

# Vivado Design Suite 移行手法ガイド

UG911 (v2012.2) 2012 年 7 月 25 日



#### Notice of Disclaimer

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

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

本資料は英語版 (v2012.2) を翻訳したもので、内容に相違が生じる場合には原文を優先します。

資料によっては英語版の更新に対応していないものがあります。

日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

この資料に関するフィードバックおよびリンクなどの問題につきましては、[jpn\\_trans\\_feedback@xilinx.com](mailto:jpn_trans_feedback@xilinx.com) までお知らせください。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。

---

## 改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	改訂内容
2012/07/25	2012.2	初版

# 目次

---

改訂履歴 .....	2
第 1 章 : ISE Design Suite からの移行	
概要 .....	5
デザイン フロー.....	5
第 2 章 : ISE Design Suite デザインの Vivado Design Suite への移行	
Project Navigator プロジェクトのインポート .....	6
PlanAhead ツールプロジェクトの変換 .....	6
XST プロジェクト ファイルのインポート .....	7
ソース ファイルの移行 .....	7
ISE Design Suite コマンド スクリプトのマッピング.....	8
makefile のマッピング .....	11
メッセージの違いについて .....	15
レポートの違いについて.....	16
ログ ファイルの違いについて.....	17
第 3 章 : UCF 制約の XDC への移行	
概要 .....	18
XDC と UCF 制約の違い .....	18
UCF から XDC へのマッピング .....	18
制約の順序.....	19
PlanAhead ツールで UCF を XDC に変換.....	19
タイミング制約 .....	20
タイムグループ .....	27
物理制約.....	28
第 4 章 : CORE Generator IP を使用したデザインの Vivado Design Suite への移行	
概要 .....	77
CORE Generator IP を Vivado Design Suite に移行 .....	77
第 5 章 : ISE Simulator の Tcl の Vivado シミュレータの Tcl への移行	
Tcl コマンドの移行.....	79
付録 A : その他のリソース	
ザイリンクスの資料 .....	82
ソリューション センター .....	82
参考資料.....	82



# ISE Design Suite からの移行

## 概要

ISE® Design Suite は業界で実績のあるデザイン ツールで、ザイリンクス デバイスのすべてのジェネレーションのソリューションを提供し、また 7 シリーズおよび Zynq™-7000 デバイスをターゲットにしたプロジェクトにもデザイン フローを提供しています。

Vivado™ Design Suite は、Virtex®-7、Kintex™-7、Artix™-7 をはじめとする 7 シリーズ デバイスをサポートし、特に大型デザインや集積度の高いデザインでのツール パフォーマンスを高めます。

ISE Design Suite と Vivado Design Suite の両方で 7 シリーズ デバイスがサポートされているため、時間の余裕をもって Vivado Design Suite に移行することができます。



**推奨 :** Kintex K410 またはそれ以上の大型デバイスの新しいデザインを開始する場合は、担当の FAE に連絡し、Vivado が適しているかどうかをご確認ください。

Vivado Design Suite では、プロジェクトおよびソースを Vivado Design Suite プロジェクトにインポートし、また Tcl スクリプトへコマンドをマッピングし、前のデザインのすべてまたは一部をスムーズに再利用できるようになっています。

## デザイン フロー

ISE Design Suite でデザイン ツールを実行するには、Project navigator やフェーズごとに使用するツールなどのグラフィカル ユーザー インターフェイス (GUI) を使用方法と、コマンド ラインでバッチ スクリプトを使用する方法があります。

Vivado Design Suite には、プロジェクト モードと非プロジェクト モードの 2 つのデザイン フロー モードがあります。

- プロジェクト モード : このツールの動作およびザイリンクスの推奨事項を理解するための簡単なモードです。このモードでは、GUI または Vivado 統合設計環境 (IDE) を使用してデザインが自動的に管理されます。
- 非プロジェクト モード : このモードでは、フロー全体を通してメモリに読み込まれているデザインをコンパイルするため Tcl コマンドを使用します。Tcl コマンドを使用することで、デザインの設定、実行、解析、デバッグを柔軟かつパワフルに実行できます。Tcl コマンドは、バッチ モードで Tcl プロンプトまたは Vivado IDE Tcl コンソールから実行できます。このフローを使用する場合は、ユーザーが各デザイン フロー段階をフル制御できますが、ソース ファイル、レポート、中間結果 (チェックポイント) を手動で管理する必要があります。

デザイン フローのモードについては、『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。

このガイドでは、Vivado Design Suite の両方のデザイン フロー モードでの移行に関する注意事項や手順を説明します。

# ISE Design Suite デザインの Vivado Design Suite への移行

## Project Navigator プロジェクトのインポート

グラフィカル ユーザー インターフェイスである Vivado™ 統合設計環境 (IDE) を使用して、XISE プロジェクト ファイルを次のようにインポートすることができます。

1. [File] → [New Project] をクリックします。
2. プロジェクト名および保存場所を選択します。
3. New Project ウィザードで [Imported Project] をオンにします。
4. [ISE] をクリックし、インポートする XISE ファイルを選択します。



**重要 :** Vivado Design Suite では、拡張子が .ise の古い ISE® Design Suite ファイルはサポートされていません。

プロジェクト ファイルをインポートしたら、次の作業を行います。

- インポートされたプロジェクトの重要な情報を確認するため、サマリ レポートを開きます。
- 選択したデバイスで要件が満たされていることを確認します。満たされていない場合は新しいデバイスを選択します。ISE プロジェクトに Vivado でサポートされている同等のデバイスがない場合は、デフォルト デバイスが選択されます。
- すべてのファイルが正しくインポートされていることを確認するため [Sources] ビューでファイルを確認します。

デザインにユーザー制約ファイル (UCF) が含まれている場合は、サポートされていない制約ファイルとしてそのファイルが表示されます。デザインにタイミング制約または物理制約を適用するには、UCF を XDC (Xilinx® Design Constraints) に変換する必要があります。詳細は、第 3 章「UCF 制約の XDC への移行」を参照してください。

デザイン作成に Vivado インターフェイスを使用する場合は、『Vivado Design Suite ユーザー ガイド : システム デザイン入力』(UG895) を参照してください。

デザイン フローの次のステップについては、『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。

## PlanAhead ツール プロジェクトの変換

PlanAhead™ ツール プロジェクトを Vivado IDE プロジェクトに変換するには、PlanAhead プロジェクト ファイル (拡張子は .ppr) を Vivado IDE で開きます。プロンプトに従い、新しいプロジェクト名および変換されたプロジェクトを保存するディレクトリを設定します。

プロジェクト変換中には次の作業が行われます。

- 7シリーズよりも古いデバイスが設定されているプロジェクトでは、Vivado のデフォルトである 7 シリーズのデバイスがターゲットになります。
- すべての run がリセットされます。run はツールでデザインをインプリメントした後に生成されます。
- 既存の run ストラテジは Vivado のデフォルト ストラテジに置き換えられます。
- UCF 制約はサポートされていないため、[Unsupported Constraints] フォルダーに移動されます。

注記：パーティションのあるデザインの変換はサポートされていません。

---

## XST プロジェクト ファイルのインポート

既存の、または最新の ISE Project Navigator のプロジェクト ファイル (.xise) または PlanAhead ツールのプロジェクト ファイル (.ppr) がない場合は、XST (Xilinx Synthesis Technology) プロジェクト ファイルを使用して、Vivado プロジェクト用に初期設定をインポートできます。XST プロジェクト ファイルをインポートするには、次の手順に従います。

1. [File] → [New Project] をクリックします。
2. プロジェクト名および保存場所を選択します。
3. New Project ウィザードで [Imported Project] をオンにします。
4. [XST] を選択し、拡張子が .xst のプロジェクト ファイルを選択します。

プロジェクト ファイルをインポートしたら、次の作業を行います。

- インポートされたプロジェクトの重要な情報を確認するため、サマリ レポートを開きます。
- 選択したデバイスで要件が満たされていることを確認します。満たされていない場合は新しいデバイスを選択します。XST プロジェクトに Vivado でサポートされている同等のデバイスがない場合は、デフォルト デバイスが選択されます。
- すべてのファイルが正しくインポートされていることを確認するため [Sources] ビューでファイルを確認します。

デザイン作成に Vivado インターフェイスを使用する場合は、『Vivado Design Suite ユーザー ガイド：システム デザイン入力』(UG895) を参照してください。

デザイン フローの次のステップについては、『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。

---

## ソース ファイルの移行

先に説明したように Vivado IDE にプロジェクトをインポートしたり変換する際に、Vivado Design Suite でサポートされているソース ファイルもすべて Vivado プロジェクトに追加できます。

- IP：既存の ISE Design Suite プロジェクトおよび IP は Vivado Design Suite のプロジェクトおよび IP に移行できます。Vivado Design Suite ではインプリメンテーション中に ISE Design Suite IP を使用できます。詳細は、[第 4 章「CORE Generator IP を使用したデザインの Vivado Design Suite への移行」](#)を参照してください。
- ソース ファイル：回路図 (SCH) および Architecture Wizard (XAW) のソース ファイルを除き、既存の ISE Design Suite プロジェクトからのソース ファイルはすべて Vivado Design Suite の新規プロジェクトに追加できます。たとえば、CORE Generator™ ツール プロジェクト ファイル (.xco) およびネットリスト ファイル (.ngc) はデザインソースとして追加できます。

- 制約：デザインまたは IP に使用されているユーザー制約ファイル (UCF) を Vivado Design Suite で使用するには、ザイリンクス デザイン制約 (XDC) フォーマットに変換する必要があります。UCF から XDC への移行については、このガイドの第 3 章「UCF 制約の XDC への移行」を参照してください。



注意：作業中の ISE Design Suite プロジェクトを途中で ISE Design Suite から Vivado Design Suite に移行しないでください。デザイン制約およびスクリプトはこの 2 つの環境間で互換性がありません。

## ISE Design Suite コマンド スクリプトのマッピング

このセクションでは、プロジェクトを使用しない Vivado Design Suite デザイン フロー モードのみを説明します。Vivado ツールで Tcl スクリプトを使用するユーザーを対象にしています。

FPGA デザインをインプリメントするための ISE Design Suite スクリプトを移行するには、Tcl スクリプトを使用することができます。ISE Design Suite と同様、Vivado Design Suite のコンパイル フローでは、デザインが変換された後、変換されたデザインがデバイス用エレメントにマップされ、最適化され、配置配線され、BIT ファイルが生成されてプログラムされます。

次の表はこの 2 つのデザイン フローの主な違いを示しています。

表 2-1 : ISE Design Suite デザイン フロー vs Vivado Design Suite デザイン フロー

ISE Design Suite	Vivado Design Suite
別々のコマンド ライン アプリケーション	シェルの Tcl コマンド
XCF/NCU/UCF/PCF 制約	XDC タイミングおよび物理制約
デザイン制約 (タイミングまたは物理制約) はフローの初期段階でのみ適用	制約 (タイミングまたは物理制約) はフローの任意点で適用、変更、削除可能
複数のデータベース ファイル (NGC、NGD、NCD) が必要	1 つのデータベース ファイル (拡張子が .dcp のチェックポイント) がフローの任意点でオンデマンドで出力される
アプリケーションによるレポート生成	適宜フローの任意点でオンデマンドでレポート生成

次の表は、ISE Design Suite のコマンドとそれに対応する Vivado Design Suite の Tcl コマンドを示します。Tcl コマンドは、次のいずれかの方法で実行できます。

- Vivado IDE の Tcl コンソール
- Tcl プロンプト (`vivado -mode tcl`)
- バッチ スクリプト (`vivado -mode batch -source my.tcl`)



