

UltraScale アーキテクチャへの移行

設計手法ガイド

UG1026 (v1.4) 2016 年 3 月 30 日

本資料は表記のバージョンの英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。資料によっては英語版の更新に対応していないものがあります。日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	内容
2016年3月30日	1.4	<p>次のザイリンクス UltraScale™ の情報を追加。</p> <ul style="list-style-type: none"> 5 ページの「UltraScale アーキテクチャの概要」を追加。 UltraRAM に関する注記を 6 ページの「UltraScale デバイスの選定」に追加。 UltraScale+ の情報を 17 ページの「7 シリーズ コンポーネントの高度なリターゲット」に追加。 トライステート用の SDR レジスタに関する情報を 19 ページの「ISERDESE2」に追加。 広い SDP モードに関する注記を 20 ページの「7 シリーズ RAM/ROM のリターゲット」に追加。 25 ページの表 3 : HP I/O バンクでキャリブレーションなしのソース終端をサポートする I/O 規格を追加。 HD I/O バンクに関する情報を 26 ページの「外部 VREF」に追加。 UltraScale+ デバイス固有の UNISIM プリミティブに関する情報を 26 ページの「SelectIO インターフェイス UNISIM プリミティブ」に追加。 UltraScale+ アーキテクチャのデータシートへのリンクを 29 ページの「参考資料」に追加。
2015年5月5日	1.3	<p>次のセクションの説明を更新。</p> <p>「レジスタまたはラッチでセットおよびリセットの両方を使用しないこと」 「制御信号の極性」</p> <p>「大きい MUX 構造」に関するセクションを削除。</p> <p>セクション「既存のソフト IP、EDIF、または NGC ネットリストの使用法」を改訂。 「ISERDESE2」および「OSERDESE2」に関するセクションで SerDes 比率の情報を修正。 レジスタ付きトライステート出力パスに関する推奨をセクション「7 シリーズ レジスタおよびラッチのリターゲット」に追加。</p>
2014年10月15日	1.2	<p>NGC から EDIF への変換に関する情報を 13 ページの「7 シリーズから UltraScale デバイスにターゲットを変更する場合の注意事項」に追加。</p> <p>DON'T_TOUCH を、15 ページの「合成属性および物理的制約の使用法」で規定された合成属性のリストに追加。</p> <p>18 ページの「7 シリーズ I/O コンポーネントのリターゲット」の IDELAY2 および ODELAY2 の説明を改訂。</p> <p>ザイリンクス トレーニング リソースへのリンクを追加。</p>
2014年4月10日	1.1	<p>詳細説明の追加と改訂。</p> <p>12 ページの「非同期データの同期」に説明を追加。</p> <p>12 ページの「クロッキングに関する注意事項」に説明を追加。</p> <p>13 ページの「UltraScale デバイスにおける IP の移行」</p> <p>IDELAY2 コンポーネントのリターゲットに関するセクションを追加 : 18 ページ</p> <p>23 ページの「SelectIO に関する考察事項」</p>
2013年12月11日	1.01	<p>付録 A 「その他のリソースおよび法的通知」の DATA_BUF_ADDR_WIDTH に関する説明およびオプションを更新。</p>
2013年12月10日	1.0	初版

目次

改訂履歴	2
UltraScale デバイスへのデザイン移行	
UltraScale アーキテクチャの概要	5
UltraScale FPGA ファミリー間の機能の違い	6
UltraScale デバイスをターゲットにする場合の注意事項	6
UltraScale デバイスの選定	6
制御信号の使用法	7
DSP およびその他の演算中心のコードの使用法	11
RAM に関する注意事項	11
FIFO の使用	11
XOR ファンクションの使用 (CRC/誤り検出/誤り訂正)	12
非同期データの同期	12
バックスルーとしての LUT 利用	12
クロッキングに関する注意事項	12
7 シリーズから UltraScale デバイスにターゲットを変更する場合の注意事項	13
既存のソフト IP、EDIF、または NGC ネットリストの使用法	13
UltraScale デバイスにおける IP の移行	13
デザイン ツールに関する注意事項	15
デザインのベースライン制約の作成	15
7 シリーズデバイスのプリミティブのリターゲットに関する注意事項	16
7 シリーズ コンポーネントの高度なリターゲット	17
7 シリーズ コンフィギュレーション/BSCAN コンポーネントのリターゲット	17
7 シリーズ I/O コンポーネントのリターゲット	18
7 シリーズ RAM/ROM のリターゲット	20
7 シリーズ クロック コンポーネントのリターゲット	21
7 シリーズ レジスタおよびラッチのリターゲット	22
スライス/CLB プリミティブ	22
SelectIO に関する考察事項	23
DCI に関する注意事項	23
T _{DCI} 規格	24
キャリブレーションなしの入力終端	25
未調整ソース終端	25
VREF ベースの入力規格	26
外部 VREF	26
SelectIO インターフェイス UNISIM プリミティブ	26
V _{CCAUX_IO}	27
SLEW 属性	27
差動入力終端	27
新規および拡張機能	27
新しいプリミティブに関する注意事項	27
スライス/CLB プリミティブ	27

付録 A: その他のリソースおよび法的通知

ザイリンクス リソース	29
ソリューション センター	29
参考資料	29
トレーニング リソース	30
お読みください: 重要な法的通知	30

UltraScale デバイスへのデザイン移行

UltraScale アーキテクチャの概要

ザイリックス UltraScale™ アーキテクチャは、ASIC クラス初の All Programmable アーキテクチャであり、スマート処理で毎秒数百ギガビットのシステム性能を実現すると共に、チップ上でデータを効率的に配線して処理します。UltraScale アーキテクチャ デバイスは、次世代配線、ASIC 方式のクロッキング、3D-on-3D IC、マルチプロセッサ SoC (MPSoC) テクノロジー、新しい消費電力削減機能など、業界最先端をいく革新的な技術によって高帯域幅、高使用率の幅広いシステム要件に対応します。これらのデバイスは多数の構築ブロックが共通となっているため、異なるプロセス ノード間や製品ファミリ間での拡張性に優れ、複数のプラットフォームに渡るシステム レベルでの投資を可能にします。

Virtex® UltraScale+™ デバイスは、最も広いシリアル I/O 帯域幅と信号処理帯域幅、最高のオンチップ メモリ集積度など、FinFET ノードで最高の性能と集積性能を実現します。業界で最高性能を誇る FPGA ファミリの Virtex UltraScale+ デバイスは、1Tb/s を超えるネットワークやデータ センターから、完全統合型レーダー / 早期警告システムにいたるまで、広範なアプリケーションに最適です。

Virtex UltraScale デバイスは、シリアル I/O 帯域幅およびロジック容量などにおいて、20nm で最高の性能と統合性を実現します。20nm プロセス ノードで業界唯一のハイエンド FPGA となるこのデバイスは、400G ネットワークから大規模 ASIC のプロトタイピングやエミュレーションなどのアプリケーションに最適です。

Kintex® UltraScale+ デバイスは、FinFET ノードで最高の価格/性能/消費電力バランスを備えており、トランシーバーやメモリ インターフェイスのライン レート、ならびに 100G コネクティビティ コアなど、ハイエンド性能に対応する最も費用対効果の高いソリューションを実現します。この最新のミッドレンジファミリは、パケット処理と DSP を多用する機能に最適であると同時に、ワイヤレス MIMO 技術、Nx100G ネットワーク、データ センターなど広範なアプリケーションにも対応します。

Kintex UltraScale デバイスは 20nm プロセスで最高の価格/性能/消費電力バランスを備え、ミッドレンジ デバイスでは最大の信号処理帯域幅、次世代トランシーバー、および低コスト パッケージなどの特長があり、性能と費用対効果の最適な調和を図っています。このファミリは、100G ネットワークやデータ センター アプリケーションでのパケット処理だけでなく、次世代の医療用画像処理、8k4k ビデオ、ヘテロジニアスなワイヤレス インフラなどで必要とされる DSP 性能を重視するアプリケーションにも最適です。

Zynq® UltraScale+ MPSoC デバイスは、64 ビットのプロセッサ スケーラビリティを備え、グラフィックス、ビデオ、波形、パケット処理にソフトおよびハード エンジンの連携によるリアルタイム制御を提供します。高度な解析が可能な ARM® ベースのシステムとタスクのアクセラレーションが可能なオンチップ プログラマブル ロジックが統合されているため、5G ワイヤレス、次世代 ADAS、インダストリアル IoT など広範なアプリケーションにおいて無限の可能性を引き出すことができます。

UltraScale FPGA ファミリの機能の違い

この資料の説明および例は、Kintex® UltraScale™ および Virtex® UltraScale™ ファミ리를ベースにしています。ここでは、Kintex UltraScale+、Virtex UltraScale+、および Zynq UltraScale+ ファミリにおけるいくつかの違いを示します。

- UltraRAM のサポート
- Zynq® UltraScale+™ MPSoC および Kintex® UltraScale+™ ファミリでは、対応するロジック リソースを備える HD (High-Density) I/O
- MIPI D-PHY トランスミッターおよびレシーバー機能は、Virtex UltraScale+ デバイス、Kintex UltraScale+ デバイス、および Zynq UltraScale+ MPSoC に固有の HP I/O でサポートされています。
- システム モニターの機能拡張

HD I/O バンクでサポートされている機能の詳細は、『UltraScale アーキテクチャ SelectIO リソース ユーザー ガイド』(UG571) [参照 1] を参照してください。

