

# Spartan-6 ライブラリ ガイド (HDL 用)

UG615 (v14.5) 2013 年 3 月 20 日

該当するソフトウェア バージョン : ISE Design Suite 14.5 および 14.6



## Notice of Disclaimer

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

© Copyright 2002–2012 Xilinx Inc. All rights reserved. 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 (v14.5) 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。

## 概要

---

HDL 用ライブラリ ガイドは、ISE® のオンライン マニュアルの 1 つです。回路図を使用して設計する場合は、回路図用ライブラリ ガイドを参照してください。

このマニュアルには、次の内容が含まれます。

- ・ 概要
- ・ 各マクロの詳細説明
- ・ このアーキテクチャでサポートされるプリミティブとマクロのファンクション別リスト
- ・ 各プリミティブの詳細説明

## デザイン エLEMENTについて

このバージョンのライブラリ ガイドでは、Spartan®-6 デバイスのデザイン エLEMENTの説明とそのインスタンス化コード例を示します。インスタンス化 テンプレートは、ISE/doc/usenglish/isehelp のインストール ディレクトリにも個別の ZIP ファイルとして含まれています。

デザイン エLEMENTは、次の 3 つのカテゴリに分類されます。

- ・ **マクロ**：ザイリンクス ツールの UniMacro ライブラリに含まれ、プリミティブだけでは複雑すぎてインスタンス化しにくいようなプリミティブをインスタンス化する際に使用します。UniMacro は、合成ツールで自動的に下位プリミティブに展開されます。
- ・ **プリミティブ**：ターゲットである FPGA デバイス用のザイリンクス コンポーネントです。プリミティブをインスタンス化して変換 (NGDBuild) プロセスを実行すると、変換後のファイルに含まれるのはまったく同じコンポーネントです。たとえば、ISERDES\_NODELAY という Virtex®-5 エLEMENTをユーザー プリミティブとしてインスタンス化し、変換 (NGDBuild) を実行すると、ISERDES\_NODELAY がそのまま残ります。一方 Virtex-5 デバイスで ISERDES を使用していると、自動的に Virtex-5 用の ISERDES\_NODELAY に変換されます。このため、「プリミティブ」の概念は、同じ分野でもその使用によって異なります。

CORE Generator では、さまざまなデバイス アーキテクチャに対応した多数のデザイン エLEMENT (UniMacro およびプリミティブ) を含むソフトウェア ライブラリを提供しています。開発システム ツールのリリースごとに、新しいデザイン エLEMENTが組み込まれます。すべてのデザイン エLEMENTを含むユニファイド ライブラリに対し、このガイドにはアーキテクチャ固有のライブラリのみが含まれています。

## デザインの入力方法

このガイドでは、各デザイン エLEMENT で 4 つの使用方法を評価して、その中から最適なソリューションを示します。この 4 つの使用方法は、次のとおりです。

- ・ **インスタンス化**：デザインにコンポーネントを直接インスタンス化します。これは、各ブロックの配置をユーザーが制御する場合に有効な方法です。
- ・ **推論**：コンポーネントはサポートされる合成ツールで推論されます。コードは柔軟性および移植性に優れているので、複数のアーキテクチャで使用できます。推論を使用すると、パフォーマンス、エリア、消費電力など、合成ツールでの指定に基づいて最適化されます。
- ・ **CORE Generator およびウィザード**：コンポーネントは CORE Generator またはウィザードから使用できます。この方法は、推論できない FPGA プリミティブを使用して大型ブロックを構築する場合に使用してください。このフローを使用する場合は、各ターゲットアーキテクチャ用にコアを再生成する必要があります。
- ・ **マクロのサポート**：使用可能な UniMacro があります。これらのコンポーネントはザイリンクス ツールの UniMacro ライブラリに含まれ、プリミティブだけでは複雑すぎてインスタンス化しにくいプリミティブをインスタンス化する際に使用します。UniMacro は、合成ツールで自動的に下位プリミティブに展開されます。