表 2-2 : ISE Design Suite コマンドと Vivado Design Suite の Tcl コマンド

ISE Design Suite コマンド	Vivado Design Suite の Tcl コマンド
xst	read_verilog read_vhdl read_xdc synth_design  注記：この順番でコマンドを実行する必要があります。
ngdbuild	read_edif read_xdc link_design  注記：サードパーティの合成からインポートする場合にのみこれらのコマンドが必要です。synth_design を使用する場合は、このステップは省略してください。
map	opt_design power_opt_design (オプション) place_design phys_opt_design (オプション)
par	route_design
trce	report_timing report_timing_summary
xpwr	read_saif report_power
drc	report_drc
netgen	write_verilog write_vhdl write_sdf
bitgen	write_bitstream

デザイン インプリメンテーションおよび解析に使用するその他の Tcl コマンドの詳細は、『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835) を参照してください。Tcl コマンド プロンプトでヘルプを表示するには、次を入力します。

- help <command>
- <command> -help

Tcl コマンドのカテゴリのヘルプを表示するには、次を入力します。

- help (カテゴリがリストされる)
- help -category <category>

注記：対話型ヘルプにはオートコンプリート機能があり、大文字/小文字は区別されません。このため、Tcl では end、EndGroup は同じカテゴリになります。コマンドやカテゴリのヘルプを表示する場合は、すべて小文字で入力してオートコンプリート機能を使用すると（「endgroup」とタイプする代わりに「end」）、時間の節約になります。

## コマンド ラインの例

次は、run.cmd ファイルに入力できる典型的な ISE Design Suite コマンド ライン run の例です。run.tcl ファイルに入力できる Vivado Design Suite の Tcl コマンドを使用した同じ run もその後に続きます。

## 例 1 : ISE Design Suite コマンドを Vivado Design Suite の Tcl コマンドへマッピング

### ISE Design Suite コマンド ライン

```
xst -ifn design_top.xst

#-ifn (input file name with project settings and options)
ngdbuild -sd ../dd -p xc7v585tffg1157-2 -uc design_top.ucf design_top.ngd

#-sd (search directory), -dd (destination directory), -p (part), -uc (UCF
#file)
map -xe c -w -pr b -ol high -t 2 design_top_map.ncd design_top.pcf
#-xe c (extra effort), -w (overwrite existing file), -pr b (push registers
#into IOBs), -ol (overall effort), -t (placer cost table)
par -xe c -w -ol high -t 2 design_top_map.ncd design_top.ncd
#-xe c (extra effort), -w (overwrite existing file), -ol (overall effort), -t
#(placer cost table)
trce -u -e 10 design_top.ncd design_top.pcf
#-u (report uncovered paths), -e (generate error report)
bitgen -w design_top.ncd design_top.pcf
```

### 同等の Vivado Design Suite の Tcl コマンド

```
set design_name design_top

#read inputs
read_verilog { $design_name.v source2.v source3.v }
read_vhdl -lib mylib { libsource1.vhdl libsource2.vhdl }
read_xdc $design_name.xdc
#run flow and save the database
synth_design -top $design_name -part xc7v585tffg1157-21
write_checkpoint -force ${design_name}_post_synth.dcp
opt_design
place_design
write_checkpoint -force ${design_name}_post_place.dcp
report_utilization -file post_place_util.txt
route_design
#Reports are not generated by default
report_timing_summary -file post_route_timing.txt
#Save the database after post route
write_checkpoint -force ${design_name}_post_route.dcp
#Check for DRC
report_drc -file post_route_drc.txt
# Write Bitstream
write_bitstream -force ${design_name}.bit
```

## 例 2 : サードパーティ合成用の Vivado Design Suite Tcl コマンド (EDIF で開始)

```
set design_name design_top
```

```
#read inputs
read_edif { source1.edf source2.edf $design_name.edf }
read_xdc $design_name.xdc
link_design -part xc7v585tffg1157-2 -top $design_name
#Reports are not generated by default
report_timing_summary -file post_synth_timing_summ.txt
opt_design
place_design
write_checkpoint -force ${design_name}_post_place.dcp
report_utilization -file post_place_util.txt
route_design
#Reports are not generated by default
report_timing_summary -file post_route_timing.txt _summ.txt
#Save the database after post route
write_checkpoint -force ${design_name}_post_route.dcp
#Check for DRC
report_drc -file post_route_drc.txt
# Write Bitstream
write_bitstream -force ${design_name}.bit
```

---

## makefile のマッピング

makefile は、make コマンドによって参照されるテキスト ファイルで、make コマンドのコンパイル方法やプログラムのリンク方法を制御します。makefile には、アクションの実行タイミングを指定するルールや、ソース レベルおよびビルド順序といった情報が含まれます。コンパイル コマンドのシーケンスを決定するため、makefile で依存ファイルのタイムスタンプがチェックされます。makefile の記述例は次のとおりです。

### 例 : ISE Design Suite の makefile を Vivado Design Suite の makefile へマッピング

#### ISE Design Suite で使用する makefile のサンプル

```
DESIGN = test
DEVICE = xc7v585tffg1157-2
UCF_FILE = ../Src/${DESIGN}.ucf
EDIF_FILE = ../Src/${DESIGN}.edf

# Make all runs to place & route
all : place_n_route

# bitstream :Creates device bitstream
bitstream :./${DESIGN}.bit
```

```

# place_n_route:Stops after place and route for analysis prior to bitstream
generation
place_n_route :${DESIGN}.ncd

# translate:Stops after full design elaboration for analysis and floorplanning prior
to place and route step
translate :${DESIGN}.ngd

# Following executes the ISE run

${DESIGN}.bit :${DESIGN}.ncd
bitgen -f ${DESIGN}.ut ${DESIGN}.ncd

${DESIGN}.ncd :${DESIGN}_map.ncd
par -w -ol high ${DESIGN}_map.ncd ${DESIGN}.ncd ${DESIGN}.pcf

${DESIGN}_map.ncd :${DESIGN}.ngd
map -w -ol high -o ${DESIGN}_map.ncd ${DESIGN}.ngd ${DESIGN}.pcf

${DESIGN}.ngd :${EDIF_FILE} ${UCF_FILE}
ngdbuild -uc ${UCF_FILE} -p ${DEVICE} ${EDIF_FILE} ${DESIGN}.ngd

# Clean up all the files from the Vivado run
clean :
rm -rf *.ncd *.ngd *.bit *.mrp *.map *.par *.bld *.pcf *.xml *.bgn *.html \
    *.lst *.ngo *.xrpt *.unroutes *.xpi *.txt *.pad *.csv *.ngm xlnx_auto* \
    _xmsgs *.ptwx

# Tar and compress all the files
tar :
tar -zcvf ${DESIGN}.tar.gz *.ncd *.ngd *.mrp *.map *.par *.bld *.pcf *.bgn \
    Makefile

```

## Vivado Design Suite で使用する同等の makefile

```

DESIGN = test
DEVICE = xc7v585tffg1157-2
XDC_FILE = ../Src/${DESIGN}.xdc
EDIF_FILE = ../Src/${DESIGN}.edf

# Make all runs to place & route
all : place_n_route

```

```
# bitstream :Creates device bitstream
bitstream :./${DESIGN}.bit

# place_n_route:Stops after place and route for analysis prior to bitstream
generation
place_n_route :./${DESIGN}_route.dcp

# translate:Stops after full design elaboration and initial optimization for analysis
and floorplanning prior to place and route step
translate :./${DESIGN}_opt.dcp

# Following calls Tcl files for each desired portion of the Vivado run
# Design checkpoint files and bit file used for dependency management

./${DESIGN}.bit :./run_vivado_place_n_route.tcl ./${DESIGN}_route.dcp
vivado -mode batch -source run_vivado_bitstream.tcl -tclargs ${DESIGN}

./${DESIGN}_route.dcp :./run_vivado_place_n_route.tcl ./${DESIGN}_opt.dcp
vivado -mode batch -source run_vivado_place_n_route.tcl -tclargs \
    ${DESIGN}

./${DESIGN}_opt.dcp :./run_vivado_opt.tcl ${EDIF_FILE} ${XDC_FILE}
vivado -mode batch -source run_vivado_opt.tcl -tclargs ${DESIGN} ${DEVICE}
${EDIF_FILE} ${XDC_FILE}

# Clean up all the files from the Vivado run
clean :
rm -f *.jou *.log *.rpt *.dcp *.bit *.xml *.html

# Tar and compress all the files
tar :
tar -zcvf ${DESIGN}.tar.gz *.jou *.log *.rpt *.dcp *.tcl Makefile
```

## Vivado makefile に関連した Tcl ファイル

### run\_vivado\_opt.tcl

```
# Gathering TCL Args
set DESIGN [lindex $argv 0]
set DEVICE [lindex $argv 1]
set EDIF_FILE [lindex $argv 2]
set XDC_FILE [lindex $argv 3]

# Reading EDIF/NGC file
```

```
read_edif ../Src/${DESIGN}.edf

# Linking Design
link_design -part ${DEVICE} -edif_top_file ../Src/${DESIGN}.edf

# Running Logic Optimization
opt_design

# Adding Constraints
read_xdc ${XDC_FILE}

# Saving Run
write_checkpoint -force ./${DESIGN}_opt.dcp

# Creating opt reports
report_utilization -file ${DESIGN}_utilization_opt.rpt
report_timing_summary -max_paths 10 -nworst 1 -input_pins -
report_io -file ${DESIGN}_io_opt.rpt
report_clock_interaction -file ${DESIGN}_clock_interaction_opt.rpt

exit
```

## run\_vivado\_place\_n\_route.tcl

```
# Gathering TCL Arg
set DESIGN [lindex $argv 0]

read_checkpoint ./${DESIGN}_opt.dcp

# Placing Design
place_design
write_checkpoint -force ./${DESIGN}_place.dcp

# Routing Design
route_design

# Saving Run
write_checkpoint -force ./${DESIGN}_route.dcp

# Creating route reports
report_timing_summary -max_paths 10 -nworst 1 -input_pins -
report_drc -file ${DESIGN}_drc_route.rpt

exit
```

## run\_vivado\_bitstream.tcl

```
# Gathering TCL Arg
set DESIGN [lindex $argv 0]

read_checkpoint ./${DESIGN}_route.dcp
# Create bitstream
write_bitstream -force ${DESIGN}.bit

exit
```

makefile で定義されているステップごとにこのフローは Vivado の終了、再起動を繰り返します。実行はうまく制御できますが、定義されているステップごとにソフトウェアが終了と再起動を繰り返し、デザインを読み込み直す必要があるため、実行時間の面からは効率のよい方法ではありません。makefile による制御よりランタイムを重視する場合は、ステップから次のステップへ移動するときにもデザインをメモリに読み込み直す必要がないので実行全体を Tcl で構築したほうがよいでしょう。

## メッセージの違いについて

Vivado Design Suite では、ISE Design Suite の情報、警告、エラー メッセージと同じコンセプトが使用され、それぞれに ID 番号が付いています (例: ngdbuild:604)。アプリケーションにも HDL-189 などの ID 番号が付いたメッセージが表示されます。Vivado Design Suite にはステータス、クリティカル警告という新しいメッセージ タイプがあります。

- ステータスは実行中のツールプロセスについての情報を表示します。
- Vivado Design Suite のクリティカル警告は ISE Design Suite のエラーと同じですが、Vivado デザイン プロセスが中断しない点が異なります。デザインのクリティカル警告は、ビットストリーム生成 (write\_bitstream) の段階でエラーにアップグレードされ、デザインプロセスが停止します。



**推奨:** デザインを続ける前にクリティカル警告を解決するようにしてください。

次の表は、Vivado Design Suite の 5 つのメッセージ タイプに関してユーザーによる処置の必要性の有無、メッセージの目的などの情報をまとめています。