UltraScale デバイスをターゲットにする場合の注意事項

このセクションでは、特定のテクノロジーに依存しない、プログラマブル ロジック デザインのインプリメンテーションについて解説します。既存のデザイン ソースを評価して UltraScale™ デバイスへの移行を検討する場合や UltraScale デバイス用に新規コードを開発する場合の参考としてください。



推奨: 新規デザインを開始する前、あるいはデザインをリターゲットする前に、ピン配置の選択、デザインの作成、インプリメンテーション、およびデバックに最適な設計手法を選択する際の参考として、『UltraFast 設計手法ガイド (Vivado Design Suite 用)』(UG949) [参照 2] を参照してください。

UltraScale デバイスの選定

UltraScale デバイスおよびその他のアーキテクチャでは、サイズの異なるデバイスにおける汎用ロジック リソースの量を定義する手段として、ロジック セル数があります。UltraScale デバイスでのロジック セル数はその他のデバイスよりも非常に大きくなります。

使用可能なプログラマブル LUT リソースを把握するにあたり、ロジック セル数はよい指標ですが、デバイスのロジック容量全体を示す指標としては必ずしも最善ではない場合があります。以前の FPGA やその他のテクノロジーでは LUT、フリップフロップ、またはほかのリソースを使用していた機能のいくつかは、UltraScale デバイスにはハードとして搭載されています。たとえば、ハード化された Interlaken コア、PCIe® Gen3 ブロック、統合 FIFO などのブロックによって、以前のインプリメンテーションよりも LUT が大幅に節約される点は、デバイスを選定する際に考慮に入れる必要があります。

デバイス選定の基準はロジック集積度のみではなく、デザインに必要なほかのリソースについても評価が必要です。



推奨: まず、『UltraScale アーキテクチャおよび製品概要』(DS890) [参照 3] に掲載されている機能一覧表を確認し、関連するリソースすべてを比較してください。

適切な UltraScale デバイスを判断する際には、次の点も含めて検討する必要があります。

- ブロック RAM/DSP

ブロック RAM または DSP を多く使用するデザイン向けには、Kintex® UltraScale デバイスが、特定のアレイサイズにおいて、より多くのブロック RAM と DSP を搭載しています。

UltraScale+ デバイスをターゲットにする場合、128Kb を超えるメモリ アレイには UltraRAM を使用することを検討してください。UltraRAM の詳細は、『UltraScale アーキテクチャ メモリ リソース ユーザー ガイド』(UG573) [参照 4] を参照してください。

- I/O

特定アレイサイズでの I/O 数、バンク サイズ、配置、機能は 7 シリーズ デバイスとは異なります。新規デザインの開始時に、より多くの機能をデバイスに盛り込むことができないかを検討してください。これは、必要な I/O 数を抑えるだけでなく、システム全体の性能、コスト、消費電力、機能の向上につながります。

- スピード グレード

多くの場合、スピード グレードを下げたデバイスを UltraScale デバイスで選択しても同様の性能が得られます。ただし、デバイスを確定させる前に、個々のブロックのスピードおよびシステム全体の性能の解析を行ってください。可能であれば、最終的なスピード グレードの決定前に、以前のデザインをインプリメントし、検討しているスピード グレードを指定してデザインがタイミングを満たすことができるかを確認します。

制御信号の使用法

制御信号(クロック、セット、リセット、クロック イネーブルなどの同期エレメントを制御する信号)の使用法が、デバイスの集積度、使用率、性能に影響を及ぼすことがあります。UltraScale アーキテクチャは、旧アーキテクチャと比較すると、デザインで指定する制御信号の数およびタイプに関して柔軟性が改善されています。しかし、制御信号と制御セットの使用法およびデザインのインプリメンテーションに与える可能性のある影響について把握しておくのが最善です。

レジスタまたはラッチでセットおよびリセットの両方を使用しないこと

旧世代の FPGA と同様、UltraScale デバイスのフリップフロップにはロジックを追加しない限りセット信号とリセット信号の両方はインプリメントできません。したがって、同期セットおよび同期リセットの両方を使用するには、データパスに信号を追加する必要があります。これは配置、ファンアウト、タイミング次第でエリアやタイミングに影響が及ぼす可能性があります。場合によっては、ロジック レベルを増加させることなく信号が追加でき、デザインへの実質的な影響はほとんどありません。これに対して非同期のセットおよびリセットは、リソース使用率やタイミングへの影響がより顕著であるため、これらの併用は避ける必要があります。

(1) 非同期リセットおよび非同期セット信号を共に備えたレジスタ (2) 動的な値を持つ非同期制御信号を備えたレジスタの両方、またはいずれか一方をインプリメントすることはできません。これらの動作のいずれかが RTL で記述されている場合、またはこの非同期動作を可能にする FDCPE または LDCPE がインスタンス化されている場合、Vivado Design Suite でエラーが発生します。

この構文は RTL で記述するか、HDL または EDIF フォーマットで FDCPE としてインスタンス化できます。

不適切/適切なコーディング スタイル例

例 1:

不適切なコーディング スタイル

次の簡単な例は、非同期のセットとリセットを記述した Verilog コードです。このコードではリソースおよびタイミング パスが増加します。

```
always @(posedge reset, posedge set, posedge clk)
  if (reset)
    a_reg <= 1'b0;
  else if (set)
    a_reg <= 1'b1;
  else
    a_reg <= A;
```

代わりに次のコーディング スタイルを使用

適切な機能にセット信号とリセット信号が必要な場合、次に示す例のように、これらの一方または両方を同期使用に変換します。

```
always @(posedge clk)
  if (reset)
    a_reg <= 1'b0;
  else if (set)
    a_reg <= 1'b1;
  else
    a_reg <= A;
```

例 2:

不適切なコーディング スタイル

次のコード例は、動的な値を持つ非同期の制御信号を記述した VHDL コードです。このコードではリソースおよびタイミング パスが増加します。

```
process (clk, initc) begin
  if initc='1' then
    data_reg <= init_signal;
  elsif (clk'event and clk='1') then
    data_reg <= data_in;
  end if;
end process;
```

代わりに次のコーディング スタイルを使用

この場合、initc を同期にして回路を改善します。

```
process (clk) begin
  if (clk'event and clk='1') then
    if (initc='1') then
      data_reg <= init_signal;
    else
      data_reg <= data_in;
    end if;
  end if;
end process;
```


レジスタの初期化

多くのエンジニアは、推論されたレジスタの初期化を明示的に指定することで、FPGA のレジスタおよびラッチがグローバルセット/リセット (GSR) 信号を介して初期化されるという本来の形を使用します。これによって、より信頼性が高く小型の回路が実現されます。初期化によって、リセット ステートとは別のステートでのスタートアップが可能となり、初期化しない場合に比べてロジック消費量が少なくなる場合があります。たとえば、スタートアップ時には動作させる必要があっても、後続のリセットには不要となり得るステートを含むステート マシンがその一例です。初期化のもう 1 つの利点として、実際の FPGA の動作により近い正確な RTL を記述できるため、回路表現の精度が高まります。このような点から、推論されたすべてのレジスタ、SRL (シフト レジスタ ルックアップ テーブル)、RAM に対して可能な限りレジスタの初期化を使用する設計手法が最善です。

次のコード例では、レジスタ reg の値を 1 に初期化しています。

```
signal reg: std_logic := '1';
...
process (clk) begin
  if (clk'event and clk='1') then
    if (rst='1') then
      reg <= '0';
    else
      reg <= val;
    end if;
  end if;
end process;
```

上記のコード例では、初期化によって、ロジック 1 の初期状態を作成するためのセット条件の指定が不要になっています。初期状態がリセット状態と同じ場合でも、レジスタは初期化してください。これにより、リセット状態を獲得しなくてもシミュレーションのスタートアップ時のステートが FPGA の初期状態をより正確に反映したものになります。

制御信号の極性

UltraScale アーキテクチャは、7 シリーズ FPGA やそれ以前のアーキテクチャよりも改良されています。新しいアーキテクチャでは、リセットにプログラム可能な反転があり、これによってエリアやタイミングにデメリットを与えることなく、アクティブ High またはアクティブ Low のリセットが可能です。選択できる場合は、使用率に影響を与える可能性を防ぎ、単極性のリセット (すべてアクティブ High またはすべてアクティブ Low) を使用してください。ただし、両極性が混ざったリセットを使用しても、UltraScale デバイス デザインにおけるマイナスの影響は旧世代のデバイスをベースとしたデザインよりも小さくなっています。

クロック イネーブルを持つコンポーネントをコーディングまたはインスタンス化する場合には、アクティブ High を使用してください。イネーブルピンにはプログラム可能な反転がなく、アクティブ Low のイネーブルを使用すると、インプリメンテーション中にエリアおよびタイミングにマイナスの影響を与える可能性があります。また、インプリメンテーションで最善の結果を得るには、IP を作成する場合や UNISIM プリミティブをインスタンス化する場合に、イネーブル信号を反転させないでください。

ファンアウトの少ない制御信号の使用を控えること

UltraScale デバイスの CLB (コンフィギュラブル ロジック ブロック) では、以前のデバイスと比較すると、制御信号を使用する際の柔軟性が改善されています。とはいえ、デザインで固有の制御信号は、デザインの機能および動作に必要なものに限って使用するようになります。ファンアウトが小さい固有の制御信号は、レジスタ、SRL、LUT RAM など CLB の使用効率を低下させる場合があります。また、配置やタイミングにも悪影響を及ぼす可能性があります。通常、セット、リセット、クロック イネーブル信号は、デザインの機能性から積極的に必要とされる場合以外、コードでインプリメントしないようにしてください。

