン、セル、またはネットのリストを指定します。

-no_header (オプション) : レポートにヘッダーを含めません。

-file *arg* (オプション) : レポートを保存するファイルを指定します。デフォルトでは、タイミング例外は標準出力または Tcl コンソールにレポートされます。**-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインのタイミング例外をすべてレポートしています。

```
report_exceptions
```

関連項目

- [create_clock](#)
- [create_generated_clock](#)
- [report_timing](#)
- [report_timing_summary](#)
- [set_false_path](#)
- [set_input_delay](#)
- [set_max_delay](#)
- [set_min_delay](#)
- [set_multicycle_path](#)
- [set_output_delay](#)

report_high_fanout_nets

ファンアウトの大きいネットをレポートします。

構文

```
report_high_fanout_nets [-file arg] [-append] [-ascending] [-histogram]
[-load_types] [-clock_regions] [-slr] [-max_nets arg] [-min_fanout arg]
[-max_fanout arg] [-cells args] [-return_string] [-quiet] [-verbose]
```

戻り値

レポート

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	既存のファイルの最後に追加します。
[-ascending]	ネットを昇順にレポートします。
[-histogram]	ファンアウトの大きいネットのヒストグラムをレポートします。
[-load_types]	ロードの詳細をレポートします。
[-clock_regions]	クロック領域でのロードの分配をレポートします。
[-slr]	SLR でのロードの分配をレポートします。
[-max_nets]	レポートするネットの最大数を指定します。デフォルトは 10 です。
[-min_fanout]	ファンアウトの数が指定した整数値以上のネットをレポートします。デフォルトは 1 です。
[-max_fanout]	ファンアウトの数が指定した整数値以下のネットをレポートします。デフォルトは INT_MAX です。
[-cells]	ネットをレポートするセル/BEL インスタンスを指定します。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

デザインのネットのファンアウトを、ファンアウトの大きいネットから順にレポートします。オプションを使用してレポートの内容を制御できます。

このコマンドは、インプリメント済みデザインまたは合成済みネットリストに対して実行できますが、インプリメント済みデザインで実行したほうがより完全な結果が得られます。

このコマンドを実行すると、デザインのファンアウトレポートが返されます。

引数

-file *arg* (オプション) : レポートを保存するファイルを指定します。**-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-ascending (オプション) : ネットを昇順にレポートします。

-histogram (オプション) : レポートをヒストグラム形式で表示し、ファンアウト数、ファンアウトの大きいネットのデザイン全体での数と割合を示します。

-load_types (オプション) : ネット上のさまざまなロード タイプをレポートします。ロード タイプ (データ、クロック、セット、リセットなど) またはロードが配置されているデバイスリソース (スライス、I/O など) を基準に並べ替えることができます。report_high_fanout_nets を配置されていない合成済みデザインに対して実行した場合は、ロード タイプのみがレポートされます。

-clock_regions (オプション) : クロック領域でのロードの分配をレポートします。配置後にネットのさまざまなロード タイプが配置されるクロック領域がレポートされます。

-slr (オプション) : SLR でのロードの分配をレポートします。配置後にネットのさまざまなロード タイプが配置される SLR がレポートされます。

-max_nets *arg* (オプション) : レポートするネット数を指定します。デフォルトは 10 です。

-min_fanout *arg* (オプション) : ファンアウト数が指定した整数値以上のネットをレポートします。これにより、指定したファンアウト ロードのネットをレポートできます。デフォルトは 1 です。

-max_fanout *arg* (オプション) : ファンアウト数が指定した整数値以下のネットをレポートします。これにより、指定したファンアウト ロードのネットをレポートできます。デフォルトでは最大値は指定されません。

-cells *args* (オプション) : ネットをレポートするセル/BEL インスタンスを指定します。デザインの指定したセルまたは BEL に接続されたネットがレポートされます。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、ファンアウトが 50 以上のネットを上位から 100 個レポートしています。

```
report_high_fanout_nets -min_fanout 50 -max_nets 100
```

report_incremental_reuse

指定のデザイン チェックポイントから現在のデザインにインクリメンタル再利用可能な量をレポートします。

構文

```
report_incremental_reuse [-file arg] [-append] [-cells args] [-netlist_only]
[-hierarchical] [-hierarchical_depth arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	既存のファイルの最後に追加します。
[-cells]	指定したセルのリストのインクリメンタル再利用をレポートします。
[-netlist_only]	ネットリスト ベースの類似性をレポートします。
[-hierarchical]	テキスト形式の階層インクリメンタル再利用レポートを生成します。
[-hierarchical_depth]	テキスト形式の階層インクリメンタル再利用レポートのレベルを指定します。デフォルトは 0 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

インクリメンタル コンパイル フローで使用し、現在のデザインと `read_checkpoint -incremental` コマンドを使用して読み込んだインクリメンタル チェックポイントで一致している量をレポートします。

読み込まれたインクリメンタル チェックポイントを現在のデザインに対して解析し、インクリメンタル配置配線を実行するのに十分な類似性があるかどうかを確認できます。現在のデザインとチェックポイントの類似性が低い場合は、チェックポイントをインクリメンタル配置配線に使用することは推奨されません。インクリメンタル配置および配線の詳細は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

現ザのデザインと読み込んでいるインクリメンタル チェックポイントの類似性が低い場合は、`open_run` または `read_checkpoint` を使用して元のデザインを復元する必要があります。または、`read_checkpoint -incremental` コマンドを再度実行して現在のデザインに新しいインクリメンタル チェックポイントを読み込むこともできます。

引数

-file *arg* (オプション) : レポートを保存するファイルを指定します。-append オプションが指定されていなければ、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。デフォルトでは、レポートは Tcl コンソールに表示されます。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : -append コマンドは、-file オプションを使用している場合にのみ使用可能です。

-cells *args* (オプション) : DCP ファイルから使用するセルを指定します。

-netlist_only (オプション) : チェックポイント間の類似性を、ネットリストのみを比較してレポートします。

-hierarchical (オプション) : テキスト形式の階層インクリメンタル再利用レポートを生成します。

-hierarchical_depth *arg* (オプション) : テキスト形式の階層インクリメンタル再利用レポートのレベルを指定します。デフォルト値は 0 です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインにインクリメンタル チェックポイントを読み込み、読み込まれたインクリメンタル チェックポイントと現在のデザインの類似性をレポートしています。

```
read_checkpoint -incremental C:/Data/reuse_checkpoint1.dcp
report_incremental_reuse
```

関連項目

- [open_run](#)
- [place_design](#)
- [read_checkpoint](#)
- [route_design](#)

report_io

デバイスのすべての I/O サイトに関する情報を表示します。

構文

```
report_io [-file arg] [-append] [-return_string] [-quiet] [-verbose]
```

戻り値

レポート

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	既存のファイルの最後に追加します。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

現在のデザインの I/O バンクの詳細をレポートします。ターゲット パーツ、パッケージ、スピード グレード、デバイス上の各ピンに関する情報などが含まれます。

引数

-file arg (オプション) : レポートを保存するファイルを指定します。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインの I/O ブロックをレポートしています。

`report_io`

関連項目

[report_route_status](#)

report_ip_status

プロジェクトの IP インスタンスのステータスをレポートします。

構文

```
report_ip_status [-name arg] [-file arg] [-append] [-return_string] [-quiet]
[-verbose]
```

戻り値

コマンドが正しく実行された場合は True

使用法

名前	説明
[-name]	GUI パネルに表示する結果の名前を指定します。
[-file]	結果を出力するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	既存のファイルの最後に追加します。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

IP フロー

report_objects

指定した HDL オブジェクト (変数、信号、ワイヤ、またはレジスタ) の詳細を表示します。

構文

```
report_objects [-quiet] [-verbose] [hdl_objects ...]
```

戻り値

HDL オブジェクトの名前、タイプ、データ型

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hdl_objects]	レポートする HDL オブジェクトを指定します。デフォルトは report_objects [get_objects *] です。

カテゴリ

シミュレーション

report_operating_conditions

消費電力予測の動作条件値を取得します。

構文

```
report_operating_conditions [-voltage] [-grade] [-process] [-junction_temp]
[-ambient_temp] [-thetaja] [-thetas] [-airflow] [-heatsink] [-thetajb]
[-board] [-board_temp] [-board_layers] [-all] [-file arg] [-return_string]
[-append] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-voltage]	電圧値を取得します。サポートされる電圧は、ファミリによって異なります。
[-grade]	電圧グレードを指定します。サポートされる値は、ファミリによって異なります。
[-process]	プロセスを取得します。
[-junction_temp]	ジャンクション温度 (C) を取得します。
[-ambient_temp]	周囲温度 (C) を取得します。
[-thetaja]	ThetaJA (C/W) を取得します。
[-thetas]	ThetaSA を取得します。
[-airflow]	エアフロー (LFM) を取得します。
[-heatsink]	ヒートシンクのサイズを取得します。
[-thetajb]	ThetaJB を取得します。
[-board]	ボード タイプを取得します。
[-board_temp]	ボード温度 (C) を取得します。
[-board_layers]	ボードの層数を取得します。
[-all]	このヘルプ メッセージにリストされるすべての動作条件を取得します。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-return_string]	動作条件を文字列として返します。
[-append]	動作条件をファイルの最後に追加します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート、SDC

説明

デザインのパフォーマンスを解析する際に使用される実際の動作条件を表示します。

デバイスの環境動作条件は、**report_power** コマンドによる消費電力解析では使用されますが、タイミング解析では使用されません。動作条件の値は、**set_operating_conditions** コマンドで定義できます。

注記：デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

-voltage (オプション)：電圧ペアのリストをレポートします。サポートされる電圧は、ファミリによって異なります。

-grade (オプション)：ターゲット デバイスの温度グレードをレポートします。

-process (オプション)：製造プロセス特性をレポートします。

-junction_temp (オプション)：モデリングに使用されるデバイスのジャンクション温度をレポートします。

-ambient_temp (オプション)：環境周囲温度をレポートします。

-thetaja (オプション)：モデリングに使用される Theta-JA 熱抵抗をレポートします。

-thetas a (オプション)：モデリングに使用される Theta-SA 熱抵抗をレポートします。

-airflow (オプション)：モデリングに使用するエアフローを LFM (リニア フィート/分) でレポートします。

-heatsink (オプション)：モデリングに使用されるヒートシンクのプタイプをレポートします。

-thetajb (オプション)：モデリングに使用される Theta-JB 熱抵抗をレポートします。

-board (オプション)：モデリングに使用されるボードのサイズをレポートします。

-board_temp (オプション)：モデリングに使用されるボードの温度を摂氏でレポートします。

-board_layers (オプション)：モデリングに使用されるボードの層数をレポートします。

-all (オプション)：動作条件の現在の値をすべてレポートします。各条件が個別にレポートされるのを避ける場合に使用します。

-file arg (オプション)：レポートを保存するファイルを指定します。

注記：パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-return_string (オプション)：出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記：このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、インダストリアル温度グレード デバイスの周囲温度を 75°C に設定し、それらの設定をディスク上のファイルに書き込んでいます。

```
set_operating_conditions -grade industrial -junction_temp 75
report_operating_conditions -grade -junction_temp -return_string -file \
~/conditions.txt
```

関連項目

[set_operating_conditions](#)

report_param

すべてのパラメーターに関する情報を取得します。

構文

```
report_param [-file arg] [-append] [-non_default] [-return_string] [-quiet]
[-verbose] [pattern]
```

戻り値

パラメーター レポート

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-non_default]	デフォルト以外の値に設定されているパラメーターのみをレポートします。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[pattern]	パターンに一致するパラメーターを取得します。デフォルトは*です。

カテゴリ

プロパティおよびパラメーター、レポート

説明

ユーザーが定義可能なパラメーターのリスト、現在の値、パラメーターのコンフィギュレーションまたは制御内容を取得します。

引数

-file arg (オプション) : レポートを保存するファイルを指定します。 **-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

pattern (オプション) : 指定したパターンに一致するパラメーターを取得します。デフォルトのパターンはワイルドカード (*) で、すべてのユーザー定義可能なパラメーターが取得されます。

例

次の例では、すべてのユーザー定義可能なパラメーターの名前、値、説明が返されます。

```
report_param
```

次の例では、指定した検索パターンに一致するユーザー定義可能なパラメーターの名前、値、および説明が返されます。

```
report_param *coll*
```

関連項目

- [get_param](#)
- [list_param](#)
- [reset_param](#)
- [set_param](#)

report_phys_opt

物理合成変換の詳細をレポートします。

構文

```
report_phys_opt [-file arg] [-append] [-return_string] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-file]	出力ファイルを指定します。
[-append]	結果をファイルの最後に追加します。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

phys_opt_design コマンドで実行されたファンアウトドライバー複製およびロード分配最適化の結果をレポートします。

引数

-file arg (オプション) : 物理最適化レポートを保存するファイルを指定します。 **-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-return_string (オプション) : レポートの出力を Tcl 文字列として返します。 Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、**phys_opt_design** コマンドで現在のデザインに対して実行された物理最適化をレポートしています。

```
report_phys_opt -file C:/Data/physOpt_Report.txt
```

関連項目

[phys_opt_design](#)

report_power

消費電力予測を実行し、レポートを表示します。

構文

```
report_power [-no_propagation] [-hier arg] [-vid] [-file arg] [-name arg]
[-format arg] [-xpe arg] [-l arg] [-return_string] [-append] [-fileset arg]
[-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-no_propagation]	ネットのスイッチング アクティビティを予測する伝搬エンジンをディスエーブルにします。
[-hier]	階層レポートの形式を指定します。有効な値は logic、power、all で、デフォルトは logic です。
[-vid]	デバイスの電圧 ID (VID) を指定します。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-name]	GUI パネルに表示する結果の名前を指定します。
[-format]	消費電力予測レポートのフォーマットを指定します。有効な値は text および xml で、デフォルトは text です。
[-xpe]	XPE にインポートできるように結果を XML ファイルに出力します。
[-l]	詳細レポートに含める最大行数を指定します。デフォルトは 10 です。
[-return_string]	レポートを文字列として返します。
[-append]	消費電力レポートをファイルの最後に追加します。
[-fileset]	ファイルセットを解析して PS7 の .xpe ファイルを取得します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート、電力

説明

現在のデザインで消費電力解析を実行し、デバイスの現在の動作条件とデザインのスイッチング レートに基づく消費電力の詳細をレポートします。動作条件は、**set operating conditions** コマンドを使用して設定できます。スイッチング アクティビティを定義するには、**set_default_switching_activity** コマンドを使用します。

スイッチング アクティビティは、**read_saif** コマンドを使用して SAIF ファイルから、**read_vcd** コマンドを使用して VCD ファイルから読み込むこともできます。SAIF または VCD ファイルからのアクティビティがデザイン ノードにアノテートされ、消費電力が正しく予測されます。

消費電力解析には、合成済みネットリストまたは配置配線済みデザインが必要です。

注記： デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

-no_propagation (オプション)：未定義のノードに対してアクティビティを予測するのにベクターなし伝搬エンジンを使用します。このオプションを使用すると、デザインを高速に解析する伝搬エンジンがディスエーブルになります。

-hier [logic | power | all] (オプション)：詳細レポートセクションに含める詳細 (ロジックまたは消費電力) を指定します。デフォルト値は **logic** です。

-vid (オプション)：ターゲット デバイスの電圧 ID ビットを使用します。電圧 ID は適応型電圧制御 (AVS) の一種で、Virtex-7 ファミリの一部のデバイスを低電圧 0.9V で動作させながら 1.0V の公称電源で動作させた場合と同じパフォーマンスを達成します。電圧 ID 対応のデバイスでは、ワースト ケース (最大) のスタティック消費電力が約 30% 低減されており、放熱量も大幅に少なくなっています。

-file arg (オプション)：レポートを保存するファイルを指定します。**-append** オプションが指定されていなければ、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。

注記： パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション)：コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記： **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-name arg (オプション)：結果の名前を指定します。

-xpe arg (オプション)：XPower Estimator または XPower Analyzer にインポートできるように結果を XML ファイルに出力します。

-l arg (オプション)：詳細レポート セクションの最大行数を指定します。0 以上の値を指定します。

注記： このオプションを指定すると、指定しないよりも詳細な情報が含まれます。

-return_string (オプション)：出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記： このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、**TCL_OK** が返されます。

注記： コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、ネット伝搬なしで消費電力解析を実行し、XPE で使用できるように結果を XML ファイルに出力します。

```
report_power -no_propagation -xpe C:/Data/design1.xpe
```

関連項目

- [read_saif](#)
- [read_vcd](#)
- [set_default_switching_activity](#)
- [set_operating_conditions](#)

report_power_opt

消費電力の最適化をレポートします。

構文

```
report_power_opt [-cell arg] [-file arg] [-format arg] [-name arg] [-append]
[-return_string] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-cell]	最上位セルを指定します。
[-file]	出力ファイルを指定します。
[-format]	レポートのフォーマットを指定します。有効な値は text および xml です。-file を使用した場合にのみ適用されます。xml を指定した場合、-append は使用できません。デフォルトは text です。
[-name]	GUI パネルに表示する結果の名前を指定します。
[-append]	既存のファイルの最後に追加します。このオプションを指定しない場合、既存のファイルは上書きされます。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

電力

説明

power_opt_design コマンドを使用してデザインに実行した消費電力最適化をレポートします。

注記： デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

-cell arg (オプション) : 指定したセル インスタンスの消費電力最適化をレポートします。デフォルトでは、デザイン全体の消費電力最適化がレポートされます。

-file arg (オプション) : レポートを保存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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインに対して実行された消費電力最適化をレポートしています。

```
report_power_opt
```

関連項目

- [power_opt_design](#)
- [report_power](#)

report_property

オブジェクトのプロパティをレポートします。

構文

```
report_property [-all] [-class arg] [-return_string] [-file arg] [-append]
[-regexp] [-quiet] [-verbose] [object] [pattern]
```

戻り値

プロパティ レポート

使用法

名前	説明
[-all]	設定していないものも含めてオブジェクトのプロパティすべてをレポートします。
[-class]	プロパティを取得するオブジェクト タイプを指定します。オブジェクトを指定する場合は使用できません。
[-return_string]	Tcl インタープリターの result 変数に report_property の実行結果を設定します。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-regexp]	検索パターンを正規表現として処理します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[object]	プロパティを取得するオブジェクトを指定します。
[pattern]	プロパティを検索するパターンを指定します。デフォルトは*です。

カテゴリ

オブジェクト、プロパティおよびパラメーター、レポート

説明

指定したオブジェクトのプロパティすべてに対してその名前、タイプ、および値を取得します。

注記： list_property でもオブジェクトのプロパティのリストが返されますが、プロパティ タイプと値は含まれません。

引数

-all (オプション)：プロパティ値が定義されていなくても、オブジェクトのプロパティすべてを返します。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

-file arg (オプション) : レポートを保存するファイルを指定します。**-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

object (オプション) : プロパティを取得するオブジェクトを 1 つ指定します。

注記 : 複数のオブジェクトを指定すると、エラー メッセージが表示されます。

例

次の例では、指定したオブジェクトのプロパティすべてが返されます。

```
report_property -all [get_cells cpuEngine]
```

ツールでサポートされる異なるデザイン オブジェクトに使用可能なプロパティを取得するには、複数の **report_property** コマンドを連続して使用します。次の例では、指定したオブジェクトのプロパティすべてが返されます。

```
report_property -all [current_project]
report_property -all [current_fileset]
report_property -all [current_design]
report_property -all [current_run]
```

関連項目

- [create_property](#)
- [get_cells](#)
- [get_property](#)
- [list_property](#)
- [list_property_value](#)
- [reset_property](#)
- [set_property](#)

report_pulse_width

パルス幅チェックをレポートします。

構文

```
report_pulse_width [-file arg] [-append] [-name arg] [-return_string]
[-all_violators] [-significant_digits arg] [-limit arg] [-min_period]
[-max_period] [-low_pulse] [-high_pulse] [-max_skew] [-clocks args] [-quiet]
[-verbose] [objects]
```

戻り値

なし

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-name]	出力を保存する結果の名前を指定します。
[-return_string]	レポートを文字列として返します。
[-all_violators]	チェック違反が発生しているピン/ポートのみをレポートします。
[-significant_digits]	有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 2 です。
[-limit]	クロックごとにレポートする特定のタイプのチェック数を指定します。デフォルトは 1 です。
[-min_period]	最小周期チェックのみをレポートします。
[-max_period]	最大周期チェックのみをレポートします。
[-low_pulse]	最小 Low パルス幅チェックのみをレポートします。
[-high_pulse]	最小 High パルス幅チェックのみをレポートします。
[-max_skew]	最大スキュー チェックのみをレポートします。
[-clocks]	最小パルス幅/最小周期チェックをレポートするクロックをリストします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	最小パルス幅をチェックするオブジェクトのリストを指定します。

カテゴリ

レポート

説明

クロック ネットワークおよびフリップフロップに到達したときの、指定したクロック信号のパルス幅をレポートします。クロックの立ち上がりエッジの最大遅延と立ち下がりエッジの最小値エンを使用して、High パルス幅のチェックも実行します。クロックの立ち上がりエッジの最小遅延と立ち下がりエッジの最大値エンを使用して、Low パルス幅のチェックを実行します。これにより、立ち上がりエッジと立ち下がりエッジのワーストケースの遅延が考慮されるため、現在の合成済みデザインまたはインプリメント済みデザインのワーストケース解析が得られます。最大スキー (クロック信号間に許容される最大のタイミング差) もレポートされます。

デフォルトでは、レポートに最小パルス幅、最大パルス幅、Low パルス幅、High パルス幅、および最大スキー チェックが含まれます。特定のチェックを指定すると、同時に指定されていない限りほかのチェックはディスエーブルになります。

デフォルトではレポートは標準出力に出力されますが、その後の処理用にファイルまたは Tcl 文字列変数に出力することもできます。レポートは、**-file**、**-return_string**、または **-name** を指定した場合を除き、デフォルトでは標準出力に返されます。

引数

-file *arg* (オプション) : レポートを保存するファイルを指定します。**-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-name *arg* (オプション) : GUI で表示する場合の結果の名前を指定します。GUI のパルス幅レポートは、**delete_timing_results** コマンドを使用して削除できます。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-all_violators (オプション) : 違反が検出されたオブジェクトのみをレポートします。

-significant_digits *arg* (オプション) : 出力結果の有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 2 です。

-limit *arg* (オプション) : クロックごとにレポートする特定のタイプのチェック数を指定します。1 以上の値を指定します。デフォルトは 1 です。

-min_period (オプション) : 最小周期チェックをレポートします。

-max_period (オプション) : 最大周期チェックをレポートします。

-low_pulse (オプション) : 最小 Low パルス幅チェックをレポートします。

-high_pulse (オプション) : 最小 High パルス幅チェックをレポートします。

-max_skew (オプション) : 2 つのクロックピン間のスキー制約をチェックします。

注記 : **report_pulse_width** コマンドのデフォルトでは、**min_period**、**max_period**、**low_pulse**、**high_pulse**、および **max_skew** がレポートされます。これらのオプションの 1 つまたは複数指定すると、指定したチェックのみがレポートされます。

-clocks arg (オプション) : パルス幅/周期チェックをレポートするクロックをリストします。-clocks オプションを指定しない場合、すべてのクロックがチェックされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (オプション) : パルス幅をレポートするピン オブジェクトを指定します。objects を指定しない場合、すべてのピンがチェックされます。

例

次の例では、最小周期および Low パルス幅チェックを実行し、結果を GUI に指定した名前で表示しています。

```
report_pulse_width -min_period -low_pulse -name timing_1
```

関連項目

[delete_timing_results](#)

report_route_status

配線のステータスをレポートします。

構文

```
report_route_status [-return_string] [-file arg] [-append] [-of_objects args]
[-route_type arg] [-list_all_nets] [-show_all] [-dump_routes] [-has_routing]
[-ignore_cache] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-return_string]	Tcl インタープリターの result 変数にレポートの実行結果を設定します。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-of_objects]	指定した配線オブジェクトの詳細な配線情報をレポートします。
[-route_type]	指定したステータスの配線のみをレポートします。有効な値は UNPLACED、NOLOADS、NODRIVER、UNROUTED、ANTENNAS、CONFLICTS、PARTIAL、INTRASITE、HIERPORT、ROUTED です。-of_objects を使用した場合は無視されます。
[-list_all_nets]	デザインの各ネットに対して完全な配線情報をレポートします。-of_objects を使用した場合は無視されます。
[-show_all]	UNPLACED または PARTIAL 配線とマークされた配線に関連するすべてのピン、ANTENNAS または CONFLICTS 配線とマークされた配線に関連するすべてのノードをリストします。デフォルトでは、1 つの配線に対して最初の 15 個のピンまたはノードがレポートされます。
[-dump_routes]	デザインの各配線済みネットに対して完全な配線ツリーをレポートします。これは非常に詳細なレポートです。
[-has_routing]	このデザインに保存されている配線がない場合は 0、ある場合は 1 を返します。その他のオプションはすべて無視されます。
[-ignore_cache]	キャッシュされている情報を破棄し、デザイン全体の配線ステータスを算出し直します。これは非常に低速です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

現在のデザインの配線ステートをレポートします。

単にデザインに配線があるかどうかを示す 1 から、デザインの各ネットの完全な配線ツリーまで、さまざまな情報を持ったステータス レポートを生成できます。

引数

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-file *arg* (オプション) : レポートを保存するファイルを指定します。**-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-of_objects *args* (オプション) : 指定した配線、ネット、または xdef_net オブジェクトの完全な配線ツリーをレポートします。

-route_type *arg* (オプション) : 指定した配線ステータスの配線のみをレポートします。有効な値は UNPLACED、NOLOADS、NODRIVER、UNROUTED、ANTENNAS、CONFLICTS、PARTIAL、INTRASITE、HIERPORT、ROUTED です。

注記 : **-of_objects** を使用した場合は無視されます。

-list_all_nets (オプション) : デザインに含まれるすべてのネットの配線ステータス サマリをレポートします。

注記 : **-of_objects** を使用した場合は無視されます。

-show_all (オプション) : UNPLACED または PARTIAL 配線とマークされた配線に関連するすべてのピン、ANTENNAS または CONFLICTS 配線とマークされた配線に関連するすべてのノードをリストします。デフォルトでは、1 つの配線に対して最初の 15 個のピンまたはノードがレポートされます。

-dump_routes (オプション) : デザインの各配線済みネットに対して完全な配線ツリーをレポートします。

注記 : このオプションを使用するとレポートが長くなり、生成するのに時間がかかる場合があります。

-has_routing (オプション) : デザインが配線されていない場合は 0、デザインに配線が含まれている場合は 1 を返します。**-has_routing** を指定した場合、その他のオプションはすべて無視されます。

注記 : 1 が返された場合でも、デザインが完全に配線されているとは限りません。

-ignore_cache (オプション) : デフォルトでは、**report_route_status** コマンドは反復的であり、デザインがインプリメントされるにつれ、新しいネットおよび配線の配線情報のみがアップデートされます。このオプションを指定すると、キャッシュに含まれる情報が無視され、デザイン全体のレポートが再生成されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、指定のネットの配線ステータスをレポートしています。

```
report_route_status -of_objects [get_nets u4*]
```

report_scopes

指定したスコープまたは現在のスコープの子スコープ (宣言領域) の名前を表示します。

構文

```
report_scopes [-quiet] [-verbose] [hdl_scopes ...]
```

戻り値

HDL スコープのプロパティのサブセット

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hdl_scopes]	レポートする HDL オブジェクトを指定します。デフォルトは report_scopes [get_scopes *] です。

カテゴリ

シミュレーション

report_simlib_info

シミュレーション ライブラリの情報をレポートします。

構文

```
report_simlib_info [-file arg] [-append] [-quiet] [-verbose] path
```

戻り値

なし

使用法

名前	説明
[-file]	出力ファイル名を指定します。デフォルトは report_simlib_info.log です。
[-append]	情報をファイルの最後に追加します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
path	コンパイル済みライブラリのパスを指定します。

カテゴリ

シミュレーション

report_ssn

現在のパッケージおよびピン配置で SSN 解析を実行します。

構文

```
report_ssn [-name arg] [-return_string] [-format arg] [-file arg] [-append]
[-phase] [-quiet] [-verbose]
```

戻り値

SSN レポート

使用法

名前	説明
[-name]	GUI パネルに表示する結果の名前を指定します。
[-return_string]	レポートを文字列として返します。
[-format]	レポートフォーマットを指定します。有効な値は CSV、HTML、TXT で、デフォルトは csv です。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	レポートを指定したファイルの最後に追加します。
[-phase]	解析で複数クロック位相を考慮します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

現在のデザインの同時スイッチノイズ (SSN) 解析を実行します。SSN 解析は、出力の切り替わりがインターフェイスのノイズマージンに与える影響を予測するのに適した方法です。予測は、デザインで発生する可能性のあるノイズに関する問題を特定するためのさまざまな変数に基づいており、最終的にデザインが完了したと判断するための条件としては使用しないでください。

SSN 解析では、同時に切り替わる出力が I/O バンク内のほかの出力ポートに与える影響を予測します。SSN 予測には、I/O バンク特定の電気特性も組み込まれており、SSN でのパッケージの影響も正確にモデリングされます。

report_ssn コマンドは、**set_operating_condition** の **-grade** オプションで指定された選択デバイスの温度グレードに影響される可能性があります。ノイズ解析を実行する前に温度グレードを設定しておくと、コマーシャル、拡張、Q グレード、またはミリタリ グレード デバイスのにおける信号のノイズを確認できます。

デフォルトでは、すべてのポートが非同期にトグルすると想定されます。そのため、ワーストケースのノイズ解析となり、不必要に悪い見積もりとなる可能性があります。**-phase** オプションを使用すると、デザインのクロック情報が考慮され、より正確な SSN ノイズがレポートされます。クロックは、**create_clock** および **create_generated_clock** コマンドを使用して定義されている必要があります。生成クロックの周期、位相シフト、デューティサイクルは、SSN 解析に大きく影響します。

report_ssn コマンドは、Virtex-7、Kintex-7、および Artix-7 デバイスの詳細な SSN 解析をレポートします。レポートは、**-file**、**-return_string**、または **-name** を指定した場合を除き、標準出力に返されます。

引数

-name arg (オプション) : GUI に出力する結果の名前を指定します。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-format [CSV | HTML | TXT] (オプション) : 出力のフォーマットを CSV、HTML、または ASCII (TXT) に指定します。デフォルトは CSV です。

注記 : フォーマットは **-file** を指定した場合にのみ適用され、それ以外は無視されます。

-file arg (オプション) : SSN レポートを保存するファイルを指定します。**-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-phase (オプション) : SSN 解析でより正確な結果を得るために、クロックスイッチサイクルを考慮します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインに対して SSN 解析を実行し、その出力を指定したファイルに HTML 形式で保存しています。

```
report_ssn -format html -file C:/Data/devSSN.html
```

次の例では、現在のデザインに対して SSN 解析を位相解析を含めて実行し、その出力を文字列として指定の変数に保存しています。

```
set devSSN [report_ssn -phase -format html -return_string]
```

注記：この例では **-file** が指定されていないので、**-format** は無視されます。

関連項目

- [create_clock](#)
- [create_generated_clock](#)
- [reset_ssn](#)

report_switching_activity

指定したオブジェクトのスイッチング アクティビティを取得します。

構文

```
report_switching_activity [-static_probability] [-signal_rate] [-file arg]
[-return_string] [-append] [-quiet] [-verbose] [objects...]
```

戻り値

なし

使用法

名前	説明
[-static_probability]	スタティック確率をレポートします。
[-signal_rate]	信号レートをレポートします。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-return_string]	スイッチング アクティビティを文字列として返します。
[-append]	スイッチング アクティビティをファイルの最後に追加します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	オブジェクトを指定します。

カテゴリ

レポート

説明

デザインのネット、ポート、ピン、およびセルの異なるスイッチング アクティビティをレポートします。レポートされるスイッチング アクティビティには、ネット、ポート、ピンの単純な信号レートと単純なスタティック確率、およびセルのステートに依存するスタティック確率が含まれます。

レポートされる値は、`set_switching_activity` コマンドで定義します。

注記： このコマンドを実行すると、指定のオブジェクトのスイッチング アクティビティが返されます。

注記： デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

-static_probability (オプション)： レポートの一部としてスタティック確率を返します。

-signal_rate (オプション) : レポートの一部として信号レートを返します。

-file filename (オプション) : レポートを指定のパスの指定のファイルに書き込みます。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-return_string (オプション) : データを Tcl 変数に代入できるようにテキスト文字列として返します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (オプション) : スイッチング アクティビティ情報をレポートするネット、ポート、またはピン オブジェクトのリストを指定します。

例

次の例では、デザインに含まれるすべての出力ポートの信号レートとスタティック確率がレポートされます。

```
report_switching_activity -signal_rate -static_probability [all_outputs]
```

関連項目

- [power_opt_design](#)
- [report_default_switching_activity](#)
- [report_power](#)
- [reset_default_switching_activity](#)
- [reset_switching_activity](#)
- [set_default_switching_activity](#)
- [set_switching_activity](#)

report_timing

タイミング パスをレポートします。

構文

```
report_timing [-from args] [-rise_from args] [-fall_from args] [-to args]
[-rise_to args] [-fall_to args] [-through args] [-rise_through args]
[-fall_through args] [-delay_type arg] [-setup] [-hold] [-max_paths arg]
[-nworst arg] [-unique_pins] [-path_type arg] [-input_pins] [-no_header]
[-label_reused] [-slack_lesser_than arg] [-slack_greater_than arg]
[-group args] [-sort_by arg] [-no_report_unconstrained] [-user_ignored]
[-match_style arg] [-of_objects args] [-significant_digits arg] [-file arg]
[-append] [-name arg] [-return_string] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-from]	タイミング パスの開始点 (ピン、ポート、セル、またはクロック) を指定します。
[-rise_from]	ピン、ポート、セル、またはクロックの立ち上がりエッジを開始点として指定します。
[-fall_from]	ピン、ポート、セル、またはクロックの立ち下がりエッジを開始点として指定します。
[-to]	タイミング パスの終点 (ピン、ポート、セル、またはクロック) を指定します。
[-rise_to]	ピン、ポート、セル、またはクロックの立ち上がりエッジを終点として指定します。
[-fall_to]	ピン、ポート、セル、またはクロックの立ち下がりエッジを終点として指定します。
[-through]	タイミング パスの通過点 (ピン、ポート、セル、またはネット) を指定します。
[-rise_through]	ピン、ポート、セル、またはネットの立ち上がりエッジを通過点として指定します。
[-fall_through]	ピン、ポート、セル、またはネットの立ち下がりエッジを通過点として指定します。
[-delay_type]	パス遅延のタイプを指定します。有効な値は max、min、min_max、max_rise、max_fall、min_rise、min_fall で、デフォルトは max です。
[-setup]	最大遅延タイミング パスをレポートします (-delay_type max と同じ)。
[-hold]	最小遅延タイミング パスをレポートします (-delay_type min と同じ)。
[-max_paths]	スラック順に並べた場合に出力するパスの最大数、またはグループごとに並べた場合のパス グループごに出力するパスの最大数を指定します。1 以上の値を指定します。デフォルト値は 1 です。

名前	説明
<code>[-nworst]</code>	エンドポイントまでのワーストパスを N 個までリストします。1 以上の値を指定します。デフォルト値は 1 です。
<code>[-unique_pins]</code>	ピンの各固有のセットに対して、パス グループに 1 つのパスを表示します。
<code>[-path_type]</code>	パス レポートのフォーマットを指定します。有効な値は end、summary、short、full、full_clock、full_clock_expanded で、デフォルトは full_clock_expanded です。
<code>[-input_pins]</code>	パスの入力ピンを表示します。
<code>[-no_header]</code>	
<code>[-label_reused]</code>	レポートのピンに再利用ステータスを示すラベルを表示します。
<code>[-slack_lesser_than]</code>	この値よりも小さいスラックのパスを表示します。デフォルトは 1e+30 です。
<code>[-slack_greater_than]</code>	この値よりも大きいスラックのパスを表示します。デフォルトは -1e+30 です。
<code>[-group]</code>	指定のグループのパスのみをレポートします。
<code>[-sort_by]</code>	パスの並べ替え順を指定します。有効な値は group、slack で、デフォルトは slack です。
<code>[-no_report_unconstrained]</code>	スラックが無限のパスはレポートしません。
<code>[-user_ignored]</code>	set_false_path または set_clock_groups タイミング制約のためにスラックが無限のパスのみをレポートします。
<code>[-match_style]</code>	パターン一致のスタイルを指定します。有効な値は ucf、sdc で、デフォルトは ucf です。
<code>[-of_objects]</code>	タイミングをレポートするパスを指定します。
<code>[-significant_digits]</code>	有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 3 です。
<code>[-file]</code>	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
<code>[-append]</code>	結果をファイルの最後に追加します。上書きはしません。
<code>[-name]</code>	GUI パネルに表示する結果の名前を指定します。
<code>[-return_string]</code>	レポートを文字列として返します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。

カテゴリ

レポート、タイミング

説明

現在の合成済みデザインまたはインプリメント済みデザインの指定のタイミング パスに対してタイミング解析を実行します。デフォルトでは、各パスグループのスラックが最大のタイミング パスがレポートされます。`-nworst` または `-max_paths` オプションを使用すると、レポートするパスおよび遅延の数を変更できます。

タイミング エンジンはクワッドタイミング モードで実行され、スロー コーナーとファーストコーナーの最小遅延と最大遅延が解析されます。**config_timing_corners** コマンドで実行する解析のタイプを設定できますが、変更するとタイミング解析の適用範囲が減少するので、デフォルトを変更することはお勧めしません。

注記：デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じて、結果を GUI に表示したり、ファイルに書き込んだり、文字列として返すこともできます。

引数

-from args (オプション)：解析するタイミング パスの開始点を指定します。ポート、ピン、またはセルをタイミング パスの開始点として指定できます。また、クロック オブジェクトを指定すると、そのクロックが供給されるオブジェクトが開始点として指定されます。

-rise_from args (オプション)：**-from** オプションと同様ですが、指定した開始点からの信号の立ち上がりエッジのみをタイミング解析で考慮します。クロック オブジェクトを指定すると、そのクロックの立ち上がりエッジで駆動されるパスのみが開始点として考慮されます。

-fall_from args (オプション)：**-from** オプションと同様ですが、指定した開始点からの信号の立ち下がりエッジのみをタイミング解析で考慮します。クロック オブジェクトを指定すると、そのクロックの立ち下がりエッジで駆動されるパスのみが開始点として考慮されます。

-to args (オプション)：解析するタイミング パスの終点 (デスティネーション オブジェクト) を指定します。ポート、ピン、またはセル オブジェクトを終点として指定できます。また、クロック オブジェクトを指定すると、そのクロックが供給されるオブジェクトが終点として指定されます。

-rise_to args (オプション)：**-to** オプションと同様ですが、指定した終点に到達する信号の立ち上がりエッジのみをタイミング解析で考慮します。クロック オブジェクトを指定すると、そのクロックの立ち上がりエッジでキャプチャされるパスのみが終点として考慮されます。

-fall_to args (オプション)：**-to** オプションと同様ですが、指定した終点に到達する信号の立ち下がりエッジのみをタイミング解析で考慮します。クロック オブジェクトを指定すると、そのクロックの立ち下がりエッジでキャプチャされるパスのみが終点として考慮されます。

-through args (オプション)：指定したピン、セル インスタンス、またはネットを通過するパスのみをタイミング解析で考慮します。個別の **-through** (または **-rise_through** および **-fall_through**) 点を順次指定し、デザインを通過する特定のパスを定義できます。特定のパスを定義するには、通過点の指定順序が重要です。通過点を複数のオブジェクトで指定することもできます。この場合、指定の通過オブジェクトのいずれかを通過するタイミング パスが考慮されます。

-rise_through args (オプション)：**-through** オプションと同様ですが、指定したオブジェクトで立ち上がるパスのみでタイミング解析を実行します。

-fall_through args (オプション)：**-through** オプションと同様ですが、指定したオブジェクトで立ち下がるパスのみでタイミング解析を実行します。

-delay_type arg (オプション)：タイミング レポートを実行する際に解析に使用する遅延のタイプを指定します。有効な値は min、max、min_max、max_rise、max_fall、min_rise、min_fall です。デフォルトは max です。

-setup (オプション)：セットアップ違反がないかどうかをチェックします。これは **-delay_type max** を指定するのと同じです。

-hold (オプション)：ホールド違反がないかどうかをチェックします。これは **-delay_type min** を指定するのと同じです。

注記：**-setup** と **-hold** の両方を指定すると、**-delay_type min_max** を指定するのと同じになります。

-max_paths arg (オプション) : **-sort_by** での指定の応じて、スラック順に並べた場合に出力するパスの最大数、またはグループごとに並べた場合にパスグループごとに出力するパスの最大数を指定します。1 以上の値を指定します。デフォルトでは、ワーストタイミング パス 1 つ、またはパスグループごとにワースト パス 1 つがレポートされます。

-nworst arg (オプション) : タイミング レポートに出力するタイミング パスの数を指定します。タイミング レポートには、指定した数のワースト パスが表示されます。1 以上の値を指定します。デフォルト値は 1 です。

-unique_pins (オプション) : ピンの各固有のセットに対して 1 つのタイミング パスのみを表示します。

-path_type arg (オプション) : タイミング レポートに出力するパス データを指定します。デフォルトフォーマットは `full_clock_expanded` です。有効なパス タイプは、次のとおりです。

- `end` : パスの終点のみをタイミング 値と共に表示します。
- `summary` : 開始点と終点をスラックと共に表示します。
- `short` : 開始点と終点をタイミング 値と共に表示します。
- `full` : 開始点、通過点、終点を含む完全なタイミング パスを表示します。
- `full_clock` : 完全なタイミング パスに加えて完全なクロック パスを表示します。
- `full_clock_expanded` : `full_clock` で表示されるタイミング パスに加え、マスター クロックおよび生成したクロック 間の完全なクロック パスを表示します。これがデフォルト設定です。

-input_pins (オプション) : タイミング パス レポートに入力ピンを表示します。**-path_type** を `full`、`full_clock`、および `full_clock_expanded` に設定した場合に使用します。

-no_header (オプション) : レポートにヘッダーを含めません。

-label_reused (オプション) : インクリメンタル配置配線 (`read_checkpoint -incremental`) を使用するデザインに対し、ピンに指定のインクリメンタル チェックポイントから再利用された物理データに関する情報を示すラベルを表示します。再利用ラベルは、次のとおりです。

- `R` : セル配置およびこのピンへの配線は再利用されています。
- `PNR` : セル配置は再利用されているが、このピンへの配線は再利用されていません。
- `NR` : セル配置およびこのピンへの配線はどちらも再利用されていません。
- `N` : セル、ネット、またはピンは新規です。インクリメンタル チェックポイントに存在しません。

-slack_lesser_than arg (オプション) : 算出されたスラック値が指定した値より小さいパスのタイミングをレポートします。**-slack_greater_than** と共に使用すると、スラック値の範囲を指定できます。

-slack_greater_than arg (オプション) : 算出されたスラック値が指定した値より大きいパスのタイミングをレポートします。**-slack_lesser_than** と共に使用すると、スラック値の範囲を指定できます。

-group arg (オプション) : 指定したパス グループのパスのタイミングをレポートします。現在定義されているパス グループを確認するには、`get_path_groups` コマンドを使用します。

注記 : 各クロックにパス グループが作成されます。パス グループは、`group_path` コマンドを使用して定義することもできます。

-sort_by [slack | group] (オプション) : レポートのタイミング パスをスラック値順またはパス グループごとリストします。有効な値は `slack` または `group` です。デフォルトでは、ワーストタイミング パス 1 つまたは **-nworst** で指定した数のタイミング パスがレポートされますが、**-sort_by group** を使用すると、各パス グループに対してワーストタイミング パス 1 つまたは **-nworst** で指定した数のタイミング パスがレポートされます。

-no_report_unconstrained (オプション) : 制約が適用されていないパスのタイミングはレポートしません。デフォルトでは、制約が適用されていないパス (スラックは無限) が含まれます。

-user_ignored (オプション) : **set_false_path** または **set_clock_groups** 制約のため通常タイミングで無視されるパスのみをレポートします。

注記 : **-user_ignored** と **-no_report_unconstrained** オプションを同時に使用することはできません。

-match_style [sdc | ucf] (オプション) : オブジェクトの検索パターンが UCF 制約または SDC 制約に一致することを示します。デフォルト値は UCF です。

-of_objects args (オプション) : 指定のタイミング パス オブジェクトのタイミングをレポートします。**get_timing_paths** コマンドと共に使用されます。

注記 : **-of_objects** オプションは、**-from**、**-to**、**-through** オプション、およびそのバリエーションと共に使用することはできません。

-significant_digits arg (オプション) : 出力結果の有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 3 です。

-file arg (オプション) : レポートを保存するファイルを指定します。**-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-name arg (オプション) : GUI で表示する場合の結果の名前を指定します。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、デザインのワースト パス 5 つの入力ピンを含む完全なタイミング パスと、そのタイミング値をレポートしています。