## UniMacro

---

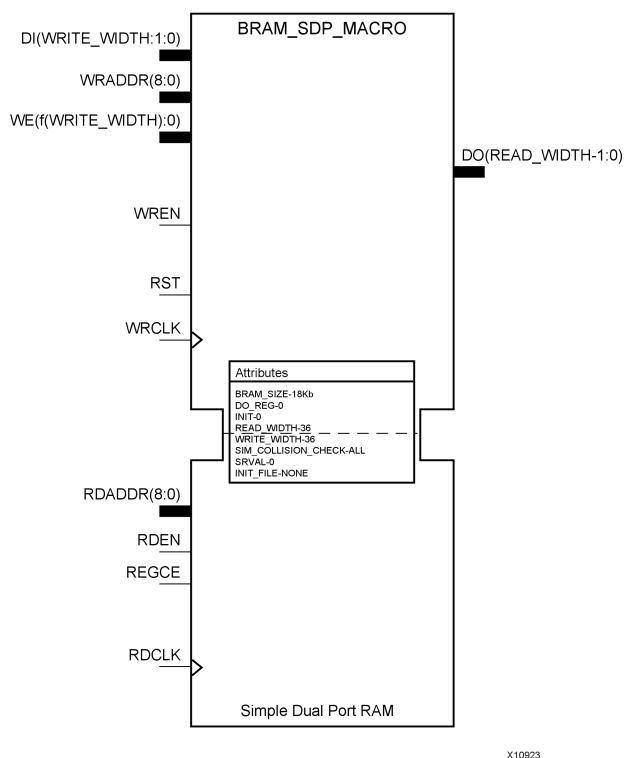
このセクションでは、Spartan®-6 デバイスで利用できる UniMacro について説明します。UniMacro は、アルファベット順に並べられています。

各 UniMacro について、次の情報を示します。

- ・ 名称
- ・ 説明
- ・ 回路図シンボル
- ・ 論理表 (該当するエレメントでのみ)
- ・ ポートの説明
- ・ デザインの入力方法
- ・ 使用可能な属性
- ・ インスタンス化コードの例
- ・ その他のリソース

## BRAM\_SDP\_MACRO

### マクロ : Simple Dual Port RAM



### 概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (18kb または 9kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。読み出しと書き込みは、コンポーネントに供給されるクロックに完全に同期して実行されます。ただし、読み出しポートと書き込みポートは完全に独立しており、お互いに非同期で、同じメモリアレイにアクセスします。バイト イネーブルの書き込みが可能で、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

**注記：** このエレメントは、読み出しおよび書き込みのポートの幅が同じになるようにコンフィギュレーションする必要があります。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
DO	出力	「コンフィギュレーション表」を参照	RDADDR で指定されたデータ出力バス
入力ポート			
DI	入力	「コンフィギュレーション表」を参照	WRADDR で指定されたデータ入力バス
WRADDR、RDADDR	入力	「コンフィギュレーション表」を参照	書き込み/読み出しアドレス入力バス
WE	入力	「コンフィギュレーション表」を参照	バイト幅ライト イネーブル

ポート名	方向	幅	機能
WREN、 RDEN	入力	1	ライト/リード イネーブル
SSR	入力	1	出力レジスタの同期リセット
REGCE	入力	1	出力レジスタのクロック イネーブル入力 (DO_REG=1 の場合にのみ有効)
WRCLK、 RDCLK	入力	1	書き込み/読み出しクロック入力

## コンフィギュレーション表

DATA_WIDTH	BRAM_SIZE	ADDR	WE
36 ~ 19	18Kb	9	4
	9Kb	8	
18 ~ 10	18Kb	10	2
	9Kb	9	
9 ~ 5	18Kb	11	1
	9Kb	10	
4 ~ 3	18Kb	12	1
	9Kb	11	
2	18Kb	13	1
	9Kb	12	
1	18Kb	14	1
	9Kb	13	

## デザインの入力方法

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメーターを設定できるようにしたものです。上記のコンフィギュレーション表を参照し、デザイン要件を満たすように設定してください。