不要なセットまたはリセットを使用しないこと

コードに不必要なセットおよびリセットがあると、SRL、RAM (LUT RAM またはブロック RAM)、その他のロジック構造の推論を妨げる可能性があります。アーキテクチャから最大限の効率を引き出すには、デザインの機能性から積極的に必要とされる場合にのみ、セットおよびリセットをコードで記述してください。次に例を挙げます。

- レジスタはコンフィギュレーションの完了時に自動的に初期化されるため、初期化にのみ使用されるリセットは不要です。
- 長期間にわたってアイドル状態となる回路にリセットは不要です。このような回路では入力レジスタをリセットすれば、その他の回路のデータはいずれフラッシュされます。
- クロックの複数サイクル間リセットが保持される場合、内部に位置するレジスタはリセット中にフラッシュされます。したがって、すべてのレジスタにリセットを付ける必要はありません。

不要なセットまたはリセットを使用しないことで、デバイスの使用率、配置、性能、消費電力を改善できます。

DSP48E2 レジスタの乗算器または加減算器のセット

UltraScale デバイスの DSP48E1 スライスに含まれるレジスタにはリセット信号だけがあり、セット信号はありません。DSP ブロックは、乗算、加減算、コンパレータ、カウンタ、汎用ロジックなど幅広い機能を実行できます。DSP48E2 を柔軟に使用するには、このリソースにマップされるレジスタに対するセット条件は記述しないようにします。つまり、乗算器、加算器、カウンタなど DSP48E2 スライスにインプリメント可能なロジック周辺では、必要な場合を除いてセット信号 (値がロジック 1 になる信号) をコード化しないでください。

同期セット/リセットの使用法

回路の適切な動作にセットまたはリセットが必要な場合は、必ず同期リセットをコード化してください。同期セット/リセットはタイミング特性および安定性に優れ、FPGA における使用リソースが少なく使用率を改善します。

同期セット/リセットは、ロジック削減 (数 LUT)、パッキング時の制約の軽減、そして多くの場合に回路の高速化をもたらします。DSP48E2 や RAMB36E2/RAMB18E2 などのブロック内にあるレジスタには、非同期セットおよびリセットはインプリメントできません。つまり、RTL コード内に非同期のセットまたはリセットがある場合、これらのレジスタは等価機能のものとしては使用できません。同期リセットを使用すれば、CLB 内のレジスタ共有に対する柔軟性が大幅に向上します。さらに、配線への影響が少ないだけでなく、多くの場合は改善されるため、必要な配線リソースが少なくなります。その他の利点としては、デバイスの集積度、性能、消費電力が総じて改善され、配線における遅延およびキャパシタンスが削減されます。

ファンアウトの大きな信号

信号のファンアウトはできる限り小さくするのが最善です。しかし、デザインによってはファンアウトの大きなネットが存在する場合があります。通常、ファンアウトの大きな信号は手動で分割または複製せずに、1 つの信号としてコード化します。タイミングなどの理由で複製が必要になった場合は、合成かその後の物理的最適化 (phys_opt_design) の実行中に処理できます。これにより、複製を多数回実行する場合でも柔軟に処理され、性能向上を目的としたリソースの追加と消費電力の増加とのトレードオフのバランスが適切に取られます。配置やその他の要因が変化することで複製に対する要件が変化することがありますが、合成およびインプリメンテーションツールでこれを管理していれば、このような変更のために元のコードを修正し、それを再検証する必要はありません。

旧アーキテクチャでは、ファンアウトの大きな信号をグローバル リソースまたは BUFG に配置することで、よい結果が得られる場合があります。しかし、UltraScale デバイスでは、通常はこのような使用法を避けることを推奨します。グローバル リソースが使用可能であり、タイミングや配線性から有益であると判断されれば、Vivado® ツールで自動的に使用されます。したがって、BUFG をインスタンス化したり、使用したりする必要はありません。

DSP およびその他の演算中心のコードの使用法

UltraScale アーキテクチャは、多くの DSP デザインに十分適しています。このアーキテクチャを活用するには、その基盤となる機能や性能を理解し、これらのリソースのメリットを最大限引き出すデザイン入力コードを作成する必要があります。

DSP48E2 ブロックは符号付きの演算インプリメンテーションを使用します。HDL ソースでは、リソースの性能と最適に整合し、通常は最も効率的なマッピングを実現するために符号付きの値を使用してコードを記述することを推奨します。合成ツールでは、コード内で符号なしのバス値を使用しているリソースも使用できますが、符号なしから符号付きの値への変換によって、コンポーネントのビット精度が完全には保たれない可能性があります。DSP48E2 スライスの乗算器は、符号付きデータの入力ビット精度が 18 ビット × 27 ビットです。したがって、符号なしデータの精度は 17 ビット × 26 ビットになります。Verilog コードでは、コード内で宣言しない限り、データはすべて符号なしの値とみなされます。

ターゲット デザインに加算器が多数含まれる場合は、DSP48E2 スライスの前置加算器および後置加算器の使用数を増やすのが最良の手法です。たとえば FIR フィルターの場合、複数の連続した加算ファンクション (加算器ツリー) ではなく、加算器のカスケード接続によってシストリック フィルターを構築できます。対称フィルターの場合は、専用の前置加算器を使用することで、機能をより少数の LUT やフリップフロップ、さらにはより少数の DSP スライスに統合できるかを評価できます (ほとんどの場合リソースが半減する)。

加算器ツリーが必要な場合は、6 入力の LUT アーキテクチャを活用すれば、単純な 2 入力加算と同じリソース量で 3 入力加算 ($A + B + C = D$) を効率的に構築できます。このような構成を取ることで、必要に応じてキャリー ロジック リソースを削減、節約できます。多くの場合はこれらの手法を使用する必要ありません。しかし、このような性能を知っておけば、あらかじめ適切なトレードオフを念頭に置いて RTL コードを作成でき、デザインの着手時から円滑で効率的なインプリメンテーションが可能になります。

RAM に関する注意事項

推論、プリミティブのインスタンスシート、あるいは Vivado IP カタログによってブロック RAM および LUT RAM をリターゲットする場合、UltraScale アーキテクチャでブロック RAM と LUT を最大限に活用するには 2 つの方法があります。次のいずれかの方法を使用してください。

- コアが RAM の生成に使用されている場合、UltraScale デバイス用の IP を再生成する
- RAM が合成で適切に推論されるよう、コードを記述し直す

いずれの方法でも、使用率および性能に良い結果をもたらします。ただし、コードおよびシミュレーションがより理解しやすいものになり、コード移植も容易になるため、メモリは可能な限り推論してください。

FIFO の使用

デザインの多くでは、データを別のクロックドメインに転送したり、デザインの別の部分で必要になるまでデータを格納するために FIFO が必要です。このようなデザインに UltraScale デバイスを使用する場合、データの転送や格納に専用 FIFO の 1 つを活用することを検討してください。通常は統合された FIFO を使用することで、ソフトウェアで作成した FIFO の場合よりも信頼性および MTBF (平均故障間隔) が改善されると同時に、低い消費電力、少ないエリアでより高速な動作が可能になります。UltraScale アーキテクチャデバイスの FIFO は、直接インスタンスシートするか、Vivado IP カタログ FIFO Generator を用いることでデザインで使用できます。

XOR ファンクションの使用 (CRC/誤り検出/誤り訂正)

誤り検出/訂正、暗号化など複数のタイプのアルゴリズムが多入力 XOR ファンクションを活用して、回路に実装されています。前世代のデバイスでは、この XOR 回路は LUT でインプリメントされることが多く、場合によってはタイミングのボトルネックとなったり、回路の消費電力を増大させる可能性があります。UltraScale アーキテクチャの場合、DSP48E2 に多ビットの XOR ファンクションが追加されたため、パリティ、CRC、チェックサム、そしてビット数の多い XOR 演算を必要とする他のファンクションを効率的にインプリメントできるようになりました。



推奨 : XOR ファンクションを最大限に活用するには、DSP ブロックのレジスタを使用するようにファンクションに対して適切なパイプライン処理を行います。

非同期データの同期

最近のデザインでは、2 以上のクロックドメインが使用されます。クロックドメインの位相関係が不明だったり、互換性のない周波数がない場合、ドメインをまたぐデータの転送は適切に管理する必要があります。データバスについては、統合された FIFO ロジックをバス幅のデータ転送に使用することが最も安全で効率的な方法です。ただし、リセットなどクロックドメイン同期しない場合のある 1 つの信号には、適切な同期テクニックを使用することを推奨します。次のようなテクニックを使用してください。

- 1 つまたは複数の同期レジスタを使用する
- ASYNC_REG プロパティが設定されている同期レジスタを特定する。これにより、ツールは配置配線テクニックを用いて MTBF を向上でき、解析ツールでは信頼性の観点から重要なクロックの載せ替えが認識される

パックスルーとしての LUT 利用

スライス/CLB のすべての回路にアクセスするには、場合によっては LUT を単純なバッファとして構成し、データが LUT 入力を通して内部回路に入るようにする必要があります。旧アーキテクチャでは、このコンフィギュレーションはルートスルーまたはパックスルーと呼ばれました。このパックスルーは、スライス内のすべてのレジスタへのアクセスに用いられる場合が少なくありませんでしたが、レジスタを多用するデザインでは、ロジックリソースの一部を消費する可能性があります。UltraScale アーキテクチャではこれが再設計され、LUT およびレジスタが互いに独立してアクセス可能になりました。このため、パックスルーを一般的に使用することがなくなり、ロジックの使用/使用率が大きく改善されました。レジスタを多く含むデザインのインプリメンテーションも、UltraScale アーキテクチャではシンプルになっています。