表 2-3: Vivado Design Suite のメッセージ タイプ

タイプ	処置	目的
STATUS	必要なし	一般的なプロセスのステータスおよびデザイン プロセスに関するユーザーへのフィードバックを示します。 重要度やメッセージ ID タグが含まれていないことを除き、STATUS メッセージは INFO メッセージと同じです。
INFO	必要なし	プロセスの一般的なステータスおよびデザイン プロセスに関するユーザーへのフィードバックを示します。 INFO メッセージは、STATUS メッセージと同じですが、フィルター処理や検索のための重要度やメッセージ ID タグが含まれる点が異なります。
WARNING	オプション	制約または仕様が意図どおりに適用されていないために、最適なデザイン結果が得られない可能性があることを示します。プロセスは完了するまで続行され、有効な結果が生成されます。

表 2-3 : Vivado Design Suite のメッセージ タイプ

タイプ	処置	目的
CRITICAL WARNING	推奨	ハードウェアが正しく動作しない可能性があり、後のフローで <b>ERROR</b> になる可能性があることを示します。プロセスは <b>ERROR</b> が発生するまで続行されます。
ERROR	必要あり	デザイン結果が使用できなくなる問題があることを示し、エラーに対処しない限り回避できません。プロセスは、ここから先に進められません。

## レポートの違いについて

ISE Design Suite では、デザイン フローで各アプリケーションが実行されるとレポートが自動的に生成されます。たとえば次の拡張子のレポートが含まれます。

- xst の場合は .syr
- ngdbuild の場合は .bld
- map の場合は .mrp
- par の場合は .par
- trce の場合は .twr
- xpwv の場合は .pwr

Vivado Design Suite では、どのデザイン段階でもレポートを生成できます。オンデマンドでレポートを作成する利点は次のとおりです。

- より優れたランタイム : 随時レポートを作成することでランタイムをより管理しやすくなります。
- より多くのレポート : デザイン フローの任意時点でレポートを作成できるため、より多くのレポートを利用できるようになります。たとえば、合成後、最適化後、配線後にデザインのリソース使用率レポートを生成して、最新情報を確認することができます。

Vivado IDE のプロジェクト モードを使用する場合は、決まった数のレポートが自動生成され、[Reports] ビューに表示されます。Tcl コマンドまたはスクリプトを使用した非プロジェクト モードを使用する場合、Tcl レポート コマンドを追加して、デザインがメモリに読み込まれている間に指定した段階のレポートが生成されるようにします。特定の `report *` コマンドを使用すると、使用率、タイミング、DRC 結果などのさまざまなタイプの情報をレポートできます。デフォルトでは、レポートはツールのログ ウィンドウと `vivado.log` ファイルに出力されますが、ファイルに出力することも可能です。レポートのリストとその説明は、Tcl コマンドプロンプトで「`help -category report`」と入力すると表示されます。

次の表は、ISE Design Suite のレポート情報がどの Vivado Design Suite コマンドに対応しているかを示しています。

表 2-4 : ISE Design Suite のレポートと Vivado Design Suite のレポート

ISE Design Suite の情報 (レポート)	Vivado Design Suite コマンド
使用率情報 (.syr, .mrp, .par)	<code>report_utilization</code> 、 <code>report_clock_utilization</code>
I/O 情報 (.pad)	<code>report_io</code>
タイミング情報 (.par, .twr)	<code>report_timing</code> 、 <code>report_timing_summary</code>
消費電力情報 (.pwr)	<code>report_power</code>



## ログ ファイルの違いについて

ISE Design Suite ツールでは、個々のコマンド ログ ファイルにステータスおよび出力情報が生成されます。たとえば、MAP の出力ステータスおよび進行状況は拡張子が `.map` のファイルに出力され、PAR の出力は拡張子が `.par` のファイルに出力されます。

Vivado Design Suite では、すべてのツール コマンドおよび出力を記録したログ ファイルが 1 つ使用されます。このファイルの名前はデフォルトで `vivado.log` ですが、`-log` オプションを使用すると変更できます。Vivado Design Suite のログ ファイルにはフェーズごとにフローの進捗状況が表示されます。各フェーズには、名前と数、1 行のパフォーマンス サマリが含まれます。次に例を示します。

```
report_timing:Time (s): cpu = 00:03:57 ; elapsed = 00:03:55 .Memory (MB): peak =  
6526.066 ; gain = 64.125
```

説明:

- `cpu`: すべてのプロセッサのランタイム合計
- `elapsed`: プロセスを実行するのに実際に使用した時間
- `peak`: その特定のデザイン段階までの最大メモリ使用量
- `gain`: 特定のデザイン段階までのピーク メモリ使用量に追加されるメモリ使用量。たとえば、上記の例では、`report_timing` によりピーク メモリ使用量に 64.125 MB 追加されています。

# UCF 制約の XDC への移行

## 概要

Vivado™ 統合設計環境 (IDE) では ISE® Design Suite で使用されていたユーザー制約ファイル (UCF) はサポートされていません。UCF 制約を使用しているデザインは、ザイリンクス デザイン制約 (XDC) に移行する必要があります。

- XDC 制約の詳細については、『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。
- UCF 制約の詳細は、『制約ガイド』(UG625) を参照してください。

UCF と同様に、XDC は次のものから構成されています。

- タイミング制約。XDC タイミング制約は Synopsys デザイン制約 (SDC) を基本にしています。
- 物理制約

## XDC と UCF 制約の違い

XDC と UCF 制約には基本的な違いがあります。

- XDC はシーケンシャル言語で、明確な優先順位ルールがあります。
- UCF は通常ネットに適用されますが、XDC は通常ピン、ポート、およびセルオブジェクトに適用されます。
- UCF の PERIOD 制約と、XDC の `create_clock` コマンドは常に同等というわけではなく、異なるタイミング結果を出力することがあります。
- UCF の場合、デフォルトでは非同期クロック グループ間のタイミングは処理されませんが、XDC の場合は制約が設定されていない限り (`set_clock_groups`)、すべてのクロックは関連しているものとしてタイミングが処理されます。
- XDC では、同じオブジェクトに複数のクロックを存在させることが可能です。

## UCF から XDC へのマッピング

表 3-1、「UCF から XDC へのマッピング」では、UCF 制約から対応する XDC コマンドへの主なマッピングを示しています。

表 3-1 : UCF から XDC へのマッピング

UCF	SDC
TIMESPEC PERIOD	create_clock create_generated_clock
OFFSET = IN <x> BEFORE <clk>	set_input_delay
OFFSET = OUT <x> BEFORE <clk>	set_output_delay
FROM:TO “TS_”*2	set_multicycle_path
FROM:TO	set_max_delay
TIG	set_false_path
NET “clk_p” LOC = AD12	set_property LOC AD12 [get_ports clk_p]
NET “clk_p” IOSTANDARD = LVDS	set_property IOSTANDARD LVDS [get_ports clk_p]

## 制約の順序

デザインで使用している XDC ファイルの数に関わらず、次の順序で制約をまとめることを推奨します。

### ## タイミング アサートのセクション

- # プライマリ クロック
- # 仮想クロック
- # 生成されたクロック
- # クロック グループ
- # 入力および出力遅延制約

### ## タイミング例外のセクション（優先度で分類）

- # フォルス パス
- # 最大遅延/最小遅延
- # マルチサイクル パス
- # ケース解析
- # ディスエーブル タイミング

### ## 物理制約のセクション

- # ファイルの任意位置にあるもの（できればタイミング制約の前後にあるもの）。
- # または別の XDC ファイルに保存されているもの

XDC コマンドの詳細は、次の資料を参照してください。

- 『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835)
- 『Vivado Design Suite ユーザー ガイド : 制約の使用』(UG903)

## PlanAhead ツールで UCF を XDC に変換

PlanAhead™ ツールで、UCF 制約を含む ISE Design Suite または PlanAhead プロジェクトを開くと、UCF 制約を XDC へ変換することができます。デザインをデータベースに読み込む際に、write\_xdc コマンドを使用すると、UCF 制約の大部分を変換できます。この出力ファイルは手動で検証する必要があります。すべてのデザイン制約が正しいことを確認するには、一部の制約を手動で XDC に変換する必要もあります。

Tcl コマンド `write_xdc` を使用するには、合成済みのネットリストが開いていて、1 つまたは複数の UCF ファイルが読み込まれている必要があります。PlanAhead ツールで次の操作を行います。

1. UCF 制約を含むプロジェクトを開きます。
2. [Open Synthesized Design] をクリックします。
3. Tcl コンソールに「`write_xdc filename.xdc`」と入力します。

`write_xdc` コマンドは、ファイル コンバーターではありません。デザインに適用することができた制約を XDC ファイルとして記述するコマンドです。出力される XDC ファイルには次のものが含まれています。

- 変換された各 UCF 制約の UCF ファイル名および行番号を含むコメント
- 変換が行われなかったものに対するコメント



**重要：**制約が変換できなかったことを示すクリティカル警告メッセージに注意してください。

この変換は XDC ベースの制約へ移行する際の起点として使用することを目的としています。



**推奨：**UCF と XDC の根本的な違いのため自動化では最適な結果を得ることはできないので、この変換プロセスを使用せずに XDC タイミング制約を作成することを推奨します。

PlanAhead ツールでの UCF 変換は、物理制約および基本的なタイミング制約で使用するのがベストです。通常、単純クロック定義および I/O 遅延のタイミング制約はうまく変換できますが、タイミング例外は手動で変換することを推奨します。多くは変換されず、また変換されたとしても最適ではない結果となる可能性があります。Vivado IDE (XDC/SDC) と ISE Design Suite (UCF) のタイミング制約には根本的な違いがあるので、直接変換することは不可能です。UCF 制約を使用していたときのタイミング制約設定を再考し、XDC を使用した新たなアプローチが必要になる場合があります。

エラボレートされた RTL デザインを使用して変換することは可能ですが、標準 UCF で参照されているオブジェクトの多くはこの段階では存在せず、データベースには適用されないことがほとんどです。データベースに適用された制約のみが XDC として出力されます。このため、通常は単純クロックおよび I/O 遅延のみをエラボレートされた RTL デザインから変換できます。

## タイミング制約

Vivado Design Suite で XDC タイミング制約として変換することができる ISE Design Suite のタイミング制約を説明します。各制約に対して UCF の記述例、同等の XDC の記述例を示します。

ポートなどデザインの境界に直接接続されていないネットにクロックを作成するとき、UCF と XDC は異なります。XDC の場合、ネットに `create_clock` を使用してプライマリ クロックを定義するときのソース起点はネットの駆動ピンで、この起点よりも前のクロック挿入遅延は無視されます。このため、ほかの関連クロックとこのクロックのタイミングを処理するときにスキューが正確にはならないので、問題になる可能性があります。`create_clock` は、デザインの途中ではなく、入力ポートや GT クロック出力ピンなどのクロック ツリーの起点に使用する必要があります。デザインの途中には生成されたクロックのみを作成します。

## クロック制約

### PERIOD

#### UCF の例

```
NET "clka" TNM_NET = "clka";  
TIMESPEC "TS_clka" = PERIOD "clka" 13.330 ns HIGH 50.00%;
```

#### XDC の例

```
create_clock -name clka -period 13.330 -waveform {0 6.665} [get_ports clka]
```

### デューティ サイクル が 50% でない場合の PERIOD 制約

#### UCF の例

```
NET "clka" TNM_NET = "clka";  
TIMESPEC "TS_clka" = PERIOD "clka" 13.330 ns HIGH 40.00%;
```

#### XDC の例

```
create_clock -name clka -period 13.330 -waveform {0 5.332} [get_ports clka]
```

### 生成されたクロックの制約

#### UCF の例

```
NET "gen_clk" TNM_NET = "gen_clk";  
TIMESPEC "TS_gen_clk" = PERIOD "gen_clk" "TS_clka" * 0.500 HIGH 50.00%;
```