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	推奨

## 使用可能な属性

属性	データ型	値	デフォルト	説明
BRAM_SIZE	文字列	"18Kb"、"9Kb"	"9Kb"	RAM を 18Kb または 9Kb メモリとしてコンフィギュレーションします。
DO_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロックサイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能です。clock-to-out タイムが長くなります。
INIT	16 進数	72 ビット値	すべて 0	コンフィギュレーション後の出力の初期値を指定します。
READ_WIDTH、WRITE_WIDTH	整数	1 ~ 72	36	DI および DO バスの幅を指定します。 次の設定が可能です。 <ul style="list-style-type: none"> <li>・ READ_WIDTH = WRITE_WIDTH</li> <li>・ READ_WIDTH と WRITE_WIDTH に異なる値を指定する場合は、比率を 2 にするか、UNISIM で許容される値 (1、2、4、8、9、16、18、32、36、64、72) にする必要があります。</li> </ul>
INIT_FILE	文字列	ファイルの名前と場所	なし	初期値を含むファイルの名前を指定します。
SIM_COLLISION_CHECK	文字列	"ALL"、 "WARNING_ONLY"、 "GENERATE_X_ONLY"、 "NONE"	"ALL"	メモリの競合が発生した場合のシミュレーションの動作を指定します。出力は次のようになります。 <ul style="list-style-type: none"> <li>・ "ALL" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。</li> <li>・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。</li> </ul> <b>注記 :</b> "ALL" 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。





[illegible]



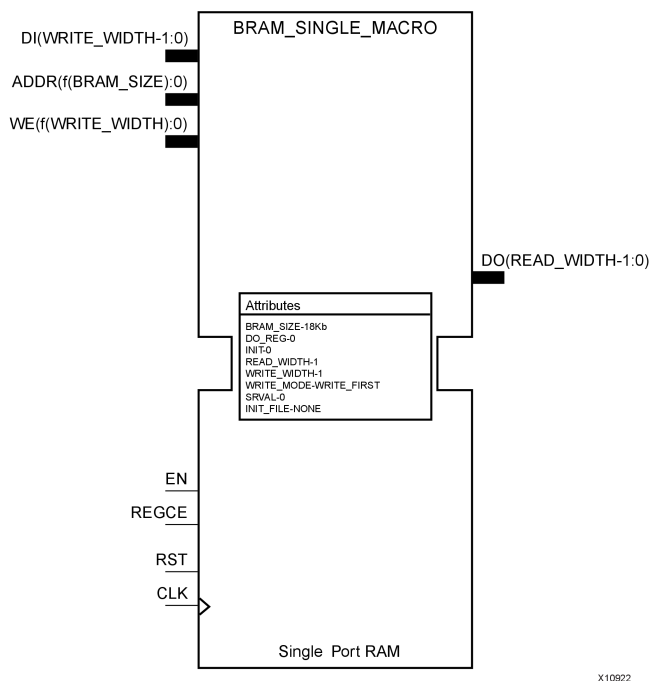


## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BRAM\_SINGLE\_MACRO

### マクロ : Single Port RAM



### 概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (18kb または 9kb) としてコンフィギュレーションできます。これらのシングル ポートのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。バイトイネーブルの書き込みが可能で、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
DO	出力	「コンフィギュレーション表」を参照	ADDR で指定されたデータ出力バス
入力ポート			
DI	入力	「コンフィギュレーション表」を参照	ADDR で指定されたデータ入力バス
ADDR	入力	「コンフィギュレーション表」を参照	アドレス入力バス
WE	入力	「コンフィギュレーション表」を参照	バイト幅ライト イネーブル
EN	入力	1	書き込み/読み出しイネーブル
RST	入力	1	出力レジスタの同期リセット
REGCE	入力	1	出力レジスタのクロック イネーブル入力 (DO_REG=1 の場合にのみ有効)
CLK	入力	1	クロック入力

## コンフィギュレーション表

WRITE_WIDTH	BRAM_SIZE	ADDR	WE
37 ~ 72	18Kb	8	8
36 ~ 19		9	4
18 ~ 10		10	2
9 ~ 5		11	1
4 ~ 3		12	1
2		13	1
1		14	1
36 ~ 19	9Kb	8	4
18 ~ 10		9	2
9 ~ 5		10	1
4 ~ 3		11	1
2		12	1
1		13	1