クロッキングに関する注意事項

UltraScale デバイスのクロッキングアーキテクチャは、大幅に変更および改善されています。このセクションでは、7 シリーズ FPGA を使用した既存のクロッキング構造を UltraScale デバイスにリターゲットする際の概要ガイドラインを示します。クロッキングアーキテクチャの変更点の詳細は、『UltraScale アーキテクチャクロッキングリソースユーザーガイド』(UG572) [参照 5] を参照してください。



推奨 : ほとんどの場合、クロックバッファおよびクロック修正ブロック (つまり、MMCM と PLL) は、自動で UltraScale デバイスのクロックネットワークにリターゲットされますが、デザインがより UltraScale アーキテクチャにフィットするように変更を加える場合などに備えて、クロックネットワークを見直すことを推奨します。

7 シリーズ FPGA のクロックバッファは、使用目的に応じて UltraScale デバイスが備える 3 つのクロックバッファ BUFGCE、BUFGCTRL、BUFCE_DIV の 1 つにリターゲットされます。多くの場合、バッファが複数あり、重複したバッファが直列または並列に現れるようなときは、Vivado Design Suite のロジック最適化アルゴリズム (logic_opt) が、クロックリソースの効率を向上させるためにこれらのバッファを最小限まで減らし、可能であればクロックドメイン間をまたぐ信号のタイミングを向上させようとします。7 シリーズ FPGA デザインの MMCM および PLL は、UltraScale デバイスの MMCME3 にリターゲットされます。これは、機能的にはクロック修正コンポーネントのいずれにとっても上位セットになります。

ギガビット トランシーバーからのクロックの場合、UltraScale アーキテクチャには FPGA リソースへのクロック供給をサポートする新しいクロッキング構造があります。これは BUFG_GT および BUFG_GT_SYNC コンポーネントを使用してクロックを接続したり、分配したりするもので、既存の 7シリーズ FPGA デザインから自動的に移行されません。UltraScale デバイスにおけるクロッキング回路の詳細は、UltraScale アーキテクチャ GT トランシーバーのユーザー ガイドを参照してください。ザイリンクス サポート ページから入手可能な資料は、付録 A 「その他のリソースおよび法的通知」 にリストされています。

7シリーズから UltraScale デバイスにターゲットを変更する場合の注意事項

このセクションでは、7シリーズ デバイスの既存デザインを UltraScale デバイスにリターゲットする際に考察すべき、デザインのインプリメンテーションおよび制約についてさらに詳しく説明します。

既存のソフト IP、EDIF、または NGC ネットリストの使用法



推奨 : デザインに含まれる既存のソフト IP またはブラック ボックス ネットリストは、UltraScale デバイスにインプリメントする前に、再生成または再合成してください。

7シリーズ アーキテクチャをターゲットにしたネットリストの多くは、UltraScale デバイスをターゲットにした場合でもエラーなくインプリメントできます。しかし、UltraScale デバイスに最も効率よくリターゲットし、新しい機能および性能を十分にいかす設計を行うために、旧アーキテクチャをターゲットとしたネットリストまたはコアはすべて再生成することを推奨します。

UltraScale アーキテクチャでは NGC フォーマットのネットリストがサポートされていません。既存のデザインに NGC ネットリストが含まれている場合、UltraScale デバイスをターゲットにする前に、それらのネットリストを EDIF (Electronic Data Interchange Format) に変換する必要があります。ザイリンクス Tcl Store には、NGC ネットリストから EDIF への変換を支援するユーティリティがあります。このユーティリティにアクセスするには、ザイリンクス Tcl Store を開き、プロジェクト ユーティリティ (projutils) アプリケーションをインストールします (プロジェクト ユーティリティ アプリケーションがまだインストールされていない場合)。インストールされると、コマンド `convert_ngc` を使用して簡単に変換できます。このコマンドの実行に関する詳細は、`convert_ngc -help` と入力して参照してください。

UltraScale デバイスにおける IP の移行

ザイリンクス IP を含む 7シリーズ FPGA デバイスを UltraScale デバイスに移行する場合、通常は IP をアップグレードしたり、新しく UltraScale デバイス版の IP を作成するだけで済みます。一部の IP は、アップグレードされた IP を使用しても適切に動作するようにデザインに変更が必要となる場合があります。アップグレード変更の詳細は IP の変更ログを確認してください。

IP を移行する際には、次の条件を踏まえる必要があります。

- [Report IP Status] を実行したときに、IP にアップグレード パスがあり、ステータスが「IP part change」(または「IP board change」) とレポートされます。IP のバージョンに変更はありません。この場合、変更ログを確認します。UltraScale アーキテクチャにおける IP の使用に特定の変更 (ポートおよび機能の変更を含む) を見つけます。たとえば、トランシーバーを 7シリーズ デバイスから UltraScale デバイスにアップデートする場合、トランシーバーのデバッグ ポートを選択していると、これらのポートの名前および機能は変更されます。このような場合は、新しいデバイス向けの製品ガイドのトランシーバー デバイス ポートに関するセクションを参照してください。
- UltraScale アーキテクチャへのアップグレード パスがない IP があります。IP によっては別のバージョンが利用可能な場合があります。詳細は、該当 IP の製品ガイドを参照してください。



推奨 : 7シリーズ FPGA デザインを移行する場合、まず UltraScale デバイスをターゲットとしたプロジェクトを作成するか、あるいはこのデバイスをターゲットとするように修正してください。次に、使用している Vivado ツールで入手可能な最新バージョンに IP をアップグレードします。その後、次の図に示すような Report IP Status レポートを作成します。表 1 に、よく使用される IP をリストします。

このレポートには、UltraScale アーキテクチャへのアップグレードパスがあるか、あるいは IP がサポートされているかを示します。このレポートでは、アップグレードパスを含む IP の変更ログを確認できます。各変更ログにはその IP の変更が含まれ、UltraScale アーキテクチャ向けの変更が適切かどうかを判断するための具体的な情報が記載される場合があります。UltraScale デバイス特定の詳細については、関連する IP の製品ガイドを参照してください。

UltraScale デバイスに変更後の IP Status レポート

次の表には、よく使用される IP とそれぞれの移行情報を示します。

表 1 : UltraScale デバイスへの移行情報

IP	説明
Ten Gig Ethernet PCS PMA	アップグレード可 ポート変更 リセット機能変更
7 Series PCIe 2.1 コア (7 Series Virtex® XT コアを除く)	アップグレード不可 UltraScale デバイス用の PCIe 3.1 を作成
7 Series Virtex XT PCIe 3.0 コア	ポートの微変更 この資料のリリース日時点では、自動アップグレード不可
MIG	アップグレード不可 UltraScale デバイス用の MIG 5.0 を作成
Transceiver Wizard	アップグレード不可 UltraScale FPGA Transceiver Wizard を使用
FIFO	アップグレード可 ポート変更 機能変更
Aurora	アップグレード可 ポート変更

デザイン ツールに関する注意事項

UltraScale デバイスは、Vivado Design Suite でサポートされています。システム要件 (オペレーティング システムやメモリの要件など)、サポートするデバイスを含むツールに関する重要な情報については、リリース ノートを確認してください。

デザインのベースライン制約の作成



推奨: デザインをリターゲットする際は初期の段階で、『UltraFast 設計手法ガイド (Vivado Design Suite 用)』(UG949) [参照 2] の説明に従ってデザインのベースライン制約を再作成してください。

まずはシンプルなデザイン制約およびデザイン フローを使用してから、新たなターゲット用に適切な制約がデザインに適用され、最適なデザイン フローが確立されているかを確認しながら、これらを拡張していきます。

合成属性および物理的制約の使用法

以前のアーキテクチャでは結果を向上させた合成属性 (KEEP、MAX_FANOUT、DONT_TOUCH など) または物理制約 (LOC、PBLOCK、RLOC、PROHIBIT など) でも、別のデバイスをターゲットとすると最適でなかったり、結果の質を低下させる場合があります。このため、すべての合成属性を再評価するのが最善で、場合によってはこれらを HDL コードと合成 XDC ファイルから削除してください。同様に、インプリメンテーションの XDC ファイルでもすべての物理制約を再評価して、場合によっては削除してください。KEEP または MARK_DEBUG などの属性が引き続き必要であるかを確認し、必要なければ、これらは最適な結果を得る妨げとなる可能性があるため削除します。最初のインプリメンテーションおよび解析後、必要に応じて削除した属性のうち再使用するものを判断します。ただし、その前にこれら制約を使用することなく、ツールによってインプリメンテーションを実行して、所望の結果となるかを確認してください。

タイミング制約の指定

Vivado 合成およびインプリメンテーション ツールは、タイミング制約の指定に SDC (Synopsys Design Constraints) 構文を使用します。7シリーズ FPGA のデザインでも Vivado Design Suite を使用していると、大抵の場合は同じタイミング制約が修正なしで、UltraScale デバイスにも適用されます。ISE® Design Suite を使用していた場合は、UCF フォーマットで指定されたタイミング制約を XDC 構文に変換する必要があります。詳細は、『ISE から Vivado Design Suite への移行ガイド』(UG911) [参照 6] を参照してください。