#### XDC の例

```
create_generated_clock -source [get_ports clka] -name gen_clk -multiply_by 2  
[get_ports gen_clk]
```

### LOW キーワードを含む PERIOD 制約

#### UCF の例

```
NET "clka" TNM_NET = "clka";  
TIMESPEC "TS_clka" = PERIOD "clka" 13.330 ns LOW 50.00%;
```

#### XDC の例

```
create_clock -name clka -period 13.330 -waveform {6.665 13.330} [get_ports clka]
```

### ネットの PERIOD 制約

#### UCF の例

```
NET "clk_bufg" PERIOD = 10 ns;
```

#### XDC の例

```
create_clock -name clk_bufg -period 10 -waveform {0 5} [get_pins clk_bufg/O]
```

注記: クロックを bufg/O に定義する特別な理由がない限り、アップストリームの最上位ポートに定義してください。

## OFFSET IN

### BEFORE

#### UCF の例

```
OFFSET = IN 8 BEFORE clka;
```

#### XDC の例

```
set_input_delay -clock clka 2 [all_inputs]
```

注記: この例では、クロック周期は 10ns であると想定しています。

### AFTER

#### UCF の例

```
OFFSET = IN 2 AFTER clka;
```

#### XDC の例

```
set_input_delay -clock clka 2 [all_inputs]
```

注記: この例では、クロック周期は 10ns であると想定しています。

### BEFORE (入力ポート ネットの前)

#### UCF の例

```
NET enable OFFSET = IN 8 BEFORE clka;
```

#### XDC の例

```
set_input_delay 2 [get_ports enable]
```

注記: この例では、クロック周期は 10ns であると想定しています。

### BEFORE (入力ポート バスの前)

#### UCF の例

```
INST "processor_data_bus[*]" TNM = "processor_bus";  
TIMEGRP "processor_bus" OFFSET = IN 8ns BEFORE "clka";
```

#### XDC の例

```
set_input_delay 2 [get_ports {processor_data_bus[*]}]
```

注記: オフセットはポートのみに適用されます。

### TIMEGROUP へ

#### UCF の例

```
INST "input_ffs[*]" TNM = "input_ffs";
```

```
OFFSET = IN 8ns BEFORE "clka" TIMEGRP "input_ffs";
```

### XDC の例

手動変換が必要です。詳細は、「[タイムグループ](#)」セクションを参照してください。

## FALLING/RISING (立ち下がり/立ち上がりエッジ)

### UCF の例

```
OFFSET = IN 8ns BEFORE "clka" FALLING;
```

### XDC の例

```
set_input_delay -clock clka 2 [all_inputs]
```

注記: この例では、クロック周期は 10ns であると想定しています。

## LOW/HIGH キーワード

### UCF の例

```
OFFSET = IN 8ns BEFORE "clka" HIGH;
```

### XDC の例

手動変換が必要です。

注記: HIGH/LOW キーワードは RISING/FALLING につながるキーワードです。RISING/FALLING を使用を推奨します。

## VALID キーワード

### UCF の例

```
OFFSET = IN 1ns VALID 2ns BEFORE clka;
```

### XDC の例

```
set_input_delay -clock clka -max 9 [all_inputs]  
set_input_delay -clock clka -min 1 [all_inputs]
```

注記: この例では、クロック周期は 10ns であると想定しています。

## OFFSET OUT

### AFTER

### UCF の例

```
OFFSET = OUT 12 AFTER clkc;
```

### XDC の例

```
set_output_delay -clock clkc 8 [all_outputs]
```

注記: この例では、クロック周期は 20ns であると想定しています。

## BEFORE

### UCF の例

```
OFFSET = OUT 8 BEFORE clk;
```

### XDC の例

```
set_output_delay -clock clk 8 [all_outputs]
```

注記: この例では、クロック周期は 20ns であると想定しています。

## 出力ネット

### UCF の例

```
NET out_net OFFSET = OUT 12 AFTER clk;
```

### XDC の例

```
set_output_delay 8 [get_port out_net]
```

注記: この例では、クロック周期は 20ns であると想定しています。

## 出力のグループ

### UCF の例

```
TIMEGRP outputs OFFSET = OUT 12 AFTER clk;
```

### XDC の例

```
set_output_delay -clock clk 8 [get_ports outputs*]
```

注記: この例では、クロック周期は 20ns であると想定しています。

## TIMEGROUP から

### UCF の例

```
OFFSET = OUT 1.2 AFTER clk TIMEGRP from_ffs;
```

### XDC の例

手動変換が必要です。

## FALLING/RISING (立ち下がり/立ち上がりエッジ) キーワード

### UCF の例

```
OFFSET = OUT 12 AFTER clk FALLING;
```

### XDC の例

```
set_output_delay -clock clk -clock_fall 8 [all_outputs]
```



## LOW キーワード

### UCF の例

```
OFFSET = OUT 12 AFTER clk LOW;
```

### XDC の例

手動変換が必要です。

**注記** : HIGH/LOW キーワードは RISING/FALLING につながるキーワードです。RISING/FALLING を使用を推奨します。

## REFERENCE\_PIN

### UCF の例

```
TIMEGRP mac_ddr_out;  
OFFSET = OUT AFTER clk REFERENCE_PIN clk_out RISING;
```

### XDC の例

手動変換が必要です。

**注記** : REFERENCE\_PIN を使用すると、TRACE でバス スキュー レポートが出力されますが、Vivado Design Suite ではこの機能はサポートされていません。

## FROM:TO 制約

通常、UCF の FROM:TO 制約は XDC の set\_max\_delay または set\_min\_delay 制約に変換され、またデザインにより異なりますが、-from、-to、および、-through という引数が使用されます。

UCF 制約は同等の XDC 制約に置き換える必要があります。ほとんどの UCF 制約はネットを基準としていますが、XDC 制約はポートおよびピンに対して作成する必要があります。

この変換には、all\_fanout、get\_cells、および get\_pins といった XDC コマンド、-from、-to、および -through といった引数が役立ちます。

## その他

### エリア グループへのタイミング グループの割り当て

#### UCF の例

```
TIMEGRP clock_grp = AREA_GROUP clock_ag;
```

#### XDC の例

Vivado Design Suite では XDC でのこの制約はサポートされていません。

## 例外

### UCF の例

```
TIMEGRP my_group = FFS EXCEPT your_group;
```

## XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

## タイミング無視 (TIG)

### グループ間

#### UCF の例

```
TIMESPEC TS_TIG = FROM reset_ff TO FFS TIG;
```

#### XDC の例

手動変換が必要です。

目的のパスを含む `set_false_path` を作成します。

### ネットで指定

#### UCF の例

```
NET reset TIG;
```

#### XDC の例

```
set_false_path -through [get_nets reset]
```

注記: プライマリ リセット ポートを検索し使用する方法の方が適切です。

```
set_false_path -from [get_ports reset_port]
```

### インスタンスで指定

#### UCF の例

```
INST reset TIG;
```

#### XDC の例

```
set_false_path -from [get_cells reset]  
set_false_path -through [get_cells reset]  
set_false_path -to [get_cells reset]
```

### ピンで指定

#### UCF の例

```
PIN ff.d TIG;
```

#### XDC の例

```
set_false_path -to [get_pins ff/d]  
set_false_path -from [get_pins ff/q]  
set_false_path -through [get_pins lut/i0]
```

## 特定のタイミング制約

### UCF の例

```
NET reset TIG = TS_fast TS_even_faster;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

注記：制約別の TIG では、ネットのタイミングが無視されますが、参照されている 2 つの制約の解析に対してのみ無視されます。

## MAXSKEW

### UCF の例

```
NET local_clock MAXSKEW = 2ns;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

## MAXDELAY

### UCF の例

```
NET local_clock MAXDELAY = 2ns;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

---

# タイムグループ

INST/TNM および TIMESPEC と同じ効果を得るため、タイミング例外を使って Tci 変数を使用することができます。次はその例です。

UCF の例：

```
INST "DUT/BLOCK_A/data_reg[*]" TNM = "from_data_reg_0";  
INST "DUT/BLOCK_A/addr_reg[*]" TNM = "from_data_reg_0";  
INST "DUT/BLOCK_B/data_sync[*]" TNM = "to_data_reg_0";  
INST "DUT/BLOCK_B/addr_sync[*]" TNM = "to_data_reg_0";  
TIMESPEC "TS_MCP" = FROM "from_data_reg_0" TO "to_data_reg_0" TS_FSCLK * 3;
```

Tcl を使用した場合 :

```
set from_data_reg_0 [get_cells {DUT/BLOCK_A/data_reg[*] DUT/BLOCK_A/addr_reg[*]}];  
set to_data_reg_0 [get_cells {DUT/BLOCK_B/data_sync[*] DUT/BLOCK_B/addr_sync[*]}];  
set_multicycle_path -setup 3 -from $from_data_reg_0 -to $to_data_reg_0;  
set_multicycle_path -hold 2 -from $from_data_reg_0 -to $to_data_reg_0;
```

---

## 物理制約

Vivado Design Suite で XDC 制約に変換できる ISE Design Suite の物理制約を説明します。各制約に対し次の点が説明されています。

- ターゲット オブジェクト タイプ
- 制約値タイプ
- UCF の例
- 同等の XDC の例

## 配置制約

### AREA\_GROUP

#### 適用箇所

セル

#### 制約値

文字列

#### UCF の例

```
INST bmg0 AREA_GROUP = AG1;
```

#### XDC の例

```
create_pblock ag1; add_cells_to_pblock [get_pblocks ag1] [get_cells [list bmg0]]
```

### AREA\_GROUP RANGE

## スライス

#### 適用箇所

エリア グループおよび Pblock

#### 制約値

SLICE\_XnYn[:SLICE\_XnYn]

### UCF の例

```
AREA_GROUP AG1 RANGE = SLICE_X0Y44:SLICE_X27Y20;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {SLICE_X0Y44:SLICE_X27Y20}
```

## RAMB18

### 適用箇所

エリア グループおよび Pblock

### 制約値

RAMB18\_XnYn:RAMB18\_XnYn

### UCF の例

```
AREA_GROUP AG1 RANGE = RAMB18_X0Y86:RAMB18_X3Y95;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {RAMB18_X0Y86:RAMB18_X3Y95}
```

## RAMB36

### 適用箇所

エリア グループおよび Pblock

### 制約値

RAMB36\_XnYn:RAMB36\_XnYn

### UCF の例

```
AREA_GROUP AG1 RANGE = RAMB36_X0Y11:RAMB36_X3Y18;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {RAMB36_X0Y11:RAMB36_X3Y18}
```

## CLOCKREGION (1)

### 適用箇所

エリア グループおよび Pblock

### 制約値

CLOCKREGION\_XnYn

### UCF の例

```
area_group ag1 range = CLOCKREGION_X0Y0;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {CLOCKREGION_X0Y0:CLOCKREGION_X0Y0}
```

### CLOCKREGION (2)

#### 適用箇所

エリア グループおよび Pblock

#### 制約値

CLOCKREGION\_XnYn[:CLOCKREGION\_XnYn]

### UCF の例

```
area_group ag1 range = CLOCKREGION_X0Y0:CLOCKREGION_X1Y0;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {CLOCKREGION_X0Y0:CLOCKREGION_X0Y0}
```

### CLOCKREGION (3)

#### 適用箇所

エリア グループおよび Pblock

#### 制約値

CLOCKREGION\_XnYn,CLOCKREGION\_XnYn, ...

### UCF の例

```
area_group ag1 range = CLOCKREGION_X0Y0, CLOCKREGION_X1Y0;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {CLOCKREGION_X0Y0:CLOCKREGION_X0Y0  
CLOCKREGION_X1Y0:CLOCKREGION_X1Y0}
```

### DSP48

#### 適用箇所

エリア グループおよび Pblock

#### 制約値

DSP48\_XnYn:DSP48\_XnYn

### UCF の例

```
AREA_GROUP D1 RANGE = DSP48_X2Y0:DSP48_X2Y9;
```

### XDC の例

```
resize_pblock [get_pblocks D1] -add {DSP48_X2Y0:DSP48_X2Y9}
```

## BUFGCTRL

### 適用箇所

エリア グループおよび Pblock

### 制約値

BUFGCTRL\_XnYn:BUFGCTRL\_XnYn

### UCF の例