## デザインの入力方法

この UniMacro はインスタンスエーションのみが可能で、プリミティブにパラメーターを設定できるようにしたものです。上記のコンフィギュレーション表を参照し、デザイン要件を満たすように設定してください。

インスタンスエーション	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	推奨

## 使用可能な属性

属性	データ型	値	デフォルト	説明
BRAM_SIZE	文字列	"18Kb"、"9Kb"	"9Kb"	RAM を 18Kb または 9Kb メモリとしてコンフィギュレーションします。
DO_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロック サイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能ですが、clock-to-out タイムが長くなります。
READ_WIDTH、WRITE_WIDTH	整数	1 ~ 36	1	DI および DO バスの幅を指定します。  READ_WIDTH と WRITE_WIDTH に同じ値を指定する必要があります。
INIT_FILE	文字列	ファイルの名前と場所	なし	初期値を含むファイルの名前を指定します。
WRITE_MODE	文字列	"READ_FIRST"、 "WRITE_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	メモリへの書き込みモードを指定します。





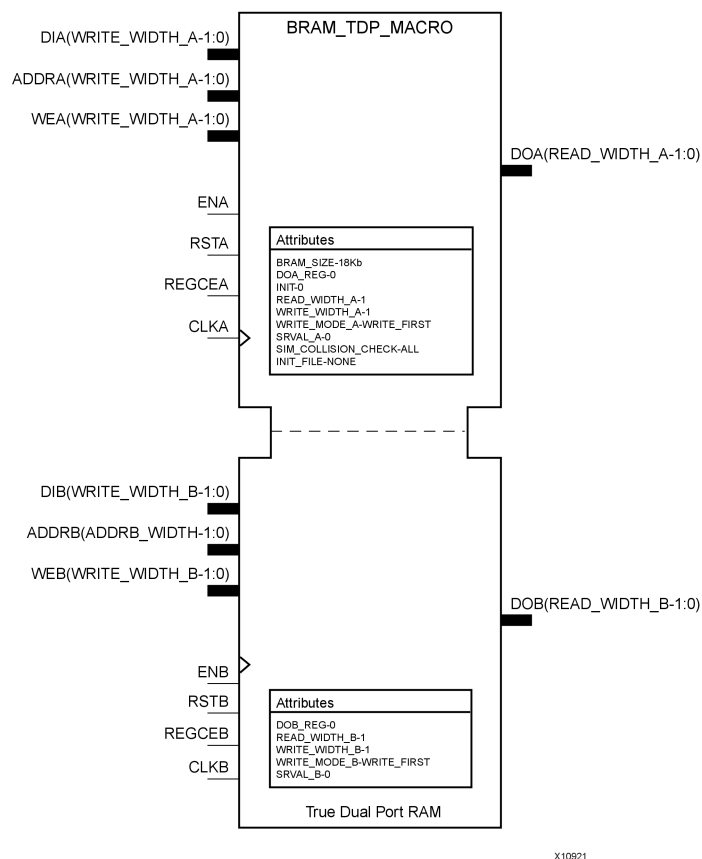






## BRAM\_TDP\_MACRO

### マクロ : True Dual Port RAM



## 概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (18kb または 9kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。読み出しと書き込みは、コンポーネントに供給されるクロックに完全に同期して実行されます。ただし、読み出しポートと書き込みポートは完全に独立しており、お互いに非同期で、同じメモリ アレイにアクセスします。バイト イネーブルの書き込みが可能になり、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

## ポートの説明

ポート名	方向	幅	機能
出力ポート			
DOA	出力	「コンフィギュレーション表」を参照	ADDRA で指定されたデータ出力バス
DOB	出力	「コンフィギュレーション表」を参照	ADDRB で指定されたデータ出力バス
入力ポート			
DIA	入力	「コンフィギュレーション表」を参照	ADDRA で指定されたデータ入力バス