```
report_timing -nworst 5 -path_type full -input_pins
```

次の例では、複数の通過点を使用して特定のパス (state_reg1 を通過) および代替パス (count_3 または count_4 を通過) を指定し、タイミング結果を指定のファイルに記述しています。

```
report_timing -from go -through {state_reg1} -through { count_3  count_4 } \
-to done -path_type summary -file C:/Data/timing1.txt
```

関連項目

- [get_path_groups](#)
- [get_timing_paths](#)
- [group_path](#)
- [place_design](#)
- [report_timing_summary](#)
- [route_design](#)
- [set_clock_groups](#)
- [set_false_path](#)
- [set_msg_limit](#)

report_timing_summary

タイミング サマリをレポートします。

構文

```
report_timing_summary [-check_timing_verbose] [-delay_type arg]
[-no_detailed_paths] [-setup] [-hold] [-max_paths arg] [-nworst arg]
[-path_type arg] [-label_reused] [-input_pins] [-slack_lesser_than arg]
[-report_unconstrained] [-significant_digits arg] [-no_header] [-file arg]
[-append] [-name arg] [-return_string] [-datasheet] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-check_timing_verbose]	発生する可能性のあるタイミング問題をチェックする際に詳細レポートを生成します。
[-delay_type]	パス遅延のタイプを指定します。有効な値は max、min、min_max で、デフォルトは min_max です。
[-no_detailed_paths]	解析された各クロックおよびパス グループのタイミング パスをレポートしません。
[-setup]	最大遅延タイミング パスをレポートします (-delay_type max と同じ)。
[-hold]	最小遅延タイミング パスをレポートします (-delay_type min と同じ)。
[-max_paths]	クロックまたはパス 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。1 以上の値を指定します。デフォルト値は 1 です。
[-nworst]	エンドポイントまでのワーストパスを N 個までリストします。1 以上の値を指定します。デフォルト値は 1 です。
[-path_type]	パス レポートのフォーマットを指定します。有効な値は end、summary、short、full、full_clock、full_clock_expanded で、デフォルトは full_clock_expanded です。
[-label_reused]	レポートのピンに再利用ステータスを示すラベルを表示します。
[-input_pins]	パスの入力ピンを表示します。
[-slack_lesser_than]	この値よりも小さいスラックのパスを表示します。デフォルトは 1e+30 です。
[-report_unconstrained]	制約が適用されていないパスをレポートします。
[-significant_digits]	有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 3 です。
[-no_header]	レポート ヘッダーを生成しません。
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。

名前	説明
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-name]	GUI パネルに表示する結果の名前を指定します。
[-return_string]	レポートを文字列として返します。
[-datasheet]	データシートレポートを含めます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート、タイミング

説明

デザインでタイミング要件が満たされているかどうかを確認するためのタイミング サマリを生成します。開いている合成済みデザインまたはインプリメンツ済みデザインに対して実行できます。

タイミング サマリ レポートには、次の情報が含まれます。

- 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タイミング エンジンの設定を示します。
- Check Timing : **check_timing** コマンドで生成されるのと同じ情報を含み、発生する可能性のあるタイミング 問題を示します。
- Design Timing Summary : 最悪の負のスラック (WNS)、負のスラックの合計 (TNS)、最悪のホールド スラック (WHS)、ホールド スラックの合計 (THS)、コンポーネントのスイッチ制限 (CSL) など、デザインのタイミングのサマリを示します。
- Clock Definitions : **report_clocks** コマンドで生成されるのと同じ情報を含み、**create_clock**、**create_generated_clock** コマンド、またはツールで自動的に生成されたすべてのクロックを示します。
- Intra-Clock Table : 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 パスのサマリを示します。
- Inter-Clock Table : 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 パスのサマリを示します。
- Path Group Table : **group_path** コマンドで作成されたデフォルトのパス グループとユーザー定義のパス グループを示します。
- Timing Details : **report_timing** コマンドで生成されるのと同様のタイミング パス (最大遅延と最小遅延の両方) および定義された各クロックのコンポーネント スイッチ制限を示します。
- Data sheet : **report_datasheet** コマンドを実行した場合と同じ情報を含み、I/O ポートでのデザインのタイミング特性を示します。データシート情報は、**-datasheet** オプションを指定するとサマリ レポートに追加されます。

このコマンドは、**launch_runs** コマンドの一部としてインプリメンテーション中に自動的に実行されます。

注記 : デフォルトでは、レポートは Tcl コンソールまたは STD 出力に表示されますが、必要に応じてファイルに書き込んだり、文字列として返すこともできます。

引数

-check_timing_verbose (オプション) : 詳細なタイミング サマリ レポートを出力します。

-delay_type *arg* (オプション) : タイミングレポートを実行する際に解析に使用する遅延のタイプを指定します。有効な値は min、max、min_max で、デフォルトは min_max です。

-no_detailed_paths (オプション) : 解析された各クロックおよびパス グループの完全なタイミング パスをレポートしません。

-setup (オプション) : セットアップ違反がないかどうかをチェックします。これは **-delay_type max** を指定するのと同じです。

-hold (オプション) : ホールド違反がないかどうかをチェックします。これは **-delay_type min** を指定するのと同じです。

注記 : **-setup** と **-hold** の両方を指定すると、**-delay_type min_max** を指定するのと同じになります。

-max_paths *arg* (オプション) : クロックまたはパス 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。1 以上の値を指定します。デフォルトでは、ワーストタイミング パス 1 つ、またはパス グループごとにワースト パス 1 つがレポートされます。

-nworst *arg* (オプション) : タイミング レポートに出力するタイミング パスの数を指定します。タイミング レポートには、指定した数のワースト パスから終点までが表示されます。1 以上の値を指定します。デフォルト値は 1 です。

-path_type *arg* (オプション) : タイミング サマリ レポートに出力するパス データを指定します。デフォルト フォーマットは **full_clock_expanded** です。有効なパス タイプは、次のとおりです。

- **end** : パスの終点のみをタイミング値と共に表示します。
- **summary** : 開始点と終点をスラックと共に表示します。
- **short** : 開始点と終点をタイミング値と共に表示します。
- **full** : 開始点、通過点、終点を含む完全なタイミング パスを表示します。
- **full_clock** : 完全なタイミング パスに加えて完全なクロック パスを表示します。
- **full_clock_expanded** : **full_clock** で表示されるタイミング パスに加え、マスター クロックおよび生成したクロック 間の完全なクロック パスを表示します。これがデフォルト設定です。

-label_reused (オプション) : インクリメンタル配置配線 (**read_checkpoint -incremental**) を使用するデザインに対し、ピンに指定のインクリメンタル チェックポイントから再利用された物理データに関する情報を示すラベルを表示します。再利用ラベルは、次のとおりです。

- R : セル配置およびこのピンへの配線は再利用されています。
- PNR : セル配置は再利用されているが、このピンへの配線は再利用されていません。
- NR : セル配置およびこのピンへの配線はどちらも再利用されていません。
- N : セル、ネット、またはピンは新規です。インクリメンタル チェックポイントに存在しません。

-input_pins (オプション) : タイミング パス レポートに入力ピンを表示します。**-path_type** を **full**、**full_clock**、および **full_clock_expanded** に設定した場合に使用します。

-slack_lesser_than *arg* (オプション) : 算出されたスラック値が指定した値より小さいパスのタイミングをレポートします。**-slack_greater_than** と共に使用すると、スラック値の範囲を指定できます。

-report unconstrained (オプション) : 現在のデザインに含まれる制約が適用されていないパスのタイミングをレポートします。デフォルトでは、制約が適用されていないパスのタイミングは含まれません。

-significant_digits *arg* (オプション) : 出力結果の有効桁数を指定します。有効な値は 0 ~ 3 で、デフォルト値は 3 です。

-no_header (オプション) : レポートにヘッダー情報を含めません。このオプションは、その後の処理用にタイミング サマリ レポートを文字列として出力する場合に便利です。

-file arg (オプション) : レポートを保存するファイルを指定します。**-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-name arg (オプション) : GUI で表示する場合の結果の名前を指定します。GUI のタイミング サマリ レポートは、**delete_timing_results** コマンドを使用して削除できます。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-datasheet (オプション) : タイミング サマリ レポートにデータシート情報を追加します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインのタイミング サマリをレポートしています。

```
report_timing_summary
```

次の例では、現在のデザインの制約が適用されていないパスを含むホールド タイミング サマリを、指定のオプションでレポートしています。

```
report_timing_summary -delay_type min -path_type full_clock_expanded \
-report_unconstrained -max_paths 2 -nworst 1 -significant_digits 2 \
-input_pins -name {timing_6}
```

関連項目

- [check_timing](#)
- [create_clock](#)
- [create_generated_clock](#)
- [delete_timing_results](#)
- [get_path_groups](#)
- [get_timing_paths](#)
- [group_path](#)
- [report_clocks](#)
- [report_timing](#)
- [set_msg_limit](#)
- [report_datasheet](#)

report_transformed_primitives

UNISIM プリミティブ変換の詳細をレポートします。

構文

```
report_transformed_primitives [-file arg] [-append] [-return_string] [-quiet]
[-verbose]
```

戻り値

なし

使用法

名前	説明
[-file]	出力ファイルを指定します。
[-append]	結果をファイルの最後に追加します。
[-return_string]	レポートを文字列として返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

現在のデザインの変換されたプリミティブをレポートします。

合成済みデザインを開き、メモリに読み込むプロセスの一部として、レガシ ネットリスト プリミティブがサポートされる Unisim プリミティブのサブセットに変換されます。

デフォルトではレポートは標準出力に表示されますが、その後の処理用にファイルまたは Tcl 文字列変数に出力することもできます。

引数

-file arg (オプション) : 変換されたプリミティブのレポートを指定のファイルに保存します。 **-append** オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : **-append** コマンドは、**-file** オプションを使用している場合にのみ使用可能です。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを **-file** オプションと共に使用することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のデザインの変換されたプリミティブをレポートし、その結果を指定した Tcl 変数に返しています。

```
set transPrim [ report_transformed_primitives -return_string ]
```

report_utilization

デバイス使用率を算出し、レポートを表示します。

構文

```
report_utilization [-file arg] [-append] [-pblocks args] [-cells args]
[-return_string] [-packthru] [-name arg] [-no_primitives] [-omit_locs]
[-hierarchical] [-hierarchical_depth arg] [-quiet] [-verbose]
```

戻り値

レポート

使用法

名前	説明
[-file]	結果を保存するファイルの名前を指定します。このオプションを使用しない場合、出力はコンソールに表示されます。
[-append]	結果をファイルの最後に追加します。上書きはしません。
[-pblocks]	指定した Pblock のリストの使用率をレポートします。
[-cells]	指定したセルのリストの使用率をレポートします。
[-return_string]	レポートを文字列として返します。
[-packthru]	パックスルーハウジングとしてのみ使用される LUT をレポートします。
[-name]	GUI パネルに表示する結果の名前を指定します。
[-no_primitives]	レポートからプリミティブのセクションを削除します。
[-omit_locs]	レポートから [Loced] 列を削除します。
[-hierarchical]	テキスト形式の階層レポートを生成します。
[-hierarchical_depth]	テキスト形式の階層レポートのレベルを指定します。デフォルトは 0 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

現在の合成済みデザインまたはインプリメント済みデザインによるターゲット デザインのリソース使用率をレポートします。レポートは、**-file**、**-return_string**、または **-name** を指定した場合を除き、標準出力に返されます。

リソース使用率は設計プロセスの初期段階でレポートすることができますが、合成からインプリメンテーションに進むにつれより正確なレポートが得られます。

引数

-file *arg* (オプション) : レポートを保存するファイルを指定します。-append オプションが指定されていなければ、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。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-append (オプション) : コマンドの出力を指定したファイルに上書きするのではなく追加します。

注記 : -append コマンドは、-file オプションを使用している場合にのみ使用可能です。

-pblocks (オプション) : デザインの 1 つまたは複数の Pblock で使用されるリソースをレポートします。

-cells (オプション) : 現在のデザインの 1 つまたは複数の階層セルで使用されるリソースをレポートします。

-return_string (オプション) : 出力を Tcl 文字列として返します。Tcl 文字列は、変数定義でキャプチャしたり、解析またはその他の処理が可能です。

注記 : このオプションを -file オプションと共に使用することはできません。

-packthru (オプション) : ルートスルーに使用される LUT をレポートします。これは、使用率レポートに「LUTs used exclusively as route-thrus」とレポートされます。

-name *arg* (オプション) : GUI に出力する結果の名前を指定します。

-no_primitives (オプション) : レポートからプリミティブ セクションを削除します。プリミティブ セクションでは、デバイスで使用されているロジック プリミティブの数とタイプがレポートされます。

-omit_locs (オプション) : レポートから [LOCed] 列を削除します。[LOCed] 列には、デバイスのファブリック上に配置されるロジック エレメントの量がレポートされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、デザインの Pblock で使用されるリソース使用率を、指定したファイルに記述しています。

```
report_utilization -pblocks [get_pblocks] -file C:/Data/FPGA_Design/pb_util.txt
```

関連項目

[delete_utilization_results](#)

report_values

指定した HDL オブジェクト (変数、信号、ワイヤ、またはレジスタ) の現在のシミュレーション値を表示します。

構文

```
report_values [-radix arg] [-quiet] [-verbose] [hdl_objects ...]
```

戻り値

HDL オブジェクトの名前と値

使用法

名前	説明
[-radix]	HDL オブジェクトの値の表示に使用する基底を指定します。有効な値は、default、dec、bin、oct、hex、unsigned、ascii です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hdl_objects]	レポートする HDL オブジェクトを指定します。デフォルトは report_objects [get_objects *] です。

カテゴリ

シミュレーション

説明

指定した HDL オブジェクトの現在のシミュレーション時間での値をレポートします。

HDL オブジェクトには、Verilog または VHDL テストベンチおよびソース ファイルで定義されている HDL 信号、変数、または定数が含まれます。HDL 信号には、Verilog の wire または reg エンティティ、および VHDL 信号が含まれます。HDL 変数には、Verilog の real、realtime、time、event などがあります。

HDL 定数には、Verilog のパラメーターおよび localparam、VHDL ジェネリックおよび定数が含まれます。HDL スコープは、Verilog のモジュール、ファンクション、タスク、プロセス、begin-end ブロックなど、HDL コードの宣言部分で定義されます。VHDL スコープには、エンティティ/アーキテクチャ定義、ファンクション、プロシージャ、およびプロセス ブロックが含まれます。

引数

-radix arg (オプション) : 指定したオブジェクトの値を表示するのに使用する基底を指定します。有効な値は、default、dec、bin、oct、hex、unsigned、ascii です。

注記 : **dec** は、符号付き 10 進数を示します。符号なしデータの場合は、**unsigned** を指定してください。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

hdl_objects (必須) : 値を取得する 1 つまたは複数の HDL オブジェクトを指定します。オブジェクトは、名前または **get_objects** コマンドで指定します。

例

次の例では、現在の時間におけるすべてのオブジェクトの値をレポートしています。

```
report_values [get_objects]
```

次の例では、指定したバスから返された値の基數 **bin**、**dec**、および **unsigned** による違いを示しています。

```
report_values -radix bin /test/bench_VStatus_pad_0_i[7:0]
Declared: {/test/bench_VStatus_pad_0_i[7:0]} Verilog 10100101
report_values -radix unsigned /test/bench_VStatus_pad_0_i[7:0]
Declared: {/test/bench_VStatus_pad_0_i[7:0]} Verilog 165
report_values -radix dec /test/bench_VStatus_pad_0_i[7:0]
Declared: {/test/bench_VStatus_pad_0_i[7:0]} Verilog -91
```

関連項目

- [current_time](#)
- [get_objects](#)
- [get_value](#)
- [set_value](#)

reset_default_switching_activity

デフォルト タイプのスイッチング アクティビティをリセットします。

構文

```
reset_default_switching_activity [-static_probability] [-toggle_rate] [-quiet]
[-verbose] type...
```

戻り値

なし

使用法

名前	説明
[-static_probability]	スタティック確率をリセットします。
[-toggle_rate]	トグル レートをリセットします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>type</i>	ベクターなしの伝搬エンジンで指定したタイプのデフォルトシード値を、ツールのデフォルトにリセットします。有効なデフォルト タイプの値は、input、input_set、input_reset、input_enable、register、dsp、bram_read_enable、bram_write_enable、output_enable、clock、all です。

カテゴリ

電力

説明

デザインに含まれるネット、ポート、ピン、およびセルのデフォルト スイッチング アクティビティの属性をリセットします。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-static_probability (オプション)：指定のタイプのスタティック確率をリセットします。

-toggle_rate (オプション)：指定のタイプのトグル レートをリセットします。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記: メッセージの制限は、**set_msg_config** コマンドで定義できます。

type : リセットするタイプを指定します。有効な値は、input、input_set、input_reset、input_enable、register、dsp、bram_read_enable、bram_write_enable、output_enable、clock、all です。

例

次の例では、すべてのデザイン出力ポートのトグル レートおよびスタティック確率をリセットしています。

```
reset_default_switching_activity -toggle_rate -static_probability all
```

関連項目

- [power_opt_design](#)
- [report_default_switching_activity](#)
- [report_power](#)
- [report_default_switching_activity](#)
- [report_switching_activity](#)
- [reset_switching_activity](#)
- [set_default_switching_activity](#)
- [set_switching_activity](#)

reset_drc

DRC レポートを消去します。

構文

```
reset_drc [-name arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-name]	削除する DRC 結果の名前を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

[DRC、レポート](#)

説明

指定の結果セットから DRC 結果を消去します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (オプション) : 消去する DRC 結果の名前を指定します。名前は、**report_drc** コマンドの **-name** オプションで指定されたものです。

例

次の例では、メモリおよび GUI から指定した結果を消去しています。

```
reset_drc DRC1
```

関連項目

[report_drc](#)

reset_drc_check

1 つまたは複数の DRC チェックをデフォルトにリセットします。

構文

```
reset_drc_check [-quiet] [-verbose] [rules...]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[rules]	リセットするチェックのリストを指定します。

カテゴリ

DRC、オブジェクト

説明

指定の DRC チェックを Vivado Design Suite のデフォルトにリセットします。DRC チェックが、IS_ENABLED、SEVERITY プロパティも含め、デフォルト設定に戻ります。

DRC チェックの IS_ENABLED プロパティを変更すると、そのチェックが **report_drc** で直接またはルール デックの一部として指定されている場合でも、実行されないようディスエーブルにできます。

SEVERITY プロパティは文字列プロパティで、**report_drc** コマンドで違反が検出された場合にその DRC ルールに関連付けられている重要度を変更できます。有効な値は、FATAL、ERROR、"CRITICAL_WARNING"、WARNING、ADVISORY です。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

rules (必須) : デフォルト設定に戻す DRC ルールのリストを指定します。

例

次の例では、ROAS-1 ルールの IS_ENABLED プロパティを変更し、RFFC-1 ルールの SEVERITY プロパティを変更した後、すべてのチェックをデフォルト設定に戻しています。

```
set_property IS_ENABLED false [get_drc_checks ROAS-1]
set_property SEVERITY "Critical Warning" [get_drc_checks RFFC-1]
reset_drc_check [get_drc_checks]
```

関連項目

- [add_drc_checks](#)
- [get_drc_checks](#)
- [report_drc](#)
- [set_property](#)

reset_hw_ilab

ハードウェア ILA の制御プロパティをデフォルト値にリセットします。

構文

```
reset_hw_ilab [-reset_compare_values arg] [-quiet] [-verbose] [hw_ilas...]
```

戻り値

なし

使用法

名前	説明
[-reset_compare_values]	関連のハードウェア プローブ比較値をリセットします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_ilas]	ハードウェア ILA オブジェクトのリストを指定します。デフォルトは、現在のハードウェア ILA です。

カテゴリ

ハードウェア

reset_hw_vio_activity

指定のハードウェア VIO オブジェクトに関連付けられているハードウェア プローブに対し、ハードウェア VIO の ACTIVITY_VALUE プロパティをリセットします。

構文

```
reset_hw_vio_activity [-quiet] [-verbose] [hw_vios...]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_vios]	ハードウェア VIO オブジェクトのリストを指定します。

カテゴリ

ハードウェア

reset_hw_vio_outputs

ハードウェア VIO コアの出力を初期値にリセットします。

構文

```
reset_hw_vio_outputs [-quiet] [-verbose] [hw_vios...]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_vios]	ハードウェア VIO オブジェクトのリストを指定します。

カテゴリ

ハードウェア

reset_msg_config

set_msg_config コマンドで定義されたメッセージ制御ルールをリセットまたは削除します。

構文

```
reset_msg_config [-string args] [-id arg] [-severity arg] [-limit] [-suppress]
[-count] [-default_severity] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-string]	選択した操作を指定の文字列を含むメッセージのみに適用します。デフォルトは空です。
[-id]	メッセージ ID を指定します。指定しない場合、すべてのメッセージ ID が指定されます。
[-severity]	選択した操作を指定の重要度のメッセージのみに適用します。
[-limit]	現在のプロジェクトで指定したメッセージの最大表示回数をリセットします。
[-suppress]	現在のプロジェクトで指定のメッセージの非表示を解除します。
[-count]	現在のプロジェクトで指定したメッセージのカウントをリセットします。メッセージが -limit 制御のために非表示になるのを、次にメッセージ カウントが指定の最大表示回数を超えるまで回避します。
[-default_severity]	現在のプロジェクトで指定のメッセージの重要度をデフォルト値にリセットします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

set_msg_config コマンドで設定したメッセージの最大表示回数またはメッセージの重要度を Vivado ツールのデフォルト設定に戻したり、非表示になっているメッセージの制限を解除します。

1 つの reset_msg_config コマンドで、1 つリセット操作のみを実行できます。1 つの reset_msg_config コマンドで複数のリセット操作を実行しようとすると、エラーが返されます。

reset_msg_config コマンドでリセットするメッセージは、-string、-id、および -severity で指定します。これらのすべてのオプションの指定に一致するメッセージのみがリセットされます。

注記：少なくとも 1 つを使用してコマンドを適用するメッセージを指定する必要があります。そうでないと、エラーが返されます。

メッセージに設定されている現在の制御ルールをレポートするには、**get_msg_config** コマンドを使用します。

引数

-string args (オプション)：指定した文字列を含むメッセージのみに選択した操作を適用します。文字列は中かっこ (()) で囲んで指定します。複数の文字列はスペースで区切って指定します。

```
{{Vivado} {All Programmable}}
```

注記：大文字/小文字が区別されます。

-id arg (オプション)：指定のメッセージ ID のメッセージをリセットします。メッセージ ID は、すべてのメッセージに含まれます。たとえば、「Common 17-54」、「Netlist 29-28」です。

注記：ワイルドカード (*) を指定すると、すべてのメッセージ ID のメッセージがリセットされます。

-severity arg：指定の重要度のメッセージをリセットします。次の 5 種類の重要度があります。

- **ERROR**：デザインの結果が使用不可となったり、ユーザーの操作がないと回避できないような問題が発生していることを示すエラー メッセージです。
- **{CRITICAL WARNING}**：入力や制約に適用されないものがあつたり、FPGA ファミリには適していないものがあることを示すクリティカル警告メッセージです。修正することが強く推奨されます。

注記：これは 2 単語の値なので、{} で囲む必要があります。

- **WARNING**：制約または仕様が指定どおりに適用されず、デザインが最適な結果にならない可能性を示す警告メッセージです。修正するかどうかは、ユーザーが判断します。
- **INFO**：STATUS メッセージと同じですが、重要度とメッセージ ID タグが含まれる点が異なります。INFO メッセージには、必要に応じてアンサー データベースで確認できるようにメッセージ ID が含まれます。
- **STATUS**：デザインの処理に関するプロセスおよびフィードバックのステータスを示すステータス メッセージです。STATUS メッセージには、メッセージ ID は含まれません。

-limit (オプション)：-string、-id、または -severity オプションの指定に一致するメッセージの最大表示回数をリセットします。

-suppress (オプション)：-string、-id、または -severity オプションの指定に一致するメッセージの制限を解除します。

-count (オプション)：-string、-id、または -severity オプションの指定に一致するメッセージのカウントをリセットします。

-default_severity (オプション)：-string、-id、または -severity オプションの指定に一致するメッセージの重要度をデフォルトにリセットします。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、指定の INFO メッセージの重要度をデフォルト値にリセットしています。

```
reset_msg_config -id "Common 17-81" -default_severity
```

次の例では、指定したメッセージ ID のメッセージの制限を解除しています。

```
reset_msg_config -id {HDL 9-1654} -suppress
```

関連項目

- [get_msg_config](#)
- [set_msg_config](#)

reset_msg_count

メッセージ カウントをリセットします。

構文

```
reset_msg_count [-quiet] [-verbose] id
```

戻り値

新しいメッセージ数

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>id</i>	リセットするメッセージ ID (「Common 17-99」など) を指定します。 「reset_msg_count *」と指定すると、すべてのカウンターがリセットされます。

カテゴリ

レポート

説明

指定したメッセージ ID のメッセージ カウントを 0 にリセットします。これにより、メッセージ カウンターが新しく開始します。最大表示回数に近づいている特定のメッセージのカウントをリセットしたり、またはすべてのメッセージのカウントをリセットしたりできます。

ツールで表示される各メッセージには、アプリケーション サブシステム コードとメッセージ識別子を含む固有のメッセージ ID が付けられています。次に、メッセージ ID の例を示します。

```
"Common 17-54"  
"Netlist 29-28"  
"Synth 8-3295"
```

特定のメッセージ ID の現在のカウントを取得するには、**get_msg_count** コマンドを使用します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記: メッセージの制限は、**set_msg_config** コマンドで定義できます。

-id (必須) : カウントを 0 にリセットするメッセージの ID を指定します。すべてのメッセージのカウントを 0 にリセットする場合は * を指定します。

例

次の例では、すべてのメッセージのカウントをリセットしています。

```
reset_msg_count *
```

関連項目

- [get_msg_count](#)
- [set_msg_config](#)

reset_msg_limit

メッセージの最大表示回数をリセットします。

構文

```
reset_msg_limit [-severity arg] [-id arg] [-quiet] [-verbose]
```

戻り値

新しいメッセージの最大表示回数

使用法

名前	説明
[-severity]	リセットするメッセージの重要度 (「ERROR」、「CRITICAL」、「WARNING」など) を指定します。-id と共に使用することはできません。デフォルトは ALL です。
[-id]	最大表示回数をリセットするメッセージの ID (「Common 17-99」など) を指定します。-severity と共に使用することはできません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

メッセージの最大表示回数をデフォルトに戻します。特定のメッセージ ID、特定の重要度、またはすべてのメッセージのデフォルトの最大表示回数を元に戻すことができます。

表示されるすべてのメッセージのデフォルトの最大表示回数は、4,294,967,295 です。

引数

-id *value* (オプション) : 指定した ID のメッセージの最大表示回数をリセットします。各メッセージには、独自の ID が含まれます。たとえば、「Common 17-54」、「Netlist 29-28」です。

-severity *value* (オプション) : メッセージの重要度を指定します。次の 5 種類の重要度があります。

- **ERROR** : デザインの結果が使用不可となったり、ユーザーの操作がないと回避できないような問題が発生していることを示すエラー メッセージです。
 - **{CRITICAL WARNING}** : 入力や制約に適用されないものがあったり、FPGA ファミリには適していないものがあることを示すクリティカル警告メッセージです。修正することが強く推奨されます。
- 注記** : これは 2 単語の値なので、{} または "" で囲む必要があります。
- **WARNING** : 制約または仕様が指定どおりに適用されず、デザインが最適な結果にならない可能性を示す警告メッセージです。修正するかどうかは、ユーザーが判断します。
 - **INFO** : STATUS メッセージと同じですが、重要度とメッセージ ID タグが含まれる点が異なります。INFO メッセージには、必要に応じてアンサー データベースで確認できるようにメッセージ ID が含まれます。
 - **STATUS** : デザインの処理に関するプロセスおよびフィードバックのステータスを示すステータス メッセージです。STATUS メッセージには、メッセージ ID は含まれません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、指定したメッセージ ID のメッセージの最大表示回数をリセットしています。

```
reset_msg_limit -id "Netlist 29-28"
```

関連項目

- [get_msg_config](#)
- [get_msg_count](#)
- [get_msg_limit](#)
- [reset_msg_config](#)
- [set_msg_config](#)
- [set_msg_limit](#)

reset_msg_severity

指定した ID のメッセージの重要度をリセットします。

構文

```
reset_msg_severity [-quiet] [-verbose] id
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>id</i>	リセットするメッセージ ID (「Common 17-99」など) を指定します。

カテゴリ

レポート

説明

指定したメッセージ ID の重要度をデフォルト設定に戻します。

set_msg_severity の後にこのコマンドを使用すると、特定のメッセージ ID の重要度を元のレベルに戻すことができます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

id (必須) : 重要度をリセットするメッセージの ID を指定します。

例

次の例では、メッセージ ID common-99 の重要度を元に戻しています。

```
reset_msg_severity common-99
```

次の例では、メッセージ ID Netlist-1129 の重要度を元に戻しています。

```
reset_msg_severity Netlist-1129
```

関連項目

[set_msg_severity](#)

reset_operating_conditions

消費電力予測の動作条件をツールのデフォルトにリセットします。

構文

```
reset_operating_conditions [-voltage args] [-grade] [-process]
[-junction_temp] [-ambient_temp] [-thetaja] [-thetasas] [-airflow] [-heatsink]
[-thetajb] [-board] [-board_temp] [-board_layers] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-voltage]	電圧値をリセットします。サポートされる電圧は、ファミリによって異なります。
[-grade]	温度グレードをリセットします。
[-process]	プロセスをリセットします。
[-junction_temp]	ジャンクション温度をリセットします。
[-ambient_temp]	周囲温度をリセットします。
[-thetaja]	ThetaJA をリセットします。
[-thetasas]	ThetaSA をリセットします。
[-airflow]	エアフローをリセットします。
[-heatsink]	ヒートシンクのサイズをリセットします。
[-thetajb]	ThetaJB をリセットします。
[-board]	ボード タイプをリセットします。
[-board_temp]	ボード温度をリセットします。
[-board_layers]	ボード レイヤーをリセットします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、電力

説明

指定の動作条件をデフォルト値にリセットします。動作条件を指定しない場合、すべての動作条件がデフォルト値にリセットされます。

動作条件は、`set_operating_conditions` コマンドを使用して設定できます。現在の値を確認するには、`report_operating_conditions` コマンドを使用します。デバイスの環境動作条件は、`report_power` コマンドによる消費電力解析では使用されますが、タイミング解析では使用されません。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-voltage args (オプション)：電源をデフォルト値にリセットします。電源およびそのデフォルト値は、デバイスファミリによって異なります。

-grade (オプション)：選択したデバイスの温度グレードをリセットします。デフォルト値は `commercial` です。

-process (オプション)：ターゲット デバイスの製造プロセスをリセットします。デフォルト値は `typical` です。

-junction_temp (オプション)：ターゲット デバイスのジャンクション温度をリセットします。デフォルト値は `auto` です。

-ambient_temp (オプション)：デザインの周囲温度をリセットします。デフォルト値は `default` です。

-thetaJA (オプション)：Theta-JA 熱抵抗をリセットします。デフォルト値は `auto` です。

-thetaSA (オプション)：Theta-SA 熱抵抗をリセットします。デフォルト値は `auto` です。

-airflow (オプション)：エアフロー (LFM (リニア フィート/分)) をリセットします。デフォルト値はデバイス ファミリによって異なります。

-heatsink (オプション)：ヒートシンクのプロファイルをリセットします。デフォルト値は `medium` です。

-thetaJB (オプション)：Theta-JB 熱抵抗をリセットします。デフォルト値は `auto` です。

-board (オプション)：モデリングに使用されるボードのサイズをリセットします。デフォルト値は `medium` です。

-board_temp (オプション)：ボード温度をデフォルト値にリセットします。

-board_layers (オプション)：モデリングに使用されるボードの層数をデフォルトの 12to15 にリセットします。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、`set_msg_config` コマンドで定義できます。

例

次の例では、デザインのすべての動作条件をデフォルト値にリセットしています。

```
reset_operating_conditions
```

次の例では、デザインのジャンクション温度、周囲温度、およびボード温度をデフォルト値にリセットしています。

```
reset_operating_conditions -junction_temp -ambient_temp -board_temp
```

次の例では、電源 `Vccint` をデフォルト値にリセットしています。

```
reset_operating_conditions -voltage Vccint
```

関連項目

- [report_operating_conditions](#)
- [report_power](#)
- [set_operating_conditions](#)

reset_param

パラメーターをリセットします。

構文

reset_param [-quiet] [-verbose] *name*

戻り値

元のパラメーター値

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	パラメータ名を指定します。

カテゴリ

プロパティおよびパラメーター

説明

set_param コマンドで変更したユーザー定義可能なコンフィギュレーション パラメーターをデフォルト値に戻します。

report_param コマンドを使用すると、現在定義されているパラメーターを表示できます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 値をリセットするパラメーターの名前を指定します。このコマンドでリセットできるのは、一度に 1 つのパラメーターのみです。

例

次の例では、tcl.statsThreshold パラメーターの値がデフォルト値にリセットされます。

```
reset_param tcl.statsThreshold
```

関連項目

- [get_param](#)
- [list_param](#)
- [report_param](#)
- [set_param](#)

reset_project

現在開いているプロジェクトをリセットします。

構文

```
reset_project [-exclude_runs] [-exclude_ip] [-exclude_sim_runs] [-quiet]
[-verbose]
```

戻り値

なし

使用法

名前	説明
[-exclude_runs]	run はリセットしません。
[-exclude_ip]	IP はリセットしません。
[-exclude_sim_runs]	シミュレーション実行はリセットしません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

説明

現在のプロジェクトを、合成、シミュレーション、インプリメンテーション、および **write_bitstream** で生成されたさまざまな出力ファイルをクリーンアップして、開始時のソースおよび制約ファイルのみの状態にリセットします。プロジェクトのステートもデザイン フローの開始にリセットします。

引数

-exclude_runs (オプション) : リセット プロセスから *project.runs* フォルダーを除外します。runs フォルダーはそのまま保持され、ほかのプロジェクト データが削除されます。

-exclude_ip (オプション) : リセット プロセスから *project.srcc/sources_1/ip* フォルダーを除外します。IP コアと生成されたターゲットを含む IP フォルダーはそのまま保持され、ほかのプロジェクト データが削除されます。

-exclude_sim_runs (オプション) : リセット プロセスから *project.sim* フォルダーを除外します。シミュレーション フォルダーはそのまま保持され、ほかのプロジェクト データが削除されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、シミュレーション run データを保持して現在のプロジェクトをリセットし、メッセージの数制限にかかわらずすべてのメッセージを返しています。

```
reset_project -exclude_sim_runs -verbose
```

関連項目

- [create_project](#)
- [current_project](#)

reset_property

オブジェクトのプロパティをリセットします。

構文

```
reset_property [-quiet] [-verbose] property_name objects...
```

戻り値

設定された値、エラーが発生した場合は “”

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>property_name</i>	リセットするプロパティの名前を指定します。
<i>objects</i>	プロパティをリセットするオブジェクトの名前を指定します。

カテゴリ

オブジェクト、プロパティおよびパラメーター

説明

指定のオブジェクトの指定のプロパティをデフォルト値にリセットします。プロパティにデフォルト値が定義されていない場合は、指定のオブジェクトにそのプロパティは割り当てられません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

property_name (必須) : リセットするプロパティの名前を指定します。

objects (必須) : プロパティをデフォルト値にリセットするオブジェクトを 1 つまたは複数指定します。

例

次の例では、すべてのセルの ALL_PROPS プロパティをリセットします。

```
reset_property ALL_PROPS [get_cells]
```

関連項目

- [create_property](#)
- [get_cells](#)
- [get_property](#)
- [list_property](#)
- [list_property_value](#)
- [report_property](#)
- [set_property](#)

reset_run

既存の run をリセットします。

構文