```
AREA_GROUP ag1 range = BUFGCTRL_X0Y24:BUFGCTRL_X0Y31;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {BUFGCTRL_X0Y24:BUFGCTRL_X0Y31}
```

## BUFHCE

### 適用箇所

エリア グループおよび Pblock

### 制約値

BUFHCE\_XnYn:BUFHCE\_XnYn

### UCF の例

```
AREA_GROUP ag1 range = BUFHCE_X0Y72:BUFHCE_X1Y77;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {BUFHCE_X0Y72:BUFHCE_X1Y77}
```

## BUFR

### 適用箇所

エリア グループおよび Pblock

### 制約値

BUFR\_XnYn:BUFR\_XnYn

### UCF の例

```
AREA_GROUP ag1 range = BUFR_X0Y20:BUFR_X1Y23;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {BUFR_X0Y0:BUFR_X1Y2}
```

## BUFIO

### 適用箇所

エリア グループおよび Pblock

### 制約値

BUFIO\_XnYn:BUFIO\_XnYn

### UCF の例

```
AREA_GROUP ag1 range = BUFIO_X0Y8:BUFIO_X0Y11;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {BUFIO_X0Y8:BUFIO_X0Y11}
```

## IOB の範囲

### 適用箇所

エリア グループおよび Pblock

### 制約値

IOB\_XnYn:IOB\_XnYn

### UCF の例

```
AREA_GROUP ag1 range = IOB_X0Y341:IOB_X1Y349;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {IOB_X0Y341:IOB_X1Y349}
```

## IN\_FIFO

### 適用箇所

エリア グループおよび Pblock

### 制約値

IN\_FIFO\_XnYn:IN\_FIFO\_XnYn

### UCF の例

```
AREA_GROUP ag1 range = IN_FIFO_X0Y24:IN_FIFO_X1Y27;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {IN_FIFO_X0Y24:IN_FIFO_X1Y27}
```



## OUT\_FIFO

### 適用箇所

エリア グループおよび Pblock

### 制約値

OUT\_FIFO\_XnYn:OUT\_FIFO\_XnYn

### UCF の例

```
AREA_GROUP ag1 range = OUT_FIFO_X0Y24:OUT_FIFO_X1Y27;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {OUT_FIFO_X0Y24:OUT_FIFO_X1Y27}
```

## ILOGIC

### 適用箇所

エリア グループおよび Pblock

### 制約値

ILOGIC\_XnYn:ILOGIC\_XnYn

### UCF の例

```
AREA_GROUP ag1 range = ILOGIC_X0Y76:ILOGIC_X0Y79;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {ILOGIC_X0Y76:ILOGIC_X0Y79}
```

## OLOGIC

### 適用箇所

エリア グループおよび Pblock

### 制約値

OLOGIC\_XnYn:OLOGIC\_XnYn

### UCF の例

```
AREA_GROUP ag1 range = OLOGIC_X0Y76:OLOGIC_X0Y79;
```

### XDC の例

```
resize_pblock [get_pblocks ag1] -add {OLOGIC_X0Y76:OLOGIC_X0Y79}
```

## LOC

### IOB

#### 適用箇所

ポート ネット

#### 制約値

IOB サイト

#### UCF の例

```
NET p[0] LOC = H1;
```

#### XDC の例

```
set_property PACKAGE_PIN H1 [get_ports p[0]]
```



---

**ヒント** : Vivado Design Suite でピンを割り当てるには、セルに使用している LOC ではなく、PACKAGE\_PIN ポート プロパティを使用します。

---

### SLICE (1)

#### 適用箇所

セル

#### 制約値

サイト 範囲

#### UCF の例

```
INST a_reg[*] LOC = SLICE_X25Y*;
```

#### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

### SLICE (2)

#### 適用箇所

セル

#### 制約値

SLICE\_XnYn

#### UCF の例

```
INST a_reg[0] LOC = SLICE_X4Y4;
```

### XDC の例

```
set_property LOC SLICE_X4Y4 [get_cells a_reg[0]]
```

### 適用箇所

セル

### 制約値

RAMB18\_XnYn

### UCF の例

```
INST ram0 LOC = RAMB18_X0Y5;
```

### XDC の例

```
set_property LOC RAMB18_X0Y5 [get_cells ram0]
```

## RAMB36

### 適用箇所

セル

### 制約値

RAMB36\_XnYn

### UCF の例

```
INST ram0 LOC = RAMB36_X0Y0;
```

### XDC の例

```
set_property LOC RAMB36_X0Y0 [get_cells ram0]
```

## DSP48

### 適用箇所

セル

### 制約値

DSP48\_XnYn

### UCF の例

```
INST dsp0 LOC = DSP48_X0Y10;
```

### XDC の例

```
set_property LOC DSP48_X0Y10 [get_cells dsp0]
```

## BUFGCTRL

### 適用箇所

セル

### 制約値

BUFGCTRL\_XnYn

### UCF の例

```
INST cb[0] LOC = BUFGCTRL_X0Y24;
```

### XDC の例

```
set_property LOC BUFGCTRL_X0Y24 [get_cells cb[0]]
```

## BUFHCE

### 適用箇所

セル

### 制約値

BUFHCE\_XnYn

### UCF の例

```
INST cb[0] LOC = BUFHCE_X0Y72;
```

### XDC の例

```
set_property LOC BUFHCE_X0Y72 [get_cells cb[0]]
```

## BUFR

### 適用箇所

セル

### 制約値

BUFR\_XnYn

### UCF の例

```
INST cb[0] LOC = BUFR_X0Y20;
```

### XDC の例

```
set_property LOC BUFR_X0Y20 [get_cells cb[0]]
```

## BUFIO

### 適用箇所

セル

### 制約値

BUFIO\_XnYn

### UCF の例

```
INST cb[0] LOC = BUFIO_X0Y8;
```

### XDC の例

```
set_property LOC BUFIO_X0Y8 [get_cells cb[0]]
```

## IOB

### 適用箇所

セル

### 制約値

IOB\_XnYn

### UCF の例

```
INST ib[0] LOC = IOB_X0Y341;
```

### XDC の例

```
set_property LOC IOB_X0Y341 [get_cells ib[0]]
```

## IN\_FIFO

### 適用箇所

セル

### 制約値

IN\_FIFO\_XnYn

### UCF の例

```
INST infifo_inst LOC = IN_FIFO_X0Y24;
```

### XDC の例

```
set_property LOC IN_FIFO_X0Y24 [get_cells infifo_inst]
```

## OUT\_FIFO

### 適用箇所

セル

### 制約値

OUT\_FIFO\_XnYn

### UCF の例

```
INST outfifo_inst LOC = OUT_FIFO_X0Y24;
```

### XDC の例

```
set_property LOC OUT_FIFO_X0Y24 [get_cells outfifo_inst]
```

## ILOGIC

### 適用箇所

セル

### 制約値

ILOGIC\_XnYn

### UCF の例

```
INST ireg LOC = ILOGIC_X0Y76;k
```

### XDC の例

```
set_property LOC ILOGIC_X0Y76 [get_cells ireg]
```

## OLOGIC

### 適用箇所

セル

### 制約値

OLOGIC\_XnYn

### UCF の例

```
INST oreg LOC = OLOGIC_X0Y76;
```

### XDC の例

```
set_property LOC OLOGIC_X0Y76 [get_cells oreg]
```

## IDELAY

### 適用箇所

セル

### 制約値

IDELAY\_XnYn

### UCF の例

```
INST idelay0 LOC = IDELAY_X0Y21;
```

### XDC の例

```
set_property LOC IDELAY_X0Y21 [get_cells idelay0]
```

## IDELAYCTRL

### 適用箇所

セル

### 制約値

IDELAYCTRL\_XnYn

### UCF の例

```
INST idelayctrl0 LOC = IDELAYCTRL_X0Y0;
```

### XDC の例

```
set_property LOC IDELAYCTRL_X0Y0 [get_cells idelayctrl0]
```

## BEL

### A5LUT、B5LUT、C5LUT、D5LUT

### 適用箇所

セル

### 制約値

A5LUT、B5LUT、C5LUT、D5LUT

### UCF の例

```
INST a0 BEL = A5LUT;
```

### XDC の例

```
set_property BEL A5LUT [get_cells a0]
```

## A6LUT、B6LUT、C6LUT、D6LUT

### 適用箇所

セル

### 制約値

A6LUT、B6LUT、C6LUT、D6LUT

### UCF の例

```
INST a0 BEL = D6LUT;
```

### XDC の例

```
set_property BEL D6LUT [get_cells a0]
```

## AFF、BFF、CFF、DFF

### 適用箇所

セル

### 制約値

AFF、BFF、CFF、DFF

### UCF の例

```
INST a_reg[0] BEL = CFF;
```

### XDC の例

```
set_property BEL CFF [get_cells a_reg[0]]
```

## A5FF、B5FF、C5FF、D5FF

### 適用箇所

セル

### 制約値

A5FF、B5FF、C5FF、D5FF

### UCF の例

```
INST a_reg[0] BEL = B5FF;
```

### XDC の例

```
set_property BEL B5FF [get_cells a_reg[0]]
```



## F7AMUX、F7BMUX

### 適用箇所

セル

### 制約値

F7AMUX、F7BMUX

### UCF の例

```
INST m0 BEL = F7BMUX;
```

### XDC の例

```
set_property BEL F7BMUX [get_cells m0]
```

## IOB

### TRUE

#### 適用箇所

FF セル

#### 制約値

TRUE

#### UCF の例

```
INST a1_reg[*] IOB = TRUE;
```

#### XDC の例

```
set_property IOB TRUE [get_cells b1_reg[*]]
```

### FALSE

#### 適用箇所

FF セル

#### 制約値

FALSE

#### UCF の例

```
INST b1_reg[*] IOB = FORCE;
```

#### XDC の例

```
set_property IOB TRUE [get_cells a1_reg[*]]
```

## FORCE

### 適用箇所

FF セル

### 制約値

FORCE

### UCF の例

```
INST q_reg[*] IOB = FALSE;
```

### XDC の例

```
set_property IOB TRUE [get_cells q_reg[*]]
```

Vivado Design Suite では XDC でこの制約はサポートされていません。TRUE を使用してください。

## H\_SET

### 適用箇所

セル

### 制約値

ツールで生成される文字列

### UCF の例

なし

### XDC の例

なし

詳細は、『制約ガイド』(UG625) の「RLOC」セクションを参照してください。

Vivado Design Suite では、H\_SET セルに RPM というプロパティがあります。

## HU\_SET

### 適用箇所

セル

### 制約値

文字列

### UCF の例

```
INST u0 HU_SET = h0;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

HU\_SET は属性として HDL コードに記述する必要があります。

詳細は、『制約ガイド』(UG625) の「RLOC」セクションを参照してください。

## U\_SET

### 適用箇所

セル

### 制約値

文字列

### UCF の例

```
INST u0 U_SET = h0; (usually set in UCF)
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

U\_SET は属性として HDL コードに記述する必要があります。

詳細は、『制約ガイド』(UG625) の「RLOC」セクションを参照してください。

## RLOC

### 適用箇所

セル

### 制約値

XnYn

### UCF の例

```
INST u0 RLOC = X2Y1;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

RLOC は属性として HDL コードに記述する必要があります。

詳細は、『制約ガイド』(UG625) の「RLOC」セクションを参照してください。

## RLOC\_ORIGIN

### 適用箇所

セル

## 制約値

XnYn

## UCF の例

```
INST u0 RLOC_ORIGIN = X144Y255;
```

## XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

RLOC\_ORIGIN は属性として HDL コードに記述する必要があります。

詳細は、『制約ガイド』(UG625) の「RLOC」セクションを参照してください。

## RPM\_GRID

### 適用箇所

セル

### 制約値

GRID

### UCF

```
INST u0 RPM_GRID = GRID;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

RPM\_GRID は属性として HDL コードに記述する必要があります。

詳細は、『制約ガイド』(UG625) の「RLOC」セクションを参照してください。

## USE\_RLOC

### 適用箇所

セル

### 制約値

TRUE、FALSE

### UCF の例