ポート名	方向	幅	機能
DIB	入力	「コンフィギュレーション表」を参照	ADDRB で指定されたデータ入力バス
ADDRA、ADDRB	入力	「コンフィギュレーション表」を参照	ポート A およびポート B のアドレス入力バス
WEA、WEB	入力	「コンフィギュレーション表」を参照	ポート A およびポート B のライトイネーブル
ENA、ENB	入力	1	ポート A およびポート B のライト/リード イネーブル
RSTA、RSTB	入力	1	ポート A およびポート B の出力レジスタの同期リセット
REGCEA、REGCEB	入力	1	ポート A および B の出力レジスタのクロック イネーブル入力 (DO_REG=1 の場合にのみ有効)
CLKA、CLKB	入力	1	ポート A および B の書き込み/読み出しクロック入力

## コンフィギュレーション表

WRITE_WIDTH_A/B-DIA/DIB	BRAM_SIZE	ADDRA/B	WEA/B
36 ~ 19	18Kb	9	4
18 ~ 10		10	2
9 ~ 5		11	1
4 ~ 3		12	1
2		13	1
1		14	1
18 ~ 10	9Kb	9	2
9 ~ 5		10	1
4 ~ 3		11	1
2		12	1
1		13	1

## デザインの入力方法

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメーターを設定できるようにしたものです。上記のコンフィギュレーション表を参照し、デザイン要件を満たすように設定してください。

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	推奨

## 使用可能な属性

属性	データ型	値	デフォルト	説明
BRAM_SIZE	文字列	"18Kb"、"9Kb"	"9Kb"	RAM を 18Kb または 9Kb メモリとしてコンフィギュレーションします。
DO_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロック サイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能です、clock-to-out タイムが長くなります。
INIT	16 進数	72 ビット値	すべて 0	コンフィギュレーション後の出力の初期値を指定します。
INIT_FILE	文字列	ファイルの名前と場所	なし	初期値を含むファイルの名前を指定します。
READ_WIDTH、WRITE_WIDTH	整数	1 ~ 72	36	DI および DO バスの幅を指定します。  READ_WIDTH と WRITE_WIDTH に同じ値を指定する必要があります。
SIM_COLLISION_CHECK	文字列	"ALL"、 "WARNING_ONLY"、 "GENERATE_X_ONLY"、 "NONE"	"ALL"	メモリの競合が発生した場合のシミュレーションの動作を指定します。出力は次のようになります。  <ul style="list-style-type: none"> <li>・ "ALL" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。</li> <li>・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。</li> </ul> <p><b>注記 :</b> "ALL" 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SIM_MODE	文字列	"SAFE"、"FAST"	"SAFE"	シミュレーションのみの属性です。"FAST" に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。
SRVAL_A、SRVAL_B	16 進数	72 ビット値	すべて 0	同期リセット信号 (RST) がアサートされたときの DO ポートの出力値を指定します。
INIT_00 ~ INIT_FF	16 進数	256 ビット値	すべて 0	16Kb または 32Kb のデータ メモリ アレイの初期値を指定します。
INITP_00 ~ INITP_0F	16 進数	256 ビット値	すべて 0	2Kb または 4Kb のパリティ データ メモリ アレイの初期値を指定します。

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
use UNISIM.vcomponents.all;
```

```
-- BRAM_TDP_MACRO: True Dual Port RAM
--                 Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

-- Note - This Unimacro model assumes the port directions to be "downto".
--         Simulation of this model with "to" in the port directions could lead to erroneous results.
```

DATA_WIDTH_A/B	BRAM_SIZE	RAM Depth	ADDR A/B Width	WEA/B Width
19-36	"18Kb"	512	9-bit	4-bit
10-18	"18Kb"	1024	10-bit	2-bit
10-18	"9Kb"	512	9-bit	2-bit
5-9	"18Kb"	2048	11-bit	1-bit
5-9	"9Kb"	1024	10-bit	1-bit
3-4	"18Kb"	4096	12-bit	1-bit
3-4	"9Kb"	2048	11-bit	1-bit
2	"18Kb"	8192	13-bit	1-bit
2	"9Kb"	4096	12-bit	1-bit
1	"18Kb"	16384	14-bit	1-bit
1	"9Kb"	8192	12-bit	1-bit

[illegible]

```
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",

-- The next set of INIT_xx are for "18Kb" configuration only
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",

-- The next set of INITP_xx are for the parity bits
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",