```
reset_run [-prev_step] [-from_step arg] [-noclean_dir] [-quiet] [-verbose] run
```

戻り値

なし

使用法

名前	説明
[-prev_step]	最後の run ステップをリセットします。
[-from_step]	リセットする最初のステップを指定します。
[-noclean_dir]	すべての出力ファイルおよびディレクトリをディスクから削除しません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
run	変更する run を指定します。

カテゴリ

プロジェクト

説明

指定した run をインプリメント前または合成前のステートにリセットします。このコマンドは、run をリセットして再実行できるようにするために使用します。

引数

-prev_step (オプション) : インプリメンテーション run を最後に完了したステップからリセットします。このオプションを使用すると、**launch_runs -to_step** コマンドを使用して実行した部分的に完了しているインプリメンテーション run をリセットできます。

-from_step arg (オプション) : インプリメンテーション run を指定のステップからリセットします。このオプションを使用すると、**launch_runs -next_step** コマンドを使用して run を指定のステップから再開できます。有効なステップは、次のとおりです。

- **opt_design** : ターゲット デバイスのリソースがより効率的に使用されるように、論理デザインを最適化します (オプション)。
- **power_opt_design** : インプリメント済み FPGA の消費電力を削減するため、ロジック デザインのエレメントを最適化します (オプション)。
- **place_design** : ロジック セルをターゲット デバイスに配置します。この段階は必須です。
- **power_opt_design (Post-Place)** : 配置済みロジック エレメントの消費電力を最適化します (オプション)。
- **phys_opt_design** : ファンアウトの大きいネットのドライバーを複製してロードを分散することにより、デザインのタイミングを最適化します (オプション)。
- **route_design** : デザインの接続をターゲット FPGA 上に配線します。この段階は必須です。
- **write_bitstream** : ザイリンクス デバイス コンフィギュレーションのビットストリーム ファイルを生成します。この段階は必須です。

-noclean_dir (オプション) : run ディレクトリのファイルを削除しません。run をリセットすると、その run を再インプリメントする際に新たに開始するため、デフォルトでは run ディレクトリとそのファイルがすべて削除されます。このオプションを使用すると、run をリセットしたときにその run のディレクトリとファイルは削除されなくなります。この場合、run を再インプリメントすると、プロジェクトの run ディレクトリに新しい run ディレクトリが作成されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : リセットする run の名前を指定します。

例

次の例では、インプリメンテーション run がリセットされます。

```
reset_run impl_1
```

注記 : **-noclean_dir** オプションが指定されていないので、run ディレクトリとその内容がハード ディスクから削除されます。

次の例では、合成 run をリセットしていますが、run ディレクトリは削除されません。

```
reset_run -noclean_dir synth_1
```

この場合、**-noclean_dir** オプションが指定されているので、synth_1 run ディレクトリは削除されず、run を実行すると、synth_1_2 という新しい run ディレクトリが作成されます。

関連項目

- [create_run](#)
- [launch_runs](#)
- [opt_design](#)
- [place_design](#)
- [route_design](#)

reset_simulation

既存のシミュレーション実行をリセットします。

構文

```
reset_simulation [-mode arg] [-type arg] [-quiet] [-verbose] [simset]
```

戻り値

なし

使用法

名前	説明
[-mode]	指定したモードの生成データを削除します。有効な値は behavioral、post-synthesis、post-implementation で、デフォルトは behavioral です。
[-type]	指定したタイプの生成データを削除します。このオプションは、-mode を post-synthesis または post-implementation に設定している場合にのみ有効です。有効な値は functional、timing です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[simset]	リセットするシミュレーション ファイルセットの名前を指定します。

カテゴリ

シミュレーション

reset_ssn

メモリから SSN 結果を消去します。

構文

```
reset_ssn [-quiet] [-verbose] name
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	結果のセット名を指定します。

カテゴリ

レポート

説明

指定の結果セットから SSN 結果を消去します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 消去する SSN 結果の名前を指定します。

例

次の例では、メモリから指定した結果を消去しています。

```
reset_ssn SSN1
```

関連項目

[report_ssn](#)

reset_switching_activity

指定したオブジェクトのスイッチング アクティビティをリセットします。

構文

```
reset_switching_activity [-static_probability] [-signal_rate] [-hier] [-quiet]
[-verbose] [objects ...]
```

戻り値

なし

使用法

名前	説明
[-static_probability]	スタティック確率をリセットします。
[-signal_rate]	信号レートをリセットします。
[-hier]	objects で指定した階層セルのすべてのレベルでスイッチング アクティビティをリセットします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	スイッチング アクティビティをリセットするオブジェクトを指定します。

カテゴリ

電力

説明

デザインに含まれるネット、ポート、ピン、およびセルのスイッチング アクティビティの属性をリセットします。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

スイッチング アクティビティを定義するには、**set_switching_activity** コマンドを使用します。特定のポート、ピン、ネット、またはセルの現在定義されているスイッチング アクティビティを確認するには、**report_switching_activity** コマンドを使用します。

引数

-static_probability (オプション)：指定のオブジェクトのスタティック確率をリセットします。

-signal_rate (オプション)：指定のオブジェクトの信号レートをリセットします。

-hier (オプション)：階層オブジェクトのすべてのレベルでスイッチング アクティビティをリセットします。**-hier** を使用しない場合、現在の階層レベルで指定の *objects* のスイッチング アクティビティがリセットされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (オプション) : スイッチング アクティビティをリセットするオブジェクトのリストを指定します。オブジェクトのリストを指定しない場合、すべてのオブジェクトのスイッチング アクティビティがリセットされます。

例

次の例では、すべての出力ポートの信号レートおよびスタティック確率をリセットしています。

```
reset_switching_activity -signal_rate -static_probability [all_outputs]
```

関連項目

- [power_opt_design](#)
- [report_default_switching_activity](#)
- [report_power](#)
- [report_switching_activity](#)
- [reset_default_switching_activity](#)
- [set_default_switching_activity](#)
- [set_switching_activity](#)

reset_target

指定したソースのターゲット データをリセットします。

構文

```
reset_target [-quiet] [-verbose] name objects
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	リセットするターゲットのリストを指定します。all を指定すると、生成されたターゲットがすべてリセットされます。
<i>objects</i>	データをリセットするオブジェクトを指定します。

カテゴリ

プロジェクト、XPS、IP フロー

説明

指定した IP コアのターゲット データを削除します。これにより、指定したターゲットの生成中に生成されたファイルが削除されます。コアは現在のプロジェクトから削除されませんが、参照場所にある関連のターゲット データが削除されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : リセットするターゲット タイプの名前を指定します。有効な値は次のとおりです。

- **all** : 指定のコアのすべてのターゲットをリセットします。
- **synthesis** : 指定したコアの合成ネットリストをリセットします。指定したコアのネットリスト ファイルが削除されます。
- **simulation** : 指定したコアのシミュレーション ネットリストをリセットします。
- **instantiation_template** : 指定したコアのインスタンシエーション テンプレートをリセットします。

objects (必須) : ターゲット データを削除する IP コア オブジェクトを指定します。

例

次の例では、指定した IP コアのインスタンシエーション テンプレートをリセットしています。

```
reset_target instantiation_template [get_ips blk_mem*]
```

関連項目

[generate_target](#)

reset_timing

現在のデザインのタイミング情報をリセットします。

構文

```
reset_timing [-invalid] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-invalid]	無効なタイミング制約もリセットします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート、タイミング

説明

現在のデザインのタイミング データをリセットします。現在メモリにあるタイミング データを消去し、タイミング エンジンでデザインを反復的にではなく包括的に再評価する場合に使用します。

注記：メモリのタイミング データは削除されますが、タイミング レポートは削除されません。レポートされたタイミング情報を削除するには、**delete_timing_results** コマンドを使用します。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、メモリから現在のタイミング データを消去しています。

```
reset_timing
```

関連項目

- [delete_timing_results](#)
- [report_timing](#)

reset_ucf

ファイルから読み込んだフロアプラン制約を消去します。

構文

```
reset_ucf [-quiet] [-verbose] file
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	リセットする UCF ファイルを指定します。

カテゴリ

フロアプラン

説明

現在のデザインから指定した UCF 制約ファイルで定義されている配置制約を削除します。このコマンドを実行するには、デザインを開いている必要があります。

指定したファイルに含まれる制約は現在のデザインから削除されますが、この変更はすぐに制約ファイルに保存されるわけではありません。制約ファイルは、**save_design** コマンドを使用したときにアップデートされます。

デザインを保存せずに制約を新しいファイルに保存するには、**write_ucf** コマンドを使用します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : リセットする UCF ファイルを指定します。

注記 : 制約ファイル セットから指定のファイルが検されます。

例

次の例では、指定したファイルに含まれる配置制約を削除しています。

```
reset_ucf top_full.ucf
```

resize_net_bus

現在のデザインに含まれるネット バスのサイズを変更します。

構文

```
resize_net_bus [-from arg] [-to arg] [-quiet] [-verbose] net_bus_name ...
```

戻り値

なし

使用法

名前	説明
[-from]	新しい開始バス インデックスを指定します。
[-to]	新しい終了バス インデックスを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>net_bus_name</i>	サイズを変更するネット バスの名前を指定します。

カテゴリ

ネットリスト

説明

既存のバス ネットのサイズを変更するか (拡大または縮小)、現在のインデックス範囲の番号を付け直します。1つのコマンドでは、拡大、縮小、または番号の付け直しのいずれかのみを実行可能です。

- バスを拡大するには、現在のインデックス範囲を超える新しいインデックス範囲を指定します。接続されている既存のビットは、そのまま保持されます。
- バスを縮小するには、現在のインデックス範囲未満の新しいインデックス範囲を指定します。ビットを削除するため接続が解除されますが、残りのビットは現在の接続が保持されます。
- 現在のバス インデックスの番号を付け直すには、現在の範囲と同じ幅の新しいインデックス範囲を指定します。バス ビットの番号は変更されますが、接続は変更されません。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソース ファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザイン チェックポイントとして保存するか、**write_*** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリストファイルにエクスポートできます。

注記： エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

このコマンドが正常に実行された場合は何も返されず、正常に実行できない場合はエラーが返されます。

引数

-from arg (オプション) : 指定のバス ネットの新しい開始インデックスを指定します。

-to arg (オプション) : 指定のバスの新しい終了インデックスを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

net_bus_name (必須) : 既存のバスネットの名前を指定します。

例

次の例では、新しい 24 ビットバスを作成し、負のインデックスを含むようにバス インデックスの番号を付け直し、その後 8 ビットバスにサイズを変更しています。

```
create_net tempBus -from 23 -to 0
resize_net_bus tempBus -from -12 -to 11
resize_net_bus tempBus -from 0 -to 7
```

関連項目

- [connect_net](#)
- [create_net](#)
- [create_pin](#)
- [create_port](#)
- [disconnect_net](#)
- [get_nets](#)
- [remove_net](#)
- [resize_pin_bus](#)
- [resize_port_bus](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

resize_pblock

サイト範囲制約を移動、サイズ変更、追加、削除します。

構文

```
resize_pblock [-add args] [-remove args] [-from args] [-to args] [-replace]
[-locs arg] [-quiet] [-verbose] pblock
```

戻り値

なし

使用法

名前	説明
[-add]	サイト範囲を追加します。
[-remove]	サイト範囲を削除します。
[-from]	移動するサイト範囲を指定します。
[-to]	サイト範囲の移動先を指定します。
[-replace]	既存の範囲をすべて削除します。
[-locs]	ロックの処理方法を指定します。デフォルトは keep_all です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>pblock</i>	サイズを変更する Pblock を指定します。

カテゴリ

フロアプラン、XDC

説明

指定した Pblock を配置、サイズ変更、移動、削除します。Pblock は、`create_pblock` コマンドを使用して作成されている必要があります。

Pblock は、独立したまたはオーバーラップした 1 つまたは複数の矩形で定義されたセルのグループで構成されます。次に定義されているさまざまなオプションを使用して、矩形にサイトを追加したり、矩形からサイトを削除したり、既存の Pblock に関連付ける新しい矩形を定義したりできます。

引数

-add arg (オプション) : Pblock に指定の範囲のサイトを追加します。スライス範囲は、矩形の 1 つの角からその対角線上の角までで指定します。たとえば、「SLICE_X0Y0:SLICE_X20Y12」のように指定します。

注記 : 複数のサイトタイプは、個別の矩形として追加します。

-remove arg (オプション) : Pblock から指定の範囲のサイトを削除します。Pblock からサイトを削除すると、Pblock を 1 つまたは複数の矩形で定義するという要件に従うため、Pblock が複数の矩形に分割される場合があります。

-from args (オプション) : **-from** と **-to** オプションはペアとして指定し、1 つの場所から別の場所に移動するサイトまたはサイト範囲を指定します。

-to args (オプション) : **-from** と **-to** オプションはペアとして指定し、1 つの場所から別の場所に移動するサイトまたはサイト範囲を指定します。

-locs args (オプション) : Pblock を移動またはサイズ変更したときに、Pblock の配置されているロジックをどのように処理するかを指定します。有効な値は次のとおりです。

- **keep_all** : すべての配置を現在設定されているとおり保持します。**-locs** を指定しない場合、これがデフォルトです。Pblock の外に配置されているロジックは、Pblock に割り当てられなくなります。
- **keep_only_fixed** : ユーザーが配置したロジック (固定) のみを保持します。自動的に配置された未固定のロジックは配置が解除されます。
- **keep_none** : すべてのロジックの配置を解除します。
- **move** : すべての配置を Pblock の座標に相対して移動します。
- **move_unfixed** : 未固定のエレメントのみを移動するよう指定します。ユーザーが配置した (固定) ロジックは移動しません。
- **trim** : 新しい Pblock の境界外となるロジックの配置を解除します。配置されているロジックで Pblock の境界内になるものは、配置が保持されます。
- **trim_unfixed** : 新しい Pblock の境界外となる未固定のエレメントのみの配置を解除します。

-replace (オプション) : Pblock に関連付けられているすべての矩形を削除します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

pblock (必須) : サイズ変更、移動、または削除する Pblock の名前を指定します。

例

次の例では、Pblock にスライス範囲を追加し、別のスライスを削除して、Pblock のサイズを変更しています。すべてのインスタンスの配置は変更しません。

```
resize_pblock block3 -add SLICE_X6Y67:SLICE_X11Y71 -remove SLICE_X6Y71:SLICE_X7Y71 \
-locs keep_all
```

次の例では、指定した Pblock にスライス範囲を追加し、既存のスライス範囲を削除して、新しい Pblock の境界外となるロジックを削除しています。その後、新しいスライス範囲とブロック RAM を別の矩形として追加します。

```
resize_pblock block3 -add SLICE_X3Y8:SLICE_X10Y3 -remove SLICE_X6Y67:SLICE_X11Y71 \
-locs trim
resize_pblock block3 -add {SLICE_X6Y67:SLICE_X11Y71 RAMB18_X0Y2:RAMB18_X1Y4}
```

関連項目

- [add_cells_to_pblock](#)
- [create_pblock](#)
- [place_pblocks](#)

resize_pin_bus

現在のデザインに含まれるピン バスのサイズを変更します。

構文

```
resize_pin_bus [-from arg] [-to arg] [-quiet] [-verbose] pin_bus_name ...
```

戻り値

なし

使用法

名前	説明
[-from]	新しい開始バス インデックスを指定します。
[-to]	新しい終了バス インデックスを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>pin_bus_name</i>	サイズを変更するピン バスの名前を指定します。

カテゴリ

ネットリスト

説明

既存のバスピンのサイズを変更するか(拡大または縮小)、現在のピン インデックス範囲の番号を付け直します。1 つのコマンドでは、拡大、縮小、または番号の付け直しのいずれかのみを実行可能です。

- バスを拡大するには、現在のピン インデックス範囲を超える新しいインデックス範囲を指定します。接続されている既存のピンは、そのまま保持されます。
- バスを縮小するには、現在のピン インデックス範囲未満の新しいインデックス範囲を指定します。バスピンを削除するため接続が解除されますが、残りのピンは現在の接続が保持されます。
- 現在のバス インデックスの番号を付け直すには、現在の範囲と同じ幅の新しいピン インデックス範囲を指定します。ピン インデックスの番号は変更されますが、接続は変更されません。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソース ファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザイン チェックポイントとして保存するか、**write_*** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリストファイルにエクスポートできます。

注記：エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

このコマンドが正常に実行された場合は何も返されず、正常に実行できない場合はエラーが返されます。

引数

-from arg (オプション) : 指定のバス ピンの新しい開始インデックスを指定します。

-to arg (オプション) : 指定のバス ピンの新しい終了インデックスを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

bus_pin_name (必須) : 変更するバス ピンの名前を指定します。ピン名は、ピンを割り当てるセル インスタンスを基準に階層で指定する必要があります。デザインの最上位に作成されたピンはポートであり、**resize_port_bus** コマンドを使用してサイズを変更する必要があります。

例

次の例では、ブラック ボックス セルを作成し、指定の階層セルに 24 ビットの双方向バスを作成し、バス ピンの幅を 32 ビットに拡張して、負のバス インデックスを含むように番号を付け直しています。

```
create_cell -reference dmaBlock -black_box usbEngine0/myDMA
create_pin -direction INOUT -from 0 -to 23 usbEngine0/myDMA/dataBus
resize_pin_bus -from 0 -to 31 usbEngine0/myDMA/dataBus
resize_pin_bus -from -16 -to 15 usbEngine0/myDMA/dataBus
```

関連項目

- [create_net](#)
- [create_pin](#)
- [create_port](#)
- [get_pins](#)
- [remove_pin](#)
- [resize_net_bus](#)
- [resize_port_bus](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

resize_port_bus

現在のデザインに含まれるポートバスのサイズを変更します。

構文

```
resize_port_bus [-from arg] [-to arg] [-quiet] [-verbose] port_bus_name ...
```

戻り値

なし

使用法

名前	説明
[-from]	新しい開始バス インデックスを指定します。
[-to]	新しい終了バス インデックスを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>port_bus_name</i>	サイズを変更するポートバスの名前を指定します。

カテゴリ

ピン配置

説明

既存のバスポートのサイズを変更するか(拡大または縮小)、現在のポートインデックス範囲の番号を付け直します。1つのコマンドでは、拡大、縮小、または番号の付け直しのいずれかのみを実行可能です。

- バスを拡大するには、現在のポートインデックス範囲を超える新しいインデックス範囲を指定します。接続されている既存のポートは、そのまま保持されます。
- バスを縮小するには、現在のポートインデックス範囲未満の新しいインデックス範囲を指定します。バスポートを削除するため接続が解除されますが、残りのポートは現在の接続が保持されます。
- 現在のバスインデックスの番号を付け直すには、現在の範囲と同じ幅の新しいポートインデックス範囲を指定します。ポートインデックスの番号は変更されますが、接続は変更されません。

ネットリストを編集すると、現在のデザインのネットリストのメモリにあるビューが変更されます。ソースファイルセットのファイルおよびディスク上のデザインは変更されません。ネットリストに加えた変更は、**write_checkpoint** コマンドを使用してデザインチェックポイントとして保存するか、**write_*** コマンドを使用して Verilog、VHDL、または EDIF 形式のネットリストファイルにエクスポートできます。

注記：エラボレート済み RTL デザインでは、ネットリストを編集することはできません。

このコマンドが正常に実行された場合は何も返されず、正常に実行できない場合はエラーが返されます。

引数

-from arg (オプション) : 指定のバス ポートの新しい開始インデックスを指定します。

-to arg (オプション) : 指定のバス ポートの新しい終了インデックスを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

bus_port_name (必須) : 変更するバス ポートの名前を指定します。

例

次の例では、32 ビットの出力バス ポートを作成し、負のバス インデックスを含めるためポートの番号を付け直し、バス幅を 32 ビットから 16 ビットに縮小します。

```
create_port -direction out -from 0 -to 31 outPorts
resize_port_bus -from -16 -to 15 outPorts
resize_port_bus -from -8 -to 7 outPorts
```

関連項目

- [create_net](#)
- [create_pin](#)
- [create_port](#)
- [get_ports](#)
- [remove_port](#)
- [resize_net_bus](#)
- [resize_pin_bus](#)
- [write_checkpoint](#)
- [write_edif](#)
- [write_verilog](#)
- [write_vhdl](#)

restart

シミュレーションをデザインを読み込んだ後の状態に戻し、時間を 0 に設定します。

構文

```
restart [-quiet] [-verbose]
```

戻り値

ブレークポイント、Tcl フォース、波形ビューアーの設定は保持し、その他の Tcl コマンドの結果をすべてクリーンアップ

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

シミュレーション

route_design

現在のデザインを配線します。

構文

```
route_design [-unroute] [-re_entrant arg] [-nets args] [-physical_nets]
[-pin arg] [-directive arg] [-no_timing_driven] [-preserve] [-delay]
[-free_resource_mode] -max_delay arg -min_delay arg [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-unroute]	デザイン全体の配線を解除するか、-nets または -pin を指定した場合は指定のネット/ピンの配線を解除します。
[-re_entrant]	on に設定すると再配線モードがオンになります。デフォルトでは、再配線モードはオフです。-pin または -nets を使用している場合は不要です。これらのオプションでは、自動的に再配線モードが使用されます。デフォルトは off です。
[-nets]	指定したネットに対して操作を実行します。
[-physical_nets]	すべての物理ネットに対して操作を実行します。
[-pin]	指定したピンに対して操作を実行します。
[-directive]	コマンドのモードを指定します。有効な値は、「引数」セクションを参照してください。デフォルトは Default です。
[-no_timing_driven]	タイミング ドリブン モードで実行しません。
[-preserve]	既存の配線を保持します。
[-delay]	-nets または -pin オプションと共に使用し、遅延ドリブン モードで配線します。
[-free_resource_mode]	配線ツールをフリー リソース モードで実行します。
-max_delay	-pin オプションと共に使用し、ピンに最大遅延制約を指定します。このオプションを指定すると、-delay も指定されます。
-min_delay	-pin オプションと共に使用し、ピンに最大遅延制約を指定します。このオプションを指定すると、-delay も指定されます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール

説明

現在のデザインに含まれるネットを配線し、ターゲット パーツでのロジック接続を完成させます。

定義済みの配線ストラテジを **route_design -directive** コマンドを使用して選択するか、または配線オプションを設定して独自の配線ストラテジを定義できます。

配線は、**route_design** コマンドを使用してすべて自動的に実行するか、さまざまなオプションを使用して反復的に完成させることができます。配線を反復的に実行すると、まずクリティカルなネットを配線してからそれほどクリティカルでないネットを配線するなどして配線プロセスを制御したり、エフォートレベルやタイミング アルゴリズムを調整したりできます。

配線はインプリメンテーションの 1 つの段階です。Vivado ツールをプロジェクトモードで実行している場合は、インプリメンテーションは **launch_runs** コマンドを使用してすべてを自動的に実行できます。

非プロジェクト モードでは、インプリメンテーション プロセスを **opt_design**、**place_design**、**phys_opt_design**、**power_opt_design**、および **route_design** の各コマンドを使用して手動で実行する必要があります。プロジェクトモードおよび非プロジェクトモードの詳細は、『Vivado Design Suite ユーザー ガイド：デザインフローの概要』(UG892) を参照してください。

配置および配線はどちらも、インクリメンタル コンパイル フローを使用して、デザイン チェックポイント ファイル (DCP) に保存されている結果に基づいて、インクリメンタルに実行できます。インクリメンタル配置および配線の詳細は、**read_checkpoint** コマンドまたは『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

このコマンドを実行するには、配置済みデザインが必要であり、**opt_design** コマンドを実行しておくことをお勧めします。

引数

-unroute arg(オプション)：デザインのネットの配線を解除します。引数が指定されていない場合は、すべてのネットの配線が解除されます。**-unroute** オプションを指定した場合、**route_design** コマンドでネットは配線されません。

- **-nets** オプションと共に使用すると、配線を解除するネットのリストを指定できます。
- **-pin** オプションと共に使用すると、指定のピンから最も近いネットの分岐までの配線が解除されます。
- **-physical_nets** オプションと共に使用すると、論理 1 および論理 0 のネットの配線を解除できます。

-re_entant [on | off](オプション)：配線を再配線モードで実行します。デフォルト値は **off** です。**route_design** コマンドを **-re_entant off** を指定して実行した場合、または配線をデフォルト モードで実行した場合は、配線の完了後にメモリ内のデータ構造がクリーンアップされます。**route_design** を **-re_entant on** を指定して実行すると、メモリ内の配線データが保持され、次回の配線の実行時間が短縮されます。配線を繰り返し実行する場合に再配線モードをオンにし、配線実行のたびに初期化が実行されないようにします。

注記： **-re_entant** オプションは、**-nets** または **-pin** オプションを指定している場合は不要です。これらのオプションを使用すると自動的に再配線モードになります。

-directive arg (オプション) : 特定のデザイン目標を達成するよう配線します。1 つの **route_design** コマンドに対して 1 つのモードのみを指定できます。値では大文字/小文字が区別されます。有効な値は次のとおりです。

- **Explore** : 初期配線の後、異なるクリティカル パス配線を試します。
- **NoTimingRelaxation** : 配線を完了するためにタイミングを緩和しないようにします。配線でタイミング満たすのが困難である場合、元のタイミング制約を満たすため実行時間が長くなります。
- **MoreGlobalIterations** : 最終段階だけでなく、すべての段階で詳細なタイミング解析を使用し、タイミングの向上が少しであってもグローバル反復を実行します。
- **HigherDelayCost** : 配線の内部コスト関数を調整して反復実行で遅延に焦点を置き、実行時間が長くなる代わりにパフォーマンスを向上します。
- **AdvancedSkewModeling** : 配線のすべての段階でより正確なスキー モデルを使用します。スキーの大きいクロック ネットワークでデザインのパフォーマンスが向上する場合があります。
- **RuntimeOptimized** : 反復回数を少なくし、デザイン パフォーマンスよりも実行時間を短縮することを優先します。
- **Quick** : 最も高速な、タイミング ドリブンでない、有効なデザインを得るために最低限の配線を実行します。
- **Default** : **route_design** をデフォルト設定で実行します。

各モードの効果は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

注記 : **-directive** オプションは全体的な配線ストラテジを制御するもので、特定の配線オプションとは互換性がありません。**-quiet** および **-verbose** とのみ使用できます。また、**read_checkpoint -incremental** を使用してインクリメンタル コンパイル フローが指定されている場合は、**-directive** オプションは無視されます。

-physical_nets (オプション) : 論理 0 または論理 1 のネットのみを配線または配線解除します。

-nets args (オプション) : 指定したネット オブジェクトのみを配線または配線解除します。ネット オブジェクトは、**get_nets** コマンドを使用して指定する必要があります。

注記 : **-nets** または **-pin** オプションを指定すると、指定のネットに対して配線ソリューションを見つけるのにクイック配線が使用され、タイミング 遅延は無視されます。遅延が最も少ない配線を見つける際に **-delay** を使用してください。

-pin arg (オプション) : 指定したピンのみを配線または配線解除します。セル入力を指定する必要があります。ピンが複数のファンアウト ネットで駆動されている場合、ネットとピンの間の配線セグメントのみに操作が実行されます。

注記 : **-nets** または **-pin** オプションを指定すると、指定のネットに対して配線ソリューションを見つけるのにクイック配線が使用され、タイミング 遅延は無視されます。遅延が最も少ない配線を見つける際に **-delay** を使用してください。

-delay (オプション) : **-nets** または **-pin** オプションを指定している場合にのみ使用できます。デフォルトでは、**-nets** または **-pin** オプションを指定すると、ネットの配線は配線ランタイムが最短になるようにタイミング制約を無視して実行されます。**-delay** オプションを指定すると、配線インターフェクト 遅延が最小になるように配線されますが、タイミング制約は無視されます。

-max_delay arg (オプション) : **-pin** オプションを使用している場合にのみ有効で、遅延が指定の遅延 (ps) 以下になるよう配線します。このオプションを指定すると、**-delay** オプションも指定されます。

-min_delay arg (オプション) : **-pin** オプションを使用している場合にのみ有効で、遅延が指定の遅延 (ps) 以上になるよう配線します。このオプションを指定すると、**-delay** オプションも指定されます。

-no_timing_driven (オプション) : デフォルトのタイミングドリブン配線アルゴリズムをディスエーブルにします。このオプションを使用すると短時間で配線結果を得ることができますが、タイミング制約は無視されます。

-preserve (オプション) : 既存の配線を保持します。IS_ROUTE_FIXED または FIXED_ROUTE プロパティを使用して固定されている配線には適用されません。配線は、現在の配線セッションでのみ保持されます。

-free_resource_mode (オプション) : 配線リソースを制御するアドバンス オプションです。Vivado 配線では、デフォルトでは、初期のネット配線が配線ソリューションを高速化するため同じ配線リソースを使用してオーバーラップしている場合があります。これらのオーバーラップは、接続を完了させるために後で解決されます。このオプションを指定すると、この初期のオーバーラップは許容されず、新しい配線に使用可能な配線リソースのみが使用されます。このオプションを後半の配線実行で使用し、新しい配線がオーバーラップしたり完成した配線に影響するのを防ぐことができます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、デザイン全体を配線し、クリティカルパス遅延を向上するために複数のアルゴリズムを試しています。

```
route_design -directive Explore
```

次の例では、インターフェクト遅延が最小になるように指定のタイミング クリティカル ネットを配線した後、その配線を保持してデザインの残りの部分をエフォート レベルを low に設定して配線しています。

```
route_design -delay -nets $myCriticalNets
route_design -preserve -directive RuntimeOptimized
```

次の例では、再配線をオンにし、現在のデザインの配線を解除した後、デフォルト設定を使用して配線を再実行しています。

```
route_design -re_entrant on
route_design -unroute
route_design
```

次の例では、指定のネットを最短のランタイムで配線しています。

```
route_design -nets [get_nets ctrl0/ctr*]
```

次の例では、指定のネットをインターフェクト遅延が最短になるように配線しています。

```
route_design -nets [get_nets ctrl0/ctr*] -delay
```

次の例では、指定のピンを配線しています。

```
route_design -pin [get_pins ctrl0/reset_reg/D]
```

次の例では、指定のピンの遅延が 500ps 以下になるよう配線しています。

```
route_design -pin [get_pins ctrl0/reset_reg/D] -max_delay 500
```

次の例では、指定のピンの遅延が 200ps より大きくなるよう配線しています。

```
route_design -pin [get_pins ctrl0/ram0/ADDRARDADDR] -min_delay 200
```

関連項目

- [get_nets](#)
- [get_pins](#)
- [launch_runs](#)
- [opt_design](#)
- [phys_opt_design](#)
- [place_design](#)
- [power_opt_design](#)
- [read_checkpoint](#)
- [write_checkpoint](#)

run

シミュレーションを指定した時間実行します。

構文

```
run [-all] [-quiet] [-verbose] [time] [unit]
```

戻り値

なし

使用法

名前	説明
[-all]	シミュレーションをブレークポイント、例外、またはキューにイベントがなくなるまで実行します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[time]	シミュレーション時間を指定します。
[unit]	時間の単位を指定します。有効な値は fs、ps、ns、us、ms、sec です。

カテゴリ

[シミュレーション](#)

run_hw_ilab

ハードウェア ILA をトリガー待機状態にします。

構文

```
run_hw_ilab [-trigger_now arg] [-quiet] [-verbose] [hw_ilas...]
```

戻り値

なし

使用法

名前	説明
[-trigger_now]	即座にトリガーおよびキャプチャを実行します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_ilas]	ハードウェア ILA を指定します。デフォルトは、現在のハードウェア ILA です。

カテゴリ

ハードウェア

run_hw_sio_scan

ハードウェア SIO スキャンを実行します。

構文

```
run_hw_sio_scan [-quiet] [-verbose] hw_sio_scans
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hw_sio_scans</i>	ハードウェア SIO スキャンを指定します。

カテゴリ

ハードウェア

save_bd_design

既存の IP サブシステム デザインをディスク ファイルに保存します。

構文

```
save_bd_design [-quiet] [-verbose] [name]
```

戻り値

デザイン オブジェクト、エラーが発生した場合はなし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[name]	保存するデザインの名前を指定します。

カテゴリ

IP インテグレーター

説明

Vivado Design Suite の IP インテグレーター機能の現在の IP サブシステムまたは指定した IP サブシステムに変更を保存します。

このコマンドを実行すると、保存されたファイルの名前が返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name : 保存する IP サブシステム デザインの名前を指定します。名前を指定しない場合は、現在の IP サブシステム デザインが保存されます。

例

次の例では、現在のプロジェクトの現在の IP サブシステム デザインを保存しています。

```
save_bd_design
```

関連項目

- [close_bd_design](#)
- [create_bd_design](#)
- [current_bd_design](#)
- [get_bd_designs](#)
- [open_bd_design](#)

save_constraints

現在のデザインの制約を保存します。

構文

```
save_constraints [-force] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-force]	制約を強制的に保存します。ターゲットおよびソース XDC を上書きします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

説明

アクティブな制約セットの制約ファイルに変更を保存します。このコマンドにより、制約ファイルへのすべての変更がハード ドライブのプロジェクト データに書き込まれ、作業中の変更がすべて保存されます。

引数

-force (オプション) : 変更されているかどうかにかかわらず、アクティブな制約ファイルを保存します。現在のターゲット制約ファイルが上書きされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、ファイルが変更されているかどうかにかかわらず、アクティブな制約セットの制約ファイルが保存されます。

```
save_constraints -force
```

関連項目

[save_constraints_as](#)

save_constraints_as

現在のデザインの制約を新しい制約ファイルのセットとして保存します。

構文

```
save_constraints_as [-dir arg] [-target_constrs_file arg] [-quiet]
[-verbose] name
```

戻り値

なし

使用法

名前	説明
[-dir]	制約を保存するディレクトリを指定します。
[-target_constrs_file]	新しいファイルセットのターゲット制約ファイルを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	新しい制約ファイルセットの名前を指定します。

カテゴリ

プロジェクト

説明

アクティブな制約セットとその制約セットに含まれる制約ファイルのローカル コピーを、新しい制約セットにコピーします。コピーした制約セットのターゲットとして使用する制約ファイルも指定できます。

このコマンドは、現在の制約ファイルを変更せずに、デザインの制約への変更を保存するために使用します。これにより、さまざまな条件のデザイン制約を試すことができます。

注記 : `save_constraints_as` コマンドで作成された新しい制約セットは、参照されますが、アクティブにはなりません。この制約セットをアクティブにするには、特定の run で constrset プロパティを新しい制約セットに設定する必要があります。例を参照してください。

引数

-dir arg (オプション) : 制約ファイルを保存するディレクトリを指定します。ディレクトリを指定しない場合、新しい制約セットはプロジェクト ソース ディレクトリに保存されます。アクティブな制約セットからの制約ファイルは、指定したディレクトリにコピーされます。

-target_constrs_file arg (オプション) : 新しい制約ファイルセットのターゲット制約ファイルを指定します。パスをファイル名の一部として指定しない場合は、ファイルセットのディレクトリにファイルが作成されます。

注記 : 拡張子 **.xdc** を指定する必要があります。指定しないと、ファイル タイプが不正であり、ターゲット制約ファイルとして設定できないという警告メッセージが表示されます。この場合、既存のターゲット制約ファイルがターゲットとして使用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 新しく作成する制約セットの名前を指定します。

例

次の例では、アクティブな制約セットを **constrs_2** という新しい制約セットに保存し、制約ファイルを指定したディレクトリにコピーし、新しい制約セットのターゲット制約ファイルを指定しています。

```
save_constraints_as -dir C:/Data/con1 -target_constrs_file rev1.xdc constrs_2
```

次の例では、アクティブな制約セットを **newCon2** という新しい制約セットに保存し、制約ファイルをプロジェクトソースの下の **newCon2** 制約ディレクトリにコピーして、指定した合成 run およびインプリメンテーション run の **constrset** プロパティを新しい制約セットに設定しています。

```
save_constraints_as newCon2
set_property CONSTRSET newCon2 [get_runs synth_1]
set_property CONSTRSET newCon2 [get_runs impl_1]
```

注記 : 制約セットは、現在の run に対してアクティブに設定されない限り、アクティブにはなりません。

関連項目

[save_constraints](#)

save_project_as

現在のプロジェクトを新しい名前で保存します。

構文

```
save_project_as [-force] [-quiet] [-verbose] name [dir]
```

戻り値

保存されたプロジェクト オブジェクト

使用法

名前	説明
[-force]	既存のプロジェクトディレクトリを上書きします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	保存するプロジェクトの新しい名前を指定します。
[<i>dir</i>]	プロジェクトファイルを保存するディレクトリを指定します。デフォルトは . です。

カテゴリ

プロジェクト

説明

現在開いているプロジェクトファイルを別の名前で指定のディレクトリまたはディレクトリを指定しない場合は現在の作業ディレクトリに保存します。

引数

-force (オプション) : 既存のプロジェクトを上書きします。指定したディレクトリに指定のプロジェクト名が既に存在する場合、**-force** オプションを使用して既存のプロジェクトを上書きする必要があります。

注記 : 既存プロジェクトを開いている場合、新しいプロジェクトでディスクの既存プロジェクトが上書きされますが、両方のプロジェクトがツールで開いたままになります。この場合、**create_project** を実行する前に **close_project** コマンドを実行しておくことをお勧めします。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : 新しいプロジェクトの名前を指定します。この引数はディレクトリの前に指定します。このコマンドにはパラメーターがないので、最初の引数は *name*、2 つ目の引数は *dir* として認識されます。プロジェクトファイルは *name.ppr* という名前で保存され、指定したディレクトリ *dir* に書き込まれます。

dir (オプション) : 新しいプロジェクトファイルを保存するディレクトリ名を指定します。指定したディレクトリが存在しない場合は、その名前の新しいディレクトリが作成されます。ディレクトリを完全なパスで指定すると、その指定したパス名が使用されます。*dir* をパスなしで指定すると、現在の作業ディレクトリまたはツールの起動ディレクトリでそのディレクトリが検索されるか、作成されます。

注記 : プロジェクトを GUI モードで作成すると、ディレクトリ名 *dir* にファイル名 *name* を付けた *dir/name* という名前のプロジェクトディレクトリが作成され、新しいプロジェクトファイルとプロジェクトデータ フォルダーが保存されます。

例

次の例では、アクティブ プロジェクトが *myProjectDir* というディレクトリに *myProject* という新しいプロジェクト名で保存されます。

```
save_project_as myProject myProjectDir
```

注記 : *dir* にはフォルダーネームのみが指定されているので、プロジェクトは現在の作業ディレクトリかツールの起動ディレクトリに作成されます。

次の例では、現在のプロジェクトが *C:/Designs/myProjectDir* というディレクトリに *myProject* という新しいプロジェクト名で保存されます。**-force** オプションを使用すると、指定したディレクトリに同名のプロジェクトがある場合はそれが上書きされます。

```
save_project_as myProject C:/Designs/myProjectDir -force
```

関連項目

- [create_project](#)
- [open_project](#)

save_wave_config

指定または現在の波形コンフィギュレーション オブジェクトを指定のファイル名で保存します。

構文

```
save_wave_config [-object args] [-quiet] [-verbose] [filename]
```

戻り値

保存された波形コンフィギュレーション オブジェクト

使用法

名前	説明
[-object]	保存する波形コンフィギュレーション (WCFG) を指定します。デフォルトは、現在の波形コンフィギュレーションです。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[filename]	指定または現在の波形コンフィギュレーション オブジェクトを保存するファイルの名前を指定します。

カテゴリ

波形

select_objects

GUI でオブジェクトを選択します。

構文

```
select_objects [-add] [-quiet] [-verbose] objects
```

戻り値

なし

使用法

名前	説明
[-add]	既存の選択リストに追加します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	選択するオブジェクトを指定します。

カテゴリ

GUI 制御

説明

GUI モードで開いている適切なビューで、指定のオブジェクトを選択します。このコマンドは、表示目的のためにのみ使用されます。ほかのコマンドで使用するために選択されているオブジェクトを返すには、**get_selected_objects** コマンドを使用します。

select_objects コマンドでは、指定されたプライマリ オブジェクトに加えてセカンダリ オブジェクトが選択される場合があります。セカンダリ オブジェクトの選択は、[Tools] → [Options] → [Selection Rules] で定義します。選択規則の設定方法は、『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。

オブジェクトの選択を解除するには、**unselect_objects** コマンドを使用します。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須) : 選択するオブジェクトを指定します。

例

次の例では、デバイス上の指定したサイトが選択されます。

```
select_objects [get_sites SLICE_X56Y214]
```

関連項目

- [get_selected_objects](#)
- [unselect_objects](#)

set_case_analysis

入力を 1、0、rising、falling のいずれかに指定します。

構文

```
set_case_analysis [-quiet] [-verbose] value objects
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>value</i>	ピンの論理値を指定します。設定可能な値は、0、1、rising、falling、zero、one、rise、fall です。
<i>objects</i>	ポートまたはピンのリストを指定します。

カテゴリ

SDC、XDC

説明

入力を 1、0、rising、falling のいずれかに指定します。このコマンドは通常、ポートの値を強制的に設定し、解析を実行するために使用します。これは、BUFMUX のセレクトラインを駆動するのに適しています。BUFMUX のセレクトラインに値を駆動すると、タイミング解析用に TIMESPEC の 1 つが選択されます。これにより、不適切な TIMESPEC の解析を回避できます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、set_msg_config コマンドで定義できます。

value (必須)：タイミング解析用にポートまたはピンに適用する値を指定します。有効な値は、0 または zero、1 または one、rise または rising、fall または falling です。デフォルト値は 1 です。

objects (必須) : *value* を適用するポートまたはピンを 1 つ以上指定します。

例

次の例では、デザインに 2 つのクロックがあり、BUFMUX を使用して選択されます。これらのクロックは、異なる周波数で動作します。このデザインのデフォルト解析は、XDC で後ほど定義されるクロックに対して実行されます。この場合、CLK_B です。set_case_analysis を使用すると、異なる解析が実行されます。SEL = 0 の場合は解析は CLK_A に基づき、SEL = 1 の場合は CLK_B に基づきます。