```
INST u0 USE_RLOC = FALSE;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

## RLOC\_RANGE

### 適用箇所

セル

### 制約値

XnYn:XnYn

### UCF の例

```
INST u0 RLOC_RANGE = X1Y1:X3Y3;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

範囲を指定して Pblock を作成し、RPM セルを Pblock に追加します。

## BLKNM

### 適用箇所

セル

### 制約値

文字列

### UCF の例

```
INST u0 BLKNM = blk0;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

可能であれば、U\_SET またはネットの重みを使用してください。

### 適用箇所

ネット

### 制約値

文字列

### UCF の例

```
NET n0 BLKNM = blk0;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

可能であれば、U\_SET またはネットの重みを使用してください。

## HBLKNM

### 適用箇所

セル

### 制約値

文字列

### UCF の例

```
INST u0 HBLKNM = blk0;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

可能であれば、H\_SET/HU\_SET またはネットの重みを使用してください。

### 適用箇所

ネット

### 制約値

文字列

### UCF の例

```
NET n0 HBLKNM = blk0;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

可能であれば、H\_SET/HU\_SET またはネットの重みを使用してください。

## XBLKNM

### 適用箇所

セル

### 制約値

文字列

### UCF の例

```
INST u0 XBLKNM = blk0;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

可能であれば、H\_SET/HU\_SET またはネット ウェイトを使用してください。

関連していないロジックを除外するには **BEL PROHIBIT** を使用します。

### 適用箇所

ネット

### 制約値

文字列

### UCF の例

```
NET n0 XBLKNM = blk0;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

可能であれば、**H\_SET/HU\_SET** またはネット ウェイトを使用してください。

関連していないロジックを除外するには **BEL PROHIBIT** を使用します。

## HLUTNM

### 適用箇所

LUT セル

### 制約値

文字列

### UCF の例

UCF では使用できません。HDL でのみ使用できます。

### XDC の例

```
set_property HLUTNM h0 [get_cells {LUT0 LUT1}]
```

## LUTNM

### 適用箇所

LUT セル

### 制約値

文字列

### UCF の例

UCF では使用できません。HDL でのみ使用できます。

### XDC の例

```
set_property LUTNM h0 [get_cells {LUT0 LUT1}]
```

## USE\_LUTNM

### 適用箇所

LUT セル

### 制約値

TRUE、FALSE

### UCF の例

```
INST lut0 USE_LUTNM = FALSE;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

## CLOCK\_DEDICATED\_ROUTE

### TRUE(1)

### 適用箇所

ネット

### 制約値

TRUE

### UCF の例

```
net clk0 CLOCK_DEDICATED_ROUTE = TRUE;
```

### XDC の例

```
set_property CLOCK_DEDICATED_ROUTE TRUE [get_nets clk0]
```

### TRUE(1)

### 適用箇所

ピン

### 制約値

TRUE

### UCF の例

```
PIN clkbuf0.0 CLOCK_DEDICATED_ROUTE = TRUE;
```

### XDC の例

```
set_property CLOCK_DEDICATED_ROUTE TRUE [get_pins clkbuf0/0]
```



## FALSE(1)

### 適用箇所

ネット

### 制約値

FALSE

### UCF の例

```
NET clk0 CLOCK_DEDICATED_ROUTE = FALSE;
```

### XDC の例

```
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets clk0]
```

## FALSE(2)

### 適用箇所

ピン

### 制約値

FALSE

### UCF の例

```
PIN clkbuf0.O CLOCK_DEDICATED_ROUTE = FALSE;
```

### XDC の例

```
set_property CLOCK_DEDICATED_ROUTE FALSE [get_pins clkbuf0/O]
```

## BACKBONE(1)

### 適用箇所

ネット

### 制約値

BACKBONE

### UCF の例

```
NET clk0 CLOCK_DEDICATED_ROUTE = BACKBONE;
```

### XDC の例

```
set_property CLOCK_DEDICATED_ROUTE BACKBONE [get_nets clk0]
```

## BACKBONE(2)

### 適用箇所

ピン

### 制約値

BACKBONE

### UCF の例

```
PIN clkbuf0.0 CLOCK_DEDICATED_ROUTE = BACKBONE;
```

### XDC の例

```
set_property CLOCK_DEDICATED_ROUTE BACKBONE [get_pins clkbuf0/0]
```

## I/O 関連の制約

### HIODELAY\_GROUP

#### 適用箇所

IDELAY および IDELAYCTRL セル

#### 制約値

文字列

#### UCF の例

```
INST idelay0 HIODELAY_GROUP = group0;
```

#### XDC の例

```
set_property HIODELAY_GROUP group0 [get_cells idelay0]
```

### IODELAY\_GROUP

#### 適用箇所

IDELAY および IDELAYCTRL セル

#### 制約値

文字列

#### UCF の例

```
INST idelay0 IODELAY_GROUP = group0;
```

#### XDC の例

```
set_property IODELAY_GROUP group0 [get_cells idelay0]
```

## DCI\_VALUE

### 適用箇所

I/O バッファード セル

### 制約値

整数値。抵抗値をオームで指定します。

### UCF の例

```
INST a_IBUF[0]_inst DCI_VALUE = 75;
```

### XDC の例

```
set_property DCI_VALUE 75 [get_cells {a_IBUF[0]_inst}]
```

## DIFF\_TERM

### 適用箇所

I/O バッファード セル

### 制約値

ブール値

### UCF の例

```
INST a_IBUF[0]_inst DIFF_TERM = TRUE;
```

### XDC の例

```
set_property DIFF_TERM true [get_cells {a_IBUF[0]_inst}]
```

## DRIVE

### 適用箇所

入出力および出力バッファード セル

### 制約値

整数値: 2、4、6、8、12、16、24

### UCF の例

```
INST q_OBUF[0]_inst DRIVE = 24;
```

### XDC の例

```
set_property DRIVE 24 [get_ports q[0]]
```

LVTTL で 24 は有効な値です。

## IOSTANDARD

### 適用箇所

I/O バッファードセル

### 制約値

I/O 規格の文字列

### UCF の例

```
INST q_OBUF[0]_inst IOSTANDARD = LVCMOS25;
```

### XDC の例

```
set_property IOSTANDARD LVCMOS25 [get_ports q[0]]
```

詳細は、『制約ガイド』(UG625) を参照してください。

## SLEW

### 適用箇所

入出力および出力バッファードセル

### 制約値

SLOW または FAST

### UCF の例

```
INST q_OBUF[0]_inst SLEW = FAST;
```

### XDC の例

```
set_property SLEW FAST [get_ports q[0]]
```

## FAST

### 適用箇所

入出力および出力バッファードセル

### 制約値

なし

### UCF の例

```
INST q_OBUF[0]_inst FAST;
```

### XDC の例

```
set_property SLEW FAST [get_ports q[0]]
```

## SLOW

### 適用箇所

入出力および出力バッファードセル

### 制約値

なし

### UCF の例

```
INST q_OBUF[0]_inst SLOW;
```

### XDC の例

```
set_property SLEW SLOW [get_ports q[0]]
```

## IN\_TERM

### 適用箇所

ポート

### 制約値

- 。 NONE
- 。 UNTUNED\_SPLIT\_40
- 。 UNTUNED\_SPLIT\_50
- 。 UNTUNED\_SPLIT\_60

### UCF の例

```
NET a[0] IN_TERM = UNTUNED_SPLIT_50;
```

### XDC の例

```
set_property IN_TERM UNTUNED_SPLIT_50 [get_ports [list clk]]
```

## OUT\_TERM

### 適用箇所

ポート

### 制約値

- NONE
- UNTUNED\_25
- UNTUNED\_50
- UNTUNED\_75

### UCF の例

```
net q[0] OUT_TERM = UNTUNED_50;
```

### XDC の例

```
set_property OUT_TERM UNTUNED_50 [get_ports q[0]]
```

## IOBDELAY

### NONE

#### 適用箇所

ポート ネット

#### 制約値

NONE

#### UCF の例

```
net b[0] IOBDELAY = NONE;
```

#### XDC の例

```
set_property IOBDELAY NONE [get_nets b[0]]
```

注記 : IOBDELAY はポートに設定することはできませんが、入力バッファなどのセルには設定できます。

### BOTH

#### 適用箇所

ポート ネット

#### 制約値

BOTH

#### UCF の例

```
net b[0] IOBDELAY = BOTH;
```

#### XDC の例

```
set_property IOBDELAY BOTH [get_nets b[0]]
```

注記 : IOBDELAY はポートに設定することはできませんが、入力バッファなどのセルには設定できます。

## IBUF

### 適用箇所

ポート ネット

### 制約値

IBUF

### UCF の例

```
net b[0] IOBDELAY = IBUF;
```

### XDC の例

```
set_property IOBDELAY IBUF [get_nets b[0]]
```

注記 : IOBDELAY はポートに設定することはできませんが、入力バッファなどのセルには設定できます。

## IFD

### 適用箇所

ポート ネット

### 制約値

IFD

### UCF の例

```
net b[0] IOBDELAY = IFD;
```

### XDC の例

```
set_property IOBDELAY IFD [get_nets b[0]]
```

注記 : IOBDELAY はポートに設定することはできませんが、入力バッファなどのセルには設定できます。

## KEEPER

### 適用箇所

ポート ネット

### 制約値

- 。 TRUE
- 。 FALSE
- 。 YES
- 。 NO

### UCF の例

```
NET n1 KEEPER = TRUE;
```

### XDC の例

```
set_property KEEPER true [get_ports n1]
```

## PULLDOWN

### 適用箇所

ポート ネット

### 制約値

- TRUE
- FALSE
- YES
- NO

### UCF の例

```
NET n1 PULLDOWN = TRUE;
```

### XDC の例

```
set_property PULLDOWN true [get_ports n1]
```

## PULLUP

### 適用箇所

ポート ネット

### 制約値

- TRUE
- FALSE
- YES
- NO

### UCF の例

```
NET n1 PULLUP = TRUE;
```

### XDC の例

```
set_property PULLUP true [get_ports n1]
```

## VCCAUX\_IO

### 適用箇所

ポート



### 制約値

- NORMAL
- HIGH
- DONTCARE

### UCF の例

```
NET d[0] VCCAUX_IO = HIGH;
```

### XDC の例

```
set_property VCCAUX_IO HIGH [get_ports d[0]]
```

## その他のネット関連の制約

### KEEP

#### 適用箇所

ネット

### 制約値

- 。 TRUE
- 。 FALSE

### UCF の例

```
net x_int KEEP = TRUE;
```

### XDC の例

```
set_property DONT_TOUCH true [get_nets x_int]
```

## SAVE NET FLAG

### 適用箇所

ネット

### 制約値

なし

### UCF の例

```
net x_int S;
```

### XDC の例

```
set_property DONT_TOUCH true [get_nets x_int]
```

## KEEP\_HIERARCHY

### 適用箇所

セル

### 制約値

- 。 TRUE
- 。 FALSE
- 。 YES
- 。 NO

### UCF の例

```
INST u1 KEEP_HIERARCHY = TRUE;
```

### XDC の例

```
set_property DONT_TOUCH true [get_cells u1]
```

## LOCK\_PINS

### 適用箇所

LUT セル

### 制約値

CSV 文字列:

I[0-5]:A[6-1]

### UCF の例

```
INST LUT1 LOCK_PINS = I3:A6, I2:A5;
```

### XDC の例

```
set_property LOCK_PINS {I3:A6 I2:A5} [get_cells LUT1]
```

## ROUTE

### 適用箇所

ネット

### 制約値

指定配線文字列 (DIRT)