-- The next set of INITP_xx are for "18Kb" configuration only
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")

port map (
DOA => DOA,      -- Output port-A data
DOB => DOB,      -- Output port-B data
ADDRA => ADDR_A, -- Input port-A address
ADDRB => ADDR_B, -- Input port-B address
CLKA => CLKA,    -- Input port-A clock
CLKB => CLKB,    -- Input port-B clock
DIA => DIA,      -- Input port-A data
DIB => DIB,      -- Input port-B data
ENA => ENA,      -- Input port-A enable
ENB => ENB,      -- Input port-B enable
REGCEA => REGCEA, -- Input port-A output register enable
REGCEB => REGCEB, -- Input port-B output register enable
RSTA => RSTA,     -- Input port-A reset
RSTB => RSTB,     -- Input port-B reset
```



```
-- End of BRAM TDP MACRO inst instantiation
```

```
// BRAM_TDP_MACRO: True Dual Port RAM
//                               Spartan-6
// Xilinx HDL Libraries Guide, version 14.5
```

Spartan-6 ライブラリ ガイド (HDL 用)  
UG615 (v14.5) 2013 年 3 月 20 日



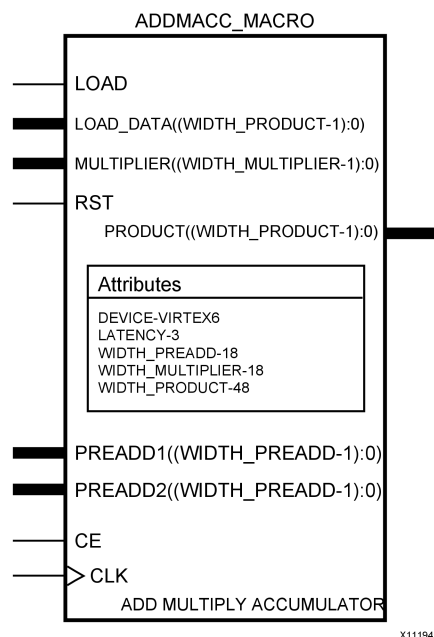
```
// End of BRAM_TDP_MACRO_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ADDMACC\_MACRO

マクロ : Adder/Multiplier/Accumulator



### 概要

ADDMACC\_MACRO を使用すると、DSP48 ブロックを前置加算、積和演算ファンクションとして使用する場合のインスタネーションが簡単になります。入力幅、出力幅、レイテンシを指定可能であり、DSP48 ブロックを HDL に統合しやすくなっています。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
PRODUCT	出力	可変 (WIDTH_A 属性値 + WIDTH_B 属性値)	プライマリ データ出力
入力ポート			
PREADD1	入力	可変 (WIDTH_PREADD 属性を参照)	前置加算データ入力
PREADD2	入力	可変 (WIDTH_PREADD 属性を参照)	前置加算データ入力
MULTIPLIER	入力	可変 (WIDTH_MULTIPLIER 属性を参照)	乗算データ入力
CARRYIN	入力	1	キャリー入力
CLK	入力	1	クロック
CE	入力	1	クロック イネーブル
LOAD	入力	1	ロード

ポート名	方向	幅	機能
LOAD_DATA	入力	可変 (WIDTH_PRODUCT 属性を参照)	DSP スライスでは、LOAD がアサートされると P に $A*B+LOAD\_DATA$ が読み込まれます。
RST	入力	1	同期リセット

## デザインの入力方法

この UniMacro はインスタンスレーションのみが可能で、プリミティブにパラメーターを設定できるようにしたものです。

インスタンスレーション	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	推奨

## 使用可能な属性

属性	データ型	値	デフォルト	説明
WIDTH_PREADD	整数	1 ~ 24	24	PREADD1 および PREADD2 入力の幅を指定します。
WIDTH_MULTIPLIER	整数	1 ~ 18	18	MULTIPLIER 入力の幅を指定します。
WIDTH_PRODUCT	整数	1 ~ 48	48	MULTIPLIER 出力の幅を指定します。
LATENCY	整数	0、1、2、3、4	3	パイプライン レジスタの数を指定します。 <ul style="list-style-type: none"> <li>・ 1 : MREG == 1</li> <li>・ 2 : AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1</li> <li>・ 3 : AREG == BREG == 1 および MREG == 1 および PREG == 1</li> <li>・ 4 : AREG == BREG == 2 および MREG == 1 および PREG == 1</li> </ul>
DEVICE	文字列	"VIRTEX6"、 "SPARTAN6"	"VIRTEX6"	ターゲットのハードウェア アーキテクチャを指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ADDMACC_MACRO: Add and Multiple Accumulate Function implemented in a DSP48E
--                Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