UltraScale デバイス デザインへ制約を移行する際は、次の変更点に注意してください。PRIMITIVE_GROUP および PRIMITIVE_SUBGROUP プロパティの両方または一方が変更されている場合があります。デザインでこれらを使用しているときは、『UltraScale アーキテクチャ ライブラリ ガイド』(UG974) [参照 7] を参照し、UNISIM コンポーネントの PRIMITIVE_GROUP/PRIMITIVE_SUBGROUP プロパティを修正してください。これにより、制約が確実に適切なセルに適用されるようになります。

リターゲットする場合のほとんどで、変更した IP およびその他のデザイン エリアが全体的なタイミング仕様に影響を与える可能性があります。つまり、タイミング パスの名前、基準となる点、タイプのすべてあるいはいくつかが変わるため、以前のターゲットでは適切に指定されていた制約が無効となる場合が生じます。特にデザインの内部ノードを参照するタイミング パスに対しては、参照されたそのノードがあり、タイミング仕様または目標とするタイミングが許容範囲にあることを確認してください。指定したタイミング制約が新規デザインに適切で、正確なものかを確認する場合は、check_timing を使用します。

シンプルなデザイン フロー

設計が進むに従い、別のオプションや指示子、インプリメンテーション手順 (例: `power_opt_design`、`phys_opt_design`)、あるいはより高度な制御 (TCL 前処理、または後処理) の使用でデザインフローが増補されることが少なくありません。あるアーキテクチャをターゲットとしたデザインに対して、そのようなフローを作り上げることは多くの場合、結果の向上につながります。ただし、デザインおよびターゲット アーキテクチャの一方または両方が変わると、別の手法が必要となる場合があります。したがって、UltraScale デバイスをターゲットとする場合に、7シリーズ デバイスと同じオプションおよびフローを使用することが常に最適なストラテジとは限りません。



推奨: デザインおよび制約の再検証後、Vivado ツールでデフォルトのフローを実行してください。デザインまたは制約を変更することで、よりよい結果を導くことができないかを解析し、それが完了した後に、異なるツールオプション、指示子、あるいはフローの使用を検討します。

7シリーズ デバイスのプリミティブのリターゲットに関する注意事項

次に列挙するプリミティブは、7シリーズと UltraScale デバイスで同じです。したがって、動作、接続、一般的な使用法に変更は不要です。

AND2B1L	FDCE	LUT3	RAM128X1D
BUFG	FDPE	LUT4	RAM256X1S
BUFGCE	FDRE	LUT5	RAM32M
BUFGCTRL	FDSE	LUT6	RAM32X1S
BUFGMUX	KEEPER	MUXF7	RAM64M
BUFGMUX_1	LDCE	MUXF8	RAM64X1D
BUFGMUX_CTRL	LDPE	OR2L	RAM64X1S
CFGLUT5	LUT1	PULLDOWN	SRL16E
EFUSE_USR	LUT2	PULLUP	SRLC32E

7シリーズ アーキテクチャ向けにインスタンス化されたデバイス プリミティブの中には、UltraScale アーキテクチャに自動的にリターゲットされないものや、ターゲットされたとしても注意すべき変更を伴うものがあります。UNISIM コンポーネントの詳細は、『UltraScale アーキテクチャ ライブラリ ガイド』(UG974) [参照 7] を参照してください。上記の各ブロックを UltraScale デバイスにリターゲットする場合の詳細情報は、それぞれのユーザー ガイドに記載されています。

次に、以前のコンポーネントの一部について説明します。

7シリーズ コンポーネントの高度なリターゲット

GTPE2_CHANNEL、GTPE2_COMMON、GTHE2_CHANNEL、GTHE2_COMMON、GTXE2_CHANNEL、GTXE2_COMMON、IBUFDS_GTE2

ギガビット トランシーバーのプリミティブおよびこれらに関連するバッファや一般的な回路は UltraScale および UltraScale+ デバイスではサポートされていません。UltraScale および UltraScale+ デバイスではシリアル トランシーバー回路の基本的な機能が変更されたため、これらはリターゲットされません。これらの機能は、UltraScale デバイスでは トランシーバー コンポーネントの E3 シリーズで、UltraScale+ デバイスでは トランシーバー コンポーネントの E4 シリーズで置き換えられました。



重要 : デザインにこれらの旧コンポーネントが含まれる場合、適切な UltraScale デバイスをターゲットとし、Vivado IP カタログから生成した同等のインターフェイスで置き換えてください。

XADC

UltraScale デバイスの SYSMONE1 および UltraScale+ デバイスの SYSMONE4 の仕様の一部は、7シリーズ デバイスの XADC コンポーネントとは異なります。インスタンス化された XADC をデザインで使用する場合、『UltraScale アーキテクチャ システム モニター ユーザー ガイド』(UG580) [参照 8] を参照し、この変更が特定のアプリケーションにおける XADC コンポーネントの使用に影響を与えるかを確認してください。



推奨 : SYSMONE1 または SYSMONE4 を使用する場合、Vivado Design Suite IP カタログの System Management Wizard を実行して、この機能をデザインに統合します。

7シリーズ コンフィギュレーション/BSCAN コンポーネントのリターゲット

注記 : コンフィギュレーション コンポーネントの詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [参照 9] を参照してください。

CAPTUREE2

UltraScale デバイスにおけるキャプチャ機能は、以前のデバイスから変更されています。したがって、UltraScale デバイスのデザインでは CAPTUREE2 プリミティブがリターゲットされないため、エラーが発生します。より包括的なソリューションとして Vivado® ロジック解析を使用してください。詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [参照 9] の「リードバック検証および CRC」のセクションを参照してください。

DNA_PORT

DNA の値は 57 ビットから 96 ビットに拡張されたため、プリミティブはこの新しい動作を反映するよう変更されています。



推奨 : DNA_PORT を含むすべてのデザインは、DNA_PORTE2 プリミティブをインスタンス化し、使用するようアップデートしてください。これにより、この機能に対して論理シミュレーションが正しく実行されます。

FRAME_ECCE2

7シリーズ デバイスと UltraScale デバイスでは、コンフィギュレーションフレームの誤り訂正回路または FRAME_ECC のピン配置および機能が異なります。このため、UltraScale デバイスのデザインに FRAME_ECCE2 コンポーネントが含まれると、これによってエラーが発生します。デザインで引き続きこの機能が必要な場合は、FRAME_ECCE2 を FRAME_ECCE3 コンポーネントで置き換えてください。



推奨 : 変更点およびこの新しいコンポーネントの使用法の詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [\[参照 9\]](#) を参照してください。

ICAPE2

内部コンフィギュレーション アクセス ポート (ICAP) は、ピン配置、機能、およびレイテンシが変更されています。このため、ICAPE2 コンポーネントは自動的にリターゲットされず、エラーが発生します。デザインで ICAP 機能が必要な場合は、ICAPE3 コンポーネントを使用してください。



推奨 : このコンポーネントの詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [\[参照 9\]](#) を参照してください。

STARTUPE2

STARTUP ブロックはピンおよび機能が追加され、STARTUPE3 ブロックとなっています。ほとんどの場合、このブロックは適切にリターゲットされます。ただし、必要に応じて STARTUPE3 ブロックをインスタンスエートするようコード修正して使用するのが最良の場合もあります。



推奨 : このコンポーネントの詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [\[参照 9\]](#) を参照してください。

7シリーズ I/O コンポーネントのリターゲット

IDELAYE2

IDELAYE2 プリミティブは、UltraScale アーキテクチャで IDELAYE3 プリミティブに置き換えられています。入力遅延ラインは従来の 32 タップではなく 512 タップで構成されており、最大遅延が減少する場合があります。仕様については、該当する UltraScale のデータシート [\[参照 11\]](#) を参照してください。必要であれば、IDELAYE3 を関連した IDELAYE3 出力遅延プリミティブとカスケード接続することで、より大きな遅延を作成可能です。FIXED モードでの遅延は、タップ値ではなく ps 単位になっています。遅延ラインは、この値および関連した基準クロックを (IDELAYCTRL を介して) 使用し、現時点の PVT (プロセス/電圧/温度) で目的の遅延を生成するために必要なタップ数を算出します。ユーザーによる操作は不要です。VAR_LOAD モードで IDELAYE3 をプログラムする場合、9 ビット値が必要です。

IDELAY は、BUFG または MMCM/PLL などのクロック リソースに直接接続できなくなりました。7シリーズ FPGA デザインでこのような接続を使用している場合、MMCM/PLL を使用して位相調整したり、データをシフトさせてクロックとデータを適切に揃えるなどの修正が必要です。

さらに、IDELAY を入力レジスタ、IDDR、または ISERDES と共に使用するときは、それぞれレジスタ、IDDR のクロック ピン (C)、あるいは ISERDES コンポーネントの CLK_DIV ピンと基準クロックを共有する必要があります。UltraScale アーキテクチャでは IDELAY の基準クロック周波数範囲が広がっているため、これらのコンポーネントに共通したクロックを使用するのに十分な柔軟性を備えています。

IN_FIFO

IN_FIFO は、任意で ISERDESE3 コンポーネント内で使用可能な 8 段の FIFO で置き換えられています。この機能にアクセスするには、ISERDESE3 をインスタンスエートし、このビルトイン機能を使用するように設定します。

ISERDESE2

ISERDESE2 は、ISERDESE3 プリミティブで置き換えられています。SerDes でサポートされる比率は、DDR クロッキングで 4 または 8 で、SDR クロッキングで 2 または 4 です。中間の比率を使用するには、FPGA ロジックのギアボックスが必要です。ビットスリップ機能はなくなりましたが、必要であれば FPGA ロジックに簡単にインプリメントできます。