```
create_clock -period 10.0 [get_ports CLK_A]
create_clock -period 15.0 [get_ports CLK_B]
set_case_analysis 0 [get_ports SEL]
```

関連項目

[report_timing](#)

set_clock_groups

排他的または非同期のクロック グループを設定します。

構文

```
set_clock_groups [-name arg] [-logically_exclusive] [-physically_exclusive]
[-asynchronous] [-group args] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-name]	クロック グループの名前を指定します。
[-logically_exclusive]	論理的に排他的なクロック グループを指定します。
[-physically_exclusive]	物理的に排他的なクロック グループを指定します。
[-asynchronous]	非同期クロック グループを指定します。
[-group]	クロック リストを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、XDC

説明

デザインのほかのクロックとは排他的または非同期のクロックまたはクロックのグループを定義します。排他的または非同期のクロックは同時にアクティブにならないので、それらのクロック間のパスはタイミング解析で無視できます。

このコマンドは、排他的または非同期のクロック グループ間を移動するデータ パスにフォルス パス制約を定義するのと似ています。

このコマンドは、1 つの BUFGMUX で駆動される複数のクロックにも使用できます。これは、両方のクロックが同時にアクティブになることはないからです。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-name group_name (オプション)：作成するクロック グループの名前を指定します。指定しない場合、名前が自動的に付けられます。

-logically_exclusive (オプション) : 指定のクロックが論理的に排他的であることを示します。

-physically_exclusive (オプション) : 指定のクロックが物理的に排他的であり、デザインに同時に存在できないことを示します。

-asynchronous (オプション) : 指定のクロックがお互いに非同期であることを示します。

注記 : **-logically_exclusive**、**-physically_exclusive**、および **-asynchronous** は、いずれか 1 つのみ使用できます。

-group args (オプション) : クロック グループに含めるクロックのリストを指定します。クロックの各グループは、その他のグループで指定されたクロックとは排他的または非同期です。クロックのグループを 1 つのみ指定している場合、そのグループはデザインの他のクロックすべてと排他的または非同期です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、src_clk および sync_clk で駆動されるすべてのエレメントをグループ化しています。2 つのクロック グループは非同期です。

```
set_clock_groups -group [get_clocks src_clk] -group [get_clocks sync_clk] \
-asynchronous
```

関連項目

[set_false_path](#)

set_clock_latency

クロック レイテンシを設定します。

構文

```
set_clock_latency [-clock args] [-rise] [-fall] [-min] [-max] [-source]
[-late] [-early] [-quiet] [-verbose] latency objects
```

戻り値

なし

使用法

名前	説明
[-clock]	相対クロックのリストを指定します。
[-rise]	クロックの立ち上がりレイテンシを指定します。
[-fall]	クロックの立ち下がりレイテンシを指定します。
[-min]	クロックの立ち上がりおよび立ち下がりの最小状態のレイテンシを指定します。
[-max]	クロックの立ち上がりおよび立ち下がりの最大状態のレイテンシを指定します。
[-source]	クロックの立ち上がりおよび立ち下がりのソース レイテンシを指定します。
[-late]	指定の遅延がクロック エッジがどれだけ遅れて到着するかを示すよう指定します。
[-early]	指定の遅延がクロック エッジがどれだけ早く到着するかを示すよう指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>latency</i>	レイテンシ値を指定します。
<i>objects</i>	クロック、ポート、またはピンのリストを指定します。

カテゴリ

SDC、XDC

説明

指定したクロック、ポート、またはピンに対して、クロックのソース レイテンシまたはネットワーク レイテンシを定義します。

注記 : このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

ソース レイテンシは、クロック信号が波形の開始点からデザインのクロック定義点まで伝搬されるのにかかる時間です。たとえば、クロックがシステム ボード上のソース (オシレーター) から FPGA 入力ポートまで伝搬されるまでの遅延です。

ネットワーク レイテンシは、クロック信号がデザインの定義点からタイミング パス上のレジスタ クロック ピンに伝搬されるのにかかる時間です。レジスタ クロック ピンの合計クロック レイテンシは、クロックのソース レイテンシとネットワーク レイテンシを加算したものです。

引数

-clock *args* (オプション) : 指定の *objects* に割り当てられた *latency* に関するクロックのリストを指定します。
-clock オプションを使用しない場合、*latency* は指定のピンおよびポートを通過するすべてのクロックに適用されます。

-rise (オプション) : 立ち上がりクロック エッジのレイテンシを定義します。

-fall (オプション) : 立ち下がりクロック エッジのレイテンシを定義します。

-min (オプション) : マルチ コーナー解析用に指定のクロックの最小レイテンシを指定します。

-max (オプション) : マルチ コーナー解析用に指定のクロックの最大レイテンシを指定します。

注記 : **-min** と **-max** オプションを同時に使用することはできません。

-source (オプション) : 指定の *latency* をソース レイテンシとして定義します。クロック ソース レイテンシは、クロック オブジェクトおよびクロック ソース ピンにのみ指定できます。

注記 : **-source** オプションを指定しない場合、*latency* はネットワーク レイテンシとして使用されます。

-late (オプション) : 指定の遅延がクロック エッジがどれだけ遅れて到着するかを示すよう指定します。

-early (オプション) : 指定の遅延がクロック エッジがどれだけ早く到着するかを示すよう指定します。

注記 : **-early** と **-late** オプションを同時に使用することはできず、**-source** を指定する場合にのみ使用できます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

latency (オプション) : 適用するクロック レイテンシの量を指定します。

objects (オプション) : レイテンシを適用するクロック、ポート、またはピン オブジェクトを指定します。ピンまたはポート オブジェクトを指定すると、そのピンまたはポートの推移的なファンアウトのレジスタ クロックすべてにレイテンシが適用されます。**-clock** を使用した場合、指定のクロックのレジスタ クロック ピンすべてにレイテンシが適用されます。

注記 : *objects* にクロックを指定した場合、**-clock** オプションは不要で、無視されます。

例

次の例では、CLK_A の立ち上がりエッジ早く到着するレイテンシ量を設定しています。

```
set_clock_latency -source -rise -early 0.4 [get_ports CLK_A]
```

関連項目

[report_timing](#)

set_clock_sense

ポートまたはピンのクロック エッジを設定します。

構文

```
set_clock_sense [-positive] [-negative] [-stop_propagation] [-pulse arg]
[-clocks args] [-quiet] [-verbose] pins
```

戻り値

なし

使用法

名前	説明
[-positive]	正ユネイト(反転なし)のクロック エッジを指定します。
[-negative]	負ユネイト(反転)のクロック エッジを指定します。
[-stop_propagation]	指定したピンからのクロック伝搬を停止します。
[-pulse]	パルス クロック エッジを指定します。
[-clocks]	クロックのリストを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
pins	ポート/ピンのリストを指定します。

カテゴリ

SDC、XDC

説明

ポートまたはピンのクロック エッジを設定します。これは、クロック オブジェクトに対してピンでの正または負のユネイト性を定義するために使用します。ただし、指定のユネイト性はクロック ネットワークの非ユネイト ポイント、つまりクロック信号が特定できないポイントにのみ適用されます。クロック信号が特定されないので、指定のピンから定義したクロック エッジが伝搬されます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-positive (オプション)：正ユネイト(反転なし)のクロック エッジを指定します。

-negative (オプション)：負ユネイト(反転)のクロック エッジを指定します。

-stop_propagation (オプション)：指定したピンまたはポートからの **-clocks** オプションのクロックの伝搬を停止します。クロックおよびデータの伝搬が停止されます。

-pulse *arg* (オプション) : パルス クロック エッジを指定します。

注記 : **-positive**、**-negative**、**-stop_propagation**、および **-pulse** は、いずれか 1 つのみ使用できます。

-clocks *args* (オプション) : 指定のピンおよびポートに対してクロック エッジを適用するクロックのリストを指定します。**-clocks** オプションを使用しない場合、クロック エッジは指定のピンおよびポートを通過するすべてのクロックに適用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

pins (必須) : クロック エッジを伝搬するポートおよびピンのリストを指定します。

例

次の例では、元のクロックと比較して、正ユネイト パスのみが XOR ゲートの出力ピンを介して伝搬されるよう指定します。

```
set_clock_sense -positive [get_pins xor_a.z]
```

関連項目

- [create_clock](#)
- [get_pins](#)

set_clock_uncertainty

クロックのばらつきを設定します。

構文

```
set_clock_uncertainty [-setup] [-hold] [-from args] [-rise_from args]
[-fall_from args] [-to args] [-rise_to args] [-fall_to args] [-quiet]
[-verbose] uncertainty [objects]
```

戻り値

なし

使用法

名前	説明
[-setup]	セットアップ チェック用にクロックのばらつきを指定します。
[-hold]	ホールド チェック用にクロックのばらつきを指定します。
[-from]	クロック間のばらつきのソース クロックを指定します。
[-rise_from]	クロック間のばらつきのソース クロックを立ち上がりエッジで指定します。
[-fall_from]	クロック間のばらつきのソース クロックを立ち下がりエッジで指定します。
[-to]	クロック間のばらつきのデスティネーション クロックを指定します。
[-rise_to]	クロック間のばらつきのデスティネーション クロックを立ち上がりエッジで指定します。
[-fall_to]	クロック間のばらつきのデスティネーション クロックを立ち下がりエッジで指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
uncertainty	クロック ネットワークのばらつきを指定します。
[objects]	クロック、ポート、またはピンのリストを指定します。

カテゴリ

SDC、XDC

説明

デザインのクロックのばらつきを定義します。クロックのばらつきは、1 つのクロック ドメイン内またはクロック ドメイン間での、クロック信号の到着時間の最大差です。これは、スキーと呼ばれます。クロックのばらつきは、セットアップおよびホールド チェックで使用されます。

クロックを作成するには、`create_clock` または `create_generated_clock` コマンドを使用するか、MMCM の出力などで、ツールで自動的に生成することもできます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-setup (オプション)：セットアップ チェックのクロックのばらつきを指定します。

-hold (オプション)：ホールド チェックのクロックのばらつきを指定します。

-from *clock_source_name* (オプション)：クロック間のばらつきのソース クロックを指定します。

-rise_from *clock_source_name* (オプション)：クロック間のばらつきのソース クロックを立ち上がりエッジで指定します。

-fall_from *clock_source_name* (オプション)：クロック間のばらつきのソース クロックを立ち下がりエッジで指定します。

-to *clock_destination_name* (オプション)：クロック間のばらつきのデスティネーション クロックを指定します。

-rise_to *destination_clock_name* (オプション)：クロック間のばらつきのデスティネーション クロックを立ち上がりエッジで指定します。

-fall_to *destination_clock_name* (オプション)：クロック間のばらつきのデスティネーション クロックを立ち下がりエッジで指定します。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、`set_msg_config` コマンドで定義できます。

***uncertainty* (必須)**：クロック ネットワークのばらつきを指定します。

***objects* (オプション)**：ばらつきを適用するクロック、セル、ポート、またはピンのリストを指定します。

例

次の例では、すべてのクロック ドメイン間のばらつきを定義しています。

```
set_clock_uncertainty 0.225 -from [get_clocks] -to [get_clocks]
```

次の例では、wbClk クロック ドメインのセットアップおよびホールドのばらつきを定義しています。

```
set_clock_uncertainty -setup 0.213 [get_clocks wbClk]
set_clock_uncertainty -hold 0.167 [get_clocks wbClk]
```

関連項目

- [create_clock](#)
- [create_generated_clock](#)
- [get_clocks](#)

set_data_check

データ間チェックを作成します。

構文

```
set_data_check [-from args] [-to args] [-rise_from args] [-fall_from args]
[-rise_to args] [-fall_to args] [-setup] [-hold] [-clock args] [-quiet]
[-verbose] value
```

戻り値

なし

使用法

名前	説明
[-from]	データ間チェックの開始ピン/ポートを指定します。
[-to]	データ間チェックの終了ピン/ポートを指定します。
[-rise_from]	データ間チェックの開始ピン/ポートの立ち上がりエッジを指定します。
[-fall_from]	データ間チェックの開始ピン/ポートの立ち下がりエッジを指定します。
[-rise_to]	データ間チェックの終了ピン/ポートの立ち上がりエッジを指定します。
[-fall_to]	データ間チェックの終了ピン/ポートの立ち下がりエッジを指定します。
[-setup]	データ間チェックのセットアップ タイムを指定します。
[-hold]	データ間チェックのホールド タイムを指定します。
[-clock]	チェックの関連ピン/ポートのクロック ドメインを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
value	定義したチェックのセットアップまたはホールド タイム

カテゴリ

SDC、XDC

説明

データピンのセットアップ チェックおよびホールド チェックを別のデータピンに対して実行します。これは、クロックピンに対して実行する従来のセットアップ チェックおよびホールド チェックとは異なります。セットアップ チェックおよびホールド チェックは、**-from** で指定された関連ピンから **-to** で指定された制約されたピンまで実行されます。関連ピンは、従来のセットアップ チェックおよびホールド チェックのクロックピンと似ています。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-from *value* (オプション)：データ間チェックの開始ピン/ポートを指定します。関連ピンを指定します。

-to *value* (オプション)：データ間チェックの終了ピン/ポートを指定します。制約されたピンを指定します。

-rise_from *value* (オプション)：データ間チェックの開始ピン/ポートの立ち上がりエッジを指定します。

-fall_from *value* (オプション)：データ間チェックの開始ピン/ポートの立ち下がりエッジを指定します。

-rise_to *value* (オプション)：データ間チェックの終了ピン/ポートの立ち上がりエッジを指定します。

-fall_to *value* (オプション)：データ間チェックの終了ピン/ポートの立ち下がりエッジを指定します。

-setup *value* (オプション)：セットアップ データ チェックを実行します。

-hold *value* (オプション)：ホールド データ チェックを実行します。

-clock *value* (オプション)：チェックの関連ピンまたはポートのクロックドメインを指定します。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

value (必須)：定義されたデータ チェックのセットアップまたはホールド タイムを定義します。

例

次の例では、ピン A_IN からピン C_IN までのセットアップ違反を調べるデータ チェックを定義します。

```
set_data_check -from A_IN -to C_IN -setup 2.0
```

この例では、A_IN が関連ピンで C_IN が制約されるピンで、この制約により A_IN に対する C_IN のセットアップ チェックが実行されます。C_IN のデータは、A_IN のエッジより 2.0ns 前に到着する必要があります。

関連項目

[report_timing](#)

set_default_switching_activity

指定したタイプのデフォルトのスイッチング アクティビティを設定します。

構文

```
set_default_switching_activity [-toggle_rate arg] [-static_probability arg]
[-quiet] [-verbose] type...
```

戻り値

なし

使用法

名前	説明
[-toggle_rate]	トグル レート値を指定します。有効な値は 0 ~ 200% です。値は、クロックの % です。デフォルトは 0.0 です。
[-static_probability]	スタティック確率値を指定します。有効な値は 0 ~ 1 で、デフォルトは 1 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>type</i>	ベクターなしの伝搬エンジンで指定したタイプのデフォルトのシード値を設定します。有効なデフォルトタイプの値は、input、input_set、input_reset、input_enable、register、dsp、bram_read_enable、bram_write_enable、output_enable、clock、all です。

カテゴリ

XDС、電力

説明

消費電力予測を実行する際に使用する、信号の広範囲のクラスに対してデフォルトのアクティビティ レートを指定します。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

デザインのスイッチング アクティビティは、スタティックおよびダイナミック消費電力の両方に影響します。スタティック消費電力はロジック ステートの遷移によって異なり、ダイナミック消費電力はトグル レートまたはスイッチング アクティビティに正比例します。

現在のデフォルトスイッチング アクティビティ属性は、**report_default_switching_activity** コマンドを使用して確認できます。スイッチング アクティビティをデフォルト値にリセットするには、**reset_default_switching_activity** コマンドを使用します。

set_switching_activity コマンドは、クラス全体ではなく、1 つ以上の信号のアクティビティを定義する際に使用してください。

引数

-toggle_rate rate (オプション) : 制御クロックに対して出力が切り替わる頻度を示すトグル レートを指定します。有効な値は 0 ~ 200% です。クロック サイクルごとに 1 回切り替わる出力のトグル レートは 100% です。トグル レートは、指定のタイプの信号の消費電力を算出するのに直接使用されます。

-static_probability value (オプション) : スタティック確率 (クロック周期に対して出力のロジック値が 1 である割合) を指定します。有効な値は 0 < value < 1 です。0 は出力が常に 0 であることを示し、1 は出力が常に 1 であることを示します。たとえば、デューティサイクルが 50% のクロック ポートの場合、値は 0.5 になります。スタティック確率は、既知のスイッチング アクティビティ (トグル レート) のデザインのすべてのノードを介した伝搬を算出するのに使用されるので、消費電力の算出に重要です。

注記 : **-static_probability** または **-toggle_rate** のいずれかを指定する必要があります。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

type (必須) : スイッチング アクティビティを定義する信号のクラスを指定します。有効な値は、input、input_set、input_reset、input_enable、register、dsp、bram_read_enable、bram_write_enable、output_enable、clock、all です。

例

次の例では、すべての DSP ブロックのトグル レートを 85% に指定しています。

```
set_default_switching_activity -toggle_rate 85 dsp
```

次の例では、サポートされるすべてのタイプのトグル レートとスイッチング確率を指定しています。

```
set_default_switching_activity -toggle_rate 19 -static_probability .22 all
```

関連項目

- [power_opt_design](#)
- [report_default_switching_activity](#)
- [report_power](#)
- [report_switching_activity](#)
- [reset_default_switching_activity](#)
- [reset_switching_activity](#)
- [set_switching_activity](#)

set_delay_model

タイミング解析用のインターフェクト遅延モデルを設定します。

構文

```
set_delay_model [-interconnect arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-interconnect]	タイミング解析に使用するインターフェクト遅延モデルを指定します。有効な値は estimated、actual (デフォルト)、none です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

タイミング

説明

タイミング解析用のインターフェクト遅延モデルを設定します。インターフェクト遅延モデルには、actual、estimated、および none の 3 つの設定があります。

- actual を選択すると、タイミング解析に配線済みインターフェクトからの実際の遅延が使用されます。デザインの一部のみが配線されている場合は、配線された部分には実際の遅延が使用され、未配線の部分には予測遅延が使用されます。タイミング レポートには、算出された遅延のソースに関する詳細が含まれます。
- estimated に設定すると、インプリメンテーション前のデバイス上のデザインの配置および接続に基づくインターフェクト遅延の予測がタイミング解析に含まれます。デザインが完全に配線されている場合でも、予測遅延を指定できます。
- none に設定すると、タイミング遅延にインターフェクト遅延は含まれず、ロジック遅延のみが使用されます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-interconnect [actual | estimated | none] (オプション)：使用する遅延モデルを指定します。デフォルトは **actual** です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、予測値であるタイミング遅延モデルが使用されます。

```
set_delay_model -interconnect estimated
```

関連項目

[report_timing](#)

set_disable_timing

タイミング アークをディスエーブルにします。

構文

```
set_disable_timing [-from arg] [-to arg] [-quiet] [-verbose] objects
```

戻り値

なし

使用法

名前	説明
[-from]	セル上の開始ピンを指定します。
[-to]	セル上の終了ピンを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	セルまたはピン、ポート、ライブラリ セル、ライブラリ ピン、ライブラリ セル/セル タイミング アークのリストを指定します。

カテゴリ

SDC、XDC、タイミング

説明

指定したセル内の出力ピンまでのタイミング アークをディスエーブルにします。クロック ポートとセルの出力の間に
ある I/O パスのみがディスエーブルになります。

タイミング アークをディスエーブルにするのは、アークのタイミング解析を実行しないようにするためです。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-from *pin_name* (オプション) : オブジェクト セルのソース ピンを指定します。

-to *pin_name* (オプション) : オブジェクト セルのデスティネーション ピンを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、
TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが
非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記: メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須) : タイミング アークをディスエーブルにするオブジェクトのリストを指定します。セル、ピン、ライブラリ セル、ライブラリ ピン、ライブラリ セル/セルのタイミング アークを指定できます。

例

次の例では、セル abc の AX から AMUX ピンまでのタイミング チェックをディスエーブルにします。

```
set_disable_timing -from AX -to AMUX abc
```

次の例では、BRAM セルの指定した入力ピンから指定した出力ピンまでのタイミング アークをディスエーブルにしています。

```
set_disable_timing -from WEBWE[3] -to CLKMEM \
[get_cells ldpc_dout360_channel/U_AP_FIFO_ldpc_dout360_channel_ram/mem_reg_0]
```

次の例では、指定したセルのタイミング アークすべてをディスエーブルにしています。

```
set arcs [get_timing_arcs -of_objects \
[get_cells ldpc_dout360_channel/U_AP_FIFO_ldpc_dout360_channel_ram/mem_reg_0]]
set_disable_timing $arcs
```

関連項目

- [get_cells](#)
- [get_timing_arcs](#)
- [report_timing](#)

set_external_delay

外部遅延を設定します。

構文

```
set_external_delay -from arg -to arg [-min] [-max] [-quiet]
[-verbose] delay_value
```

戻り値

なし

使用法

名前	説明
-from	出力ポートを指定します。
-to	入力ポートを指定します。
[-min]	最小遅延を指定します。
[-max]	最大遅延を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
delay_value	外部 (フィードバック) 遅延値を指定します。

カテゴリ

xdc

説明

出力ポートと入力ポート間の外部 (フィードバック) 遅延を設定します。外部遅延は、外部フィードバックを使用する PLL/MMCM の PLL/MMCM 補正遅延を算出するために使用されます。

最小値または最大値を指定できます。デフォルトでは、指定した値は最小 (ホールド) および最大 (セットアップ) 補正遅延の両方に適用されます。

このコマンドを実行すると、指定した遅延が返されます。

引数

-from arg (必須) : 出力ポート名を指定します。

-to arg (必須) : 入力ポート名を指定します。

-min (オプション) : 遅延値がホールド タイム解析用の最小遅延値であることを指定します。

-max (オプション) : 遅延値がセットアップ解析用の最大遅延値であることを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

delay_value (必須) : 外部遅延値を指定します。デフォルト値は 0 です。

例

次の例では、ClkOut と ClkFb 間の外部フィードバック遅延を 1.0ns に設定しています。

```
set_external_delay -from [get_ports ClkOut] -to [get_ports ClkFb] 1.0
```

関連項目

- [report_timing](#)
- [set_input_delay](#)
- [set_output_delay](#)

set_false_path

フォルス パスを定義します。

構文

```
set_false_path [-setup] [-hold] [-rise] [-fall] [-reset_path] [-from args]
[-rise_from args] [-fall_from args] [-to args] [-rise_to args]
[-fall_to args] [-through args] [-rise_through args] [-fall_through args]
[-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-setup]	指定のパスに必要なクロック サイクル数を指定します。
[-hold]	パスのホールド タイミング解析を除外します。
[-rise]	指定のパスの立ち上がり遅延のみを除外します。
[-fall]	指定のパスの立ち下がり遅延のみを除外します。
[-reset_path]	フォルス パスを設定する前にこのパスをリセットします。
[-from]	パスの開始点またはクロックのリストを指定します。
[-rise_from]	開始点またはクロックのリストから立ち上がるパスに適用します。
[-fall_from]	開始点またはクロックのリストから立ち下がるパスに適用します。
[-to]	パスの終点またはクロックのリストを指定します。
[-rise_to]	終点またはクロックのリストで立ち上がるパスに適用します。
[-fall_to]	終点またはクロックのリストで立ち下がるパスに適用します。
[-through]	通過ピン、セル、またはネットのリストを指定します。
[-rise_through]	ピン、セル、またはネットを立ち上がりで通過するパスに適用します。
[-fall_through]	ピン、セル、またはネットを立ち下がりで通過するパスに適用します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、XDC

説明

タイミング解析で無視するフォルス タイミング パスを設定します。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-setup (オプション)：指定のタイミング パスに対してセットアップ タイミング解析を実行しないようにします。

-hold (オプション)：指定のタイミング パスに対してホールドタイミング解析を実行しないようにします。

-rise (オプション)：指定のタイミング パスの立ち上がり遅延を除外します。

-fall (オプション)：指定のタイミング パスの立ち下がり遅延を除外します。

-reset_path (オプション)：フォルス パスを設定する前にタイミング パスをリセットします。これにより、指定のタイミング パスから除外ベースのタイミング制約がすべて消去されます。

-from *element_name* (オプション)：パスの開始点またはクロックのリストを指定します。

-rise_from *element_name* (オプション)：開始点またはクロックのリストから立ち上がるパスに適用します。

-fall_from *element_name* (オプション)：開始点またはクロックのリストから立ち下がるパスに適用します。

-to *element_name* (オプション)：パスの終点またはクロックのリストを指定します。

-rise_to *element_name* (オプション)：終点またはクロックのリストで立ち上がるパスに適用します。

-fall_to *element_name* (オプション)：終点またはクロックのリストで立ち下がるパスに適用します。

-through *element_name* (オプション)：通過するピン、セル、またはネットのリストを指定します。

-rise_through *element_name* (オプション)：ピン、セル、またはネットを立ち上がりで通過するパスに適用します。

-fall_through *element_name* (オプション)：ピン、セル、またはネットを立ち上がりで通過するパスに適用します。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、bftClk からのパスのセットアップ タイミング解析を除外します。

```
set_false_path -setup -from bftClk
```

次の例では、タイミング解析から指定した 2 つのクロックの間にあるパスを除外します。

```
set_false_path -from [get_clocks GT0_RXUSRCLK2_OUT] -to [get_clocks DRPCLK_OUT]
```

関連項目

- [get_clocks](#)
- [get_pins](#)
- [get_ports](#)
- [report_timing](#)

set_hierarchy_separator

階層区切り文字を設定します。

構文

```
set_hierarchy_separator [-quiet] [-verbose] [separator]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[separator]	階層区切り文字を指定します。デフォルトはスラッシュ (/) です。

カテゴリ

SDC、XDC

説明

デザインで使用する階層区切り文字を設定します。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

separator (オプション)：階層区切り文字として使用する文字を指定します。有効な文字は、/、@、^、#、..、および|です。デフォルトはスラッシュ (/) で、*separator*を指定しない場合に使用されます。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、階層区切り文字を | に変更しています。

```
set_hierarchy_separator |
```

次の例では、デフォルトの階層区切り文字 (/) に戻しています。

```
set_hierarchy_separator
```

関連項目

[get_hierarchy_separator](#)

set_input_delay

ポートの入力遅延を設定します。

構文

```
set_input_delay [-clock args] [-reference_pin args] [-clock_fall]
[-rise] [-fall] [-max] [-min] [-add_delay] [-network_latency_included]
[-source_latency_included] [-quiet] [-verbose] delay objects
```

戻り値

なし

使用法

名前	説明
[-clock]	相対クロックを指定します。
[-reference_pin]	相対ピンまたはポートを指定します。
[-clock_fall]	遅延がクロックの立ち下がりエッジを基準にすることを示します。
[-rise]	立ち上がり遅延を指定します。
[-fall]	立ち下がり遅延を指定します。
[-max]	最大遅延を指定します。
[-min]	最小遅延を指定します。
[-add_delay]	既存の入力遅延を削除しません。
[-network_latency_included]	クロックのネットワーク レイテンシが既に含まれていることを示します。
[-source_latency_included]	クロックのソース レイテンシが既に含まれていることを示します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
delay	遅延値を指定します。
objects	ポートのリストを指定します。

カテゴリ

[SDC](#)、[XDC](#)

説明

ポートの入力遅延を設定します。

注記 : このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-clock *arg* (オプション) : 入力遅延が指定のクロックを基準にしていることを指定します。デフォルトでは立ち上がりエッジが使用されます。 **-clock_fall** を使用すると、立ち下がりエッジを使用するよう指定できます。

-reference_pin *arg* (オプション) : 遅延がクロックではなく指定のピンに現れるクロックのアクティブ エッジを基準としていることを指定します。

-clock_fall (オプション) : 遅延が指定のクロックの立ち上がりエッジではなく立ち下がりエッジを基準にすることを示します。

-rise (オプション) : 入力遅延が指定したポートの立ち上がり遷移に適用されることを指定します。デフォルトでは、立ち上がり遷移と立ち下がり遷移の両方に適用されます。

-fall (オプション) : 入力遅延が指定したポートの立ち下がり遷移に適用されることを指定します。デフォルトでは、立ち上がり遷移と立ち下がり遷移の両方に適用されます。

-max (オプション) : 指定の入力遅延が最大パス遅延の算出にのみ使用されるよう指定します。

-min (オプション) : 指定の入力遅延が最小パス遅延の算出にのみ使用されるよう指定します。

-add_delay (オプション) : 指定の遅延をポート上の既存の遅延に追加します。デフォルトでは、既存の遅延が置き換えられます。

-network_latency_included (オプション) : 遅延値に基準クロックのネットワーク レイテンシが含まれることを示します。

-source_latency_included (オプション) : 指定の遅延値に相対クロックのソース レイテンシが含まれることを示します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

delay (必須) : 指定のポートに適用する入力遅延を指定します。

objects (必須) : 遅延値を適用するポートのリストを指定します。

例

次の例では、ポート DIN 上の入力遅延を指定しています。入力遅延は 3 で、クロック clk1 の立ち上がりエッジを基準としています。

```
set_input_delay -clock clk1 3 DIN
```

次の例では、ポート DIN 上の入力遅延を指定しています。入力遅延は 2 で、クロック clk1 の立ち下がりエッジを基準としています。

```
set_input_delay -clock_fall -clock clk1 2 DIN
```

次の例では、ポート reset 上の入力遅延を指定しています。入力遅延は 2 で、クロック wbClk から派生する wbClk_IBUF_BUFG_inst/O ピンに現れるクロックの立ち上がりエッジを基準としています。

```
set_input_delay -clock wbClk 2 -reference_pin [get_pin wbClk_IBUF_BUFG_inst/O] reset
```

次の例では、デザインのクロック以外の入力ポートすべてに入力遅延を指定しています。all_inputs ではクロックポートを含むデザインのすべてのポートが返されますが、set_input_delay ではクロックポートに入力遅延を設定していません。入力遅延は 1 で、クロック wbClk の立ち上がりエッジを基準としています。

```
set_input_delay -clock wbClk 1 [all_inputs]
```

次の例では、reset および wbDataForInput に対して、クロック wbClk の立ち上がりエッジを基準とした入力遅延を 4 に設定しています。

```
set_input_delay -clock wbClk 4 [list reset wbDataForInput]
```

関連項目

- [all_clocks](#)
- [all_inputs](#)
- [check_timing](#)
- [create_clock](#)
- [get_ports](#)
- [report_timing](#)
- [set_output_delay](#)

set_input_jitter

クロック オブジェクトの入力ジッターを設定します。

構文

```
set_input_jitter [-quiet] [-verbose] clock input_jitter
```

戻り値

クロック

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>clock</i>	クロック
<i>input_jitter</i>	入力ジッター値を指定します。有効な値は 0 以上です。

カテゴリ

xdc

説明

入力ジッターとは、理想的なクロック到達時間と比較した連続クロック エッジ間のばらつきです。このコマンドは、**create_clock** コマンドで定義された指定のプライマリ クロックの入力ジッターを設定します。このコマンドでは 1 つのクロックしか指定できないので、各プライマリ クロックにジッターを個別に設定する必要があります。

set_input_jitter コマンドでは、プライマリ クロックの入力ジッターのみを設定可能です。生成クロックまたは自動生成クロックの入力ジッターは設定できません。入力ジッターは、マスター クロックから生成クロックおよび自動生成クロックに伝搬されます。

set_input_jitter コマンドは、エラボレートされたデザインまたは合成ではサポートされません。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

clock_name (必須) : **create_clock** コマンドで定義されたプライマリ クロックの名前を指定します。

input_jitter (必須) : 指定のクロック オブジェクトの入力ジッターを指定します。有効な値は 0 以上です。

例

次の例では、2 つのクロック sysClk および procClk の入力ジッターを 0.3 に設定しています。ジッター値は同じですが、**set_input_jitter** コマンドでは 1 つのクロックしか指定できないので、クロックごとに個別に設定する必要があります。

```
set_input_jitter sysClk 0.3
set_input_jitter procClk 0.3
```

次の例では、プライマリ クロック sysClk を定義し、このプライマリ クロックを 2 で分周したクロック sysClkDiv2 を生成した後、プライマリ クロックの入力ジッターを 0.15 に設定しています。入力ジッターは、生成クロックに自動的に伝搬されます。

```
create_clock -period 10 -name sysClk [get_ports sysClk]
create_generated_clock -name sysClkDiv2 -source [get_ports sysClk] -divide_by 2 \
[get_pins clkgen/sysClkDiv/Q]
set_input_jitter sysClk 0.15
```

関連項目

- [all_clocks](#)
- [check_timing](#)
- [create_clock](#)
- [create_generated_clock](#)
- [report_clocks](#)
- [report_timing](#)
- [set_clock_latency](#)
- [set_system_jitter](#)

set_load

ポートおよびネットの容量を設定します。

構文

```
set_load [-rise] [-fall] [-max] [-min] [-quiet]
[-verbose] capacitance objects
```

戻り値

なし

使用法

名前	説明
[-rise]	立ち上がり容量値を指定します (ポートのみ)。
[-fall]	立ち下がり容量値を指定します (ポートのみ)。
[-max]	最大容量値を指定します。
[-min]	最小容量値を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
capacitance	容量値を指定します。
objects	ポートまたはネットのリストを指定します。

カテゴリ

SDC、XDC

説明

出力ポートの負荷容量を指定の値に設定します。負荷容量は、デバイスの環境動作条件は、**report_power** コマンドによる消費電力解析では使用されますが、タイミング解析では使用されません。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-max (オプション)：最大負荷容量値を指定します。

-min (オプション)：最小負荷容量値を指定します。

-rise (オプション)：指定したポートの立ち上がりエッジ負荷容量を定義します。

-fall (オプション)：指定したポートの立ち下がりエッジ負荷容量を定義します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

capacitance (必須) : 負荷容量の値を指定します。値は、0 以上の浮動小数点値で指定します。デフォルト値は 0 です。

注記 : デフォルトの容量の単位はピコファラド (pF) です。

objects (必須) : 負荷容量を適用する出力ポート オブジェクトのリストを指定します。デザインのすべての出力を取得するには、**all_outputs** コマンドを使用します。

例

次の例では、すべてのポートの負荷容量が指定の値に設定されます。

```
set_load 5.5 [all_outputs]
```

次の例では、指定の出力ポートの立ち上がりエッジおよび立ち下がりエッジ負荷容量が設定されます。

```
set_load -rise -fall 8 [get_ports wbOutput*]
```

関連項目

- [all_outputs](#)
- [get_ports](#)
- [report_power](#)

set_logic_dc

ポート/ピンをドントケアに設定します。

構文

```
set_logic_dc [-quiet] [-verbose] objects
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	ポートまたはピンのリストを指定します。

カテゴリ

SDC、XDC

説明

指定の入力ポートまたは入力ピンを論理値 X (ドントケア) に設定します。このコマンドは合成ではサポートされません。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須)：入力ポートまたはピンのリストを指定します。

例

次の例では、指定したポートを X に設定しています。

```
set_logic_dc [get_ports reset]
```

関連項目

- [all_inputs](#)
- [get_pins](#)
- [get_ports](#)
- [set_logic_one](#)
- [set_logic_unconnected](#)
- [set_logic_zero](#)

set_logic_one

ポート/ピンを論理 1 に設定します。

構文

```
set_logic_one [-quiet] [-verbose] objects
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	ポートまたはピンのリストを指定します。

カテゴリ

SDC、XDC

説明

指定の入力ポートまたは入力ピンを論理 1 に設定します。このコマンドは合成ではサポートされません。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須)：入力ポートまたはピンのリストを指定します。

例

次の例では、指定の入力ポートを論理 1 に設定しています。

```
set_logic_one [get_ports reset]
```

次の例では、入力ポート reset および wbDataForInput を論理 1 に設定しています。

```
set_logic_one [list [get_ports reset] [get_ports wbDataForInput]]
```

次の例では、インスタンス reset_IBUF の入力ピン I を論理 1 に設定しています。

```
set_logic_one [get_pins reset_IBUF_inst/I]
```

関連項目

- [all_inputs](#)
- [get_pins](#)
- [get_ports](#)
- [set_logic_dc](#)
- [set_logic_unconnected](#)
- [set_logic_zero](#)

set_logic_unconnected

ポート/ピンを未接続に設定します。

構文

```
set_logic_unconnected [-quiet] [-verbose] objects
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	ポートまたはピンのリストを指定します。

カテゴリ

SDC、XDC

説明

指定した出力ポートまたはピンを未接続に指定します。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須)：出力ポートまたはピンのリストを指定します。

例

次の例では、指定したポートを未接続に設定しています。

```
set_logic_unconnected [get_ports OUT1]
```

関連項目

- [set_logic_dc](#)
- [set_logic_one](#)
- [set_logic_zero](#)

set_logic_zero

ポート/ピンを論理 0 に設定します。

構文

```
set_logic_zero [-quiet] [-verbose] objects
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	ポートまたはピンのリストを指定します。

カテゴリ

SDC、XDC

説明

指定の入力ポートおよびピンを論理 0 に設定します。このコマンドは合成ではサポートされません。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須)：入力ポートまたはピンのリストを指定します。

例

次の例では、指定したポートを論理 0 に設定しています。

```
set_logic_zero [get_ports reset]
```

関連項目

- [all_inputs](#)
- [get_pins](#)
- [get_ports](#)
- [set_logic_one](#)
- [set_logic_unconnected](#)
- [set_logic_zero](#)

set_max_delay

タイミング パスの最大遅延を指定します。

構文

```
set_max_delay [-rise] [-fall] [-reset_path] [-from args] [-rise_from args]
[-fall_from args] [-to args] [-rise_to args] [-fall_to args] [-through args]
[-rise_through args] [-fall_through args] [-datapath_only] [-quiet]
[-verbose] delay
```

戻り値

なし

使用法

名前	説明
[-rise]	遅延値が立ち上がりパス遅延に適用されます。
[-fall]	遅延値が立ち下がりパス遅延に適用されます。
[-reset_path]	最大遅延を設定する前にこのパスをリセットします。
[-from]	パスの開始点またはクロックのリストを指定します。
[-rise_from]	開始点またはクロックのリストから立ち上がるパスに適用します。
[-fall_from]	開始点またはクロックのリストから立ち下がるパスに適用します。
[-to]	パスの終点またはクロックのリストを指定します。
[-rise_to]	終点またはクロックのリストで立ち上がるパスに適用します。
[-fall_to]	終点またはクロックのリストで立ち下がるパスに適用します。
[-through]	通過ピン、セル、またはネットのリストを指定します。
[-rise_through]	ピン、セル、またはネットを立ち上がりで通過するパスに適用します。
[-fall_through]	ピン、セル、またはネットを立ち下がりで通過するパスに適用します。
[-datapath_only]	遅延の算出からクロック スキューおよびジッターを除外します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
delay	遅延値を指定します。

カテゴリ

SDC、XDC

説明

タイミング パスに許容される最大遅延を時間の単位で指定します。**-rise** または **-fall** オプションを使用しない場合、指定の遅延値は指定のタイミング パスの立ち上がりエッジと立ち下がりエッジの両方に割り当てられます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

最大立ち上がりおよび立ち下がり遅延は、同じパスの最小立ち上がりおよび立ち下がり遅延以上である必要があります。最大遅延が最小遅延より小さい場合、タイミング パスから以前に割り当てられていた遅延値が削除され、0 にリセットされます。

遅延値は、**-from**、**-through**、または **-to** オプションの少なくとも 1 つを使用して定義されたタイミング パスに割り当てる必要があります。**-to** 終点などの汎用パス遅延は、**-from/-to** または **-from/-through/-to** パス定義などのような特定のパス定義により上書きされます。

引数

-rise (オプション)：遅延値をタイミング パスの立ち上がりエッジに適用するよう指定します。

-fall (オプション)：遅延値をタイミング パスの立ち下がりエッジに適用するよう指定します。

注記：**-rise** または **-fall** のどちらも指定しない場合、遅延は立ち上がりエッジおよび立ち下がりエッジ両方のパス遅延に適用されます。

-reset_path (オプション)：新しいパス遅延を適用する前に既存の立ち上がりエッジまたは立ち下がりエッジ最大遅延を消去するよう指定します。**-to** のみを指定した場合、指定の終点に到達するすべてのパスで遅延が消去されます。**-from** のみを指定した場合、指定の開始点からのすべてのパスで遅延が消去されます。**-from/-to** または **-from/-through/-to** を指定した場合、指定のパスがリセットされます。

注記：このオプションは、**set_max_delay** の前に **reset_paths** を実行するのと同じです。

-from value (オプション)：パスの開始点またはクロックのリストを指定します。有効な開始点は、プライマリ入力または入出力ポート、あるいは同期エレメントのクロック ピンです。クロックが指定されている場合は、そのクロックおよびそのクロックに接続されているレジスタのすべてのクロック ピンに関連するプライマリ入力および入出力ポートが開始点として指定されます。

-rise_from element_name (オプション)：最大遅延を開始点またはクロックのリストから立ち上がるパスに適用します。

-fall_from element_name (オプション)：最大遅延を開始点またはクロックのリストから立ち下がるパスに適用します。

-to element_name (オプション)：パスの終点またはクロックのリストを指定します。有効な終点は、プライマリ出力または入出力ポート、あるいは同期エレメントのデータ ピンです。クロックが指定されている場合は、そのクロックおよびそのクロックに接続されているレジスタのすべてのデータ ピンに関連するプライマリ入力および入出力ポートが終点として指定されます。

-rise_to element_name (オプション)：最大遅延を終点またはクロックのリストで立ち上がるパスに適用します。

-fall_to element_name (オプション)：最大遅延を終点またはクロックのリストで立ち下がるパスに適用します。

-through element_name (オプション)：通過するピン、セル、またはネットのリストを指定します。

-rise_through element_name (オプション)：最大遅延をピン、セル、またはネットを立ち上がりで通過するパスに適用します。

-fall_through element_name (オプション)：最大遅延をピン、セル、またはネットを立ち下がりで通過するパスに適用します。

-datapath_only (オプション) : 指定のバスの遅延算出からクロック スキューとジッターを除外します。このオプションは、異なるクロックを持つ順次エレメント間の遅延を制約する際に遅延の算出にクロック スキューおよびジッターを考慮しない場合に使用します。最初のフリップフロップの clock-to-Q 遅延、フリップフロップ間のワイヤ遅延、および 2 番目のフリップフロップのセットアップ タイムのみが考慮されます。指定のバスは直接接続されており、開始点と終点の間にロジックがないことが必要です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

delay (必須) : 最大遅延値を時間の単位で指定します。delay は 0 以上の浮動小数点値で指定します。デフォルト値は 0 です。

例

次の例では、すべての入力ポートと出力ポートの間の最大遅延を 60ns に設定しています。