ADDMACC_MACRO_inst : ADDMACC_MACRO
generic map (
    DEVICE => "SPARTAN6", -- Target Device: "VIRTEX6", "SPARTAN6"
    LATENCY => 4,          -- Desired clock cycle latency, 1-4
    WIDTH_PREADD => 25,    -- Pre-Adder input bus width, 1-25
    WIDTH_MULTIPLIER => 18, -- Multiplier input bus width, 1-18
    WIDTH_PRODUCT => 48)  -- MACC output width, 1-48
port map (
    PRODUCT => PRODUCT,    -- MACC result output, width defined by WIDTH_PRODUCT generic
    MULTIPLIER => MULTIPLIER, -- Multiplier data input, width determined by WIDTH_MULTIPLIER generic
    PREADDER1 => PREADDER1, -- Preadder data input, width determined by WIDTH_PREADDER generic
    PREADDER2 => PREADDER2, -- Preadder data input, width determined by WIDTH_PREADDER generic
    CARRYIN => CARRYIN,    -- 1-bit carry-in input
    CE => CE,              -- 1-bit input clock enable
    CLK => CLK,            -- 1-bit clock input
    LOAD => LOAD,          -- 1-bit accumulator load input
    LOAD_DATA => LOAD_DATA, -- Accumulator load data input, width defined by WIDTH_PRODUCT generic
    RST => RST             -- 1-bit input active high synchronous reset
);
-- End of ADDMACC_MACRO_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// ADDMACC_MACRO: Variable width & latency - Pre-Add -> Multiplier -> Accumulate
//                function implemented in a DSP48E
//                Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

ADDMACC_MACRO #(
    .DEVICE("SPARTAN6"), // Target Device: "VIRTEX6", "SPARTAN6"
    .LATENCY(4),          // Desired clock cycle latency, 0-4
    .WIDTH_PREADD(18),    // Pre-adder input width, 1-18
    .WIDTH_MULTIPLIER(18), // Multiplier input width, 1-18
    .WIDTH_PRODUCT(48)    // MACC output width, 1-48
) ADDMACC_MACRO_inst (
    .PRODUCT(PRODUCT),    // MACC result output, width defined by WIDTH_PRODUCT parameter
    .CARRYIN(CARRYIN),    // 1-bit carry-in input
    .CLK(CLK),            // 1-bit clock input
    .CE(CE),              // 1-bit clock enable input
    .LOAD(LOAD),          // 1-bit accumulator load input
    .LOAD_DATA(LOAD_DATA), // Accumulator load data input, width defined by WIDTH_PRODUCT parameter
    .MULTIPLIER(MULTIPLIER), // Multiplier data input, width defined by WIDTH_MULTIPLIER parameter
    .PREADD2(PREADD2),    // Preadder data input, width defined by WIDTH_PREADD parameter
    .PREADD1(PREADD1),    // Preadder data input, width defined by WIDTH_PREADD parameter
    .RST(RST)             // 1-bit active high synchronous reset
);

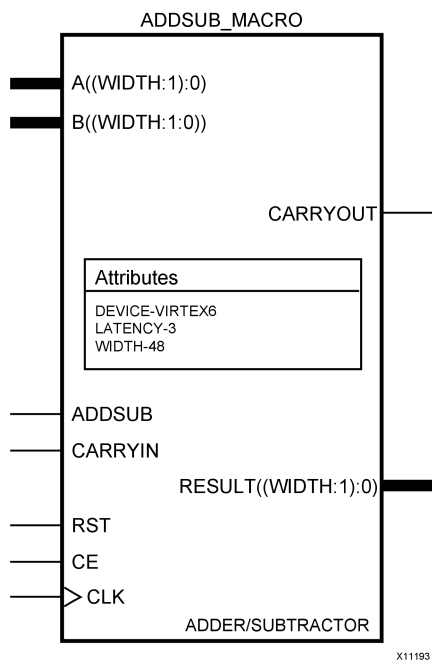
// End of ADDMACC_MACRO_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ADDSUB\_MACRO