トライステート用の SDR レジスタはなくなりました。トライステートの SDR 双方向信号または出力をサポートするには、データ フリップフロップを IOLOGIC (IOB 制約を使用した OFD) に配置し、トライステート フリップフロップをファブリックに配置して、できるだけ IOB の近くに割り当てます。1 つのフリップフロップが複数のトライステートを駆動する場合、そのフリップフロップを複製し、個々のトライステート フリップフロップをできるだけ IOB の近くに割り当てる必要があります。

ISERDESE3 には 8 段で、バイパス可能な FIFO が追加されているため、サンプリング クロックを分数分周したクロックを使用して、プリミティブからパラレル データを読み出すことができます。UltraScale デバイスが持つ I/O データキャプチャ機能を最大限に活用するには、デザインで ISERDESE3 をインスタンスエートして使用してください。

ODELAYE2

ODELAYE2 プリミティブは UltraScale アーキテクチャで ODELAYE3 プリミティブに置き換えられています。出力遅延ラインは、従来の 32 タップではなく、512 タップで構成され、最大遅延は 1.25ns になりました。遅延のカスケードはサポートされていません。FIXED モードでの遅延は、タップ値ではなく ps 単位になりました。遅延ラインは、この値および関連した基準クロックを (IDELAYCTRL を介して) 使用し、現時点の PVT で目的の遅延を生成するために必要なタップ数を算出します。ユーザーによる操作は不要です。VAR_LOAD モードで IDELAYE3 をプログラムする場合、9 ビット値が必要です。属性値 DELAY_SRC="CLKIN" はサポートされていません。

さらに、ODELAY を出力レジスタ、ODDR、または OSERDES と共に使用するときは、それぞれレジスタ、ODDR のクロック ピン (C)、あるいは OSERDES コンポーネントの CLK_DIV ピンと基準クロックを共有する必要があります。UltraScale アーキテクチャでは ODELAY の基準クロック周波数範囲が広がっているため、これらのコンポーネントに共通したクロックを使用するのに十分な柔軟性があります。

OSERDESE2

OSERDESE2 は、OSERDESE3 プリミティブで置き換えられています。SerDes でサポートされる比率は、DDR クロッキングで 4 または 8 で、SDR クロッキングで 2 または 4 です。中間の比率を使用するには、FPGA ファブリックのギアボックスが必要です。UltraScale デバイスが持つ I/O データキャプチャ機能を最大限に活用するには、デザインで OSERDESE3 をインスタンスエートして使用してください。

OUT_FIFO

UltraScale アーキテクチャには OUT_FIFO に直接相当する機能はありません。同様の機能が必要な場合は、IP カタログを用いて LUT ベースの FIFO を生成して OUT_FIFO の機能を置き換えることを検討してください。

IDDR

IDDR は、UltraScale アーキテクチャで IDDRE1 に置き換えられています。これらの機能は同じですが、クロック イネーブルおよびセット機能はなくなったため、これらのピン (CE および S) が削除されています。リセットはありますが、非同期リセットとしてのみ動作可能です。SRTYPE が SYNC に設定されていると、リターゲットされません。3 つの既存のキャプチャモード (SAME_EDGE、OPPOSITE_EDGE、SAME_EDGE_PIPELINED) はすべてサポートされます。

UltraScale デバイスが持つ I/O データ キャプチャ機能を最大限に活用するには、デザインで IDDR1 をインスタンスエートして使用してください。

ODDR

ODDR は、UltraScale アーキテクチャで ODDR1 に置き換えられています。これらの機能は同じですが、クロック イネーブル、リセット、およびセット機能はなくなったため、これらのピン (CE、R、S) が削除されています。R および S ピンは SR ピンで置き換えられています。このピンは、プログラムによってコンポーネントを非同期でセットまたはリセット可能です。INIT ファンクションはセットまたはリセット機能の設定と一致しなければなりません。つまり、リセットを指定したときは INIT をゼロに設定し、セットを指定したときは 1 に設定する必要があります。新しいプリミティブでは、SAME_EDGE モードのみサポートされています。別の相違点として、set/reset ファンクションがリリースされた場合、クロックの 3 サイクル後までコンポーネントはデータのキャプチャ/送信はできません。

UltraScale デバイスが持つ I/O データ キャプチャ機能を最大限に活用するには、デザインで ODDR1 をインスタンスエートして使用してください。

7シリーズ RAM/ROM のリターゲット

FIFO18E1/FIFO36E1

UltraScale アーキテクチャには、新しい機能および性能を備え、ソフト FIFO コアよりも互換性に優れた、ハード化された FIFO ブロックがあります。結果としてピン配置、機能、およびレイテンシが変わっているため、このコンポーネントは自動的にリターゲットされません。



推奨 : FIFOE1 ブロックを含むデザインの場合、『UltraScale アーキテクチャ メモリ リソース ユーザー ガイド』(UG573) [参照 4] を参照し、このブロックの変更点および新しい FIFOE2 ブロックの適切な使用法の詳細を確認してください。また、Vivado IP カタログの FIFO Generator を使用すると、このブロックを容易にデザインにインスタンスエートできます。

RAMB18E1/RAMB36E1

ブロック RAM コンポーネントは、UltraScale アーキテクチャでその機能が大きく改善されています。ただし、これは 7シリーズ FPGA の RAMBE1 ブロックが持つ特長および機能の上位セットです。通常、RAMBE1 ブロックを含むデザインは、UltraScale アーキテクチャに同じ機能を維持して適切にリターゲットされ、レイテンシも変わりません。しかし、UltraScale デバイスの RAMBE2 ブロックで改善された機能によってエリア、性能、消費電力などが向上したソリューションが得られる可能性もあるため、以前のコンポーネントの使用を見直してください。

注記 : UltraScale デバイスの場合に限り、広い SDP (Simple Dual Port) モード (RAMB36 では x72、RAMB18 では x36) で出力レジスタを使用している場合、1つのポートでオプションのクロック反転を使用し、もう1つのポートは反転しないとはできません。UltraScale+ デバイスがターゲットの場合、4つ以上のブロック RAM を必要とする大きいメモリ アレイでは、使用可能であれば、1つまたは複数の UltraRAM に移行することを検討してください。

7シリーズ クロック コンポーネントのリターゲット

BUFG

デザインに含まれる BUFG はすべて、クロック イネーブル (CE) が常時有効な BUFGCE に変換されます。これらの機能は同じで、通常は UltraScale デバイスで最適なリソースです。

BUFH/BUFHCE

UltraScale デバイスではクロック ネットワークが再設計されているため、7シリーズ FPGA の HROW バッファまたは BUFH はなくなりました。デザインに含まれる BUFH は BUFG にリターゲットされ、このまま最終的なインプリメンテーションに残るか、その他の BUFG と組み合わせられて適切なクロック ネットワークを形成します。同様に、BUFHCE は BUFGCE にリターゲットされます。ほとんどの場合、このリターゲットによって十分なクロッキング構造がもたらされますが、さらに改善あるいは強化できる点がないかどうか、UltraScale デバイスのクロッキング アーキテクチャを念頭に置いて、クロッキング トポロジを見直してください。クロッキング リソースの詳細は、『UltraScale アーキテクチャ クロッキング リソース ユーザー ガイド』(UG572) [参照 5] を参照してください。

BUFIO

UltraScale デバイスではクロック ネットワークが再設計されているため、7シリーズ FPGA の I/O クロック バッファまたは BUFIO はなくなりました。デザインに含まれる BUFIO は BUFG にリターゲットされ、このまま最終的なインプリメンテーションに残るか、その他の BUFG と組み合わせられて適切なクロック ネットワークを形成します。ほとんどの場合、このリターゲットによって低スキューで高速な I/O クロッキングをサポートする十分なクロッキング構造がもたらされますが、UltraScale デバイスのクロッキング アーキテクチャを念頭に置いて、クロッキング トポロジを見直してください。非常に高速な I/O インターフェイスのインプリメンテーション向けには、ビット スライス コンポーネントに専用で低ジッターのクロックがあります。これは BUFIO の代わりになりますが、ビット スライス コンポーネントと共に使用する必要があります。クロッキング リソースの詳細は、『UltraScale アーキテクチャ クロッキング リソース ユーザー ガイド』(UG572) [参照 5] を参照してください。

BUFR/BUFMR/BUFMRCE

UltraScale デバイスではクロック ネットワークが再設計されているため、7シリーズ FPGA のリージョナル クロック バッファ (BUFR、BUFMR、BUFMRCE) はなくなりました。デザインに含まれるこれらのバッファは、BUFGCE_DIV にリターゲットされ、適切な分周機能が維持されます。多くの場合、このリターゲットによって十分なクロッキング構造がもたらされますが、さらに改善あるいは強化できる点がないかどうか、UltraScale デバイスのクロッキング アーキテクチャを念頭に置いて、クロッキング トポロジを見直してください。クロッキング リソースの詳細は、『UltraScale アーキテクチャ クロッキング リソース ユーザー ガイド』(UG572) [参照 5] を参照してください。

MMCME2_ADV/MMCME2_BASE

UltraScale デバイスでは MMCM に機能が追加され、コンポーネントが MMCME3 に変更されました。ただし、MMCME3 は MMCME2 の完全な上位セットです。したがって、すべての MMCME2 は UltraScale デバイスに自動的にリターゲットされるものです。MMCM が Clocking Wizard または他の IP で作成されている場合は、UltraScale デバイスをターゲットにして作成し、このリソースが最新の形で表現されるようにしてください。デザインで MMCME2 が直接インスタンスシートされている場合、MMCME3 へのアップデートを推奨しますが、実際にこれを行うかどうかは設計者の判断に委ねられます。