```
set_max_delay 60 -from [all_inputs]
```

次の例では、既存の最大遅延を消去し、指定のクロックが供給される終点へのバスに新しい最大遅延を設定しています。

```
set_max_delay -reset_path 50 -to [get_clocks cpuClk]
```

set_max_delay コマンドは、単純なシンクロナイザーが使用されている場合に、クロック ドメインをまたぐタイミング制約を定義するのによく使用されます。次の例では、2 つのフリップフロップ (FF1 および FF2) に異なるクロックが供給されており、FF1/Q が net1 を介して FF2/D に直接接続されています。この接続の遅延を 1.0 ns に制限するには、次のいずれかの制約を使用します。

```
set_max_delay -from FF1 -to FF2 -datapath_only 1.0
set_max_delay -from FF1/Q -to FF2/D -datapath_only 1.0
set_max_delay -through net1 -datapath_only 1.0
```

関連項目

- [get_clocks](#)
- [get_nets](#)
- [get_ports](#)
- [report_timing](#)
- [set_min_delay](#)
- [set_units](#)

set_max_time_borrow

ラッチ用に借りることのできる最大時間を設定します。

構文

```
set_max_time_borrow [-quiet] [-verbose] delay objects
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>delay</i>	遅延値を指定します。有効な値は 0 以上です。
<i>objects</i>	クロック、セル、データピン、またはクロックピンのリストを指定します。

カテゴリ

SDC、XDC

説明

ラッチのタイミングを解析する際に、ネット間で借りることのできる最大時間を指定します。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

delay (必須)：指定のオブジェクトに適用する遅延を指定します。*delay* は 0 以上の浮動小数点値で指定します。デフォルト値は 0 です。

注記：時間の単位は、**set_units** コマンドで設定します。デフォルトの時間の単位はナノ秒 (ns) です。

objects (必須)：制限を適用するクロック、セル、データピン、またはクロックピンのリストを指定します。

例

次の例では、すべてのクロックに接続されているラッチで借りることのできる時間を 0 に設定しています。デザイン全体で時間の借用がディスエーブルになります。

```
set_max_time_borrow 0.0 [all_clocks]
```

次の例では、最上位階層のネットで借りることのできる時間を 20 に設定しています。

```
set_max_time_borrow 20 {top/*}
```

関連項目

- [all_clocks](#)
- [get_clocks](#)
- [get_nets](#)

set_min_delay

タイミング パスの最小遅延を指定します。

構文

```
set_min_delay [-rise] [-fall] [-reset_path] [-from args] [-rise_from args]
[-fall_from args] [-to args] [-rise_to args] [-fall_to args] [-through args]
[-rise_through args] [-fall_through args] [-quiet] [-verbose] delay
```

戻り値

なし

使用法

名前	説明
[-rise]	遅延値が立ち上がりパス遅延に適用されます。
[-fall]	遅延値が立ち下がりパス遅延に適用されます。
[-reset_path]	最小遅延を設定する前にこのパスをリセットします。
[-from]	パスの開始点またはクロックのリストを指定します。
[-rise_from]	開始点またはクロックのリストから立ち上がるパスに適用します。
[-fall_from]	開始点またはクロックのリストから立ち下がるパスに適用します。
[-to]	パスの終点またはクロックのリストを指定します。
[-rise_to]	終点またはクロックのリストで立ち上がるパスに適用します。
[-fall_to]	終点またはクロックのリストで立ち下がるパスに適用します。
[-through]	通過ピン、セル、またはネットのリストを指定します。
[-rise_through]	ピン、セル、またはネットを立ち上がりで通過するパスに適用します。
[-fall_through]	ピン、セル、またはネットを立ち下がりで通過するパスに適用します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
delay	遅延値を指定します。

カテゴリ

SDC、XDC

説明

タイミング パスに許容される最小遅延を時間の単位で指定します。-rise または -fall オプションを使用しない場合、指定の遅延値は指定のタイミング パスの立ち上がりエッジと立ち下がりエッジの両方に割り当てられます。

注記： このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

最小立ち上がりおよび立ち下がり遅延は、同じパスの最大立ち上がりおよび立ち下がり遅延以下である必要があります。最大遅延が最小遅延より小さい場合、タイミング パスから以前に割り当てられていた遅延値が削除され、0 にリセットされます。

遅延値は、**-from**、**-through**、または **-to** オプションの少なくとも 1 つを使用して定義されたタイミング パスに割り当てる必要があります。**-to** 終点などの汎用パス遅延は、**-from/-to** または **-from/-through/-to** パス定義などのような特定のパス定義により上書きされます。

引数

-rise (オプション) : 遅延値をタイミング パスの立ち上がりエッジに適用するよう指定します。

-fall (オプション) : 遅延値をタイミング パスの立ち下がりエッジに適用するよう指定します。

-reset_path (オプション) : 新しいパス遅延を適用する前に既存の立ち上がりエッジまたは立ち下がりエッジ最小遅延を消去します。**-to** のみを指定した場合、指定の終点に到達するすべてのパスで遅延が消去されます。**-from** のみを指定した場合、指定の開始点からのすべてのパスで遅延が消去されます。**-from/-to** または **-from/-through/-to** を指定した場合、指定のパスがリセットされます。

注記 : このオプションは、**set_min_delay** の前に **reset_paths** を実行するのと同じです。

-from objects (オプション) : 指定の遅延を割り当てるタイミング パスの開始点を指定します。

-rise_from objects (オプション) : 立ち上がりエッジに指定の遅延を割り当てるタイミング パスの開始点を指定します。

-fall_from objects (オプション) : 立ち下がりエッジに指定の遅延を割り当てるタイミング パスの開始点を指定します。

-to objects (オプション) : 最小遅延を割り当てるパスのデスティネーション オブジェクトを指定します。

-rise_to objects (オプション) : 最小遲延を割り当てる立ち上がりエッジ パスのデスティネーション オブジェクトを指定します。

-fall_to objects (オプション) : 最小遲延を割り当てる立ち下がりエッジ パスのデスティネーション オブジェクトを指定します。

-through objects (オプション) : 最小遲延を割り当てるパスが通過するピン、セル、またはネットのリストを指定します。

-rise_through objects (オプション) : 最小遲延を割り当てる立ち上がりパスが通過するピン、セル、またはネットのリストを指定します。

-fall_through objects (オプション) : 最小遲延を割り当てる立ち下がりパスが通過するピン、セル、またはネットのリストを指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

delay (必須) : 最小遅延値を時間の単位で指定します。*delay* は 0 以上の浮動小数点値で指定します。デフォルト値は 0 です。

注記 : 時間の単位は、**set_units** コマンドで設定します。デフォルトの時間の単位はナノ秒 (ns) です。

例

次の例では、すべての入力ポートからの最小遅延を 20ns に設定しています。

```
set_min_delay 20 -from [all_inputs]
```

次の例では、すべての出力ポートが終点となるタイミング パスの最小遅延を 20ns に設定しています。

```
set_min_delay 20 -to [get_ports -filter {DIRECTION == out}]
```

関連項目

- [get_clocks](#)
- [get_nets](#)
- [get_ports](#)
- [report_timing](#)
- [set_units](#)

set_msg_config

メッセージ ID、文字列、または重要度で指定したメッセージの表示および制御方法を設定します。

構文

```
set_msg_config [-id arg] [-string args] [-severity arg] [-limit arg]
[-new_severity arg] [-suppress] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-id]	選択した操作を指定のメッセージ ID のメッセージのみに適用します (例 : -id {Common 17-35})。デフォルトでは、すべての ID が指定されます。
[-string]	選択した操作を指定の文字列を含むメッセージのみに適用します。デフォルトはなしです。
[-severity]	選択した操作を指定の重要度のメッセージのみに適用します (例 : -severity INFO)。デフォルトでは、すべての重要度のメッセージが指定されます。
[-limit]	現在のプロジェクトで、指定のメッセージの最大表示回数を整数で指定します。
[-new_severity]	現在のプロジェクトで、指定のメッセージの重要度を変更します。
[-suppress]	現在のプロジェクトで、指定のメッセージを非表示にします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

レポート

説明

現在のプロジェクトで Vivado ツールにより返されるメッセージ制御ルールを設定します。メッセージの重要度を変更したり、メッセージが表示される回数を制限したり、メッセージを非表示にできます。1 つの **set_msg_config** で実行できるのは、いずれか 1 つの操作のみです。

- ツールで表示されるメッセージの重要度を特定のレベルに変更します。たとえば、指定したメッセージ ID の重要度を「WARNING」から「ERROR」に変更できます。
- デザイン セッション中または 1 回の起動中にツールでメッセージが表示される最大回数を設定します。特定のメッセージ ID の最大表示回数、または特定の重要度のメッセージの最大表示回数を設定できます。
- 特定のメッセージ ID がツールでレポートされないようにします。非表示になっているメッセージが表示されるようにするには、**reset_msg_config** コマンドを使用します。

1 つの **set_msg_config** コマンドで複数の操作を実行しようとすると、エラーが返されます。

set_msg_config コマンドで設定するメッセージは、-string、-id、および -severity で指定します。少なくとも 1 つを使用してコマンドを適用するメッセージを指定する必要があります。これらのすべてのオプションの指定に一致するメッセージのみに設定が適用されます。

注記 : **set_msg_config** では、メッセージの指定にワイルドカードは使用できません。

メッセージ制御ルールはプロジェクト特定であり、プロジェクトを閉じて開き直しても継続されます。現在のプロジェクトで特定のメッセージに対して定義されている現在の制御ルールを確認するには、**get_msg_config** コマンドを使用します。

デフォルト設定に戻すには、**reset_msg_config** コマンドを使用します。

注記 : すべてのメッセージ ID のデフォルトの最大表示回数は 100 で、**messaging.defaultLimit** パラメーターで定義されています。この最大表示回数が、ツールで返される各メッセージに適用されます。このパラメーターの現在の値を取得するには **get_param** コマンドを使用し、変更するには **set_param** コマンドを使用します。

引数

-id arg (オプション) : 設定を適用するメッセージの ID を指定します。ツールで表示される各メッセージには、アプリケーション サブシステム コードとメッセージ識別子を含む固有のメッセージ ID が付けられています。次に、メッセージ ID の例を示します。

```
"Common 17-54"
"Netlist 29-28"
"Synth 8-3295"
```

-string args (オプション) : 指定した文字列を含むメッセージのみに選択した操作を適用します。文字列は中かっこ (()) で囲んで指定します。複数の文字列はスペースで区切って指定します。

```
{}{Vivado} {All Programmable}}
```

注記 : 大文字/小文字が区別されます。

-severity args (オプション) : メッセージの重要度を指定します。次の 5 種類の重要度があります。

- **ERROR** : デザインの結果が使用不可となったり、ユーザーの操作がないと回避できないような問題が発生していることを示すエラー メッセージです。

- **{CRITICAL WARNING}** : 入力や制約に適用されないものがあったり、FPGA ファミリには適していないものがあることを示すクリティカル警告メッセージです。修正することが強く推奨されます。

注記 : これは 2 単語の値なので、{} または "" で囲む必要があります。

- **WARNING** : 制約または仕様が指定どおりに適用されず、デザインが最適な結果にならない可能性を示す警告メッセージです。修正するかどうかは、ユーザーが判断します。

- **INFO** : STATUS メッセージと同じですが、重要度とメッセージ ID タグが含まれる点が異なります。INFO メッセージには、必要に応じてアンサー データベースで確認できるようにメッセージ ID が含まれます。

- **STATUS** : デザインの処理に関するプロセスおよびフィードバックのステータスを示すステータス メッセージです。STATUS メッセージには、メッセージ ID は含まれません。

注記 : STATUS メッセージにはメッセージ ID がないので、重要度を変更することはできません。

-limit arg (オプション) : メッセージの最大表示回数を 1 以上の整数で指定します。-1 に設定すると、**messaging.defaultLimit** で指定されたデフォルトの制限数にリセットされます。

-new_severity arg (オプション) : 指定したメッセージの新しい重要度を指定します。有効な値は、**-severity** オプションに示されています。

-suppress (オプション) : 指定したメッセージがレポートされないようにします。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、Common 17-81 メッセージをクリティカル警告に変更しています。

```
set_msg_config -id "Common 17-81" -new_severity "CRITICAL WARNING"
```

次の例では、指定したメッセージ ID のメッセージがレポートされないようにしています。

```
set_msg_config -suppress -id {HDL 9-1654}
```

次の例では、メッセージ ID 「17-35」の警告メッセージで、文字列「clk」が含まれるメッセージをエラーに変更しています。

```
set_msg_config -severity warning -string "clk" -id "17-35" -new_severity error
```

次の例では、指定のメッセージ ID に設定されている現在のデフォルトのメッセージ最大表示回数を取得し、新しい最大表示回数を指定しています。

```
get_param messaging.defaultLimit
100
set_param messaging.defaultLimit 1000
set_msg_config -id "common 17-81" -limit 1500
```

関連項目

- [get_msg_config](#)
- [get_param](#)
- [reset_msg_config](#)
- [set_param](#)

set_msg_limit

メッセージの最大表示回数を設定します。

構文

```
set_msg_limit [-severity arg] [-id arg] [-quiet] [-verbose] count
```

戻り値

新しいメッセージの最大表示回数

使用法

名前	説明
[-severity]	メッセージの重要度 (「ERROR」、「CRITICAL WARNING」など) を指定します。-id と共に使用することはできません。デフォルトは ALL です。
[-id]	「Common 17-99」など、メッセージの ID を指定します。-severity と共に使用することはできません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
count	メッセージの新しい最大表示回数を指定します。

カテゴリ

レポート

説明

デザイン セッション中または 1 回の起動中にツールで表示されるメッセージの最大回数を設定します。特定のメッセージ ID (「Common 17-54」など) または特定の重要度 (「CRITICAL WARNING」など) の最大表示回数を指定できます。

ツールで表示される各メッセージには、アプリケーション サブシステム コードとメッセージ識別子を含む固有のメッセージ ID が付けられています。次に、メッセージ ID の例を示します。

```
"Common 17-54"  
"Netlist 29-28"  
"Synth 8-3295"
```

メッセージの重要度または ID 別の現在の最大表示回数は、**get_msg_limit** コマンドを使用して取得できます。メッセージの最大表示回数をデフォルトに戻すには、**reset_msg_limit** コマンドを使用します。

メッセージ ID または重要度を指定しない場合、エラーが返されます。

すべてのメッセージ ID のデフォルトの最大表示回数は、**messaging.defaultLimit** パラメーターで定義されています。この最大表示回数が、ツールで返される各メッセージに適用されます。このパラメーターの現在の値を取得するには **get_param** コマンドを使用し、変更するには **set_param** コマンドを使用します。

set_msg_config を使用すると、メッセージの重要度の変更、最大表示回数の設定、メッセージの非表示を実行できます。

引数

-id value (オプション) : メッセージ ID を指定します。各メッセージには、独自の ID が含まれます。たとえば、「Common 17-54」、「Netlist 29-28」です。

-severity value (オプション) : メッセージの重要度を指定します。次の 5 種類の重要度があります。

- **ERROR** : デザインの結果が使用不可となったり、ユーザーの操作がないと回避できないような問題が発生していることを示すエラー メッセージです。
- **{CRITICAL WARNING}** : 入力や制約に適用されないものがあったり、FPGA ファミリには適していないものがあることを示すクリティカル警告メッセージです。修正することが強く推奨されます。

注記 : これは 2 単語の値なので、{} で囲む必要があります。

- **WARNING** : 制約または仕様が指定どおりに適用されず、デザインが最適な結果にならない可能性を示す警告メッセージです。修正するかどうかは、ユーザーが判断します。
- **INFO** : STATUS メッセージと同じですが、重要度とメッセージ ID タグが含まれる点が異なります。INFO メッセージには、必要に応じてアンサー データベースで確認できるようにメッセージ ID が含まれます。
- **STATUS** : デザインの処理に関するプロセスおよびフィードバックのステータスを示すステータス メッセージです。STATUS メッセージには、メッセージ ID は含まれません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

count (必須) : メッセージの最大表示回数を指定します。1 以上の値を指定します。-1 に設定すると、**messaging.defaultLimit** で指定されたデフォルトの制限数にリセットされます。

例

次の例では、指定したメッセージ ID のメッセージの最大表示回数を設定しています。

```
set_msg_limit -id "Netlist 29-28" 10000
```

次の例では、指定した重要度のメッセージの最大表示回数を設定しています。

```
set_msg_limit -severity WARNING 50000
```

次の例では、メッセージの現在のデフォルト最大表示回数を取得し、新しいデフォルト値を指定して、指定したメッセージ ID の最大表示回数をデフォルトから変更しています。

```
get_param messaging.defaultLimit
100
set_param messaging.defaultLimit 1000
set_msg_limit -id "common 17-81" 1500
```

関連項目

- [get_msg_config](#)
- [get_param](#)
- [reset_msg_limit](#)
- [set_msg_config](#)
- [set_param](#)

set_msg_severity

指定した ID のメッセージの重要度を設定します。

構文

```
set_msg_severity [-quiet] [-verbose] id severity
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>id</i>	メッセージ ID (「Common 17-54」など) を指定します。
<i>severity</i>	メッセージの重要度 (「ERROR」または「CRITICAL WARNING」など) を指定します。

カテゴリ

レポート

説明

set_msg_config コマンドではメッセージの重要度、最大表示回数を設定し、メッセージを非表示にできますが、このコマンドでは指定したメッセージ ID の重要度のみを変更します (「WARNING」から「ERROR」など)。

ツールで表示される各メッセージには、アプリケーション サブシステム コードとメッセージ識別子を含む固有のメッセージ ID が付けられています。次に、メッセージ ID の例を示します。

```
"Common 17-54"  
"Netlist 29-28"  
"Synth 8-3295"
```

このコマンドを使用すると、ツールで表示されるメッセージの重要度を特定のレベルに変更できます。

注記：特定の ID のメッセージの重要度は、**reset_msg_severity** コマンドを使用して元の設定に戻すことができます。

引数

id (必須) : メッセージ ID を指定します。各メッセージには、独自の ID が含まれます。たとえば、「Common 17-54」、「Netlist 29-28」です。

severity (必須) : メッセージの重要度を指定します。次の 5 種類の重要度があります。

- **ERROR** : デザインの結果が使用不可となったり、ユーザーの操作がないと回避できないような問題が発生していることを示すエラー メッセージです。

- **{CRITICAL WARNING}** : 入力や制約に適用されないものがあったり、FPGA ファミリには適していないものがあることを示すクリティカル警告メッセージです。修正することが強く推奨されます。

注記 : これは 2 単語の値なので、{} または "" で囲む必要があります。

- **WARNING** : 制約または仕様が指定どおりに適用されず、デザインが最適な結果にならない可能性を示す警告メッセージです。修正するかどうかは、ユーザーが判断します。

- **INFO** : STATUS メッセージと同じですが、重要度とメッセージ ID タグが含まれる点が異なります。INFO メッセージには、必要に応じてアンサー データベースで確認できるようにメッセージ ID が含まれます。

- **STATUS** : デザインの処理に関するプロセスおよびフィードバックのステータスを示すステータス メッセージです。STATUS メッセージには、メッセージ ID は含まれません。

注記 : STATUS メッセージにはメッセージ ID がないので、重要度を変更することはできません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、メッセージ ID 「**Common 17-54**」の重要度を CRITICAL WARNING から WARNING に変更し、該当する問題が発生したときにそれほど注意しなくてもいいようにしています。

```
set_msg_severity "Common 17-54" WARNING
```

次の例では、Common 17-81 メッセージをクリティカル警告に変更しています。

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

関連項目

- [get_msg_config](#)
- [reset_msg_severity](#)
- [set_msg_config](#)

set_multicycle_path

複数サイクル パスを定義します。

構文

```
set_multicycle_path [-setup] [-hold] [-rise] [-fall] [-start] [-end]
[-reset_path] [-from args] [-rise_from args] [-fall_from args] [-to args]
[-rise_to args] [-fall_to args] [-through args] [-rise_through args]
[-fall_through args] [-quiet] [-verbose] path_multiplier
```

戻り値

なし

使用法

名前	説明
[-setup]	セットアップのみの乗数を設定します。
[-hold]	ホールド タイムの算出に path_multiplier の値を適用します。
[-rise]	パスの終点の立ち上がりエッジ遅延にパスの乗数を適用します。
[-fall]	パスの終点の立ち下がりエッジ遅延にパスの乗数を適用します。
[-start]	パスの乗数を適用するパスの開始点を指定します。
[-end]	パスの乗数を適用するパスの終点を指定します。
[-reset_path]	複数サイクルを適用する前に、指定のパスをリセットします。
[-from]	パスの開始点またはクロックのリストを指定します。
[-rise_from]	開始点またはクロックのリストから立ち上がるパスに適用します。
[-fall_from]	開始点またはクロックのリストから立ち下がるパスに適用します。
[-to]	パスの終点またはクロックのリストを指定します。
[-rise_to]	終点またはクロックのリストで立ち上がるパスに適用します。
[-fall_to]	終点またはクロックのリストで立ち下がるパスに適用します。
[-through]	通過ピン、セル、またはネットのリストを指定します。
[-rise_through]	ピン、セル、またはネットを立ち上がりで通過するパスに適用します。
[-fall_through]	ピン、セル、またはネットを立ち下がりで通過するパスに適用します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
path_multiplier	サイクル数を指定します。

カテゴリ

SDC、XDC

説明

デフォルトでは、Vivado タイミング エンジンでシングル サイクル解析が実行され、セットアップ チェックが受信エッジのデスティネーションで送信クロックのエッジの 1 クロック サイクル後に実行されます。これは、タイミング パスによっては適切でない場合があります。たとえば、データがエンドポイントで安定するまでに複数クロック サイクル必要なロジック パスなどです。

このコマンドは、パスの乗数 *N* を指定し、タイミング パスで送信クロック エッジから受信クロック エッジまでに *N* クロック サイクルかかるることを示します。パスの乗数は、信号を開始点から終点に伝搬するために複数クロック サイクル必要な場合に、信号の伝搬に必要なクロック サイクル数を指定します。このコマンドの使用法の詳細は、『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。

set_multicycle_path コマンドは、セットアップ解析またはホールド解析用に、ソース クロックまたはデスティネーション クロックに対して、立ち上がりエッジまたは立ち下がりエッジにパスの乗数を指定します。このコマンドには、3 つの要素が含まれます。

- 複数サイクル パスが適用されるタイミング解析の指定
- 複数サイクル パスが適用されるタイミング パスのデスティネーション
- タイミング解析に適用するクロック サイクル数を定義するパスの乗数

パスの乗数は、セットアップ解析およびホールド解析の両方に適用されます。ホールド解析はセットアップ解析から導出されるので、セットアップ解析と共に移動されます。これにより、ホールド タイミング エラーが発生することがよくあります。2 つ目の **set_multicycle_path** コマンドを **-hold** オプションを指定して使用し、ホールド解析を元の位置に戻すことができます。

デフォルトでは、セットアップ パスの乗数はデスティネーション クロックに対して適用され、ホールド パスの乗数はソース クロックに対して適用されます。**-start** または **-end** オプションを使用すると、デフォルトのセットアップ 解析またはホールド解析をソース クロックまたはデスティネーション クロックを基準にするよう変更できます。

このコマンドが正常に実行された場合は何も返されず、正常に実行できない場合はエラーが返されます。

引数

-setup (オプション)：パスの乗数をセットアップ チェックにのみ適用します。

-hold (オプション)：パスの乗数をホールドチェックにのみ適用します。

注記： **-setup** または **-hold** のどちらも使用しない場合は、パスの乗数はセットアップ チェックおよびホールド チェックの両方に適用されます。

-rise (オプション)：パスの終点の立ち上がりエッジ遅延にパスの乗数を適用します。

-fall (オプション)：パスの終点の立ち下がりエッジ遅延にパスの乗数を適用します。

注記： **-rise** または **-fall** のどちらも指定しない場合、乗数は立ち上がりエッジ遅延および立ち下がりエッジ遅延の両方に適用されます。

-start (オプション)：デフォルトでは、セットアップ パスの乗数はデスティネーション クロック (**-end**) に対して定義されます。セットアップ要件をソース クロックに対して変更するには、**-start** オプションを使用する必要があります。

-end (オプション) : デフォルトでは、ホールド パスの乗数はソース クロックに対して定義されます。ホールド要件をデスティネーション クロックに対して変更するには、**-end** オプションを使用する必要があります。

注記 : 同じクロック、同じ波形の 2 つのクロック、または位相シフトのないクロックが供給されるパスに複数サイクル パス制約を適用する場合は、**-start/-end** オプションは影響しません。

-reset_path (オプション) : 複数サイクルの `path_multiplier` 値を適用する前に、指定のパスをリセットします。

-from args (オプション) : パスの乗数を適用するパスの開始点のリストを指定します。

-rise_from args (オプション) : 複数サイクル パスの乗数を適用する立ち上がりエッジ パスの開始点のリストを指定します。

-fall_from args (オプション) : 複数サイクル パスの乗数を適用する立ち下がりエッジ パスの開始点のリストを指定します。

-to args (オプション) : 複数サイクル パスの乗数を適用するパスの終点のリストを指定します。

-rise_to args (オプション) : 複数サイクル パスの乗数を適用する立ち上がりエッジ パスの終点のリストを指定します。

-fall_to args (オプション) : 複数サイクル パスの乗数を適用する立ち下がりエッジ パスの終点のリストを指定します。

-through args (オプション) : 複数サイクル パスの乗数を割り当てるパスが通過するピン、セル、またはネットのリストを指定します。

-rise_through args (オプション) : 複数サイクル パスの乗数を割り当てる立ち上がりパスが通過するピン、セル、またはネットのリストを指定します。

-fall_through args (オプション) : 複数サイクル パスの乗数を割り当てる立ち下がりパスが通過するピン、セル、またはネットのリストを指定します。

重要 : **-to**、**-through**、および **-from** (およびそのバリエーション) はオプションですが、**set_multicycle_path** 制約のタイミング パスを定義するため **-from**、**-to**、または **-through** のいずれかを指定する必要があります。指定しないと、エラーが返されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

path_multiplier (必須) : タイミング解析を移動するクロック サイクル数を指定します。

例

次の例では、**-from/-to** オプションで定義したタイミング パスのセットアップ チェックに対して 3 クロック サイクルのパスの乗数を定義しています。パスの乗数 $N-1$ (この例では 2) が、同じタイミング パスのホールド チェックに使用されます。

```
set_multicycle_path 3 -setup -from [get_pins data0_reg/C] -to [get_pins data1_reg/D]
set_multicycle_path 2 -hold -from [get_pins data0_reg/C] -to [get_pins data1_reg/D]
```

注記：セットアップ解析とホールド解析の関係については、『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。

関連項目

- [report_timing](#)
- [report_timing_summary](#)
- [set_input_delay](#)
- [set_output_delay](#)

set_operating_conditions

消費電力予測の動作条件を設定します。

構文

```
set_operating_conditions [-voltage args] [-grade arg] [-process arg]
[-junction_temp arg] [-ambient_temp arg] [-thetaJA arg] [-thetaSA arg]
[-airflow arg] [-heatsink arg] [-thetaJB arg] [-board arg] [-board_temp arg]
[-board_layers arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-voltage]	電圧ペアのリストを指定します (例 : {name value})。サポートされる電圧は、ファミリによって異なります。
[-grade]	電圧グレードを指定します。サポートされる値は、ファミリによって異なります。デフォルトは commercial です。
[-process]	プロセス データを指定します。有効な値は typical または maximum で、デフォルトは typical です。
[-junction_temp]	ジャンクション温度 (C) を指定します。有効な値は auto または実際の温度 (C) で、デフォルトは auto です。
[-ambient_temp]	周囲温度 (C) を指定します。有効な値は default または実際の温度 (C) で、デフォルトは default です。
[-thetaJA]	ThetaJA (C/W) を指定します。有効な値は auto または実際の値 (C/W) で、デフォルトは auto です。
[-thetaSA]	ThetaSA (C/W) を指定します。有効な値は auto または実際の値 (C/W) で、デフォルトは auto です。
[-airflow]	エアフロー (LFM) を指定します。有効な値は 0 ~ 750 で、デフォルトはファミリによって異なります。
[-heatsink]	ヒートシンクのサイズを指定します。有効な値は none、low、medium、high、custom で、デフォルトは medium です。
[-thetaJB]	ThetaJB (C/W) を指定します。有効な値は auto または実際の値 (C/W) で、デフォルトは auto です。
[-board]	ボード タイプを指定します。有効な値は jedec、small、medium、large、custom で、デフォルトは medium です。
[-board_temp]	ボード温度 (C) を指定します。
[-board_layers]	ボードの層数を指定します。有効な値は 4to7、8to11、12to15、16+ で、デフォルトは 8to11 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、XDC、電力

説明

デザインのパフォーマンスを解析する際に使用される実際の動作条件を設定します。デバイスの環境動作条件は、**report_power** コマンドによる消費電力解析では使用されますが、タイミング解析では使用されません。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

動作条件をデフォルト値に戻すには、**reset_operating_conditions** コマンドを使用します。

動作条件を確認するには、**report_operating_conditions** コマンドを使用します。

引数

-voltage arg (オプション)：電圧ペアのリストを指定します。サポートされる電圧は、ファミリによって異なります。

-grade arg (オプション)：ターゲット デバイスの温度グレードを指定します。サポートされる値は、ファミリによって異なります。デフォルト値は **commercial** です。

-process arg (オプション)：製造プロセス特性を指定します。有効な値は、**typical** または **maximum** です。デフォルト値は **typical** です。

-junction_temp arg (オプション)：モデリングに使用するデバイスのジャンクション温度を指定します。有効な値は **auto** または実際の温度 (摂氏) で、デフォルト値は **auto** です。

-ambient_temp arg (オプション)：環境周囲温度を摂氏で指定します。デフォルト値は **default** です。

-thetaja arg (オプション)：モデリングに使用する Theta-JA 熱抵抗を C/W で指定します。デフォルト値は **auto** です。

-thetas a arg (オプション)：モデリングに使用する Theta-SA 熱抵抗を C/W で指定します。デフォルト値は **auto** です。

-airflow [0:750] (オプション)：モデリングに使用するエアフローを LFM (リニア フィート/分) で指定します。デフォルト値はデバイス ファミリによって異なります。

-heatsink arg (オプション)：モデリングに使用するヒートシンクのプロファイルを指定します。有効な値は **none**、**low**、**medium**、**high**、**custom** で、デフォルト値は **medium** です。

-thetajb arg (オプション)：モデリングに使用する Theta-JB 熱抵抗を C/W で指定します。デフォルト値は **auto** です。

-board arg (オプション)：モデリングに使用するボードのサイズを指定します。有効な値は **jedec**、**small**、**medium**、**large**、**custom** で、デフォルト値は **medium** です。

-board_temp arg (オプション)：モデリングに使用するボードの温度を摂氏で指定します。

-board_layers arg (オプション)：モデリングに使用するボードの層数を指定します。有効な値は、4 ～ 7 層のボードでは 4to7、8 ～ 11 層のボードでは 8to11、12 ～ 15 層のボードでは 12to15、16 層以上のボードでは 16+ です。デフォルト値は 8to11 です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、インダストリアル グレードのデバイスに対して周囲動作温度を 75°C に設定しています。

```
set_operating_conditions -grade industrial -ambient_temp 75
```

関連項目

- [report_operating_conditions](#)
- [reset_operating_conditions](#)

set_output_delay

ポートの出力遅延を設定します。

構文

```
set_output_delay [-clock args] [-reference_pin args] [-clock_fall]
[-rise] [-fall] [-max] [-min] [-add_delay] [-network_latency_included]
[-source_latency_included] [-quiet] [-verbose] delay objects
```

戻り値

なし

使用法

名前	説明
[-clock]	相対クロックを指定します。
[-reference_pin]	相対ピンまたはポートを指定します。
[-clock_fall]	遅延がクロックの立ち下がりエッジを基準にすることを示します。
[-rise]	立ち上がり遅延を指定します。
[-fall]	立ち下がり遅延を指定します。
[-max]	最大遅延を指定します。
[-min]	最小遅延を指定します。
[-add_delay]	既存の入力遅延を削除しません。
[-network_latency_included]	クロックのネットワーク レイテンシが既に含まれていることを示します。
[-source_latency_included]	クロックのソース レイテンシが既に含まれていることを示します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
delay	遅延値を指定します。
objects	ポートのリストを指定します。

カテゴリ

SDC、XDC

説明

指定のポートの出力遅延を設定します。

注記： このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

- clock *arg* (オプション) : 遅延が指定のクロックの立ち上がりエッジを基準にすることを指定します。
 - reference_pin *arg* (オプション) : 遅延がクロックではなく指定のピンを基準にすることを指定します。
 - clock_fall (オプション) : 遅延が指定のクロックの立ち上がりエッジではなく立ち下がりエッジを基準にすることを示します。
 - rise (オプション) : 遅延が立ち上がりエッジ用であることを示します。
 - fall (オプション) : 遅延が立ち下がりエッジ用であることを示します。
 - max (オプション) : 指定の遅延を最大しきい値として処理します。
 - min (オプション) : 指定の遅延を最小しきい値として処理します。
 - add_delay (オプション) : 指定の遅延をパス上の既存の遅延と置き換えるのではなく、既存の遅延に追加するよう指定します。
 - network_latency_included (オプション) : 指定の遅延値に基準クロックのネットワークレイテンシが含まれることを示します。
 - source_latency_included (オプション) : 指定の遅延値に基準クロックのソース レイテンシが含まれることを示します。
 - quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。
- 注記** : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。
- verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。
- 注記** : メッセージの制限は、`set_msg_config` コマンドで定義できます。

delay (オプション) : 指定のポートに適用する遅延を指定します。有効な値は 0 以上の浮動小数点値で、デフォルト値は 1.0 です。

objects (必須) : 遅延値を適用するポートのリストを指定します。

例

次の例では、指定のクロックに対してポートの出力遅延を設定しています。

```
set_output_delay 5.0 -clock [get_clocks cpuClk] [get_ports]
```

次の例は上記の例とほぼ同じですが、ネットワークレイテンシが含まれています。

```
set_output_delay 5.0 -clock [get_clocks cpuClk] -network_latency_included [get_ports]
```

関連項目

- [get_ports](#)
- [set_input_delay](#)

set_package_pin_val

1 つまたは複数のパッケージ ピンのユーザー列を設定します。

構文

```
set_package_pin_val [-quiet] [-verbose] column value package_pins ...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
column	ユーザー列名を指定します。
value	設定する値を指定します。
package_pins	パッケージ ピン名を指定します。

カテゴリ

xdc、ピン配置

説明

ユーザー定義のパッケージ ピン属性を作成し、値をパッケージの特定のピンに割り当てます。

ユーザー定義のピン属性は CSV ファイルで定義でき、**read_csv** コマンドを使用して I/O ピン配置プロジェクトにインポートするか、このコマンドを使用してプロジェクトで変更できます。

注記： パッケージ ピンのツールで定義されたプロパティを設定するには、**set_property** コマンドを使用します。

引数

column (必須) : ユーザー定義の列名を指定します。列名では、大文字/小文字が区別されます。指定した列が存在しない場合は、パッケージ ピンに新しい属性が作成されます。指定したユーザー定義列名が既に存在する場合は、指定した値が指定したピンに割り当てられます。

注記： 列は、ツールの GUI の [Package Pins] ビューに表示される属性の列です。指定したパッケージ ピンの属性は、**write_csv** でエクスポートするなどできます。

value (必須) : 指定の列に割り当てる値を指定します。同じ列に対して異なるピンに異なる値を割り当てる場合は、**set_package_pin_val** コマンドを繰り返し使用します。

package_pins (必須) : 値を割り当てるパッケージ ピンを指定します。パッケージ ピンの指定には **get_package_pins** コマンドを使用できます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、[Package Pins] ビューに新しいユーザ一定義列を追加し、指定のピンに値 **true** を割り当てています。

```
set_package_pin_val -column track1 -value true -package_pins AK27
```

次の例では、**Test** というユーザ一定義列を作成し、すべての AK パッケージピンに値 RED を割り当て、指定した 3 つのピンの値を GREEN に変更しています。

```
set_package_pin_val -column Test -value RED -package_pins [get_package_pins AK*]
set_package_pin_val -column Test -value GREEN -package_pins {AK1 AK2 AK3}
```

関連項目

- [get_package_pins](#)
- [set_property](#)
- [write_csv](#)

set_param

パラメーター値を設定します。

構文

```
set_param [-quiet] [-verbose] name value
```

戻り値

新しく設定されたパラメーター値

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	パラメータ名を指定します。
<i>value</i>	パラメーター値を指定します。

カテゴリ

プロパティおよびパラメーター

説明

ユーザーが定義可能なコンフィギュレーション パラメーターの値を設定します。これらのパラメーターを使用すると、ツールのさまざまな動作をコンフィギュレーションおよび制御できます。現在定義されているパラメーターの説明は、**report_param** を参照してください。

reset_param コマンドを使用すると、変更されたパラメーターの値をデフォルト設定に戻すことができます。

注記： 指定したパラメーターの値を -1 に設定すると、その機能をオフにできます。

引数

name (必須)： 値を設定するパラメーターの名前を指定します。このコマンドで値を設定できるのは、一度に 1 つのパラメーターのみです。

value (必須)： 指定したパラメーターに設定する値を指定します。

-quiet (オプション)： コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記： コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、配置、配線、タイミング解析で、マルチスレッド プロセスで実行するスレッド数を定義するパラメーターを設定しています。

```
set_param general.maxThreads 4
```

注記 : Vivado ツールでは、1 ~ 4 のスレッドがサポートされます。現在の設定を確認するには、**get_param** を使用します。

次の例では、メッセージの最大表示回数の新しいデフォルト値を設定しています。

```
set_param messaging.defaultLimit 1000
```

関連項目

- [get_param](#)
- [list_param](#)
- [report_param](#)
- [reset_param](#)

set_power_opt

消費電力の最適化用に制約を設定します。

構文

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

戻り値

なし

使用法

名前	説明
[-include_cells]	クロック ゲーティングに含めるインスタンスを指定します。デフォルトは all です。
[-exclude_cells]	クロック ゲーティングから除外するインスタンスを指定します。デフォルトはなしです。
[-clocks]	指定したクロックが供給されるインスタンスにクロック ゲーティングを適用します。デフォルトでは、すべてのクロックが指定されます。
[-cell_types]	クロック ゲーティングを適用するタイプを指定します。all、bram、reg、srl、none のいずれかを指定します。デフォルトは all です。デフォルトは all です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