### UCF の例

```
NET n85 ROUTE={2;1;-4!-1;-53320; ...16;-8!};
```

### XDC の例

```
set_property FIXED_ROUTE {EE2BEG0 NR1BEG0 CLBLL_LL_AX} [get_nets n85]
```

注記: ISE Design Suite の指定配線文字列と Vivado Design Suite のネット配線プロパティは互換性がありません。Vivado では、一意のエンコードされていないフォーマットが使用されます。

## コンフィギュレーション関連の制約

## CONFIG PROHIBIT

### ピン サイト

### 適用箇所

サイト

### 制約値

ピン サイト

### UCF の例

```
CONFIG PROHIBIT = K24, K26, K27, K28;
```

### XDC の例

```
set_property PROHIBIT true [get_sites {K24 K26 K27 K28}]
```

## バンク番号

### 適用箇所

サイト

### 制約値

バンク番号

### UCF の例

```
CONFIG PROHIBIT = BANK34, BANK35, BANK36;
```

### XDC の例

```
set_property PROHIBIT true [get_sites -of [get_iobanks 34 35 36]]
```

## RAM(1)

### 適用箇所

サイト

### 制約値

RAM

### UCF の例

```
CONFIG PROHIBIT = RAMB18_X0Y0;
```

### XDC の例

```
set_property PROHIBIT true [get_sites RAMB18_X0Y0]
```

## RAM(2)

### 適用箇所

サイト

### 制約値

RAM

### UCF の例

```
CONFIG PROHIBIT = RAMB18_X0Y1, RAMB18_X0Y3, RAMB18_X0Y5;
```

### XDC の例

```
set_property PROHIBIT true [get_sites {RAMB18_X0Y1 RAMB18_X0Y3 RAMB18_X0Y5}]
```

注記: 上記では、カンマで区切られた RAM サイトを使用していますが、サポートされているどのサイト タイプでも使用できます。

### RAM(3)

#### 適用箇所

サイト

#### 制約値

RAM

### UCF の例

```
CONFIG PROHIBIT = RAMB36_X1Y1:RAMB36_X2Y2;
```

### XDC の例

```
set_property PROHIBIT true [get_sites -range {RAMB36_X1Y1 RAMB36_X2Y2}]
```

### RAM(4)

#### 適用箇所

サイト

#### 制約値

RAM

### UCF の例

```
CONFIG PROHIBIT = RAMB36_X3Y*;
```

### XDC の例

```
set_property PROHIBIT true [get_sites RAMB36_X3Y*]
```

### DSP48

#### 適用箇所

サイト

#### 制約値

DSP48

### UCF の例

```
CONFIG PROHIBIT = DSP48_X0Y*;
```

## XDC の例

```
set_property PROHIBIT true [get_sites DSP48_X0Y*]
```

## SLICE

### 適用箇所

サイト

### 制約値

スライス

## UCF の例

```
CONFIG PROHIBIT = SLICE_X0Y0:SLICE_X47Y49;
```

## XDC の例

```
set_property PROHIBIT true [get_sites -range {SLICE_X0Y0 SLICE_X47Y49}]
```

## ILOGIC

### 適用箇所

サイト

### 制約値

ILOGIC

## UCF の例

```
CONFIG PROHIBIT = ILOGIC_X0Y0:ILOGIC_X0Y49;
```

## XDC の例

```
set_property PROHIBIT true [get_sites -range {ILOGIC_X0Y0 ILOGIC_X0Y49}]
```

## OLOGIC

### 適用箇所

サイト

### 制約値

OLOGIC

## UCF の例

```
CONFIG PROHIBIT = OLOGIC_X0Y0:OLOGIC_X0Y49;
```

## XDC の例

```
set_property PROHIBIT true [get_sites -range {OLOGIC_X0Y0 OLOGIC_X0Y49}]
```

## BUFGCTRL

### 適用箇所

サイト

### 制約値

BUFGCTRL

### UCF の例

```
CONFIG PROHIBIT = BUFGCTRL_X0Y0:BUFGCTRL_X0Y15;
```

### XDC の例

```
set_property PROHIBIT true [get_sites -range {BUFGCTRL_X0Y0 BUFGCTRL_X0Y15}]
```

## BUFR

### 適用箇所

サイト

### 制約値

BUFR

### UCF の例

```
CONFIG PROHIBIT = BUFR_X0Y0:BUFR_X0Y3;
```

### XDC の例

```
set_property PROHIBIT true [get_sites -range {BUFR_X0Y0 BUFR_X0Y3}]
```

## BUFIO

### 適用箇所

サイト

### 制約値

BUFIO

### UCF の例

```
CONFIG PROHIBIT = BUFIO_X0Y0:BUFIO_X0Y3;
```

### XDC の例

```
set_property PROHIBIT true [get_sites -range {BUFIO_X0Y0 BUFIO_X0Y3}]
```

## BUFHCE

### 適用箇所

サイト

### 制約値

BUFHCE

### UCF の例

```
CONFIG PROHIBIT = BUFHCE_X0Y0:BUFHCE_X1Y11;
```

### XDC の例

```
set_property PROHIBIT true [get_sites -range {BUFHCE_X0Y0 BUFHCE_X1Y11}]
```

## CONFIG INTERNAL\_VREF\_BANK

### 電圧

### 適用箇所

I/O バンク

### 制約値

電圧

### UCF の例

```
CONFIG INTERNAL_VREF_BANK14 = 0.75;
```

### XDC の例

```
set_property INTERNAL_VREF 0.75 [get_iobanks 14]
```

## NONE

### 適用箇所

I/O バンク

### 制約値

NONE

### UCF の例

```
CONFIG INTERNAL_VREF_BANK0 = NONE;
```

### XDC の例

```
reset_property INTERNAL_VREF [get_iobanks 0]
```



## CONFIG DCI\_CASCADE

### 適用箇所

I/O バンク

### 制約値

バンクの順序

### UCF の例

```
CONFIG DCI_CASCADE = 17 15 14;
```

### XDC の例

```
set_property DCI_CASCADE {15 14} [get_iobanks 17]
```

## CONFIG CONFIG\_MODE

### M\_SERIAL

### 適用箇所

グローバル

### 制約値

M\_SERIAL

### UCF の例

```
CONFIG CONFIG_MODE = M_SERIAL;
```

### XDC の例

```
set_property CONFIG_MODE M_SERIAL [current_design]
```

### S\_SERIAL

### 適用箇所

グローバル

### 制約値

S\_SERIAL

### UCF の例

```
CONFIG CONFIG_MODE = S_SERIAL;
```

### XDC の例

```
set_property CONFIG_MODE S_SERIAL [current_design]
```

## B\_SCAN

### 適用箇所

グローバル

### 制約値

B\_SCAN

### UCF の例

```
CONFIG CONFIG_MODE = B_SCAN;
```

### XDC の例

```
set_property CONFIG_MODE B_SCAN [current_design]
```

## B\_SCAN+READBACK

### 適用箇所

グローバル

### 制約値

B\_SCAN+READBACK

### UCF の例

```
CONFIG CONFIG_MODE = B_SCAN+READBACK;
```

### XDC の例

```
set_property CONFIG_MODE B_SCAN+READBACK [current_design]
```

## M\_SELECTMAP

### 適用箇所

グローバル

### 制約値

M\_SELECTMAP

### UCF の例

```
CONFIG CONFIG_MODE = M_SELECTMAP;
```

### XDC の例

```
set_property CONFIG_MODE M_SELECTMAP [current_design]
```

## M\_SELECTMAP+READBACK

### 適用箇所

グローバル

### 制約値

M\_SELECTMAP+READBACK

### UCF の例

```
CONFIG CONFIG_MODE = M_SELECTMAP+READBACK;
```

### XDC の例

```
set_property CONFIG_MODE M_SELECTMAP+READBACK [current_design]
```

## S\_SELECTMAP

### 適用箇所

グローバル

### 制約値

S\_SELECTMAP

### UCF の例

```
CONFIG CONFIG_MODE = S_SELECTMAP;
```

### XDC の例

```
set_property CONFIG_MODE S_SELECTMAP [current_design]
```

## S\_SELECTMAP+READBACK

### 適用箇所

グローバル

### 制約値

S\_SELECTMAP+READBACK

### UCF の例

```
CONFIG CONFIG_MODE = S_SELECTMAP+READBACK;
```

### XDC の例

```
set_property CONFIG_MODE S_SELECTMAP+READBACK [current_design]
```

## S\_SELECTMAP16

### 適用箇所

グローバル

### 制約値

S\_SELECTMAP16

### UCF の例

```
CONFIG CONFIG_MODE = S_SELECTMAP16;
```

### XDC の例

```
set_property CONFIG_MODE S_SELECTMAP16 [current_design]
```

## S\_SELECTMAP16+READBACK

### 適用箇所

グローバル

### 制約値

S\_SELECTMAP16+READBACK

### UCF の例

```
CONFIG CONFIG_MODE = S_SELECTMAP16+READBACK;
```

### XDC の例

```
set_property CONFIG_MODE S_SELECTMAP16+READBACK [current_design]
```

## S\_SELECTMAP32

### 適用箇所

グローバル

### 制約値

S\_SELECTMAP32

### UCF の例

```
CONFIG CONFIG_MODE = S_SELECTMAP32;
```

### XDC の例

```
set_property CONFIG_MODE S_SELECTMAP32 [current_design]
```

## S\_SELECTMAP32+READBACK

### 適用箇所

グローバル

### 制約値

S\_SELECTMAP32+READBACK

### UCF の例

```
CONFIG CONFIG_MODE = S_SELECTMAP32+READBACK;
```

### XDC の例

```
set_property CONFIG_MODE S_SELECTMAP32+READBACK [current_design]
```

## SPIx1

### 適用箇所

グローバル

### 制約値

SPIx1

### UCF の例

```
CONFIG CONFIG_MODE = SPIx1;
```

### XDC の例

```
set_property CONFIG_MODE SPIx1 [current_design]
```

## SPIx2

### 適用箇所

グローバル

### 制約値

SPIx2

### UCF の例

```
CONFIG CONFIG_MODE = SPIx2;
```

### XDC の例

```
set_property CONFIG_MODE SPIx2 [current_design]
```

## SPIx4

### 適用箇所

グローバル

### 制約値

SPIx4

### UCF の例

```
CONFIG CONFIG_MODE = SPIx4;
```

### XDC の例

```
set_property CONFIG_MODE SPIx4 [current_design]
```

## BPI8

### 適用箇所

グローバル

### 制約値

BPI8

### UCF の例

```
CONFIG CONFIG_MODE = BPI8 ;
```

### XDC の例

```
set_property CONFIG_MODE BPI8 [current_design]
```

## BPI16

### 適用箇所

グローバル

### 制約値

BPI16

### UCF の例

```
CONFIG CONFIG_MODE = BPI16;
```

### XDC の例

```
set_property CONFIG_MODE BPI16 [current_design]
```

## CONFIG POST\_CRC

### ENABLE

#### 適用箇所

グローバル

#### 制約値

ENABLE

#### UCF の例

```
CONFIG POST_CRC = ENABLE;
```

#### XDC の例

```
set_property POST_CRC ENABLE [current_design]
```

### DISABLE

#### 適用箇所

グローバル

#### 制約値

DISABLE

#### UCF の例

```
CONFIG POST_CRC = DISABLE;
```

#### XDC の例

```
set_property POST_CRC DISABLE [current_design]
```

## CONFIG POST\_CRC\_ACTION

### HALT

#### 適用箇所

グローバル

#### 制約値

HALT

#### UCF の例

```
CONFIG POST_CRC_ACTION = HALT;
```

#### XDC の例

```
set_property POST_CRC_ACTION HALT [current_design]
```

## CONTINUE

### 適用箇所

グローバル

### 制約値

CONTINUE

### UCF の例

```
CONFIG POST_CRC_ACTION = CONTINUE;
```

### XDC の例

```
set_property POST_CRC_ACTION CONTINUE [current_design]
```

## CORRECT\_AND\_CONTINUE

### 適用箇所

グローバル

### 制約値

CORRECT\_AND\_CONTINUE

### UCF の例

```
CONFIG POST_CRC_ACTION = CORRECT_AND_CONTINUE;
```

### XDC の例

```
set_property POST_CRC_ACTION CORRECT_AND_CONTINUE [current_design]
```

## CORRECT\_AND\_HALT

### 適用箇所

グローバル

### 制約値

CORRECT\_AND\_HALT

### UCF の例

```
CONFIG POST_CRC_ACTION = CORRECT_AND_HALT;
```

### XDC の例

```
set_property POST_CRC_ACTION correct_and_halt [current_design]
```



## CONFIG POST\_CRC\_FREQ

### 適用箇所

グローバル