PLLE2_ADV/PLLE2_BASE

UltraScale デバイスでは PLL が変更されており、その機能は 7シリーズ FPGA の PLLE2 に直接マップされません。このため、ツールでは PLL のインプリメンテーションを自動的に UltraScale デバイスにリターゲットはしますが、PLLE3 ではなく、PLLE2 機能の上位セットである MMCME3 にリターゲットします。ほとんどの場合、これで十分です。一方、UltraScale デバイスで PLLE3 を使用するには、デザインに直接インスタンスシートする必要があります。別の方法として、Clocking Wizard を使用することで、PLLE3 を含む適切な IP を使用することもできます。

7シリーズ レジスタおよびラッチのリターゲット

FDCE、FDPE、FDRE、FDSE

7シリーズ FPGA と比較すると、UltraScale デバイスのレジスタは次のような点で大きく改善されています。

- 7シリーズ デバイスのレジスタは、アクティブ High のリセットのみをサポートします。
- UltraScale デバイスのセット/リセット ピンには極性を反転させるオプションがあります。これにより、プリミティブにアクティブ High またはアクティブ Low のリセットがインプリメント可能です。この反転は、FD コンポーネントに対する次の属性で表され、制御されます。

```
FDCE: IS_CLR_INVERTED
FDPE: IS_PRE_INVERTED
FDRE: IS_R_INVERTED
FDSE: IS_S_INVERTED
```

インバーターがセットまたはリセットの前に記述される場合、この属性を設定する必要はありません。正しく最適な結果となるように、ツールが自動的に属性を設定します。



推奨: レジスタ付きトライステート出力パスについて: UltraScale デバイスには、トライステート パス上の専用レジスタはありません。トライステートには CLB レジスタを使用してください。詳細は、ザイリンクス アンサー #62490 (UltraScale I/O - SDR トライステート フリップフロップに推奨される設計手法) [参照 10] を参照してください。

スライス/CLB プリミティブ

CARRY4

UltraScale デバイスの CLB のサイズは 7シリーズ デバイスの 2 倍になったため、キャリー ロジックも同様に 2 倍になり、CARRY8 という新しいコンポーネントが追加されています。デザインで使用されている CARRY4 コンポーネントは、自動的かつ効率的に新しい CARRY8 コンポーネントにマップされます。



推奨: 最終的には、デザインにインスタンス化された CARRY4 はすべて、ネイティブ CARRY8 コンポーネントに変更してください。ほとんどのデザインですぐに変更が必要なわけではありませんが、単体の CARRY4 を複数含み、高密度のパッキングが求められるデザインでは変更することが推奨されます。この場合、2つの優先する CARRY4 のパッキングが定義されているコードで、CARRY4 ではなく CARRY8 を使用するよう修正します。これにより、複数の小さなキャリー チェーンを使用する密度の高いデザインが作成されます。

SelectIO に関する考察事項

UltraScale デバイスの SelectIO™ インターフェイス アーキテクチャは再設計され、改善されています。このセクションでは、7 シリーズ デバイスを使用した既存のデザインを UltraScale デバイスにリターゲットする際の SelectIO インターフェイスの概要ガイドラインを示します。SelectIO アーキテクチャの変更点の詳細は、『UltraScale アーキテクチャ SelectIO リソース ユーザー ガイド』(UG571) [参照 1] を参照してください。

DCI に関する注意事項

DCI のサポート

7 シリーズおよび UltraScale デバイスの両方で、DCI (デジタル制御インピーダンス) をサポートするのは HP I/O のみです。

UltraScale デバイスの場合は、使用する DCI 終端のタイプおよび値にかかわらず、VRP ピンに GND への 240Ω 抵抗を 1 つ使用するだけでこの機能がサポートされます。

終端回路の抵抗要件

7 シリーズ FPGA の DCI では、外部抵抗と同じ値になるように分割終端回路の各側がキャリブレーションされます。たとえば、VCCO/2 への並列終端が 50Ω のとき、7 シリーズ デバイスでは、VRN および VRP ピンに 100Ω の外部抵抗が必要になります。

UltraScale デバイスでは、VRP ピンに 240Ω の抵抗が 1 つしか必要ありません。分割終端抵抗またはシングル終端抵抗の正確な値は、ユーザー制御可能な ODT 属性で指定します。

分割終端 DCI 規格について ODT で設定可能な値 (HSTL および SSTL)

RTT_40	RTT_48	RTT_60	RTT_NONE
--------	--------	--------	----------



重要：分割終端 DCI 規格の場合、ODT 値は必要な VCCO/2 へのテブナン抵抗を示します。

シングル終端 POD 規格について ODT で設定可能な値

RTT_40	RTT_48	RTT_60	RTT_NONE
--------	--------	--------	----------

シングル終端 HSUL 規格について ODT で設定可能な値

RTT_120	RTT_240	RTT_NONE
---------	---------	----------



重要：シングル終端 DCI 規格の場合、ODT 値は必要な VCCO への抵抗を示します。

ソースドライバー終端

ソースドライバー終端は、制御インピーダンスドライバーをサポートする I/O 規格の OUTPUT_IMPEDANCE 属性で指定する必要があります。

制御インピーダンスドライバーのサポートは、次の I/O 規格まで拡張されています。

HSTL_I_DCI	DIFF_HSTL_I_DCI_18	HSLVDCI_18
DIFF_HSTL_I_DCI	LVDCI_15	POD10_DCI
LVDCI_18	POD12_DCI	DIFF_POD10_DCI
HSUL_12_DCI	DIFF_POD12_DCI	SSTL135_DCI
DIFF_HSUL_12_DCI	SSTL15_DCI	DIFF_SSTL135_DCI
SSTL18_I_DCI	DIFF_SSTL15_DCI	HSLVDCI_15
DIFF_SSTL18_I_DCI	HSTL_I_DCI_12	SSTL12_DCI
HSTL_I_DCI_18	DIFF_HSTL_I_DCI_12	DIFF_SSTL12_DCI

OUTPUT_IMPEDANCE 属性の詳細は、『UltraScale アーキテクチャ SelectIO リソース ユーザー ガイド』(UG571) [参照 1] を参照してください。

T_DCI 規格

7 シリーズ アーキテクチャは、トライステート時に内部入力分割終端がサポートされる、双方向 I/O コンフィギュレーションの T_DCI 規格に対応していました。UltraScale デバイスでは、このような T_DCI 規格はサポートされていません。

ただし、UltraScale アーキテクチャの DCI 規格の多くは同様の双方向コンフィギュレーションに対応可能です。次の表に、Vivado Design Suite を用いて設計した場合に、相当する UltraScale アーキテクチャの規格へ透過的に移植または移行される T_DCI 規格を示します。

表 2: アーキテクチャ間で移行される T_DCI I/O 規格

7 シリーズ アーキテクチャの I/O 規格	UltraScale アーキテクチャでの I/O 規格
DIFF_HSTL_II_T_DCI	移行なし
DIFF_HSTL_II_T_DCI_18	移行なし
DIFF_SSTL18_II_T_DCI	移行なし
DIFF_SSTL15_T_DCI	DIFF_SSTL15_DCI
DIFF_SSTL135_T_DCI	DIFF_SSTL135_DCI
DIFF_SSTL12_T_DCI	DIFF_SSTL12_DCI
HSTL_II_T_DCI	移行なし
HSTL_II_T_DCI_18	移行なし
SSTL18_II_T_DCI	移行なし
SSTL15_T_DCI	SSTL15_DCI
SSTL135_T_DCI	SSTL135_DCI
SSTL12_T_DCI	SSTL12_DCI

キャリブレーションなしの入力終端

UltraScale デバイスでは、IN_TERM 属性はサポートされていません。HR I/O の入力終端の値を指定するには、ODT を使用してください。UltraScale デバイスの HP I/O では、ODT 属性でアクセスされる未調整の入力終端もサポートされます。

未調整ソース終端

DCI I/O 規格と同様に、UltraScale では non-DCI バージョンの SSTL、HSTL、HSUL、POD 規格でもドライバー終端の値をオプションとして選択できます。これは HP I/O でのみサポートされています。この機能は OUTPUT_IMPEDANCE 属性を使用してアクセスします。

表 3 に、キャリブレーションなしのソース終端を HP I/O バンクでサポートする I/O 規格をすべて示しています。

表 3: HP I/O バンクでキャリブレーションなしのソース終端をサポートする I/O 規格

HSTL_I	DIFF_HSTL_I	SSTL18_I	DIFF_SSTL18_I	POD12	DIFF_POD12
HSTL_I_18	DIFF_HSTL_I_18	SSTL15	DIFF_SSTL15	POD10	DIFF_POD10
HSTL_I_12	DIFF_HSTL_I_12	SSTL135	DIFF_SSTL135	HSUL_12	DIFF_HSUL_12
		SSTL12	DIFF_SSTL12		

サポート対象外の I/O 規格

UltraScale デバイスでは、次の I/O 規格はサポートされていません。

DIFF_HSTL_II_T_DCI	DIFF_SSTL12_T_DCI	SSTL135_T_DCI
DIFF_HSTL_II_T_DCI_18	HSTL_II_T_DCI	SSTL12_T_DCI
DIFF_SSTL18_II_T_DCI	HSTL_II_T_DCI_18	MOBILE_DDR
DIFF_SSTL15_T_DCI	SSTL18_II_T_DCI	DIFF_MOBILE_DDR
DIFF_SSTL135_T_DCI	SSTL15_T_DCI	PCI33_3