電力、[XDC](#)

説明

消費電力最適化に含めるセル インスタンスを指定します。指定したセルは、[power_opt_design](#) コマンドで最適化されます。

set_power_opt コマンドを複数回実行した場合、設定は累積的であり、まず最適化するセル タイプを幅広く設定し、特定の階層セルを含めて、その階層内のセルを除外するなど、消費電力最適化の対象を調整できます。

実行された消費電力最適化は、[report_power_opt](#) コマンドを使用してレポートできます。

引数

-include_cells args (オプション)： 指定したインスタンスのみをクロック ゲーティングに含めます。このオプションは、[power_opt_design](#) を使用して最適化する特定のセルまたはブロックをリストするために使用します。デフォルトでは、すべてのセルで消費電力最適化が実行されます。

-exclude_cells args (オプション) : 指定したインスタンスをクロック ゲーティングから除外します。デフォルトでは、消費電力最適化から除外されるセルはありません。**-exclude_cells** オプションは、現在含まれているセルから除外します。デフォルトではすべてのセルが含まれますが、**-include_cells** が指定されている場合は、**-exclude_cells** は現在含まれているセルのみに適用されます。

-clocks args (オプション) : 指定のクロックが供給されるインスタンスのみに消費電力最適化を実行します。デフォルトでは、デザインのすべてのクロックが含まれます。

注記 : **-clocks** と **-include_cells** の両方を使用して、指定のクロックが供給されないセルのリストを生成できますが、消費電力最適化は実行されません。

-cell_types [all | bram | reg | slr | none] (オプション) : 消費電力最適化を指定したセル タイプでのみ実行します。デフォルトでは、すべてのタイプのセルに対して消費電力最適化が実行されます。**all** または **none** を使用して、以前の **set_power_opt** コマンドの設定をリセットまたはクリアできます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、BRAM セルのみに対して消費電力最適化を設定し、消費電力最適化を実行しています。

```
set_power_opt -cell_types bram
power_opt_design
```

次の例では、BRAM および REG セルに対して消費電力最適化を設定し、SRL を追加して、消費電力化を実行しています。その後セルをクリアし、SRL のみを含めて消費電力化を再実行しています。

```
set_power_opt -cell_types { bram reg }
set_power_opt -cell_types { srl }
power_opt_design
set_power_opt -cell_types { none }
set_power_opt -cell_types { srl }
power_opt_design
```

次の例では、BRAM セルにのみに対して消費電力最適化を設定し、cpuEngine ブロックを最適化から除外し、cpuEngine/cpu_dbg_dat_i ブロックを追加して消費電力化を実行しています。

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

関連項目

- [power_opt_design](#)
- [report_power_opt](#)

set_propagated_clock

伝搬されるクロック レイテンシを指定します。

構文

```
set_propagated_clock [-quiet] [-verbose] objects
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>objects</i>	クロック、ポート、またはピンのリストを指定します。

カテゴリ

SDC、XDC

説明

クロック レイテンシをクロック ネットワーク全体に伝搬します。これにより、クロック ネットワーク全体でより正確なスキューホーリティおよびタイミング結果が得られます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須)：レイテンシを伝搬するクロック オブジェクトを指定します。

例

次の例では、最上位のプライマリ システム クロックを伝搬するよう指定しています。

```
set_propagated_clock [get_clocks top/clk]
```

次の例では、sublevel1 からのすべてのクロックを伝搬するよう指定しています。

```
set_propagated_clock [get_clocks sublevel1/*]
```

関連項目

- [get_clocks](#)
- [create_clock](#)

set_property

オブジェクトのプロパティを設定します。

構文

```
set_property [-dict args] [-quiet] [-verbose] name value objects...
```

戻り値

設定された値、エラーが発生した場合は ""

使用法

名前	説明
[-dict]	設定するプロパティの名前と値のペアのリストを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	設定するプロパティの名前を指定します。-dict オプションと共に使用することはできません。
<i>value</i>	設定するプロパティの値を指定します。-dict オプションと共に使用することはできません。
<i>objects</i>	プロパティを設定するオブジェクトの名前を指定します。

カテゴリ

オブジェクト、プロパティおよびパラメーター、XDC

説明

指定のオブジェクトにプロパティと値を設定します。

このコマンドは、デザインのオブジェクトに対してプロパティを定義するために使用できます。各オブジェクトにはあらかじめ定義されたプロパティがあり、指定可能な値または値の範囲があります。これらのプロパティの値を指定するには、set_property コマンドを使用します。オブジェクトのプロパティに設定されている値を確認するには、report_property、list_property、または list_property_values コマンドを使用します。

オブジェクトには、固有の *name* と *value* のペアを指定することにより、カスタム プロパティも設定できます。オブジェクトにカスタム プロパティがある場合、report_property または list_property コマンドのレポートに含まれます。

引数

-dict (オプション) : 1 つの set_property コマンドで、1 つのオブジェクトに対して複数のプロパティ (*name* と *value* のペア) を指定する際に使用します。*name* と *value* のペアは、中かっこ ({}) またはダブルクオーテーション (") で囲んで指定します。

```
-dict "name1 value1 name2 value2 ... nameN valueN"
```

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name (必須) : オブジェクトに設定するプロパティの名前を指定します。*name* では大文字/小文字が区別されるので、適切に指定してください。

value (必須) : 指定のオブジェクトの *name* プロパティに設定する値を指定します。プロパティタイプに対して値が有効であるかどうかがチェックされます。値が指定のプロパティに有効でない場合は、エラーが返されます。

重要 : プロパティの値にダッシュ (-) などの特殊文字が含まれる場合があり、これによってダッシュ以降の値がコマンドの別のオプションと解釈されてしまう場合があります。この場合、ここに説明するように、位置引数 (name、value、objects) ではなく明示的なオプション (**-name**、**-value**、**-objects**) を使用してください。これを「例」に示します。

objects (必須) : プロパティを割り当てる 1 つまたは複数のオブジェクトを指定します。

例

次の例では、セル オブジェクトに TRUTH というユーザー定義のブール型プロパティを作成し、セルに設定しています。

```
create_property -type bool truth cell
set_property truth false [lindex [get_cells] 1]
```

次の例では、**-dict** オプションを使用して、現在のデザインに複数のプロパティを一度に設定しています。

```
set_property -dict "POST_CRC enable POST_CRC_ACTION correct_and_continue" \
[current_design]
```

次の例では、現在のファイルセットの TOP プロパティを設定し、プロジェクトの最上位モジュールを定義しています。

```
set_property top fftTop [current_fileset]
set_property top_file {C:/Data/sources/fftTop.v} [current_fileset]
```

注記 : 最上位モジュールを定義するには、現在のプロジェクトのソース ファイルセットに含まれる階層ブロックに TOP プロパティを設定する必要があります。上記の例では、TOP がプロパティ名、fftTop が値、current_fileset がオブジェクトです。さらに、TOP_FILE プロパティを最上位モジュールのデータソースに指定する必要があります。

この例では、ダッシュ (-) を含むプロパティ値の設定方法を示します。ダッシュのために、ダッシュ以降の値が値の一部ではなく新しいコマンド オプションと解釈され、エラーが発生することがあります。この場合、set_property コマンドの位置引数の代わりに明示的なオプションを使用する必要があります。

```
set_property {XELAB.MORE_OPTIONS} {-pulse_e_style ondetect} [get_filesets sim_1]
ERROR: [Common 17-170] Unknown option '-pulse_e_style ondetect',
please type 'set_property -help' for usage info.
set_property -name {XELAB.MORE_OPTIONS} -value {-pulse_e_style ondetect}\
-objects [get_filesets sim_1]
```

次の例では、指定の I/O バンクに内部 VREF プロパティ値を設定しています。

```
set_property internal_vref {0.75} [get_iobanks 0]
```

次の例では、指定の I/O バンクの SLAVE_BANKS プロパティを設定することにより、DCI カスケードを定義しています。

```
set_property slave_banks {14} [get_iobanks 0 ]
```

次の例では、synth_1 run をコンフィギュレーションし、Vivado Synthesis 2013 オプションを設定し、その合成 run を実行しています。

```
set_property flow {Vivado Synthesis 2013} [get_runs synth_1]
set_property STEPS.SYNTH_DESIGN.ARGS.FANOUT_LIMIT 500 [get_runs synth_1]
set_property STEPS.SYNTH_DESIGN.ARGS.GATED_CLOCK_CONVERSION on [get_runs synth_1]
set_property STEPS.SYNTH_DESIGN.ARGS.FSM_EXTRACTION one_hot [get_runs synth_1]
launch_runs synth_1
```

関連項目

- [current_fileset](#)
- [create_property](#)
- [create_run](#)
- [get_cells](#)
- [get_property](#)
- [get_runs](#)
- [launch_runs](#)
- [list_property](#)
- [list_property_value](#)
- [report_property](#)
- [reset_property](#)

set_speed_grade

タイミング解析に使用するスピード グレードを指定します。

構文

```
set_speed_grade [-quiet] [-verbose] value
```

戻り値

結果文字列

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
value	タイミング解析に使用するスピード グレードを指定します。

カテゴリ

プロジェクト

説明

現在のデザインのターゲット デバイスのスピード グレードを設定します。このコマンドは、ターゲット デバイスのスピード グレードをタイミング解析用に変更するために使用し、開いている合成済みデザインまたはインプリメント済みデザインで実行する必要があります。通常は、report_timing コマンドまたはほかのタイミング コマンドの前に実行して、解析用のスピード グレードを変更します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、set_msg_config コマンドで定義できます。

value (必須) : ターゲット デバイスのスピード グレードを指定します。有効な値は、-1、-2、-3 です。

例

次の例では、現在のデザインのデバイスのスピード グレードを -1 に設定しています。

```
set_speed_grade -1
```

関連項目

[set_property](#)

set_switching_activity

指定したオブジェクトまたはデフォルトのタイプのスイッチング アクティビティを設定します。

構文

```
set_switching_activity [-type args] [-static_probability arg]
[-signal_rate arg] [-hier] [-quiet] [-verbose] [objects...]
```

戻り値

なし

使用法

名前	説明
[-type]	RTL 消費電力予測で使用します。有効なタイプ値は、registers、inputs、outputs、inouts、ports、outputEnable、three_states、dsps、brams、bramWrite、bramEnable、clockEnable です。
[-static_probability]	スタティック確率値を指定します。有効な値は 0 ~ 1 で、デフォルトは 1 です。
[-signal_rate]	エレメントのステートが 1 秒間に変化する回数を指定します。ザイリンクス ツールでは、秒ごとの遷移数が百万単位で示されます (Mtr/s)。デフォルトは 0.0 です。
[-hier]	<i>objects</i> で指定した階層インスタンスのすべてのレベルでスイッチング アクティビティを設定します。このオプションは、 <i>objects</i> と共に使用する必要があります。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[<i>objects</i>]	スイッチング アクティビティを設定するオブジェクトを指定します。

カテゴリ

xdc、電力

説明

消費電力予測を実行する際に使用する信号レートおよびスイッチング確率を設定します。レポートされるスイッチング アクティビティには、ネット、ポート、ピンの単純な信号レートと単純なスタティック確率、およびセルのステートに依存するスタティック確率が含まれます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

デザインのスイッチング アクティビティは、スタティックおよびダイナミック消費電力の両方に影響します。スタティック消費電力はロジック ステートの遷移によって異なり、ダイナミック消費電力はトグル レートに正比例します。

このコマンドは、消費電力予測を実行する際に使用する、信号の広範囲のクラスに対してデフォルトのアクティビティ レートを指定します。クラス全体ではなく、1 つ以上の信号のアクティビティを定義します。

現在のスイッチング アクティビティ属性は、**report_switching_activity** コマンドを使用して確認できます。スイッチング アクティビティをデフォルト値にリセットするには、**reset_switching_activity** コマンドを使用します。

引数

-toggle_rate rate (オプション) : 制御クロックに対して出力が切り替わる頻度を示すトグル レートを指定します。有効な値は 0 ~ 200% です。クロック サイクルごとに 1 回切り替わる出力のトグル レートは 100% です。デフォルト値は 0 です。

-type value (オプション) : スイッチング アクティビティを定義するロジック エンティティのタイプを指定します。有効な値は registers、inputs、outputs、inouts、ports、outputEnable、three_states、dsps、brams、bramWrite、bramEnable、clockEnable です。

-static_probability value (オプション) : 解析で使用するスイッチング確率を指定します。有効な値は 0 < value < 1 です。デフォルト値は 0 です。

-signal_rate value (オプション) : 解析で使用する信号周波数を指定します。デフォルト値は 0 です。

注記 : **-static_probability** または **-signal_rate** のいずれか、あるいは両方を指定する必要があります。

-hier (オプション) : スイッチング アクティビティを階層インスタンスのすべてのレベルの信号に適用します。**-hier** を使用しない場合、現在の階層レベルで指定の *objects* のスイッチング アクティビティがリセットされます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (オプション) : スイッチング アクティビティ設定を適用するポート、ピン、ネット、およびクロック オブジェクトのリストを指定します。

例

次の例では、すべてのポートに信号レートとスイッチング確率を指定し、ポートのスイッチング属性をレポートしています。

```
set_switching_activity -signal_rate 55 -static_probability .27 [get_ports]
report_switching_activity [get_ports]
```

関連項目

- [power_opt_design](#)
- [report_default_switching_activity](#)
- [report_power](#)
- [report_switching_activity](#)
- [reset_default_switching_activity](#)
- [reset_switching_activity](#)
- [set_default_switching_activity](#)

set_system_jitter

システム ジッターを設定します。

構文

```
set_system_jitter [-quiet] [-verbose] system_jitter
```

戻り値

システム ジッター

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
system_jitter	システム ジッターを設定します。有効な値は 0 以上です。

カテゴリ

xdc

説明

デザインのプライマリ クロックおよび生成クロックを含むすべてのクロックに対し、システム ジッターを時間の単位で設定します。システム ジッターは、電源ノイズやボード ノイズなど、FPGA 内のすべてのクロックに影響するノイズを考慮するために使用します。

set_system_jitter コマンドは、デザインのすべてのクロックに適用されます。特定のプライマリ クロックに追加のジッターを設定するには、**set_input_jitter** コマンドを使用します。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

jitter (必須)：システム全体に適用するシステム ジッターを時間の単位で指定します。デフォルトのシステム ジッターは、ザイリンクス FPGA のスピード ファイルで定義されています。**set_system_jitter** コマンドでジッターを指定すると、デフォルト値の代わりに指定した値が使用されます。

例

次の例では、デザインのすべてのクロックに対してシステム全体のジッター 0.2 を設定しています。

```
set_system_jitter 0.2
```

次の例では、プライマリ クロック sysClk を定義し、システム全体のジッターを 0.1 に設定しています。

```
create_clock -period 10 -name sysClk [get_ports sysClk]
set system_jitter 0.1
```

次の例では、プライマリ クロック sysClk を定義し、このプライマリ クロックを 2 で分周したクロック sysClkDiv2 を生成した後、デザインのすべてのクロックに対してシステム ジッター 0.2 を設定しています。プライマリ クロックにのみ、追加の入力ジッター 0.09 を指定しています。

```
create_clock -period 10 -name sysClk [get_ports sysClk]
create_generated_clock -name sysClkDiv2 -source [get_ports sysClk] -divide_by 2 \
[get_pins clkgen/sysClkDiv/Q]
set_system_jitter 0.2
set_input_jitter sysClk 0.09
```

次の例では、sysClk および procClk というプライマリ クロックを定義し、システムのすべてのクロックに対してシステム ジッター 0.2 を設定しています。クロック procClk にのみ、追加の入力ジッター 0.05 を指定しています。

```
create_clock -period 10 -name sysClk [get_ports sysClk]
create_clock -period 25 -name procClk [get_ports procClk]
set_system_jitter 0.2
set_input_jitter procClk 0.05
```

関連項目

- [report_timing](#)
- [set_input_jitter](#)
- [set_input_delay](#)
- [set_clock_uncertainty](#)

set_units

チェックに使用する単位を設定します。

構文

```
set_units [-capacitance arg] [-current arg] [-voltage arg] [-power arg]
[-resistance arg] [-suffix arg] [-digits arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-capacitance]	キャパシタンスの単位 (ファラード) を指定します。有効な値は kF ～ fF で、デフォルトは pF です。
[-current]	電流の単位 (アンペア) を指定します。有効な値は kA ～ fA で、デフォルトは mA です。
[-voltage]	電圧の単位 (ボルト) を指定します。有効な値は kV ～ fV で、デフォルトは V です。
[-power]	電力の単位 (ワット) を指定します。有効な値は kW ～ fW で、デフォルトは mW です。
[-resistance]	抵抗の単位 (オーム) を指定します。有効な値は kOhm ～ fOhm で、デフォルトは ohm です。
[-suffix]	単位の接尾辞を指定します。
[-digits]	有効桁数を指定します。デフォルトは 1 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

SDC、XDC

説明

デザインの解析で使用するデフォルトの単位を指定します。report_power コマンドで返される値は、-current、-voltage、-power、および -resistance オプションの設定により変更されます。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-capacitance *value* (オプション) : キャパシタンスの単位 (ファラード) を指定します。有効な値の範囲はキロファラード (kF) からフェムトファラード (fF) までで、デフォルトの容量の単位はピコファラード (pF) です。

-current *value* (オプション) : デフォルトの電流の単位 (アンペア) を指定します。有効な値の範囲はキロアンペア (kA) からフェムトアンペア (fA) までで、デフォルトの電流の単位はミリアンペア (mA) です。

-voltage *value* (オプション) : デフォルトの電圧の単位 (ボルト) を指定します。有効な値の範囲はキロボルト (kV) からフェムトボルト (fV) までで、デフォルトの電圧の単位はボルト (V) です。

-power *value* (オプション) : デフォルトの電力の単位 (ワット) を指定します。有効な値の範囲はキロワット (kW) からフェムトワット (fW) までで、デフォルトの電力の単位はミリワット (mW) です。

-resistance *value* (オプション) : デフォルトの抵抗の単位 (オーム) を指定します。有効な値の範囲はキロオーム (kOhm) からフェムトオーム (fOhm) までで、デフォルトの抵抗の単位はオーム (Ohm) です。

-suffix *value* (オプション) : 指定の単位に使用する接尾辞を指定します。

-digits *value* (オプション) : 単位を表示する桁数を指定します。デフォルト値は 1 です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

例

次の例では、電圧の単位をミリボルトに設定し、すべての値に 3 桁を使用するよう設定しています。

```
set_units -voltage mV -digits 3
```

次の例では、電流のデフォルト単位をアンペアに変更しています。

```
set_units -current A
```

2 つ目の `set_units` コマンドを実行した際、最初のコマンドで設定された値が無効になったり変更されたりすることはありません。

関連項目

- [report_power](#)
- [set_operating_conditions](#)

set_value

HDL オブジェクト (変数、信号、ワイヤ、またはレジスタ) の現在の値を指定の値に設定します。

構文

```
set_value [-radix arg] [-quiet] [-verbose] hdl_object value
```

戻り値

なし

使用法

名前	説明
[-radix]	値の解釈に使用する基底を指定します。有効な値は、default、dec、bin、oct、hex、unsigned、ascii です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
hdl_object	値を設定する HDL オブジェクトを指定します。
value	

カテゴリ

シミュレーション

説明

現在のシミュレーション時間における 1 つの HDL オブジェクトの値を設定します。

HDL オブジェクトには、Verilog または VHDL テストベンチおよびソース ファイルで定義されている HDL 信号、変数、または定数が含まれます。HDL 信号には、Verilog の wire または reg エンティティ、および VHDL 信号が含まれます。HDL 変数には、Verilog の real、realtime、time、event などがあります。

HDL 定数には、Verilog のパラメーターおよび localparam、VHDL ジェネリックおよび定数が含まれます。HDL スコープは、Verilog のモジュール、関数、タスク、プロセス、begin-end ブロックなど、HDL コードの宣言部分で定義されます。VHDL スコープには、エンティティ/アーキテクチャ定義、関数、プロシージャ、およびプロセスブロックが含まれます。

引数

-radix arg (オプション) : 指定したオブジェクトの値を表示するのに使用する基底を指定します。有効な値は、default、dec、bin、oct、hex、unsigned、ascii です。

注記 : **dec** は、符号付き 10 進数を示します。符号なしデータの場合は、**unsigned** を指定してください。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

hdl_object (必須) : 値を取得する 1 つの HDL オブジェクトの名前を指定します。オブジェクトは、名前または **get_objects** コマンドで指定します。

value (必須) : 指定したオブジェクトに設定する値を指定します。指定の値は、*hdl_object* のタイプによって異なります。HDL オブジェクトタイプには、logic、浮動小数点、VHDL 列挙型、VHDL 整数があります。logic 以外では、-radix オプションは無視されます。

- logic は、実際の HDL オブジェクトタイプを示すのではなく、次のような VHDL std_logic と同様の値を持つオブジェクトを示します。
 - Verilog 暗示 4 値ビット型
 - VHDL ビットおよび std_logic 既定義型
 - std_logic のサブセットである VHDL 列挙型 (文字リテラル 0 および 1 を含む)
- logic 型では、値は基数によって異なります。
 - 指定の値のビット数が logic 型で予測されるビット数より少ない場合、値は予測される長さにあわせて 0 で拡張されます (符号拡張されない)。
 - 指定の値のビット数が logic 型で予測されるビット数より多い場合、MSB 側の余分なビットはすべて 0 である必要があり、そうでない場合は Vivado シミュレータでサイズ不一致エラーが返されます。
- 浮動小数点オブジェクトの値は浮動小数点値です。
- logic でない VHDL 列挙型の値は、値の列挙セットからのスカラー値 (文字の場合はシングル クオーテーションなし) です。
- VHDL 整数型の値は、その型で許容される範囲の符号付き 10 進整数です。

例

次の例では、sysClk 信号の値を設定しています。

```
set_value sysClk z
```

次の例では、**bin**、**dec**、および **unsigned** 基数を使用して、指定のバスに同じ値を設定しています。

```
set_value -radix bin /test/bench_VStatus_pad_0_i[7:0] 10100101
set_value -radix unsigned /test/bench_VStatus_pad_0_i[7:0] 165
set_value -radix dec /test/bench_VStatus_pad_0_i[7:0] -91
```

次の例では、指定の値がロジック タイプに予測されるビット数よりも少ない場合にビット拡張が実行されることを示しています。

```
set_value -radix bin /test/bench_VStatus_pad_0_i[7:0] 101
get_value -radix bin /test/bench_VStatus_pad_0_i[7:0]
00000101
```

次の例では、指定の値がロジック タイプに予測されるビット数よりも多い場合にビット切り捨てが実行されることを示しています。

```
set_value -radix bin /test/bench_VStatus_pad_0_i[7:0] 0010100101
get_value -radix bin /test/bench_VStatus_pad_0_i[7:0]
10100101
set_value -radix bin /test/bench_VStatus_pad_0_i[7:0] 1110100101
ERROR: [#UNDEF] Object size 8 does not match size of given value 1110100101
```

注記：2 番目の **set_value** コマンドでは、余分なビットは 0 ではないので、エラーが返されます。

関連項目

- [current_time](#)
- [get_objects](#)
- [get_value](#)
- [report_values](#)

show_objects

[Find Results] ビューにオブジェクトを表示します。

構文

```
show_objects [-name arg] [-quiet] [-verbose] objects
```

戻り値

なし

使用法

名前	説明
[-name]	タブのタイトルを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	[Find Results] ビューに表示するオブジェクトを指定します。

カテゴリ

GUI 制御

show_schematic

ネットリスト アイテムを回路図で表示します。

構文

```
show_schematic [-add] [-remove] [-regenerate] [-pin_pairs] [-name arg]
[-quiet] [-verbose] objects
```

戻り値

なし

使用法

名前	説明
[-add]	既存の回路図に追加します。
[-remove]	既存の回路図から削除します。
[-regenerate]	回路図のレイアウトを再生成します。
[-pin_pairs]	オブジェクトを接続されたピンのペアとして処理します。これは、パスを表示する際に有益な場合があります。
[-name]	[Schematic] ビューのタイトルを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	回路図で表示するネットリスト アイテムを指定します。

カテゴリ

GUI 制御

説明

ツールを GUI モードで起動している場合に、指定のデザイン オブジェクトを含む回路図を作成します。

表示される回路図の範囲は、指定したオブジェクトによって異なります。セルから作成された回路図には、指定したセルとセル間の接続が表示されます。ピンから作成された回路図には、ピン オブジェクトが表示され、**-pin_pairs** を指定した場合はペアピンの接続が表示されます。ネットから作成された回路図には、指定したネットと、ネットに接続されているセルおよびポートが表示されます。

複数の階層レベルの回路図を表示するには、**current_instance** コマンドを使用して最上位階層または必要な階層レベルを設定し、**get_*** コマンドでデザイン オブジェクトを指定する際に **-hierarchical** オプションを使用します。

注記：このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-add (オプション)：指定のオブジェクトを回路図に追加します。

-remove (オプション) : 指定のオブジェクトを回路図から削除します。

-regenerate (オプション) : 回路図を再生成します。

-pin_pairs (オプション) : 接続されたピン オブジェクトのペアを指定した場合に、回路図にピンとその間のワイヤを表示します。**-pin_pairs** オプションを指定しない場合、または接続されていないピンを指定した場合は、ワイヤは表示されません。

-name arg (オプション) : GUI に表示される [Schematic] ビューの名前を指定します。回路図にオブジェクトを追加する場合、回路図からオブジェクトを削除する場合、回路図を再生成する場合は、この名前を使用します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (必須) : 回路図に表示するネットリスト オブジェクトを指定します。

例

次の例では、デザインの最上位の回路図を作成し、ネットおよびネットに接続されているポートおよびセルを表示しています。

```
show_schematic [get_nets]
```

次の例では、階層レベルを指定し、そのレベルから下の階層レベルを含む回路図を作成しています。

```
current_instance A
show_schematic [get_nets -hier]
```

次の例では、指定のピンと、それらを接続するワイヤを表示する回路図を作成しています。

```
show_schematic -pin_pairs [get_pins {data0_i/O data_reg/D}]
```

関連項目

- [current_instance](#)
- [get_cells](#)
- [get_nets](#)
- [get_pins](#)
- [get_ports](#)

split_diff_pair_ports

差動ペアの 2 つのポートをシングルエンド ポートに分割します。

構文

```
split_diff_pair_ports [-quiet] [-verbose] ports...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>ports</i>	分割するポートを指定します。

カテゴリ

ピン配置

説明

既存の差動ペア ポートを 2 つのシングルエンド ポートに分割します。

注記： このコマンドを実行しても、その動作に関するメッセージや戻り値は返されません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記： コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記： メッセージの制限は、**set_msg_config** コマンドで定義できます。

ports (必須)：シングル ポートに分割する差動ペアの 2 つのポート名を指定します。

例

次の例では、指定の差動ペア ポートを 2 つのシングルエンド ポートに分離しています。

```
split_diff_pair_ports PORT_N PORT_P
```

関連項目

- [make_diff_pair_ports](#)
- [create_port](#)
- [create_interface](#)

start_gui

GUI を起動します。

構文

start_gui [-verbose]

戻り値

なし

使用法

名前	説明
[-verbose]	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。

カテゴリ

[GUI 制御](#)

説明

ツールを Vivado Design Suite Tcl シェルで実行している場合に GUI を起動します。GUI は、現在のプロジェクト、デザイン、run の情報を読み込んで起動します。

引数

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、ツールを Tcl モードで実行しているときに、GUI を起動しています。

```
Vivado% start_gui
```

関連項目

[stop_gui](#)

start_vcd

VCD 出力の記述を開始します (\$dumpon Verilog システム タスクと同等)。open_vcd で開始した VCD の生成を stop_vcd Tcl コマンドで停止した後に使用できます。

構文

```
start_vcd [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

シミュレーション

説明

VCD (Value Change Dump) 情報の指定した VCD オブジェクトへの記述を開始します。この Tcl コマンドは、Verilog の \$dumpon システム タスクと同様の操作を実行します。

注記： start_vcd コマンドを実行する前に、open_vcd コマンドを実行する必要があります。

戻り値はありません。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、set_msg_config コマンドで定義できます。

例

次の例では、HDL 信号の現在の VCD ファイルへの記述を開始しています。

```
start_vcd
```

関連項目

- [close_vcd](#)
- [open_vcd](#)
- [stop_vcd](#)

startgroup

グループ単位で実行を取り消し/やり直しできるコマンド シーケンスを開始します。

構文

```
startgroup [-try] [-quiet] [-verbose]
```

戻り値

Int

使用法

名前	説明
[-try]	既にグループが開始されている場合は、開始しません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

GUI 制御

説明

グループ単位で実行を取り消し/やり直しできるコマンド シーケンスを開始します。コマンド シーケンスを終了するには、**endgroup** コマンドを使用します。

注記 : **undo** または **redo** で一度に取り消し/やり直しできるコマンド グループは複数作成できますが、コマンド グループをネストさせることはできません。**startgroup** を使用して新しいコマンド シーケンスを作成する前に、**endgroup** を使用してコマンド シーケンスを終了する必要があります。

startgroup コマンドを実行したとき、グループが既に開始している場合は整数値 0 が返され、新しいグループが開始された場合は整数値 1 が返されます。

引数

-try (オプション) : 新しいグループが開始された場合に 1 を返します。グループが既に開始している場合は 0 が返され、新しいグループは開始できません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、まず startgroup を実行し、関連するコマンドのシーケンスを実行して、endgroup を実行します。このコマンド シーケンスは、グループ単位で実行を取り消したりやり直したりすることができます。

```
startgroup
create_pblock pblock_wbArbEngine
create_pblock pblock_usbEngnSRM
add_cells_to_pblock pblock_wbArbEngine [get_cells [list wbArbEngine]] -clear_locs
add_cells_to_pblock pblock_usbEngnSRM [get_cells [list usbEnginel/usbEngineSRAM]] \
-clear_locs
endgroup
```

関連項目

- [endgroup](#)
- [redo](#)
- [undo](#)

step

シミュレーションを次の文に進めます。

構文

step [-quiet] [-verbose]

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

シミュレーション

stop

条件内で使用し、条件が満たされた場合にシミュレーションを一時停止するよう指定します。

構文

```
stop [-quiet] [-verbose]
```

戻り値

シミュレーションでの停止は終了ではなく一時停止

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

シミュレーション

stop_gui

GUI を閉じます。

構文

stop_gui [-verbose]

戻り値

なし

使用法

名前	説明
[-verbose]	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。

カテゴリ

[GUI 制御](#)

説明

GUI モードを停止し、ツールを Tcl モードにして、Vivado Design Suite Tcl シェルを実行します。Tcl モードでは、すべてのコマンドは Tcl コマンドまたは Tcl スクリプトで入力する必要があります。

引数

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、ツールの GUI モードを終了し、Tcl モードにしています。

`stop_gui`

関連項目

[start_gui](#)

stop_hw_sio_scan

ハードウェア SIO スキャンを停止します。

構文

```
stop_hw_sio_scan [-quiet] [-verbose] hw_sio_scans
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hw_sio_scans</i>	ハードウェア SIO スキャンを指定します。

カテゴリ

ハードウェア

stop_vcd

VCD 出力の記述を停止します (\$dumpoff Verilog システム タスクと同等)。VCD 出力の記述を再開するには、**start_vcd** Tcl コマンドを使用します。

構文

stop_vcd [-quiet] [-verbose]

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

シミュレーション

説明

現在の VCD (Value Change Dump) ファイルへのシミュレーション値の記述を停止します。シミュレーション情報のファイルへの出力が、**start_vcd** コマンドで再開されるまで停止します。

この Tcl コマンドは、Verilog の **\$dumpoff** システム タスクと同様の操作を実行します。

注記 : **stop_vcd** コマンドを実行する前に、**open_vcd** コマンドを実行する必要があります。

戻り値はありません。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、シミュレーション値の現在の VCD ファイルへの記述を停止しています。

```
stop_vcd
```

関連項目

- [close_vcd](#)
- [open_vcd](#)
- [start_vcd](#)

swap_locs

2 つの位置を入れ替えます。

構文

```
swap_locs [-quiet] [-verbose] aloc bloc
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
aloc	1 つ目の位置 (ポート、セル、サイト) を指定します。bloc と同じタイプである必要があります。
bloc	2 つ目の位置 (ポート、セル、サイト) を指定します。aloc と同じタイプである必要があります。

カテゴリ

フロアプラン

説明

2 つの類似したロジック エレメントに設定されている LOC 制約を入れ替えます。ロジック エレメントとは、FPGA のデバイス リソースに配置可能なエレメントです。

swap_locs コマンドを実行すると、選択された 2 つのエレメントを新しいロケーションに配置できるかどうかを確認するため、一部の DRC チェックが実行されます。いずれかのエレメントのロケーションが何らかの理由で無効である場合、**swap_locs** コマンドでエラーが返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

aloc (必須) : 入れ替える 1 つ目のロジックのロケーションを指定します。ポート、セル、またはデバイス サイトとして指定できます。

bloc (必須) : 入れ替える 2 つ目のロジックのロケーションを指定します。ポート、セル、またはデバイス サイトとして指定できます。*aloc* で指定したタイプと一致している必要があります。

例

次の例では、指定した 2 つのデバイス サイトに割り当てられているインスタンスを入れ替えています。

```
swap_locs [get_sites {OLOGIC_X2Y1}] [get_sites {OLOGIC_X2Y0}]
```

関連項目

- [get_cells](#)
- [get_ports](#)
- [get_sites](#)

synth_design

Vivado 合成を使用してデザインを合成し、デザインを開きます。

構文

```
synth_design [-name arg] [-part arg] [-constrset arg] [-top arg]
[-include_dirs args] [-generic args] [-verilog_define args]
[-flatten_hierarchy arg] [-gated_clock_conversion arg] [-directive arg] [-rtl]
[-bufg arg] [-no_lc] [-fanout_limit arg] [-shreg_min_size arg] [-mode arg]
[-fsm_extraction arg] [-keep_equivalent_registers] [-resource_sharing arg]
[-control_set_opt_threshold arg] [-quiet] [-verbose]
```

戻り値

デザイン オブジェクト

使用法

名前	説明
[-name]	デザイン名を指定します。
[-part]	ターゲット パーツを指定します。
[-constrset]	使用する制約ファイルセットを指定します。
[-top]	最上位モジュール名を指定します。
[-include_dirs]	Verilog 検索ディレクトリを指定します。
[-generic]	ジェネリック パラメーターを指定します。構文は、-generic <name>=<value> -generic <name>=<value> ... です。
[-verilog_define]	Verilog の ‘define 文の値を指定します。構文は、-verilog_define <macro_name>[=<macro_text>] -verilog_define <macro_name>[=<macro_text>] ... です。
[-flatten_hierarchy]	LUT マップ中に階層をフラット化します。有効な値は full、auto、none、rebuilt で、デフォルトは rebuilt です。
[-gated_clock_conversion]	クロック ゲーティング ロジックをフリップフロップ イネーブルに変換します。有効な値は off、on、auto で、デフォルトは off です。
[-directive]	合成の目標を指定します。有効な値は default、runtimeoptimized で、デフォルトは default です。
[-rtl]	エラボレーションを実行し、RTL デザインを開きます。
[-bufg]	合成で使用するグローバル クロック バッファーの最大数を指定します。デフォルトは 12 です。
[-no_lc]	LUT の結合をディスエーブルにします。LUT ペアを 1 つのデュアル出力 LUT に結合しません。
[-fanout_limit]	最大ファンアウトを指定します。デフォルトは 10000 です。
[-shreg_min_size]	SRL にマップされるレジスタ チェーンの最小長さを指定します。デフォルトは 3 です。

名前	説明
[-mode]	デザイン モードを指定します。有効な値は default、out_of_context で、デフォルトは default です。
[-fsm_extraction]	FSM 抽出エンコードを指定します。有効な値は off、one_hot、sequential、johnson、gray、auto で、デフォルトは auto です。
[-keep_equivalent_registers]	同じロジックがソースとなるレジスタが統合されないようにします。合成の KEEP 属性を使用してもレジスタの統合を回避できます。
[-resource_sharing]	算術演算子を共有します。有効な値は auto、on、off で、デフォルトは auto です。
[-control_set_opt_threshold]	同期制御セット最適化のしきい値の制御セット数を削減します。デフォルトは 1 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ツール

説明

Vivado® 合成エンジンを直接起動し、プロジェクト モードまたは非プロジェクト モードのデザインをコンパイルおよび合成します。プロジェクト モードおよび非プロジェクト モードの詳細は、『Vivado Design Suite ユーザー ガイド：デザイン フローの概要』(UG892) を参照してください。

非プロジェクト モードでは、**synth_design** コマンドで Vivado 合成を直接起動できます。

プロジェクト モードでは、**create_run** コマンドを使用して作成した合成 run から合成を実行します。**launch_runs** コマンドを使用して run を実行すると、Vivado 合成に対しては **synth_design** が呼び出されます。

引数

-name arg (オプション) : **synth_design** コマンドが完了したときに開くデザインの名前を指定します。

-part arg (オプション) : デザインに使用するターゲット ザイリンクス デバイスを指定します。このオプションを指定しない場合、プロジェクトで設定されたデフォルトのパートが使用されます。

-constrset arg (オプション) : デザインを合成する際に使用する XDC 制約の名前を指定します。Vivado 合成では XDC を使用する必要があり、UCF はサポートされません。**-constrset** オプションには、既存の制約ファイルセットを指定する必要があります。新しいファイルセットを作成することはできません。新しいファイルセットを作成する場合は、**create_fileset** コマンドを使用します。

-top arg (オプション) : デザイン階層の最上位モジュールを指定します。

注記 : **find_top** コマンドを使用して **-top** オプションを指定する場合は、**find_top** で複数の候補が返された場合に 1 つのみが指定されるようにしてください。例を参照してください。

-include_dirs args (オプション) : Verilog インクルード ファイルを検索するディレクトリを指定します。

-generic name=value (オプション) : VHDL ジェネリック エンティティまたは Verilog パラメーターの値を定義します。**-generic** オプションの構文は `name=value` で、ジェネリックまたはパラメーターの名前とその値を指定します。**synth_design** コマンドで、各ジェネリックまたはパラメーター値に対してそれぞれ **-generic** オプションを指定します。

```
synth_design -generic width=32 -generic depth=512 ...
```

重要 : ブール型または std_logic VHDL ジェネリック型に 2 進数値を指定する場合は、標準 VHDL フォーマットではなく Verilog ビット フォーマットを使用して値を指定する必要があります。

```
0 = 1'b0
01010000 = 8'b01010000
```

-verilog_define name=text (オプション) : Verilog の ‘define 文および ‘ifdef 文の値を指定します。**-verilog_define** オプションの構文は `name=text` で、define 文の名前とその値を指定します。この引数は、1 つの **synth_design** コマンドで複数回使用できます。

```
synth_design -verilog_define name=value -verilog_define name=value ...
```

-flatten_hierarchy arg (オプション) : LUT マップ中にデザインの階層をフラット化します。有効な値は、次のとおりです。

- **rebuilt** : 合成が完了した後に RTL デザインの元の階層を再構築するよう試みます。これがデフォルト設定です。
- **full** (オプション) : デザインの階層をフラットにします。
- **none** (オプション) : デザインの階層をフラットにしません。これによりデザインの階層が保持され増すが、合成ツールで実行できるデザインの最適化が制限されます。

-gated_clock_conversion arg (オプション) : クロック ゲーティング ロジックを変換してフリップフロップ イネーブルを使用します。この最適化によりロジックが削除され、ネットリストが簡略化されます。有効な値は **off**、**on**、**auto** です。デフォルト値は **off** です。この最適化は、**opt_design** コマンドを使用して合成済みネットリストに対して実行することも可能です。

-directive [default | runtimeoptimized] (オプション) : 特定のデザイン目標を達成するよう合成します。1 つの **synth_design** コマンドに対して 1 つのモードのみを指定できます。値では大文字/小文字が区別されます。有効な値は **default** または **runtimeoptimized** です。**runtimeoptimized** に設定すると、実行されるタイミング最適化は少なくなり、一部の RTL 最適化が実行されます。

-rtl (オプション) : HDL ソース ファイルをエラボレートし、RTL デザインを開きます。

-bufg arg (オプション) : 合成で使用するグローバル クロック バッファーの最大数を指定します。1 からターゲット デバイスに含まれる BUFG の数までの値を指定します。デフォルト値は 12 です。

-no_lc (オプション) : デフォルトの LUT の結合機能をディスエーブルにします。

-fanout_limit arg (オプション) : 合成中に適用される最大ネット ファンアウトを指定します。1 以上の値を指定します。デフォルト値は 10,000 です。