### 制約値

整数値: 周波数を MHz で指定

### UCF の例

```
CONFIG POST_CRC_FREQ = 50;
```

### XDC の例

```
set_property POST_CRC_FREQ 50 [current_design]
```

## CONFIG POST\_CRC\_INIT\_FLAG

### ENABLE

### 適用箇所

グローバル

### 制約値

ENABLE

### UCF の例

```
CONFIG POST_CRC_INIT_FLAG = ENABLE;
```

### XDC の例

```
set_property POST_CRC_INIT_FLAG ENABLE [current_design]
```

### DISABLE

### 適用箇所

グローバル

### 制約値

DISABLE

### UCF の例

```
CONFIG POST_CRC_INIT_FLAG = DISABLE;
```

### XDC の例

```
set_property POST_CRC_INIT_FLAG DISABLE [current_design]
```

## CONFIG POST\_CRC\_SOURCE

### FIRST\_READBACK

#### 適用箇所

グローバル

#### 制約値

FIRST\_READBACK

#### UCF の例

```
CONFIG POST_CRC_SOURCE = FIRST_READBACK;
```

#### XDC の例

```
set_property POST_CRC_SOURCE FIRST_READBACK [current_design]
```

### PRE\_COMPUTED

#### 適用箇所

グローバル

#### 制約値

PRE\_COMPUTED

#### UCF の例

```
CONFIG POST_CRC_SOURCE = PRE_COMPUTED;
```

#### XDC の例

```
set_property POST_CRC_SOURCE PRE_COMPUTED [current_design]
```

## CONFIG VCCOSENSEMODE $n$

#### 適用箇所

I/O バンク

#### 制約値

OFF、ALWAYSACTIVE、FREEZE

#### UCF の例

```
CONFIG VCCOSENSEMODE15 = ALWAYSACTIVE;
```

#### XDC の例

```
set_property VCCOSENSEMODE ALWAYSACTIVE [get_iobanks 15]
```

## CONFIG VREF

### 適用箇所

グローバル

### 制約値

ピン サイト

### UCF の例

```
CONFIG VREF = E11, F11;
```

### XDC の例

```
set_property VREF {E11 F11} [current_design]
```

## DEFAULT FLOAT

### 適用箇所

グローバル

### 制約値

ブール値

### UCF の例

```
DEFAULT FLOAT = TRUE;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

注記 : DEFAULT はサポートされていません。I/O ポートを個別にコンフィギュレーションする必要があります。

## DEFAULT KEEPER

### 適用箇所

グローバル

### 制約値

ブール値

### UCF の例

```
DEFAULT KEEPER = TRUE;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

注記 : DEFAULT はサポートされていません。I/O ポートを個別にコンフィギュレーションする必要があります。

## DEFAULT PULLDOWN

### 適用箇所

グローバル

### 制約値

ブール値

### UCF の例

```
DEFAULT PULLDOWN = TRUE;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

注記 : DEFAULT はサポートされていません。I/O ポートを個別にコンフィギュレーションする必要があります。

## DEFAULT PULLUP

### 適用箇所

グローバル

### 制約値

ブール値

### UCF の例

```
DEFAULT PULLUP = TRUE;
```

### XDC の例

Vivado Design Suite では XDC でこの制約はサポートされていません。

注記 : DEFAULT はサポートされていません。I/O ポートを個別にコンフィギュレーションする必要があります。

# CORE Generator IP を使用したデザインの Vivado Design Suite への移行

---

## 概要

Vivado™ Design Suite で、CORE Generator™ IP を使用したデザインを移行させることができます。Vivado Design Suite では次のソースからの IP を再利用できます。

- CORE Generator IP を使用した ISE Design Suite プロジェクト
- CORE Generator IP を使用した PlanAhead プロジェクト
- CORE Generator プロジェクトからの IP



**重要 :** Vivado Design Suite にデザインを移行する前に、CORE Generator IP カタログの最新版 IP がデザインに使用されていることを確認してください。

---

## CORE Generator IP を Vivado Design Suite に移行

CORE Generator IP は Vivado Design Suite に次の 2 ステップで移行できます。

1. CORE Generator IP を使用したデザインを移行
2. IP を最新版に移行

## ステップ 1 : CORE Generator IP ソースを使用したデザインを移行

Vivado Design Suite に IP を含むプロジェクトを次のいずれかの方法で移行することができます。

- 。 ISE Design Suite プロジェクトを Vivado Design Suite プロジェクトにインポートします (「[Project Navigator プロジェクトのインポート](#)」を参照)。
- 。 PlanAhead ツール プロジェクトを Vivado Design Suite プロジェクトに変換します (「[PlanAhead ツール プロジェクトの変換](#)」を参照)。
- 。 CORE Generator プロジェクトからの IP コア ソース ファイルを Vivado Design Suite プロジェクトに追加します。

## ステップ 2 : IP を最新版に移行

デザインで最新版の IP を使用する必要もあります。IP を移行するには、現在の IP をアップデートするか、IP を再カスタマイズします。



**重要 :** IP カタログで IP が利用できなくなっている場合は、NGC ネットリストやシミュレーション ファイルなど、既存の IP ネットリストおよびソースを Vivado 合成およびインプリメンテーション フローで再利用することができます。

### IP のアップデート

次の手順に従って既存 IP をアップデートします。

1. [Sources] ビューで [IP Sources] タブをクリックします。
2. IP コア ソースを右クリックします。
3. ポップアップ メニューで [Upgrade IP] をクリックします。

### IP の再カスタマイズ

次の手順に従って既存のカスタマイズ パラメーターに一致するように手動で IP をカスタマイズします。

1. [Sources] ビューで [IP Sources] タブをクリックします。
2. IP コア ソースを右クリックします。
3. ポップアップ メニューで [Re-customize IP] をクリックします。

適宜ダイアログ ボックスのオプションをアップデートします。

## ISE Simulator の Tcl の Vivado シミュレータの Tcl への移行

### Tcl コマンドの移行

次の表は ISE Simulator (ISim) の Tcl コマンドに対応する Vivado™ シミュレータの Tcl コマンドをリストしています。

表 5-1 : ISE Simulator (ISim) Tcl から Vivado Tcl へのマッピング

ISim Tcl	Vivado Design Suite Tcl
bp add <file_name> <line_number>	add_bp file_name linnumber
bp clear	remove_bps
bp del <index> [<index>...]	remove_bp indexlist...
bp list	report_bps
bp remove <file_name> <line_number>	remove_bps [get_bps -filter {file_name==<file_name> && line_number == <line_number>}]
describe <name>	describe name
dump	report_values
dump -p <process_scope_name>	report_values process_scope_name/*
isim condition add <condition_expression> <command> [-label <label_name>]	add_condition [-label name] <condition_expression> <command>
isim condition remove [<label_names>...] [<indexlist>...] [-all]	remove_conditions [names_indices_objects...]
isim condition list	report_conditions
isim force add <object_name> <value> [-radix <radix>] [-time <time_offset>] { [ -value <value> [-radix <radix>] -time <time_offset>] } [-cancel <time_offset>] [-repeat <time_offset>]	add_force [-radix radix] [-cancel_after <time_offset>] [-repeat_every <time_duration>] <object_name> {<value> [<time>]} [{ <value> <time>}...]
isim force remove	remove_force
isim get <property> Properties: arraydisplaylength, radix, userunit, maxtraceablesizes, ltrace, ptrace	get_property property_name [current_sim] Properties: array_display_limit, radix, time_unit, trace_limit, line_tracing, process_tracing

表 5-1 : ISE Simulator (ISim) Tcl から Vivado Tcl へのマッピング

ISim Tcl	Vivado Design Suite Tcl
isim set <property> <value> properties: arraydisplaylength, radix, userunit, maxtraceablesizes, ltrace, ptrace	set_property property_name property_value [current_sim] Properties: array_display_limit, radix, time_unit, trace_limit, line_tracing, process_tracing
onerror {tcl_commands}	onerror {tcl_commands}
put [-radix <radix>] name value	set_value [-radix radix] Design_object value
quit [-f -force] [-s -sim]	quit [-f -force]
restart	restart
resume	resume
run [all   continue   <time> <unit>]	run [-all] [time unit]
saif open [-scope <path_name>] [-file <file_name>] [-level <nesting_level>] [-allnets]	open_saif file_name; log_saif hdl_objects
saif_close	close_saif [SaifObj]
scope [<path>]	current_scope hdl_scope
sdfanno	SDF アノテーションは xelab (エラボレータ) に移動して います。このためこのコマンドは今はサポートされていません。
show time	current_time
show port	report_objects [get_objects * -filter {type == port}]
show scope	report_scope
show signal	report_objects [get_objects * -filter {type == signal}]
show variable	report_objects [get_objects * -filter {type == variable}]
show constant	report_objects [get_objects * -filter {type == constant}]
show child [-r]	report_scopes [get_scopes -r *]
show driver <hdl_object_name>	report_drivers hdl_object
show load <hdl_object_name>	report_readers hdl_object
show value [-radix <radix>] <hdl_object_name>	report_value [-radix radix] hdl_object
step	step [-over]
test [-radix radix] <hdl_object_name> <test_value>	現在はサポートされていません。「expr {[get_value -radix radix hdl_object] == test_value}」のように ビルトインされている Tcl を使用してください。
vcd dumpfile <file_name>	open_vcd file_name
vcd dumpvars -m <hdl_scope_name> [-l <level>]	log_vcd hdl_objects
vcd dumplimit <size>	limit_vcd [VCDObject] filesize
vcd dumpon	start_vcd [VCDObject]



表 5-1 : ISE Simulator (ISim) Tcl から Vivado Tcl へのマッピング

ISim Tcl	Vivado Design Suite Tcl
vcd dumpoff	stop_vcd [VCDOBJECT]
vcd dumpflush	flush_vcd [VCDOBJECT]
wave log [-r] name	log_wave hdl_objects

## その他のリソース

---

### ザイリンクスの資料

アンサー レコード、資料、ダウンロード、フォーラムなどのサポート資料は、ザイリンクスのサポート ウェブサイトをご覧ください。

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

ザイリンクスの資料で使用されている技術用語の用語集はこちらをご覧ください。

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

---

### ソリューション センター

[ザイリンクス ソリューションセンター](#)では、設計サイクルのあらゆる段階におけるデバイス、ソフトウェア ツール、およびIP コアのサポートを提供しています。デザイン アシスタント、デザイン アドバイザリ、トラブルシューティングなどが含まれます。631

---

### 参考資料

このガイドの補足としてこちらの資料を参照してください。

- Vivado Design Suite 2012.2 の資料 ([http://japan.xilinx.com/support/documentation/dt\\_vivado\\_vivado2012-2.htm](http://japan.xilinx.com/support/documentation/dt_vivado_vivado2012-2.htm))
- 『制約ガイド』(UG625)  
([http://japan.xilinx.com/support/documentation/sw\\_manuals/xilinx14\\_2/cgd.pdf](http://japan.xilinx.com/support/documentation/sw_manuals/xilinx14_2/cgd.pdf))