HP I/O では、次の Class II I/O 規格はサポートされていません。

DIFF_HSTL_II	DIFF_SSTL18_II	HSTL_II_18
DIFF_HSTL_II_18	HSTL_II	SSTL18_II

VREF ベースの入力規格

7 シリーズ デバイスでは、VREF ベースの入力規格に VCCO 要件がなく、オンダイ入力終端機能を使用しない場合はこれらの入力を任意の VCCO のバンクに配置できます。一方、UltraScale アーキテクチャでは、次の VREF ベースの入力規格に VCCO 要件があり、これらは適合する VCCO があるバンクにのみ配置できます。この要件は、オンダイ入力終端 (ODT) や内部 VREF 機能を使用しない場合でも適用されます。

HSUL_12	HSTL_I_DCI_18	SSTL12
HSUL_12_DCI	HSTL_I_12	SSTL18_I_DCI
HSLVDCI_18	HSTL_I_DCI_12	SSTL15_DCI
HSLVDCI_15	SSTL18_I	SSTL135_DCI
HSTL_I	SSTL18_II	SSTL12_DCI
HSTL_II	SSTL15	POD10
HSTL_I_DCI	SSTL15_R	POD12
HSTL_I_18	SSTL135	POD10_DCI
HSTL_II_18	SSTL135_R	POD12_DCI
		MIPI_DPHY_DCI*

注記 : * UltraScale+ にのみ適用されます。

外部 VREF

UltraScale デバイスの VREF は専用ピンです。各バンク (場合によってミニバンク) に 1 つの VREF があります。

HD I/O バンクでは、外部 VREF はサポートされておらず、代わりにデバイス内部で生成されます。内部生成された VREF は VCCO 電源レールをソースとしています。HD I/O バンクは、Zynq UltraScale+ MPSoC デバイスおよび Kintex UltraScale+ デバイスでのみサポートされています。

SelectIO インターフェイス UNISIM プリミティブ

7 シリーズ FPGA の SelectIO インターフェイス UNISIM プリミティブはすべて、UltraScale デバイスでサポートされています。さらに、新しい機能をサポートする UltraScale デバイス固有の UNISIM プリミティブとして、IBUFE3、IBUFDSE3、IOBUFE3、IOBUFDSE3、HPIO_VREF、IBUF_ANALOG があります。

加えて、新しい MIPI D-PHY 機能をサポートする新しい UltraScale+ デバイス固有の UNISIM プリミティブとして、IBUFDS_DPHY、OBUFDSDPHY があります。

- DCITERMDISABLE および INTERMDISABLE ポート :
 - ダイナミック入力終端制御ポート DCITERMDISABLE は、UltraScale デバイスの HP I/O にある DCI および未調整のオンダイ入力終端の両方を制御します。
 - UltraScale デバイスでは 7 シリーズ デバイスと同様に、ダイナミック入力終端制御ポート INTERMDISABLE が HR I/O の未調整終端を制御します。
- IBUFDISABLE ポート :
 - UltraScale デバイスの場合、IBUFDISABLE ポートが有効 (適切な場合には T が 1) で IBUFDISABLE が UNISIM でアサートされると、内部ロジックへの出力 0 はロジック Low になります。これは、IBUFDISABLE のアサートによって内部ロジックへの出力 0 がロジック High となる 7 シリーズ デバイスと異なる動作です。

V_{CCAUX_IO}

7 シリーズ FPGA と異なり、UltraScale デバイスの補助 I/O (V_{CCAUX_IO}) 電源レールは 2 つの電圧レベルをサポートしません。サポートされるのは、1.8V の標準電圧レベルのみです。

SLEW 属性

UltraScale デバイスでは、SLEW 属性が HP I/O で MEDIUM という新しい値をサポートします。

差動入力終端

DIFF_TERM 属性は UltraScale デバイスでサポートされますが、DIFF_TERM_ADV という新しい属性に自動的にマップされます。

- DIFF_TERM = TRUE は自動的に DIFF_TERM_ADV = TERM_100 にマップされます (100Ω 差動終端)
- DIFF_TERM = FALSE は自動的に DIFF_TERM_ADV = TERM_NONE にマップされます (デフォルト、終端なし)

DIFF_TERM_ADV 属性で設定可能な値は次のとおりです。

- DIFF_TERM_ADV = TERM_NONE (デフォルト)
- DIFF_TERM_ADV = TERM_100

新規および拡張機能

新機能および拡張機能の詳細は、『UltraScale アーキテクチャ SelectIO リソース ユーザー ガイド』(UG571) [参照 1] を参照してください。

新しいプリミティブに関する注意事項

このセクションでは、UltraScale アーキテクチャに新たに追加されたプリミティブについて説明します。7 シリーズ FPGA にはこれらに直接相当するプリミティブはありません。これらのプリミティブは、特定の条件でより優れた特性を達成するため、リターゲットしたデザインへの優先追加順に示します。これらのコンポーネントを使用する際の詳細は、『UltraScale アーキテクチャ ライブラリ ガイド』(UG974) [参照 7] を参照してください。

スライス/CLB プリミティブ

MUXF9

UltraScale デバイスでは CLB のサイズが 7 シリーズ デバイスと比較して 2 倍になっており、MUXF9 が追加されています。この MUX を使用すると、1 つの CLB 内に 1 段の LUT ロジック レベルで 32:1 の MUX をインプリメントできます。あるいは、9 入力 LUT または 1 x 52 ビット ROM など多入力の条件を作成することも可能です。これによって、CLB 内のすべての LUT を効率的に使用できるようになります。

RAM512X1S/RAM256X1D

CLB が多いことで、大きな分散 RAM または LUT RAM の構築が容易になり、非同期の読み出しおよび書き込み動作に使用できるようになりました。UltraScale では、深さ 512 ビットでシングルポートのファンクションおよび深さが 256 ビットでシングルデュアルポートのファンクションが 1 つの CLB で構築できるようになりました。

RAM32M16/RAM64M8

マルチポート (書き込みポートが1つ、読み出しポートが複数) の大きな構成も、UltraScale デバイスでは1つの CLB で可能です。マルチポート RAM へのアクセスが必要な場合は、RAM32M16 または RAM64M8 の使用を検討してください。

その他のリソースおよび法的通知

ザイリンクス リソース

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

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。デザイン アシスタント、デザイン アドバイザリ、トラブルシュートのヒントなどが含まれます。

参考資料

注記：日本語版のバージョンは、英語版より古い場合があります。

- 『UltraScale アーキテクチャ SelectIO リソース ユーザー ガイド』(UG571 : [英語版](#)、[日本語版](#))
- 『UltraFast 設計手法ガイド (Vivado Design Suite 用)』(UG949 : [英語版](#)、[日本語版](#))
- 『UltraScale アーキテクチャおよび製品概要』(DS890 : [英語版](#)、[日本語版](#))
- 『UltraScale アーキテクチャ メモリ リソース ユーザー ガイド』(UG573 : [英語版](#)、[日本語版](#))
- 『UltraScale アーキテクチャ クロッキング リソース ユーザー ガイド』(UG572 : [英語版](#)、[日本語版](#))
- 『ISE から Vivado Design Suite への移行ガイド』(UG911 : [英語版](#)、[日本語版](#))
- 『UltraScale アーキテクチャ ライブラリ ガイド』(UG974 : [英語版](#)、[日本語版](#))
- 『UltraScale アーキテクチャ システム モニター ユーザー ガイド』(UG580 : [英語版](#)、[日本語版](#))
- 『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570 : [英語版](#)、[日本語版](#))
- [ザイリンクス アンサー 62490](#)
- UltraScale アーキテクチャのデータシートは、[ザイリンクスのサポートサイト](#)で参照できます。
- UltraScale+ アーキテクチャのデータシートは、[ザイリンクスのサポートサイト](#)で参照できます。

トレーニング リソース

ザイリンクスは、この文書で説明した概念をさらに詳しく学習するためのさまざまなトレーニング コースと QuickTake ビデオを提供しています。関連トレーニング リソースには、次のリンクからアクセスできます。

1. [UltraFast 設計手法トレーニング コース](#)
2. [FPGA デザイン トレーニング コースの概要](#)
3. [Vivado Design Suite QuickTake ビデオ チュートリアル](#)

法的通知

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

自動車用のアプリケーションの免責条項

ザイリンクスの製品は、フェイルセーフとして設計されたり意図されたりはならず、また、フェイルセーフの動作を要求するアプリケーション (具体的には、(I) エアバッグの展開、(II) 車のコントロール (フェイルセーフまたは余剰性の機能 (余剰性を実行するためのザイリンクスの装置にソフトウェアを使用することは含まれません) および操作者がミスをした際の警告信号がある場合を除きます)、(III) 死亡や身体傷害を導く使用、に関するアプリケーション) を使用するために設計されたり意図されたりもしていません。顧客は、そのようなアプリケーションにザイリンクスの製品を使用する場合のリスクと責任を単独で負います。

© Copyright 2013-2016 Xilinx, Inc. Xilinx, Xilinx のロゴ、Artix、ISE、Kintex、Spartan、Virtex、Vivado、Zynq、およびこの文書に含まれるその他の指定されたブランドは、米国およびその他の各国のザイリンクス社の商標です。PCI、PCIe、PCIExpress は PCI-SIG の商標であり、使用にはライセンスが必要です。すべてのその他の商標は、それぞれの保有者に帰属します。

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