-shreg_min_size arg (オプション) : SRL にマップされるレジスタ チェーンの最小長さを整数で指定します。デフォルト値は 3 です。

-mode [default | out_of_context] (オプション) : out_of_context モードを指定すると、階層デザイン フローで使用するモジュールが合成されます。このモードでは、モジュールに I/O バッファーは挿入されず、モジュールが OOC とマークされ、HD フローで使用できるようになります。詳細は、『Vivado Design Suite ユーザー ガイド：階層デザイン』(UG905) を参照してください。

-fsm_extraction arg (オプション) : Vivado 合成では、デフォルトで有限ステートマシン (FSM) エンコードがディスエーブル (off) になっています。このオプションはステートマシンの識別を有効にし、適用するエンコードを指定します。有効な値は off、one_hot、sequential、johnson、gray、auto です。自動エンコード (auto) に設定すると、各ステートマシンに最適なエンコードが選択されます。この場合、同じデザインの異なる FSM に異なるエンコードが使用されることがあります。

-keep_equivalent_registers (オプション) : KEEP 合成属性のように機能し、最適化中にレジスタが統合されないようにします。

-resource_sharing arg (オプション) : 加算器や減算器などの算術演算子を異なる信号間で共有します。このオプションをオンにすると、エリア使用率が向上する場合があります。有効な値は auto、on、off で、デフォルトは auto です。

-control_set_opt_threshold arg (オプション) : 同期制御セット最適化のしきい値の制御セット数を削減します。制御セットとして使用し始める前に、制御セットのファンアウトの大きさを指定します。たとえば、**-control_set_opt_threshold** を 10 に設定すると、3 つのレジスタにのみファンアウトする同期リセットは、レジスタのリセット ラインを使用するのではなく、D 入力ロジックに移動されます。1 に設定すると、リセット ラインが使用されます。有効な値は 1 以上の整数で、デフォルトは 1 です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、**set_property** コマンドを使用してアクティブ プロジェクトのターゲット パーツを指定し、ソース ファイルをエラボレートして、RTL デザインを開いています。

```
set_property part xc7vx485tffg1158-1 [current_project]
synth_design -rtl -name rtl_1
```

注記 : この例では、デフォルトのソース セット、制約セット、パーツが使用されます。

次の例では、**find_top** コマンドを使用して合成用に現在のデザインの最上位モジュールを定義しています。

```
synth_design -top [lindex [find_top] 0]
```

注記 : **find_top** では複数の候補が返されますが、インデックス 0 を指定すると合成に最適な候補が選択されます。

次の例では、最上位モジュールとターゲット デバイスを定義し、階層をフラット化しないように指定して、現在のデザインを合成します。合成 run の結果がネットリスト デザインとして開きます。

```
synth_design -top top -part xc7k70tfg676-2 -flatten_hierarchy none
open_run synth_1 -name netlist_1
```

関連項目

- [create_run](#)
- [current_design](#)
- [current_project](#)
- [find_top](#)
- [open_run](#)
- [opt_design](#)
- [set_property](#)

tie_unused_pins

未使用のセル ピンを接続します。

構文

```
tie_unused_pins [-of_objects args] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-of_objects]	指定したセルの未使用ピンを接続します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

ネットリスト

説明

開いている合成済みデザインまたはインプリメント済みデザインで、セルの未接続のピンを接続します。このコマンドでは、内部プロセスを使用して、ピンを High に接続するか Low に接続するかを特定します。

このコマンドは、**create_cell** コマンドでネットリストに追加されたセルの未接続ピンを接続するために使用します。

引数

-of_objects args (オプション) : 現時点では無視されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、セルの未使用のピンを、その使用状況によって High または Low に接続しています。

```
tie_unused_pins
```

関連項目

- [create_cell](#)
- [create_pin](#)
- [get_cells](#)
- [get_pins](#)

undo

前のコマンドの実行を取り消します。

構文

```
undo [-list] [-quiet] [-verbose]
```

戻り値

取り消すことができるタスクのリスト (-list を使用した場合)

使用法

名前	説明
[-list]	取り消すことができるタスクのリストを表示します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

GUI 制御

説明

前に実行したコマンドを取り消します。このコマンドを繰り返して使用し、一連のコマンドを取り消すことができます。

startgroup および **endgroup** コマンドを使用してコマンド グループを作成した場合、**undo** コマンドでコマンド グループがシーケンスとして取り消されます。**undo** コマンドは **endgroup** コマンドから開始し、**startgroup** コマンドに到達するまで繰り替えされます。

あるコマンドに対して **undo** を実行し、その後そのコマンドを実行することにした場合は、**redo** コマンドを使用できます。

引数

-list (オプション) : 取り消すことができるコマンドのリストを返します。**undo** コマンドを使用すると、コマンドのリストを順にさかのぼってコマンドが取り消されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、**TCL_OK** が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、取り消すことが可能なコマンドのリストが返されます。

```
undo -list
```

関連項目

- [redo](#)
- [startgroup](#)
- [endgroup](#)

ungroup_bd_cells

階層セル内のセルのグループを親セルに移動し、階層セルを削除します。セル間の接続は保持されます。これらのセルとその他のセルの接続は、階層セルをまたいで保持されます。

構文

```
ungroup_bd_cells [-prefix arg] [-quiet] [-verbose] [cells...]
```

戻り値

正しく処理された場合は 0

使用法

名前	説明
[-prefix]	セルに追加する接頭辞を指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[cells]	セル名を検索するパターンを指定します。デフォルトは*です。

カテゴリ

IP インテグレーター

説明

group_bd_cells または **move_bd_cells** コマンドで階層モジュールにグループ化された IP インテグレーター セルのグループを解除します。選択した階層モジュール内のセルを親セルのレベルに移動し、階層モジュールを削除します。

選択したセル間の接続は保持されます。これらのセルとほかのセルの間の接続は、不必要的サブシステム ポートおよびピンを削除することにより自走的に保持されます。

このコマンドが正常に実行された場合は 0 が返され、正常に実行されなかった場合はエラー メッセージが返されます。

引数

-prefix arg (オプション) : 階層を 1 レベル上に移動するセルに適用する接頭辞を指定します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

cells (必須) : **group_bd_cells** または **move_bd_cells** コマンドで定義された階層モジュールを指定します。このコマンドで一度に指定できるのは 1 つの階層モジュールのみです。

例

次に例を示します。

```
ungroup_bd_cells -prefix up2_ [get_bd_cells myMod2]
```

関連項目

- [get_bd_cells](#)
- [group_bd_cells](#)
- [move_bd_cells](#)

unhighlight_objects

現在ハイライトされているオブジェクトのハイライトを解除します。

構文

```
unhighlight_objects [-color_index arg] [-rgb args] [-color arg] [-quiet]
[-verbose] [objects]
```

戻り値

なし

使用法

名前	説明
[-color_index]	色を色インデックスで指定します。
[-rgb]	色を RGB で指定します。
[-color]	有効な値は、red、green、blue、magenta、yellow、cyan、および orange です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	ハイライトを解除するオブジェクトを指定します。

カテゴリ

[GUI 制御](#)

説明

このコマンドは GUI モードで使用します。highlight_objects コマンドでハイライトしたオブジェクトのハイライトを解除します。

このコマンドでは、下に説明する色オプションがサポートされており、指定の色でハイライトされているオブジェクトすべてのハイライトを解除するために使用できます。例を参照してください。

引数

-color_index arg (オプション) : 有効な値は 1 ~ 19 の整数で、ハイライトを解除する色を色インデックスで指定します。色インデックスは、[Tools] → [Options] → [Themes] の [Highlight] の下に定義されています。テーマの設定方法の詳細は、『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。

-rgb args (オプション) : ハイライトを解除する色を、RGB コードを使用して {R G B} の形式で指定します。たとえば、{255 255 0} は黄色を指定します。

-color arg (オプション) : ハイライトを解除する色を色の名前で指定します。指定可能な値は、red、green、blue、magenta、yellow、cyan、および orange です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (オプション) : ハイライトを解除するオブジェクトを指定します。オブジェクトを指定しない場合は、指定した色でハイライトされているオブジェクトすべてのハイライトが解除されます。色を指定しない場合は、すべてのオブジェクトのハイライトが解除されます。

例

次の例では、選択したオブジェクトのハイライトが解除されます。

```
unhighlight_objects [get_selected_objects]
```

次の例では、現在黄色でハイライトされているオブジェクトのハイライトが解除されます。

```
unhighlight_objects -color yellow
```

関連項目

- [get_selected_objects](#)
- [highlight_objects](#)

unmark_objects

現在マークされているアイテムのマークを解除します。

構文

```
unmark_objects [-rgb args] [-color arg] [-quiet] [-verbose] [objects]
```

戻り値

なし

使用法

名前	説明
[-rgb]	色を RGB で指定します。
[-color]	有効な値は、red、green、blue、magenta、yellow、cyan、および orange です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	マークを解除するオブジェクトを指定します。

カテゴリ

GUI 制御

説明

mark_objects コマンドでマークしたオブジェクトのマークを解除します。このコマンドは GUI モードで使用します。

このコマンドでは、下に説明する色オプションがサポートされますが、これらのオプションは指定のオブジェクトのマークを解除するには必要ありません。これらの色オプションは、指定の色でマークされているオブジェクトすべてのマークを解除するために使用できます。例を参照してください。

引数

-rgb args (オプション) : マークを解除する色を、RGB コードを使用して {R G B} の形式で指定します。たとえば、{255 255 0} は黄色を指定します。

-color arg (オプション) : マークを解除する色を色の名前で指定します。指定可能な値は、red、green、blue、magenta、yellow、cyan、および orange です。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (オプション) : マークを解除する 1 つまたは複数のオブジェクトを指定します。オブジェクトを指定しない場合は、指定した色でマークされているオブジェクトすべてのマークが解除されます。色を指定しない場合は、すべてのオブジェクトのマークが解除されます。

例

次の例では、選択したオブジェクトのマークが解除されます。

```
unmark_objects [get_selected_objects]
```

次の例では、現在黄色でマークされているオブジェクトのマークが解除されます。

```
unmark_objects -color yellow
```

関連項目

- [get_selected_objects](#)
- [mark_objects](#)

unplace_cell

1 つまたは複数のインスタンスの配置を解除します。

構文

```
unplace_cell [-quiet] [-verbose] cell_list...
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
cell_list	配置を解除するセルのリストを指定します。

カテゴリ

フロアプラン

説明

指定したセルを現在配置されている配置サイトから解除します。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

cell_list (必須) : デバイスから配置を解除するセルを 1 つまたは複数指定します。

例

次の例では、指定したセルの配置を解除しています。

```
unplace_cell {fftEngine/fftInst/ingressLoop[6].ingressFifo/buffer_fifo/i_4773_12897}
```

次の例では、複数のセルの配置を解除しています。

```
unplace_cell {div_cntr_reg_inferredi_4810_15889 div_cntr_reg[0] div_cntr_reg[1]}
```

関連項目

- [create_cell](#)
- [place_cell](#)
- [remove_cell](#)

unselect_objects

現在選択されているアイテムの選択を解除します。

構文

```
unselect_objects [-quiet] [-verbose] [objects]
```

戻り値

なし

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	選択を解除するオブジェクトを指定します。

カテゴリ

GUI 制御

説明

select_objects コマンドで選択したオブジェクトの選択を解除します。

プライマリ オブジェクトとセカンダリ オブジェクト両方の選択が解除されます。セカンダリ オブジェクトの選択は、[Tools] → [Options] → [Selection Rules] で定義します。選択規則の設定方法は、『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。

引数

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (オプション)：選択を解除する 1 つまたは複数のオブジェクトを指定します。オブジェクトを指定しない場合は、すべてのオブジェクトの選択が解除されます。

例

次の例では、デバイス上の指定したサイトの選択が解除されます。

```
unselect_objects [get_sites SLICE_X56Y214]
```

次の例では、現在選択されているオブジェクトすべての選択が解除されます。

```
unselect_objects
```

関連項目

- [get_selected_objects](#)
- [select_objects](#)

update_compile_order

デザイン グラフに基づいて、ファイルセットのコンパイル順および場合によっては最上位をアップデートします。

構文

```
update_compile_order [-fileset arg] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-fileset]	デザイン グラフに基づいてアップデートするファイルセットを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

説明

現在のプロジェクトまたは指定のファイルセットでデザイン ソースのコンパイル順をアップデートします。

引数

-fileset arg : 指定のファイルセットのコンパイル順をアップデートします。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、シミュレーション ファイルセットのソース ファイルのコンパイル順をアップデートしています。

```
update_compile_order -fileset sim_1
```

関連項目

- [add_files](#)
- [import_files](#)

update_design

現在のデザインのネットリストをアップデートします。

構文

```
update_design -cells args [-strict] [-from_file arg] [-from_design arg]
[-from_cell arg] [-black_box] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
-cells	新しいサブネットリストでアップデートするセルのリストを指定します。
[-strict]	セルを置き換えるのに完全に一致したポートを必要とします。このオプションを設定しない場合、余分なポートが許可されます。
[-from_file]	新しいサブネットリストを含むファイルの名前を指定します。
[-from_design]	新しいサブネットリストを含む、開いているネットリスト デザインの名前を指定します。
[-from_cell]	新しいサブネットリストを定義する from_design のセルの名前を指定します。
[-black_box]	セルをブラック ボックスにアップデートします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

説明

メモリ内のデザインをアップデートし、指定のセルの現在のネットリストを指定のファイルまたは開いている別のデザインからのネットリストに置き換えるか、またはセルをブラック ボックス セルに変換します。1 つのインスタンス、複数のインスタンス、またはマスター セルのすべてのインスタンスをアップデートできます。

メモリ内のデザインのみが新しいネットリストに変更されます。**write_checkpoint** コマンドを使用してデザインを保存してください。そうしないと、プロジェクトを閉じたとき、またはツールを終了したときにアップデートは失われます。

引数

-cells args (必須) : 指定のネットリストでアップデートする階層インスタンス名またはセルのリストを指定します。セルのすべてのインスタンスをアップデートするには、**get_cells** コマンドで **-filter** を使用してマスター セルを指定します。

```
get_cells -hier -filter {ref_name==aMasterCellName} <cell_name>
```

-strict (オプション) : 新しいネットリストがインポート先のセルと同じポートを持つことを必須にします。新しいネットリストに指定のセルに必要なポートがすべてあるかどうかがチェックされます。**-strict** オプションを使用しない場合、追加ポートは許容されます。

-from_file (オプション) : 新しいネットリストを含むファイルの名前を指定します。構造 Verilog ネットリスト (.v) または EDIF ネットリスト (.edf) ファイルを指定できます。

注記 : **-from_file** と **-from_design** オプションを同時に使用することはできません。

-from_design (オプション) : 現在のプロジェクトの開いている別のデザインからネットリストをインポートします。デザインを別のプロセスではなく現在のツール セッションで開いている必要があります。

-from_cell (オプション) : **-from_design** で指定したデザインのセルの名前を指定します。現在のデザインのセルが指定したセルからのネットリストでアップデートされます。デフォルトでは、**-from_design** で指定したデザインの最上位セルが使用されます。

注記 : このオプションは、**-from_design** オプションを使用している場合にのみ使用可能です。

-black_box (オプション) : 指定のセルをブラック ボックス セルにアップデートします。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、arnd4 セルを指定した Verilog ネットリストでアップデートしています。

```
update_design -cells arnd4 -from_file C:/Data/round_4.v
```

次の例では、現在のデザインに含まれる arnd4 セルを、指定したデザインの同じセルからのネットリストでアップデートしています。

```
update_design -cells arnd4 -from_design netlist_2 -from_cell arnd4
```

関連項目

- [get_cells](#)
- [write_checkpoint](#)

update_files

プロジェクトのファイルを、指定のファイルまたはディレクトリからの同じ名前のファイルでアップデートします。

構文

```
update_files [-from_files args] [-norecurse] [-to_files args]
[-filesets args] [-force] [-report_only] [-quiet] [-verbose]
```

戻り値

アップデートされたファイルのリスト

使用法

名前	説明
[-from_file]	アップデートに使用するファイルおよびディレクトリを指定します。
[-norecurse]	下位ディレクトリでは検索を実行しないよう指定します。
[-to_files]	アップデートする既存のプロジェクトファイルおよびディレクトリを指定します。
[-filesets]	ファイルセット名を指定します。
[-force]	プロジェクトのインポートされたファイルを、読み取り専用であっても可能な限り上書きします。
[-report_only]	実際のファイル アップデートは実行せず、実行されるはずのアップデートをレポートします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

プロジェクト

update_ip_catalog

IP カタログをアップデートします。このコマンドを実行する前に、オプションで「`set_property ip_repo_paths <repo_path_list> [current_fileset]`」を使用してリポジトリを設定できます。

構文

```
update_ip_catalog [-rebuild] [-add_ip arg] [-delete_ip arg]
[-delete_mult_ip args] [-create_index] [-repo_path arg] [-quiet] [-verbose]
```

戻り値

正しくアップデートされた場合は True

使用法

名前	説明
<code>[-rebuild]</code>	指定したリポジトリのインデックス ファイルを再構築します。リポジトリが指定されていない場合は、すべてのリポジトリを再構築します。
<code>[-add_ip]</code>	指定した IP を指定したリポジトリに追加します。IP の component.xml へのパスまたは IP を含む ZIP ファイルへのパスを指定します。
<code>[-delete_ip]</code>	指定した IP を指定したリポジトリから削除します。IP の component.xml へのパスまたは VLVN を指定します。
<code>[-delete_mult_ip]</code>	指定した IP を指定したリポジトリから削除します。IP のリストを component.xml ファイルへのパスまたは VLVN で指定します。
<code>[-create_index]</code>	指定したリポジトリのデータをディスクにキャッシュし、読み込み時間を短縮します。
<code>[-repo_path]</code>	<code>-rebuild</code> 、 <code>-add_ip</code> 、 <code>-delete_ip</code> 、 <code>-delete_mult_ip</code> 、または <code>-create_index</code> オプションと共に使用し、操作を実行するリポジトリへのパスを指定します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。

カテゴリ

IP フロー

説明

現在のデザインに関連付けられている IP カタログをアップデートします。

デフォルトの IP カタログは、ツールのインストール ディレクトリにありますが、`set_property` コマンドを使用して現在のソース ファイルセットの IP_REPO_PATHS プロパティを設定して、IP カタログに新しい IP リポジトリのパスを追加できます。

ザイリンクス IP カタログ (リポジトリ) は、使用しているツール バージョンのインストール ディレクトリにあります。また、下の例に示すように、**set_property** コマンドを使用してソース ファイルセットの IP_REPO_PATHS プロパティをカスタム IP の場所に設定して、カスタム IP をリポジトリに追加できます。

引数

-rebuild (オプション) : IP カタログ インデックス全体を再構築するか、**-repo_path** で指定された IP リポジトリのインデックスのみを再構築します。

-add_ip *arg* (オプション) : IP コアを指定した IP リポジトリに追加します。このオプションを使用する場合は、**-repo_path** オプションも指定する必要があります。IP は、IP の `component.xml` へのパスまたは IP を含む ZIP ファイルへのパスで指定します。

-delete_ip *arg* (オプション) : IP コアを指定した IP リポジトリから削除します。このオプションを使用する場合は、**-repo_path** オプションも指定する必要があります。IP は、IP の `component.xml` へのパスまたは IP の VLVN プロパティで指定します。VLVN プロパティは Vendor:Library:Name:Version 形式の文字列で、カタログ内の IP を識別します。

-delete_mult_ip *arg* (オプション) : 指定した IP コアを IP リポジトリから削除します。このオプションを使用する場合は、**-repo_path** オプションも指定する必要があります。IP は、`component.xml` ファイルへのパスまたは IP の VLVN プロパティで指定します。VLVN プロパティは Vendor:Library:Name:Version 形式の文字列で、カタログ内の IP を識別します。

-create_index (オプション) : 指定のリポジトリのデータをディスクにキャッシュし、読み込み時間を短縮します。このオプションを使用する場合は、**-repo_path** オプションも指定する必要があります。

-repo_path *arg* (オプション) : **-rebuild**、**-add_ip**、**-delete_ip**、**-delete_mult_ip**、または **-create_index** オプションと共に使用し、操作を実行するリポジトリのディレクトリ名を指定します。

注記 : IP リポジトリは、**set_property** コマンドを使用して現在のソース ファイルセットに追加しておく必要があります。例を参照してください。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、現在のソース ファイルセットの IP_REPO_PATHS プロパティを設定し、IP リポジトリを追加して、IP カタログ全体のインデックスを再構築しています。

```
set_property IP_REPO_PATHS C:/Data/IP_LIB [current_fileset]
update_ip_catalog -rebuild
```

関連項目

- [create_ip](#)
- [import_ip](#)
- [generate_target](#)
- [validate_ip](#)

update_macro

マクロをアップデートします。

構文

```
update_macro [-absolute_grid] [-quiet] [-verbose] macro rlocs
```

戻り値

なし

使用法

名前	説明
<code>[-absolute_grid]</code>	相対ロケーションに絶対座標を使用します。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>macro</code>	アップデートするマクロを指定します。
<code>rlocs</code>	インスタンスとサイトの名前を交互に指定します。

カテゴリ

xdc

説明

作成済みのマクロに最下位セルおよび相対配置を指定します。

マクロは、プリミティブまたは最下位ロジック セル、その接続、配置座標での位置で構成されます。指定の相対ロケーション (*rlocs*) は、相対座標または絶対座標 (RPM_GRID) に基づいて指定できます。絶対および相対配置座標の詳細は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

1 つのセルは、1 つのマクロにのみ割り当てることができます。最下位セルを複数のマクロに割り当てようとすると、エラーが返されます。プリミティブでないセルをマクロに割り当てようとすると、エラーが返されます。

既存のマクロの内容を変更するには、`delete_macro` を使用してマクロを削除し、`create_macro` で再作成し、新しい内容でアップデートする必要があります。既存のマクロを単に上書きしたり変更したりすることはできません。

引数

`-absolute_grid` (オプション) : *rlocs* を絶対 RPM_GRID で指定することを示します。

`-quiet` (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

macro (必須) : アップデートするマクロの名前を指定します。

rlocs (必須) : マクロに含める最下位セルとその相対ロケーション (RLOC) を指定します。**-absolute_grid** を指定している場合、ロケーションは相対ロケーションではなく実際のデバイス座標に基づきます。セルおよび RLOC は名前と値のペアで指定し、複数の最下位セルと RLOC を 1 つのマクロに割り当てるには次のように指定します。

```
{cell0 XmYn cell1 XmYn ... cellN XmYn}
```

説明 :

- m : その前にリストされているセルの相対または絶対 X 座標を示す整数
- n : その前にリストされているセルの相対または絶対 Y 座標を示す整数
- **-absolute_grid** オプションを指定していない場合、セルの座標はお互いのセルに相対します。
- 相対座標は、それぞれに相対して配置されたスライスの仮想配列に基づきます。
- 絶対座標は、ターゲット デバイスの実際のスライスの RPM_X および RPM_Y プロパティにより決定されます。これらのプロパティは、選択したサイトに対して **report_property** コマンドを使用して確認できます。

例

次の例では、usbMacro0 という名前のマクロを作成し、現在のインスタンスを usbEngine0/u0 モジュールに設定して、3 つのセルをマクロに割り当て、相対配置で 2 つのセルを同じスライスに、3 つ目のセルを縦に隣接するスライスに配置しています。

```
create_macro usbMacro0
current_instance usbEngine0/u0
update_macro usbMacro0 {rx_active_reg X0Y0 rx_err_reg X0Y0 rx_valid_reg X0Y1}
```

次の例では、usbMacro1 という名前のマクロを作成し、最下位セルに階層パスを使用して 3 つのセルをマクロに割り当て、絶対座標を使用してセルをマクロ内に配置しています。

```
create_macro usbMacro1
set Site1 [get_sites SLICE_X8Y77]
set Site2 [get_sites SLICE_X9Y77]
set Site3 [get_sites SLICE_X8Y78]
set RPM1 X[get_property RPM_X $Site1]Y[get_property RPM_Y $Site1]
set RPM2 X[get_property RPM_X $Site2]Y[get_property RPM_Y $Site2]
set RPM3 X[get_property RPM_X $Site3]Y[get_property RPM_Y $Site3]
update_macro usbMacro1 -absolute_grid "usbEngine1/u0/rx_active_reg $RPM1 \
usbEngine1/u0/rx_err_reg $RPM2 usbEngine1/u0/rx_valid_reg $RPM3"
```

注記 : 上記の例では、Tcl 変数を使用してサイトをキャプチャし、それらのサイトの RPM_X および RPM_Y プロパティの値を **update_macro** コマンドで使用しています。**update_macro** コマンドで中かっこ {} の代わりにダブルクオーテーション ("") を使用しており、これによって Tcl シェルでコマンドの変数置換を実行できます。変数および変数置換の詳細は、『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。

次の例では、usbMacro1 マクロのプロパティをレポートし、セルに割り当てられている絶対座標を確認しています。

```
report_property -all [get_macros usbMacro1]
```

関連項目

- [create_macro](#)
- [delete_macros](#)
- [get_macros](#)
- [get_property](#)
- [get_sites](#)
- [place_design](#)
- [report_property](#)

update_timing

タイミングをアップデートします。

構文

```
update_timing [-full] [-quiet] [-verbose]
```

戻り値

なし

使用法

名前	説明
[-full]	インクリメンタル なタイミング アップデートではなく、完全なタイミング アップデートを実行します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

[タイミング](#)

upgrade_bd_cells

コンフィギュラブル IP インテグレーター セルを最新バージョンにアップグレードします。

構文

```
upgrade_bd_cells [-latest arg] [-quiet] [-verbose] objects...
```

戻り値

アップグレードされた IP インテグレーター セルの名前、エラーが発生した場合は ""

使用法

名前	説明
[-latest]	IP インテグレーター ブロックを最新バージョンにアップグレードします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
objects	アップグレードする IP インテグレーター セルを指定します。

カテゴリ

IP インテグレーター

説明

IP インテグレーター セルを IP インテグレーター カタログで使用可能な最新バージョンにアップグレードします。

このコマンドを使用すると、IP インテグレーター サブシステム デザインで IP コアを以前のリリースから最新のカタログの IP コアにアップグレードできます。

このコマンドが正常に実行された場合はアップグレードされた IP インテグレーター セルのリストが返され、正常に実行されなかった場合はエラーが返されます。

引数

-latest (オプション) : 指定したセルを IP カタログで使用可能な最新バージョンにアップグレードします。これがデフォルトの動作であり、指定する必要はありません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects: 最新バージョンにアップグレードする IP インテグレーター セルを指定します。オブジェクトは、`get_bd_cells` コマンドを使用して指定する必要があります。名前で指定することはできません。

例

次の例では、指定したセルを IP インテグレーター カタログで使用可能な最新バージョンにアップグレードしています。

```
upgrade_bd_cells [get_bd_cells {vidOut1 cmpy_1 newMod1}]
INFO: [BD 41-1162] The cell '/vidOut1' is already at its latest version.
INFO: [BD 41-1162] The cell '/cmpy_1' is already at its latest version.
WARNING: [BD 41-1082] Hierarchy block (/newMod1) cannot be directly upgraded. Please dive into
the hierarchy and select individual cells to upgrade.
```

注記: ユーザー定義の階層モジュールに対しては、警告メッセージが返されます。

関連項目

[get_bd_cells](#)

upgrade_ip

コンフィギュラブル IP を最新バージョンにアップグレードします。

構文

```
upgrade_ip [-srcset arg] [-vlnv arg] [-quiet] [-verbose] [objects...]
```

戻り値

正常に完了したか、エラーが発生したかを示す戻りコード

使用法

名前	説明
[-srcset]	アップグレードするオブジェクトを含むソースファイル セットを指定します。デフォルトは、現在のソースファイルセットです。
[-vlnv]	アップグレードした IP の IP カタログの VLNV 文字列を指定します。VLNV 文字列は IP コアの IPDEF プロパティにマップされています。デフォルトは現在の IP の最新バージョンです。<vendor>:<library>:<name>:<version> という形式の文字列で指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[objects]	アップグレードする IP を指定します。デザイン内の IP インスタンスは、get_ip <instance name> で指定します。

カテゴリ

IP フロー

説明

指定した IP コアを IP カタログの最新バージョンにアップグレードします。

アップグレードできるのは、アップグレードが明示的にサポートされている IP のみです。ipdef オブジェクトの UPGRADE VERSIONS プロパティを確認すると、IP コアのアップグレード バージョンがあるかどうかがわかります。

注記：一部の IP ではアップグレードはサポートされていません。

引数

-srcset arg (オプション) : IP ファイルをアップグレードするソースファイルセットを指定します。指定しない場合、デフォルトのソースファイルセットは sources_1 です。

-vlnv arg (オプション) : IP カタログからアップグレードする IP の Vendor:Library:Name:Version 属性を指定します。VLNV 属性は、IP カタログ内のオブジェクトを識別します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

objects (オプション) : アップグレードする IP コアを指定します。

例

次の例では、現在のプロジェクトに含まれるすべての IP を最新バージョンにアップグレードしています。

```
upgrade_ip
```

関連項目

- [create_ip](#)
- [import_ip](#)

upload_hw_ila_data

キャプチャを停止し、キャプチャされたハードウェア ILA データをアップロードします。

構文

```
upload_hw_ila_data [-quiet] [-verbose] [hw_ilas...]
```

戻り値

ハードウェア ILA データ オブジェクト

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_ilas]	ハードウェア ILA オブジェクトのリストを指定します。デフォルトは、現在のハードウェア ILA です。

カテゴリ

ハードウェア

validate_bd_design

指定のデザインまたは指定のセルに対してパラメーターの伝搬を実行します。

構文

```
validate_bd_design [-design arg] [-cell arg] [-quiet] [-verbose]
```

戻り値

TCL_OK、エラーが発生した場合は TCL_ERROR

使用法

名前	説明
[-design]	デザイン名を指定します。指定しない場合、パラメーターの伝搬は現在のデザインに対して実行されます。
[-cell]	セル名を指定します。指定しない場合、パラメーターの伝搬はデザイン全体に対して実行されます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

IP インテグレーター

説明

IP インテグレーター サブシステム デザイン、IP セル、または階層モジュールを検証します。

このコマンドが正常に実行された場合は TCL_OK が返され、正常に実行されなかった場合は、-quiet が指定されていなければ TCL_ERROR が返されます。

引数

-design arg (オプション) : 検証する IP インテグレーター サブシステム デザインを指定します。指定しない場合、現在の IP インテグレーター サブシステム デザインが検証されます。

-cell arg (オプション) : 検証する IP インテグレーター セルまたは階層モジュールを指定します。指定しない場合、デザイン全体が最上位から検証されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記: メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、IP インテグレーター サブシステム デザインを検証し、デザインに関する警告およびエラーを返しています。

```
validate_bd_design
INFO: [IP_Flow 19-3236] Customization errors found. Restoring to previous valid configuration.
ERROR: [IP_Flow 19-508] Validation failed for parameter 'Write_Depth_A'. Value '268435456' is out
of the range (2,9011200)
ERROR: [Common 17-39] 'set_property' failed due to earlier errors.
ERROR: [BD 41-66] Error running post_propagate TCL procedure: ERROR: [Common 17-39] 'set_property'
failed due to earlier errors.
    while executing
"rdi::set_property CONFIG.WRITE_DEPTH_A 268435456 /microblaze_1_local_memory/lmb_bram"
    invoked from within
"set_property CONFIG.WRITE_DEPTH_A $mem_depth_a $ip"
    (procedure "::xilinx.com_ip_blk_mem_gen_8.0::post_propagate" line 49)
    invoked from within
"::xilinx.com_ip_blk_mem_gen_8.0::post_propagate /microblaze_1_local_memory/lmb_bram {}"
WARNING: [BD 41-680] Ranges of mapped paired address blocks </microblaze_1_local_memory/
ilmb_bram_if_cntlr/SLMB/Mem> and </microblaze_1_local_memory/dlmb_bram_if_cntlr/SLMB/Mem>
do not match.
ERROR: [BD 41-241] Message from IP propagation TCL of /microblaze_1_local_memory/lmb_bram: set_property
error: Validation failed for parameter 'Write_Depth_A'. Value '268435456' is out of the range (2,9011200)
INFO: [xilinx.com:ip:axi_intc:3.1-1] /microblaze_1_axi_intc: The AXI INTc core has been configured
to operate with synchronous clocks.
ERROR: [Common 17-39] 'validate_bd_design' failed due to earlier errors.
```

関連項目

- [create_bd_design](#)
- [open_bd_design](#)
- [save_bd_design](#)

validate_ip

IP が有効であることを確認します。

構文

```
validate_ip [-save_ip] [-quiet] [-verbose] [ips]
```

戻り値

なし

使用法

名前	説明
<code>[-save_ip]</code>	IP ファイルをディスクに書き込みます。
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>[ips]</code>	確認する IP を指定します。

カテゴリ

IP フロー

説明

IP に対して DRC チェックを実行し、正しく構築されているかどうかを確認します。すべての IP が正しく構築されている場合は 1、問題がある場合は 0 が返されます。

引数

`-save_ip` (オプション) : 検証した後に既存の IP ファイルをアップデートします。新しいファイルは作成されず、コアの XCI および BOM ファイルがアップデートされます。

`-quiet` (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

`-verbose` (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

`ips` (オプション) : 検証する IP コアを指定します。

例

次の例では、現在のプロジェクトに含まれる IP を検証し、IP のファイルをアップデートしています。

```
validate_ip -save_ip [get_ips]
```

関連項目

- [create_ip](#)
- [generate_target](#)
- [upgrade_ip](#)
- [update_ip_catalog](#)
- [import_ip](#)

version

Vivado のバージョンおよびバージョンの日付を返します。

構文

```
version [-short] [-quiet] [-verbose]
```

戻り値

Vivado のバージョン

使用法

名前	説明
[-short]	バージョン番号のみを返します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。

カテゴリ

[レポート](#)

説明

ザイリンクス ツールのバージョン番号を返します。バージョン番号、ビルドの番号と日付、および著作権情報が含まれます。

引数

-short (オプション) : ツールのバージョン番号のみを返します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

例

次の例では、ツールのバージョン番号のみが返されます。

```
version -short
```

wait_on_hw_ilab

すべてのハードウェア ILA データがキャプチャされるまで待機します。

構文

```
wait_on_hw_ilab [-timeout arg] [-quiet] [-verbose] [hw_ilas...]
```

戻り値

なし

使用法

名前	説明
[-timeout]	タイムアウト時間を分で指定します。10 進数を指定します。デフォルトではタイムアウトはありません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[hw_ilas]	ハードウェア ILA オブジェクトを指定します。デフォルトは、現在のハードウェア ILA です。

カテゴリ

ハードウェア

wait_on_hw_sio_scan

ハードウェア VIO スキャンが完了するまで待機します。

構文

```
wait_on_hw_sio_scan [-timeout arg] [-quiet] [-verbose] hw_sio_scans ...
```

戻り値

なし

使用法

名前	説明
[-timeout]	タイムアウト時間を分で指定します。10 進数を指定します。デフォルトではタイムアウトはありません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>hw_sio_scans</i>	ハードウェア SIO スキャン オブジェクトのリストを指定します。

カテゴリ

ハードウェア

wait_on_run

指定した run が終了するまで Tcl コマンドの実行を停止します。

構文

```
wait_on_run [-timeout arg] [-quiet] [-verbose] run
```

戻り値

なし

使用法

名前	説明
[-timeout]	run が完了するまで待機する最大時間 (分) を指定します。デフォルトは -1 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
run	待機する run を指定します。

カテゴリ

プロジェクト

説明

指定した run が完了するまで、または指定した時間が経過するまで、Tcl コマンドの実行を停止します。

wait_on_run コマンドは、起動されている run に使用できます。**wait_on_run** コマンドを実行したときに指定した run が起動されていない場合は、エラー メッセージが表示されます。既に完了している run に大してはエラー メッセージは表示されません。

注記：このコマンドは、バッチ モードまたは Tcl スクリプトでツールを実行する場合に使用します。GUI からインタラクティブに実行した場合は、無視されます。

引数

-timeout arg (オプション) : run が終了するまで **wait_on_run** コマンドで待機する時間 (分) を指定します。これにより、指定した run が終了していないくとも、この時間を越えるとツールが Tcl の実行を再開します。タイムアウトを指定しない場合 (run が終了するまでのツールの待機時間を制限しない場合)、デフォルト値の -1 が使用されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

run (必須) : 待機する run の名前を指定します。

例

次の例では、`impl_1` という run が起動され、指定した run が終了するまでか、1 時間経過するか、どちらかが発生するまで Tcl の実行が停止されます。

```
launch_runs impl_1
wait_on_run -timeout 60 impl_1
```

関連項目

[launch_runs](#)

write_bd_tcl

現在のデザインをディスク上の Tcl ファイルにエクスポートします。

構文

```
write_bd_tcl [-quiet] [-verbose] name
```

戻り値

TCL_OK、エラーが発生した場合は TCL_ERROR

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>name</i>	エクスポート先の Tcl ファイルの名前を指定します。

カテゴリ

IP インテグレーター

説明

現在の IP インテグレーター サブシステム デザインをディスク上の Tcl スクリプトファイルにエクスポートします。

Tcl スクリプト ファイルを使用すると、IP インテグレーター サブシステム デザインの再作成、再使用、カスタマイズを実行でき、元のサブシステム デザインをアーカイブする必要はありません。

このコマンドが正常に実行された場合は TCL_OK が返され、正常に実行されなかった場合は、-quiet が指定されていなければ TCL_ERROR が返されます。

引数

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

name : 記述する Tcl ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、現在の IP インテグレーター サブシステム デザインから Tcl スクリプトを作成しています。

```
write_bd_tcl C:/Data/myDesign.tcl
```

関連項目

- [create_bd_design](#)
- [open_bd_design](#)
- [save_bd_design](#)

write_bitstream

現在のデザインのビットストリームを生成します。

構文

```
write_bitstream [-force] [-raw_bitfile] [-no_binary_bitfile] [-mask_file]
[-logic_location_file] [-bin_file] [-reference_bitfile arg] [-quiet]
[-verbose] file
```

戻り値

なし

使用法

名前	説明
[-force]	既存のファイルを上書きします。
[-raw_bitfile]	ロー ビット ファイル (.rbt) を生成します。
[-no_binary_bitfile]	バイナリ ビット ファイル (.bit) を生成しません。
[-mask_file]	マスク ファイル (.msk) を生成します。
[-logic_location_file]	ロジック ロケーション ファイル (.ll) を生成します。
[-bin_file]	ヘッダーのないバイナリ ビット ファイル (.bin) を生成します。
[-reference_bitfile]	パーシャル ビットストリームを生成する際に使用する基準 ビット ファイルを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	生成する BIT ファイルの名前を指定します。

カテゴリ

ファイル入力および出力

説明

現在のプロジェクトのビットストリーム ファイルを生成します。このコマンドは、インプリメント済み デザインで実行する必要があります。ビットストリームは、開いているインプリメント済み デザインに基づいて出力されます。

引数

-force (オプション) : 同じ名前の既存のビットストリーム ファイルを上書きします。

-raw_bitfile (オプション) : ロー ビット ファイル (.rbt) を生成します。ロー ビット ファイルには、バイナリ ビットストリーム ファイルと同じ情報が ASCII 形式で含まれます。出力ファイル名は *file.rbt* となります。

-no_binary_bitfile (オプション) : バイナリビットストリーム ファイル (.bit) を生成しません。このオプションは、バイナリビットストリーム ファイルを生成せずに、ASCII 形式のビットストリーム ファイルまたはマスク ファイルを生成したり、ビットストリーム レポートを生成したりする場合に使用します。

-mask_file (オプション) : マスク ファイル (.msk) を生成します。マスク ファイルには、ビットストリーム ファイルのコンフィギュレーション データが含まれる場所を示すマスク データが含まれます。このファイルは、検証時にビットストリームのどのビットをリードバック データと比較するべきかを判断するために使用します。マスク ビットが 0 の場合はそのビットはビットストリーム データに対して検証され、マスク ビットが 1 の場合はそのビットは検証されません。出力ファイル名は *file.msk* となります。

-logic_location_file (オプション) : ラッチ、フリップフロップ、LUT、ブロック RAM、I/O ブロック入力および出力のビットストリーム位置を示す ASCII 形式のロジック ロケーション ファイル (.ll) を生成します。このロケーション ファイルではビットがフレームおよびビット番号で示され、FPGA レジスタの内容を調べるために役立ちます。

-bin_file (オプション) : デバイス プログラム データのみを含むバイナリファイル (.bin) を生成します。通常のビットストリーム ファイル (.bit) に含まれるヘッダー情報は含まれません。

-reference_bitfile *arg* (オプション) : 参照ビットストリーム ファイルを読み込み、指定した参照ファイルからの変更部分のみを含むインクリメンタル ビットストリーム ファイルを生成します。このパーシャル ビットストリームは、既存のデバイスをアップデートされたデザインでインクリメンタルにプログラムする場合に使用できます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 生成するビットストリーム ファイル (.bit) の名前を指定します。ファイルの拡張子を指定しない場合は、.bit が追加されます。.bit 以外の拡張子を指定することはできません。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、指定した名前のビットストリーム ファイルが生成されます。

```
write_bitstream design1.bit
```

次の例では、バイナリおよび ASCII 形式のビットストリーム ファイルを生成しています。

```
write_bitstream -raw_bitfile C:/Data/design
```

注記 : 適切なファイル拡張子が追加されます。

関連項目

[launch_runs](#)

write_bmm

BMM ファイルを生成します。

構文

```
write_bmm [-force] [-quiet] [-verbose] file
```

戻り値

BMM ファイル名

使用法

名前	説明
[-force]	既存の BMM ファイルを上書きします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	デザインの BMM ファイルを指定します。英数字を使用した名前に拡張子 (.bmm) を付けて指定します。

カテゴリ

ファイル入力および出力

write_checkpoint

現在のデザインのチェックポイントを書き出します。

構文

```
write_checkpoint [-force] [-quiet] [-verbose] file
```

戻り値

チェックポイント ファイル名

使用法

名前	説明
[-force]	既存のチェックポイント ファイルを上書きします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>file</i>	デザイン チェックポイント ファイルを指定します。英数字を使用した名前に拡張子 (.dcp) を付けて指定します。

カテゴリ

ファイル入力および出力

説明

デザインをデザイン プロセスの任意の段階で保存し、必要に応じてツールにすばやくインポートし直せるようにします。デザイン チェックポイント (DCP) には、ネットリスト、制約、およびインプリメント済みデザインの配置配線情報が含まれます。

チェックポイント ファイルをインポートするには、**read_checkpoint** コマンドを使用します。

引数

-force (オプション) : 同じ名前のチェックポイント ファイルが存在する場合に上書きします。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 作成するチェックポイントファイルの名前を指定します。拡張子を指定しない場合、.dcp という拡張子が使用されます。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、指定のチェックポイントファイルを作成しています。同じ名前のチェックポイントファイルが存在する場合は上書きします。

```
write_checkpoint C:/Data/checkpoint1 -force
```

注記 : 指定のファイル名に拡張子 .dcp が付けられ、既存の `checkpoint1.dcp` ファイルが上書きされます。

関連項目

[read_checkpoint](#)

write_chipscope_cdc

デバッグ ポートに接続されているネットをエクスポートします。

構文

```
write_chipscope_cdc [-quiet] [-verbose] file
```

戻り値

出力ファイル名

使用法

名前	説明
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	ChipScope CDC ファイル名を指定します。

カテゴリ

ファイル入力および出力、ChipScope

説明

現在のプロジェクトで定義されたレガシ ICON 1.x および ILA 1.x デバッグ コア、ポート、信号を含む CDC (ChipScope™ Definition and Connection) ファイルを生成します。CDC ファイルには、ChipScope Pro Analyzer のソース ファイル、デスティネーション ファイル、コア パラメーター、コア 設定などの情報が含まれます。

注記 : `write_chipscope_cdc` コマンドは、ICON 1.x および ILA 1.x コアに対してのみ使用できます。新しい ILA 2.x または VIO 2.x 以降の Vivado ラボ ツールのデバッグ コアには使用できません。

レガシ ChipScope デバッグ コアは、HDL デザインに ICON 1.x および ILA 1.x コアをインスタンシエートし、対応するネットリストおよび制約ファイルを追加することによりプロジェクトに追加できます。

この CDC ファイルを ChipScope Analyzer にインポートし、ILA コア データおよびトリガー ポートのネット名を自動的に設定できます。

引数

`-quiet` (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、`TCL_OK` が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

`-verbose` (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、`set_msg_config` コマンドで定義できます。

file (必須) : 生成する CDC ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、bft.cdc いう CDC ファイルが生成されます。

```
write_chipscope_cdc bft.cdc
```

この CDC ファイルには、ChipScope でデバッグされる信号と、信号のクロック ドメイン、ChipScope で使用されるその他の設定が含まれます。

関連項目

[create_debug_core](#)

write_csv

パッケージ ピンとポート配置情報をエクスポートします。

構文

```
write_csv [-force] [-quiet] [-verbose] file
```

戻り値

出力ファイル名

使用法

名前	説明
[-force]	既存のファイルを上書きします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	ピン配置 CSV ファイルを指定します。

カテゴリ

ファイル入力および出力

説明

パッケージ ピンおよびポート配置情報を CSV (Comma Separated Value) ファイルに記述します。

CSV ファイルの形式および要件は、『Vivado Design Suite ユーザー ガイド：I/O およびクロックの配置』(UG899) を参照してください。

引数

-force (オプション)：同じ名前の CSV が存在する場合に上書きします。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須)：生成する CSV ファイルの名前を指定します。

注記：パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、現在のプロジェクトから CSV ファイルがエクスポートされます。

```
write_csv C:/Data/pinList.csv
```

関連項目

[read_csv](#)

write_debug_probes

デバッグ プローブをファイルに記述します。

構文

```
write_debug_probes [-force] [-quiet] [-verbose] file
```

戻り値

出力ファイル名

使用法

名前	説明
[-force]	既存のファイルを上書きします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>file</i>	デバッグ プローブ ファイル名 (デフォルト拡張子 .ltx) を指定します。

カテゴリ

ファイル入力および出力

説明

ILA v2.0 コアを使用する現在のデザインでプローブされたネットに関する情報を含む ILA プローブ ファイルを記述します。デバッグ プローブ データ ファイルの拡張子は通常 .ltx です。

ILA コアは **create_debug_core** コマンドを使用してデザインに追加し、**connect_debug_core** コマンドを使用してデザインのネットに接続します。

デバッグ プローブ ファイルの詳細および使用方法は、『Vivado Design Suite ユーザー ガイド：プログラムおよびデバッグ』(UG908) を参照してください。

引数

-force (オプション)：指定のファイルが存在する場合に上書きします。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 記述するデバッグ プローブ ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、現在のデザインからデバッグ プローブ ファイルを記述しています。

```
write_debug_probes C:/Data/designProbes.ltx
```

関連項目

- [create_debug_core](#)
- [implement_debug_core](#)

write_edif

現在のネットリストを EDIF ファイルとしてエクスポートします。

構文

```
write_edif [-pblocks args] [-cell arg] [-force] [-security_mode arg] [-quiet]
[-verbose] file
```

戻り値

出力ファイル名またはディレクトリ名

使用法

名前	説明
[-pblocks]	指定した Pblock のネットリストをエクスポートします。-cell と共に使用することはできません。
[-cell]	指定したセルのネットリストをエクスポートします。-pblocks と共に使用することはできません。
[-force]	既存のファイルを上書きします。
[-security_mode]	all に設定すると、デザインの一部を暗号化する必要がある場合に、デザイン全体が 1 つの暗号化ファイルに記述されます。デフォルトは multifile です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	出力ファイルを指定します。-pblocks、-cell を指定している場合は、ディレクトリを指定します。

カテゴリ

ファイル入力および出力

説明

現在のネットリストを EDIF ファイルとして書き出すか、指定の Pblock または階層セルの内容を EDIF ネットリスト ファイルとして出力します。

-pblock または -cell オプションのいずれかを使用する場合、この引数で各 Pblock またはセルの EDIF ネットリスト ファイルを保存するディレクトリ名を指定します。EDIF ネットリスト ファイルの名前は、その Pblock またはセルと同じになります。指定したディレクトリが存在しない場合は、エラーが返されます。

引数

-pblocks arg (オプション) : 指定した Pblock の内容を EDIF ネットリスト ファイルとして出力します。各 Pblock の内容が個別の EDIF ファイルに書き込まれます。

-cell *arg* (オプション) : 指定した階層セルの内容を EDIF ネットリスト ファイルとして出力します。出力するように指定できるセルは 1 つのみです。

注記 : -pblock と -cell オプションを同時に使用することはできません。これらはオプションですが、1 回に指定できるのはどちらか 1 つのみです。

-force (オプション) : 同じ名前の EDIF が存在する場合に上書きします。

-security_mode [*multifile* | *all*] (オプション) : デザインに暗号化された IP がある場合に、複数の EDIF ファイルを記述するか、デザイン全体を 1 つのファイルに記述するかを指定します。

- ***multifile*** : これがデフォルト設定です。デフォルトではデザイン全体のネットリストが指定したファイルに記述されますが、デザインに保護された IP が含まれている場合、保護されているモジュールの内容を含む暗号化されたファイルが作成されます。これにより、デザインの保護されたエレメントと保護されていないエレメントを含む、複数の EDIF ファイルが生成されます。
- ***all*** : 暗号化されたセルと暗号化されていないセルの両方を指定した 1 つのファイルに記述します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンドラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 生成する EDIF ファイルの名前を指定します。EDIF ネットリストのデフォルトのファイル拡張子は、.edn です。-pblocks または -cell オプションを使用している場合は、ここで指定した名前はディレクトリ名を指します。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、デザイン全体の EDIF ネットリスト ファイルを指定したファイル名で書き出します。

```
write_edif C:/Data/edifOut.edn
```

次の例では、デザインのすべての Pblock の EDIF ネットリストが output されます。ファイルは、指定したディレクトリに書き込まれます。

```
write_edif -pblocks [get_pblocks] C:/Data/FPGA_Design/
```

関連項目

[read_edif](#)

write_hw_ilab_data

ハードウェア ILA データをファイルに記述します。

構文

```
write_hw_ilab_data [-force] [-quiet] [-verbose] file [hw_ilab_data]
[hw_ilab_data]
```

戻り値

出力ファイル名

使用法

名前	説明
[-force]	既存のファイルを上書きします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	ハードウェア ILA データ ファイル名を指定します。
[hw_ilab_data]	ハードウェア ILA データ オブジェクトを指定します。デフォルトは、現在のハードウェア ILA データです。

カテゴリ

ハードウェア

write_hw_sio_scan

スキヤン データをファイルに記述します。

構文

```
write_hw_sio_scan [-force] [-quiet] [-verbose] file hw_sio_scan
```

戻り値

出力ファイル名

使用法

名前	説明
[-force]	既存のファイルを上書きします。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	ハードウェア SIO スキヤン ファイルの名前を指定します。
hw_sio_scan	ハードウェア SIO スキヤン データ オブジェクトを指定します。

カテゴリ

ハードウェア

write_ibis

現在のプロアプランの IBIS モデルを書き出します。

構文

```
write_ibis [-force] [-allmodels] [-nopin] [-truncate arg] [-ibs arg]
[-pkg arg] [-quiet] [-verbose] file
```

戻り値

出力ファイル名

使用法

名前	説明
[-force]	既存の .ibs ファイルを上書きします。
[-allmodels]	アーキテクチャで使用可能なバッファー モデルをすべて含みます。デフォルトでは、プロアプランで使用されるバッファー モデルのみが含まれます。

カテゴリ

ファイル入力および出力

説明

現在のデザインのターゲット デバイスの IBIS モデルを書き出します。デザインからのネットリストおよびインプリメンテーションの詳細と使用可能なピンごとの寄生パッケージ情報がまとめられ、デザイン専用のカスタム IBIS ファイルが作成されます。

write_ibis コマンドではデザイン情報が IBIS モデルにまとめられるので、このコマンドを実行する際には RTL、ネットリスト、またはインプリメント済みデザインが開いている必要があります。

引数

-allmodels (オプション) : ターゲット デバイスのバッファー モデルをすべてエクスポートします。デフォルトでは、デザインで使用されるバッファー モデルのみが記述されます。

-nopin : ダイ パッドからパッケージ ピンへのパスのピンごとのコード記述をオフにします。IBIS モデルには、[Package] セクションのすべてのピンが 1 本の RLC 伝送ラインで記述されます。デフォルトでは、データが存在する場合、パッケージのピンごとのモデリングが RLC メトリックとして [Define Package Model] セクションに含められます。

-truncate arg (オプション) : 出力ファイルの信号名の最大長を指定します。最大長を超えるとその部分は表示されません。有効な値は、20、40、0 (制限なし) です。デフォルトでは、IBIS バージョン 4.2 の仕様に準拠するよう信号名は 40 文字に切り詰められます。

-ibis arg (オプション) : アップデートされた汎用 IBIS モデル ファイルを指定します。これは、ツールのインストールで含まれる parts ディレクトリの IBIS モデルを上書きするために使用します。このオプションは、インストール ディレクトリに汎用 モデルのない パーツすべてで必要となります。

-pkg arg (オプション) : アップデートされたピンごとの寄生パッケージ データ ファイルを指定します。これは、ツールのインストールで含まれる parts ディレクトリの 寄生パッケージ ファイルを上書きするために使用します。このオプションは、インストール ディレクトリに汎用 モデルのない パーツすべてで必要となります。

file (必須) : 生成する IBIS ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

例

次の例では、ターゲット デバイスのバッファー モデルすべてがエクスポートされ、信号名が切り詰められないよう設定され、書き込むファイル名およびパスが指定されています。

```
write_ibis -allmodels -truncate 0 C:/Data/FPGA_Design/ibisOut.txt
```

write_project_tcl

現在のプロジェクトを再作成する Tcl スクリプトを生成します。

構文

```
::write_project_tcl [-force arg] [-all_properties arg] [-no_copy_sources arg]
[-dump_project_info arg] [-quiet] [-verbose] file
```

戻り値

True

使用法

名前	説明
[-force]	既存の Tcl スクリプト ファイルを上書きします。デフォルトは 0 です。
[-all_properties]	プロジェクト オブジェクトのすべてのプロパティ (デフォルトおよびデフォルトでないもの) を記述します。デフォルトは 0 です。
[-no_copy_sources]	元のプロジェクトのローカルであってもソースをインポートしません。デフォルトは 1 です。
[-dump_project_info]	オブジェクトの値を記述します。デフォルトは 0 です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
<i>file</i>	生成する Tcl スクリプト ファイルの名前を指定します。

説明

現在のプロジェクトを再作成する Tcl スクリプトを生成します。

生成されたスクリプトには、プロジェクトの作成、プロジェクトタイプの設定、ファイルセットの作成、ソース ファイルの追加/インポート、run および run プロパティの定義を実行する Tcl コマンドが含まれます。

この Tcl プロジェクト スクリプトとさまざまなデザイン ソースを、ソース ファイル 管理およびプロジェクトのアーカイブ用にバージョン管理システムに保存できます。

引数

-force (オプション) : 指定したプロジェクト スクリプト ファイルが存在する場合に上書きします。スクリプト ファイルが存在する場合に **-force** が指定されていないと、エラー メッセージが表示されます。

-all_properties (オプション) : プロジェクトのすべてのプロパティを記述します。デフォルトでは、デフォルトでないプロパティのみが記述されます。プロジェクト、ファイルセット、ファイル、run などのすべてのオブジェクトに対して **set_property** コマンドが記述されます。

注記 : **-all_properties** オプションを指定しない場合、デフォルトでないプロパティのみがスクリプトに記述されます。

-no_copy_sources (オプション) : 元のプロジェクトにローカルであっても、ソースをインポートしません。元のプロジェクトのローカルにあるファイルは新しいプロジェクトにインポートされません。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : **write_project_tcl** コマンドで作成するスクリプト ファイルの名前を指定します。ファイル拡張子を指定しない場合、.tcl が付けられます。

例

次の例では、現在のプロジェクトに対して recreate.tcl という Tcl スクリプトを生成しています。

```
write_project_tcl recreate.tcl
```

次の例では、現在のプロジェクトの Tcl スクリプトを生成し、デフォルト値のものおよびデフォルト値でないものを含め、すべてのプロパティを記述しています。

```
write_project_tcl -all_properties recreate.tcl
```

次の例では、現在のプロジェクトの Tcl スクリプトを生成していますが、このプロジェクトにローカルのファイルはインポートしません。再作成されたプロジェクトでは、これらのファイルが参照されます。

```
write_project_tcl -no_copy_sources recreate.tcl
```

次の例では、現在の作業ディレクトリにある現在のプロジェクトの recreate.tcl スクリプトを生成し、./my_test ディレクトリにプロジェクトを再作成して、そのプロジェクトに含まれるファイルのリストを表示した後、そのプロジェクトを閉じています。

```
open_project ./test/test.xpr
write_project_tcl -force recreate.tcl
close_project
file mkdir my_test
cd my_test
source ../recreate.tcl
get_files -of_objects [get_filesets sources_1]
report_property [current_project]
close_project
```

次の例では、bft_test という新しいプロジェクトを作成し、ファイルを追加してファイルセットプロパティを設定し、現在の作業ディレクトリの bft.tcl という Tcl スクリプトを生成した後、./my_bft ディレクトリにプロジェクトを再作成して、そのプロジェクトのファイル (test_1.v および test_2.v) をリストし、verilog_define プロパティの値を表示して、そのプロジェクトを閉じています。

```
create_project bft_test ./bft_test
add_files test_1.v
add_files test_2.v
set_property verilog_define {a=10} [get_filesets sources_1]
write_project_tcl -force bft.tcl
close_project
file mkdir my_bft
cd my_bft
source ../bft.tcl
get_files -of_objects [get_filesets sources_1]
get_property verilog_define [get_filesets sources_1]
close_project
```

関連項目

- [add_files](#)
- [archive_project](#)
- [close_project](#)
- [create_project](#)
- [current_project](#)
- [get_files](#)
- [get_property](#)
- [open_project](#)
- [report_property](#)
- [set_property](#)

write_schematic

回路図をエクスポートします。

構文

```
write_schematic [-force] [-format arg] [-orientation arg] [-scope arg]
[-name arg] [-quiet] [-verbose] file
```

戻り値

出力ファイル名

使用法

名前	説明
[-force]	既存のファイルを上書きします。
[-format]	フォーマット指定します。有効な値は native または pdf です。ネイティブ フォーマットは、read_schematic を使用して表示できます。デフォルトは native です。
[-orientation]	回路図の向きを指定します。有効な値は landscape または portrait です。
[-scope]	エクスポートする範囲を指定します。有効な値は current_page、visible、all で、デフォルトは current_page です。
[-name]	[Schematic] ビューのタイトルを指定します。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	出力ファイルを指定します。

カテゴリ

ファイル入力および出力

説明

Vivado IDE で現在開いている [Schematic] ビューまたは指定した [Schematic] ビューをファイルに出力します。ファイルは、read_schematic コマンドを使用して Vivado IDE に読み込み可能なネイティブ ASCII ファイルとして記述するか、Vivado Design Suite 外で使用可能な PDF ファイルとして記述できます。これは、IP パッケージャーフローまたは Vivado IP インテグレーターからの IP コアを記録する場合に有益です。

引数

-force (オプション) : 指定の回路図ファイルが存在する場合に上書きします。

-format [native | pdf] (オプション) : 回路図を Vivado Design Suite にネイティブで **read_schematic** コマンドを使用して Vivado IDE に読み込み可能な ASCII ファイルとして記述するか、Vivado Design Suite 外で表示できる PDF ファイルとして記述するかを指定します。デフォルトは **native** です。

注記 : 大文字/小文字が区別されます。

-orientation [landscape | portrait] (オプション) : 回路図を縦長 (portrait) または横長 (landscape) のどちらの向きで記述するかを指定します。デフォルトは **portrait** です。

-scope [current_page | visible | all] (オプション) : 回路図の現在のページ (current_page)、[Schematic] ビューで表示されている部分 (visible)、または回路図すべて (all) を記述するかを指定します。デフォルトは **current_page** です。

-name arg (オプション) : ファイルに記述する、開いている [Schematic] ビューの名前を指定します。複数の [Schematic] ビューが開いている場合に、このオプションを使用して記述する [Schematic] ビューを指定します。

注記 : 複数の [Schematic] ビューが開いていて **-name** を指定しない場合、最初に開いたビューが記述されます。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 記述する回路図ファイルのパスと名前を指定します。パスの指定はオプションですが、パスを指定しない場合は、現在の作業ディレクトリまたは Vivado ツールを起動したディレクトリに回路図ファイルが作成されます。

例

次の例では、Vivado IDE に表示されている指定の [Schematic] ビューを、Vivado Design Suite に読み込み可能なネイティブ フォーマットで記述しています。ファイルが存在する場合は、上書きします。

```
write_schematic -name Schematic C:/Data/mySchematic.txt -force
```

次の例では、指定の [Schematic] ビューを横長の PDF ファイルに記述しています。

```
write_schematic -format pdf -name "Schematic (2)" C:/Data/mySchematic.pdf \
-orientation landscape
```

関連項目

[read_schematic](#)

write_sdf

イベント シミュレーション用のフラットな SDF 遅延ファイルを生成します。

構文

```
write_sdf [-process_corner arg] [-cell arg] [-rename_top arg] [-force]
[-mode arg] [-quiet] [-verbose] file
```

戻り値

なし

使用法

名前	説明
[-process_corner]	SDF 遅延の必要とされるプロセスコーナーを指定します。有効な値は slow、fast で、デフォルトは slow です。
[-cell]	書き込むデザインのルートを指定します (例 : des.subblk.cpu)。デフォルトはデザイン全体です。
[-rename_top]	最上位モジュール名を netlist などのカスタム名に置き換えます。デフォルトは、新しい最上位モジュールの名前です。
[-force]	既存の SDF ファイルを上書きします。
[-mode]	SDF のモードを指定します。有効な値は sta (スタティック タイミング解析) または timesim (タイミング シミュレーション) で、デフォルトは timesim です。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	ファイル名を指定します。

カテゴリ

ファイル入力および出力、シミュレーション

説明

デザインのセルのタイミング遅延を標準遅延フォーマット (SDF) ファイルに記述します。

出力される SDF ファイルは、**write_verilog** コマンドで使用して、スタティック タイミング解析およびタイミング シミュレーション用の Verilog ネットリストを作成できます。

引数

-process_corner [fast | slow] (オプション) : 指定したプロセスコーナーの遅延を記述します。遅延は fast プロセスコーナーよりも slow プロセスコーナーの方が大きくなります。有効な値は、slow または fast です。デフォルトでは、SDF ファイルは slow プロセスコーナー用に記述されます。

-cell *arg* (オプション) : デザイン階層の指定したセルから SDF ファイルを記述します。デフォルトでは、デザイン全体の SDF ファイルが作成されます。

-rename_top *arg* (オプション) : 出力 SDF ファイルの最上位モジュールの名前を変更します。

-force (オプション) : 同じ名前の既存の SDF ファイルを上書きします。

-mode [**timesim** | **sta**] (オプション) : SDF ファイルを記述する際に使用するモードを指定します。有効な値は次のとおりです。

- **timesim** : タイミング シミュレーションで使用する SDF ファイルを出力します。これがデフォルト設定です。
- **sta** : スタティック タイミング解析 (STA) で使用する SDF ファイルを出力します。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 生成する SDF ファイルの名前を指定します。SDF ファイルは、**write_verilog** コマンドで **-sdf_anno** および **-sdf_file** オプションを使用することにより Verilog ネットリストで参照されます。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、指定したディレクトリに SDF ファイルが生成されます。

```
write_sdf C:/Data/FPGA_Design/designOut.sdf
```

関連項目

[write_verilog](#)

write_verilog

現在のネットリストを Verilog 形式でエクスポートします。

構文

```
write_verilog [-cell arg] [-mode arg] [-lib] [-port_diff_buffers]
[-write_all_overrides] [-rename_top arg] [-sdf_anno arg] [-sdf_file arg]
[-force] [-include_xilinx_libs] [-quiet] [-verbose] file
```

戻り値

出力ファイル名またはディレクトリ名

使用法

名前	説明
[-cell]	書き込むデザインのルートを指定します (例 : des.subblk.cpu)。デフォルトはデザイン全体です。
[-mode]	有効な値は design、port、sta、funcsim、timesim で、デフォルトは design です。
[-lib]	各ライブラリを個別のファイルに記述します。
[-port_diff_buffers]	-mode が port に設定されている場合に差動バッファーを出力します。
[-write_all_overrides]	パラメーター値がプリミティブのデフォルト値と同じであっても、デザインのすべてのパラメーター値を記述します。
[-rename_top]	最上位モジュール名を netlist などのカスタム名に置き換えます。デフォルトは、新しい最上位モジュールの名前です。
[-sdf_anno]	sdf_annotate システム タスク文を生成するかどうかを指定します。
[-sdf_file]	SDF ファイルの完全なパスを指定します。デフォルトは <file>.sdf です。
[-force]	既存のファイルを上書きします。
[-include_xilinx_libs]	シミュレーション モデルをライブラリにリンクするのではなく、直接ネットリストに含めます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	記述するファイルを指定します。

カテゴリ

ファイル入力および出力、シミュレーション

説明

現在のデザインまたは指定したセルの Verilog ネットリストを指定したファイルまたはディレクトリに記述します。出力は IEEE 1364-2001 準拠の Verilog HDL ファイルで、入力デザイン ファイルからのネットリスト情報が含まれます。

デザインまたは指定のセルの完全なネットリスト、デザインのポートリスト、シミュレーションまたはスタティック タイミング解析用の Verilog ネットリストを出力できます。

引数

-cell arg (オプション) : 指定したセルまたはデザイン階層のブロック レベルから Verilog ネットリストを記述します。出力される Verilog ファイルには、指定したセルまたはモジュール内の情報だけが含まれます。

-mode arg (オプション) : Verilog ファイルを記述する際に使用するモードを指定します。デフォルトでは、デザイン全体の Verilog ネットリストが記述されます。有効なモードは、次のとおりです。

- **design** : デザイン全体の Verilog ネットリストを出力します。これは、すべての配置、インプリメンテーション、配線情報を含むデザインのスナップショットとして使用されます。
- **port** : デザインの最上位の I/O ポートのみを出力します。
- **sta** : スタティック タイミング解析 (STA) に使用する Verilog ネットリストを出力します。
- **funcsim** : 論理シミュレーションに使用する Verilog ネットリストを出力します。この出力ネットリストは、合成には使用できません。
- **timesim** : タイミング シミュレーションに使用する Verilog ネットリストを出力します。この出力ネットリストは、合成には使用できません。

-lib (オプション) : デザインで使用される各ライブラリに個別の Verilog ファイルを作成します。

注記 : このオプションは、以前のリリースの **-nolib** オプションに置き換わるもので、動作は反対になります。以前のリリースでは、**write_verilog** を実行したときに、**-nolib** を使用しなければ、デザインで使用される各ライブラリにデフォルトで個別の Verilog ファイルが出力されていました。今リリースでは、各ライブラリに個別の Verilog ファイルを生成するには **-lib** オプションを指定する必要があります。

-port_diff_buffers (オプション) : 差動ペア バッファーとそれらのバッファーに関連付けられている内部ワイヤを出力ポートリストに追加します。このオプションは、**-mode port** オプションを使用している場合にのみ有効です。

-write_all_overrides [true | false] (オプション) : パラメーター値が定義されたプリミティブのデフォルト値と同じであっても、デザインのすべてのパラメーター値を Verilog 出力に記述します。false に設定すると、パラメーター値がプリミティブのデフォルト値と同じである場合は Verilog ファイルに記述されません。このオプションを true に設定しても結果は変わらぬが、出力される Verilog ファイルがより詳細なものとなります。

-rename top arg (オプション) : 出力ファイルの最上位モジュールの名前を変更します。このオプションは **-mode funcsim** または **-mode timesim** を指定した場合にのみ有効で、Verilog ネットリストを最上位シミュレーション テストベンチに組み込むことができます。

-sdf_anno [true | false] (オプション) : Verilog ネットリストに **\$sdf_annotation** 文を追加します。有効な値は true (または 1) および false (または 0) です。このオプションは、**-mode timesim** を指定している場合にのみ有効で、デフォルトでは false に設定されています。

-sdf_file arg (オプション) : 出力 Verilog ファイルに **\$sdf_annotation** 文を記述する場合に使用する SDF ファイルのパスとファイル名を指定します。指定しない場合、SDF ファイルのパスと名前は *file* で指定した Verilog ファイルと同じになり、ファイル拡張子は *.sdf* になります。SDF ファイルは、**write_sdf** コマンドを使用して指定のファイル パスおよび名前で別に記述する必要があります。

-force (オプション)：同じ名前の Verilog ファイルが存在する場合に上書きします。

-include_xilinx_libs (オプション)：シミュレーション モデルをライブラリから参照するのではなく、出力ネットリスト ファイルに直接記述します。

-quiet (オプション)：コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくとも、TCL_OK が返されます。

注記：コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション)：メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記：メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須)：記述する Verilog ファイルのパスと名前を指定します。パスの指定はオプションですが、パスを指定しない場合は、現在の作業ディレクトリまたは Vivado ツールを起動したディレクトリに Verilog ファイルが作成されます。

例

次の例では、デザイン全体の Verilog シミュレーション ネットリスト ファイルを指定したファイル名とパスで記述します。

```
write_verilog C:/Data/my_verilog.v
```

次の例では、**-mode timesim** と **-sdf_anno** オプションが指定されているので、Verilog ネットリストに **\$sdf_annotation** 文が追加されます。**-sdf_file** は指定されていないので、SDF ファイルの名前は Verilog ファイルと同じになり、ファイル拡張子は **.sdf** になります。

```
write_verilog C:/Data/my_verilog.net -mode timesim -sdf_anno true
```

注記：**\$sdf_annotation** 文に記述される SDF ファイル名は **my_verilog.sdf** です。

関連項目

- [write_sdf](#)
- [write_vhdl](#)

write_vhdl

現在のネットリストを VHDL 形式でエクスポートします。

構文

```
write_vhdl [-cell arg] [-mode arg] [-lib] [-port_diff_buffers]
[-write_all_overrides] [-rename_top arg] [-arch_only] [-force]
[-include_xilinx_libs] [-quiet] [-verbose] file
```

戻り値

出力ファイル名またはディレクトリ名

使用法

名前	説明
[-cell]	書き込むデザインのルートを指定します (例 : des.subblk.cpu)。デフォルトはデザイン全体です。
[-mode]	出力モードを指定します。有効な値は funcsim、port で、デフォルトは funcsim です。
[-lib]	各ライブラリを個別のファイルに記述します。
[-port_diff_buffers]	-mode が port に設定されている場合に差動バッファーを出力します。
[-write_all_overrides]	パラメーター値がプリミティブのデフォルト値と同じであっても、デザインのすべてのパラメーター値を記述します。
[-rename_top]	最上位モジュール名を netlist などのカスタム名に置き換えます。デフォルトは、新しい最上位モジュールの名前です。
[-arch_only]	アーキテクチャのみを記述し、最上位セルにエンティティ宣言を記述しません。
[-force]	既存のファイルを上書きします。
[-include_xilinx_libs]	シミュレーション モデルをライブラリにリンクするのではなく、直接ネットリストに含めます。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
file	記述するファイルを指定します。

カテゴリ

ファイル入力および出力、シミュレーション

説明

現在のデザインまたは指定したセルの VHDL ネットリストを指定したファイルまたはディレクトリに記述します。

出力は VHDL IEEE 1076.4 VITAL-2000 準拠の VHDL ファイルで、入力デザイン ファイルからのネットリスト情報が含まれます。デザインまたは指定のセルの完全なネットリスト、またはデザインのポートリストを出力できます。

引数

-cell arg (オプション) : 指定したセルまたはデザイン階層のブロック レベルから VHDL ネットリストを記述します。出力される VHDL ファイルには、指定したセルまたはモジュール内の情報だけが含まれます。

-mode arg (オプション) : VHDL ファイルを記述する際に使用するモードを指定します。デフォルトでは、デザイン全体のシミュレーション ネットリストが記述されます。有効なモードは、次のとおりです。

- funcsim : 論理シミュレーション モデルとして使用する VHDL ネットリストを出力します。この出力ネットリストは、合成には使用できません。これがデフォルト設定です。
- port : 最上位モジュールのエンティティ宣言に含まれる I/O ポートのみを出力します。

-lib (オプション) : デザインで使用される各ライブラリに個別の VHDL ファイルを作成します。

注記 : このオプションは、以前のリリースの **-nolib** オプションに置き換わるもので、動作は反対になります。以前のリリースでは、**write_vhdl** を実行したときに、**-nolib** を使用しなければ、デザインで使用される各ライブラリにデフォルトで個別の VHDL ファイルが output されていました。今リリースでは、各ライブラリに個別の VHDL ファイルを生成するには **-lib** オプションを指定する必要があります。

-port_diff_buffers (オプション) : 差動ペア バッファーとそれらのバッファーに関連付けられている内部ワイヤを出力ポートリストに追加します。このオプションは、**-mode port** オプションを使用している場合にのみ有効です。

-write_all_overrides [true | false] (オプション) : パラメーター値が定義されたプリミティブのデフォルト値と同じであっても、デザインのすべてのパラメーター値を VHDL 出力に記述します。false に設定すると、パラメーター値がプリミティブのデフォルト値と同じである場合は VHDL ファイルに記述されません。このオプションを true に設定しても結果は変わらぬが、出力されるネットリスト ファイルがより詳細なものとなります。

-rename_top arg (オプション) : 出力ファイルの最上位モジュールの名前を変更します。このオプションは **-mode funcsim** を指定した場合にのみ有効で、VHDL ネットリストを最上位シミュレーション テストベンチに組み込むことができます。

-arch_only (オプション) : 最上位モジュールのエンティティ定義を含めず、アーキテクチャのみを出力します。これにより、出力 VHDL ネットリストを別のテストベンチと使用する場合に、出力 VHDL ネットリストの使用が簡略化されます。

-include_xilinx_libs (オプション) : シミュレーション モデルをライブラリから参照するのではなく、出力ネットリスト ファイルに直接記述します。

-force (オプション) : 同じ名前の VHDL ファイルが存在する場合に上書きします。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 生成する VHDL ファイルの名前を指定します。ファイル拡張子 .vhd または .vhdl を指定しない場合、指定した名前はディレクトリであると判断され、VHDL ファイルの名前は最上位モジュールと同じになり、指定したディレクトリに出力されます。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、デザイン全体の VHDL シミュレーション ネットリスト ファイルを指定したファイル名とパスで記述します。

```
write_vhdl C:/Data/bft_top.vhd
```

次の例では、最上位モジュールのエンティティ定義は VHDL ネットリストに含まれません。

```
write_vhdl C:/Data/vhdl_arch_only.vhd -arch_only
```

関連項目

[write_verilog](#)

write_xdc

制約をザイリンクス デザイン制約 (XDC) ファイルに記述します。XDC ファイルのデフォルトのファイル拡張子は .xdc です。

構文

```
write_xdc [-no_fixed_only] [-constraints arg] [-cell arg] [-sdc]
[-no_tool_comments] [-force] [-exclude_timing] [-quiet] [-verbose] [file]
```

戻り値

なし

使用法

名前	説明
[-no_fixed_only]	配置が固定されているかどうかに問わらず、すべての配置をエクスポートします。デフォルトでは、固定された配置のみがエクスポートされます。
[-constraints]	無効と示されている制約を含めます。有効な値は valid、invalid、all で、デフォルトは valid です。
[-cell]	配置をエクスポートするセルを指定します。
[-sdc]	すべてのタイミング制約をエクスポートします。
[-no_tool_comments]	UCF から変換する際にツールで生成された詳細コメントを XDC に記述しません。
[-force]	既存のファイルを上書きします。
[-exclude_timing]	タイミング制約をエクスポートしません。
[-quiet]	コマンド エラーを無視します。
[-verbose]	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。
[file]	制約を記述する XDC ファイルを指定します。

カテゴリ

ファイル入力および出力

説明

制約をザイリンクス デザイン制約 (XDC) ファイルに記述します。XDC は、最上位からエクスポートするか (デフォルト)、指定の階層セルからエクスポートできます。

このコマンドは、UCF ファイルを含むデザインから XDC ファイルを作成するために使用できます。アクティブな制約ファイルセットからの制約はすべて、それらが複数ファイルからの制約であっても、XDC にエクスポートされます。

引数

-no_fixed_only (オプション) : 配置が固定されているかどうかに関わらず、すべての LOC を制約ファイルにエクスポートします。デフォルトでは、固定された LOC 制約のみが XDC ファイルに書き込まれます。固定された LOC はユーザーの割り当てた配置で、固定されていない LOC はツールで自動的に割り当てられた配置です。

-constraints arg (オプション) : 有効 (valid)、無効 (invalid) と指定した制約、またはすべての制約 (valid と invalid の両方) をエクスポートします。デフォルトでは、有効な制約のみが XDC ファイルにエクスポートされますが、有効な値は VALID、INVALID、ALL です。

-cell arg (オプション) : 制約を記述する現在のデザインの階層セルの名前を指定します。制約は、指定したセルに相対的に指定した XDC ファイルに書き込まれます。

注記 : このオプションを指定する場合は、デザインを開いておく必要があります。

-sdc (オプション) : 現在のデザインからタイミング制約のみを SDC フォーマットでエクスポートします。その他の定義された制約は、エクスポートされません。

-force (オプション) : 同じ名前の XDC ファイルが存在する場合に上書きします。

-quiet (オプション) : コマンドをメッセージを表示せずに実行します。実行中にエラーが発生してもしなくても、TCL_OK が返されます。

注記 : コマンドの起動中にコマンド ラインで発生したエラーは返されます。コマンド内で発生したエラーのみが非表示になります。

-verbose (オプション) : メッセージの制限を一時的に解除し、コマンドからのすべてのメッセージを返します。

注記 : メッセージの制限は、**set_msg_config** コマンドで定義できます。

file (必須) : 生成する XDC ファイルの名前を指定します。

注記 : パスをファイル名の一部として指定しない場合は、現在の作業ディレクトリまたはツールを起動したディレクトリにファイルが作成されます。

例

次の例では、固定されているセルと固定されていないセルの両方を含む、有効な制約と無効な制約を指定のファイルに記述しています。

```
write_xdc -no_fixed_only -constraints all C:/Data/design.xdc
```

関連項目

[read_xdc](#)

xsim

シミュレーションのシミュレーション スナップショットを読み込み、シミュレーション オブジェクトを返します。

構文

```
xsim [-maxdeltaid arg] [-nolog] [-onfinish arg] [-onerror arg]
[-tclbatch arg] [-t arg] [-runall] [-R] [-testplusarg args] [-vcdfile arg]
[-vcdunit arg] [-view arg] [-wdb arg] [-tp] [-tl] [-nosignalhandlers] [-quiet]
[-verbose] snapshot
```

戻り値

現在のシミュレーション オブジェクト

使用法

名前	説明
[-maxdeltaid]	最大差異を指定します。同じ時間に最大シミュレーション ループ回数を超えると、エラーがレポートされます。デフォルトは 10000 です。
[-nolog]	無視されます (xsim コマンド ライン ツールとの互換性のため)。
[-onfinish]	シミュレーションが終了したときの動作を指定します。有効な値は quit および stop で、デフォルトは stop です。
[-onerror]	シミュレーション ランタイム エラーが発生した場合の動作を指定します。有効な値は quit および stop で、デフォルトは stop です。
[-tclbatch]	バッチ モード実行用の Tcl ファイルを指定します。
[-t]	バッチ モード実行用の Tcl ファイルを指定します。
[-runall]	シミュレーションを最後まで実行し、終了します (『run -all; exit』を実行)。
[-R]	シミュレーションを最後まで実行し、終了します (『run -all; exit』を実行)。
[-testplusarg]	plusargs が \$test\$plusargs および \$value\$plusargs システム関数で使用されるように指定します。
[-vcdfile]	VCD 出力ファイルを指定します。
[-vcdunit]	VCD 出力の単位を指定します。デフォルトは、エンジンの精度ユニットと同じです。
[-view]	波形コンフィギュレーション ファイルを開きます。このオプションは、-gui オプションと共に使用する必要があります。
[-wdb]	波形データベース出力ファイルを指定します。
[-tp]	実行中のプロセスの階層名が表示されるようにします。
[-tl]	実行中の文の行番号が表示されるようにします。
[-nosignalhandlers]	信号ハンドラーを使用せずに実行し、セキュリティ ソフトウェアとの競合を回避します。

名前	説明
<code>[-quiet]</code>	コマンド エラーを無視します。
<code>[-verbose]</code>	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。
<code>snapshot</code>	デザインインスナップショットの名前を指定します。

カテゴリ

シミュレーション

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される用語集は、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ト、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

Vivado Design Suite 2013.2 の資料

http://japan.xilinx.com/support/documentation/dt_vivado2013-2.htm

Tcl Developer Xchange

Tcl リファレンス資料は、インターネットから入手できます。ザイリンクスでは、Tcl のオープンソースベースを管理する Tcl Developer Xchange サイトをお勧めしています。

<http://www.tcl.tk>

入門用チュートリアルは、次から入手できます。

<http://www.tcl.tk/man/tcl8.5/tutorial/tcltutorial.html>

SDC について

SDC (Synopsys Design Constraints) は、特にタイミング解析において設計の要件をツールに渡すための業界標準です。SDC 仕様のリファレンスコピーは、次の Synopsys 社のサイトから登録をすると入手できます。

<http://www.synopsys.com/Community/Interoperability/Pages/TapinSDC.aspx>