マクロ : Adder/Subtractor



### 概要

ADDSUB\_MACRO を使用すると、DSP48 ブロックを単純な加減算器として使用する場合のインスタンス化が簡単になります。入力幅、出力幅、レイテンシを設定可能であり、DSP48 ブロックを HDL に統合しやすくなっています。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
CARRYOUT	出力	1	キャリー出力
RESULT	出力	可変 (WIDTH 属性を参照)	RDADDR で指定されるデータ出力バス
入力ポート			
ADDSUB	入力	1	High の場合は RESULT は加算結果で、Low の場合は減算結果です。
A	入力	可変 (WIDTH 属性を参照)	加算/減算のデータ入力
B	入力	可変 (WIDTH 属性を参照)	加算/減算のデータ入力
CE	入力	1	クロック イネーブル
CARRYIN	入力	1	キャリー入力
CLK	入力	1	クロック
RST	入力	1	同期リセット

## デザインの入力方法

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメータを設定できるようにしたものです。

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	推奨

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DEVICE	文字列	"VIRTEX6"、 "SPARTAN6"	"VIRTEX6"	ターゲットのハードウェア アーキテクチャを指定します。
LATENCY	整数	0、1、2	2	パイプライン レジスタの数を指定します。 ・ 1 : PREG == 1 ・ 2 : AREG == BREG == CREG == PREG
WIDTH	整数	1 ～ 48	48	A、B、RESULT ポートの幅を指定します。B および RESULT ポート幅は、ほかのパラメータを使用して変更できます。
WIDTH_RESULT	整数	1 ～ 48	48	WIDTH で設定された RESULT ポートの幅を変更します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ADDSUB_MACRO: Variable width & latency - Adder / Subtractor implemented in a DSP48E
--                Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

ADDSUB_MACRO_inst : ADDSUB_MACRO
generic map (
    DEVICE => "SPARTAN6", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    LATENCY => 2,          -- Desired clock cycle latency, 0-2
    WIDTH => 48)           -- Input / Output bus width, 1-48
port map (
    CARRYOUT => CARRYOUT, -- 1-bit carry-out output signal
    RESULT => RESULT,     -- Add/sub result output, width defined by WIDTH generic
    A => A,                -- Input A bus, width defined by WIDTH generic
    ADD_SUB => ADD_SUB,    -- 1-bit add/sub input, high selects add, low selects subtract
    B => B,                -- Input B bus, width defined by WIDTH generic
    CARRYIN => CARRYIN,    -- 1-bit carry-in input
    CE => CE,              -- 1-bit clock enable input
    CLK => CLK,            -- 1-bit clock input
    RST => RST             -- 1-bit active high synchronous reset
);
-- End of ADDSUB_MACRO_inst instantiation
```



## Verilog 記述 (インスタンスレーション)

```
// ADDSUB_MACRO: Variable width & latency - Adder / Subtractor implemented in a DSP48E
//                      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

ADDSUB_MACRO #(
    .DEVICE("SPARTAN6"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    .LATENCY(2),          // Desired clock cycle latency, 0-2
    .WIDTH(48)            // Input / output bus width, 1-48
) ADDSUB_MACRO_inst (
    .CARRYOUT(CARRYOUT), // 1-bit carry-out output signal
    .RESULT(RESULT),     // Add/sub result output, width defined by WIDTH parameter
    .A(A),               // Input A bus, width defined by WIDTH parameter
    .ADD_SUB(ADD_SUB),   // 1-bit add/sub input, high selects add, low selects subtract
    .B(B),               // Input B bus, width defined by WIDTH parameter
    .CARRYIN(CARRYIN),   // 1-bit carry-in input
    .CE(CE),              // 1-bit clock enable input
    .CLK(CLK),            // 1-bit clock input
    .RST(RST)             // 1-bit active high synchronous reset
);

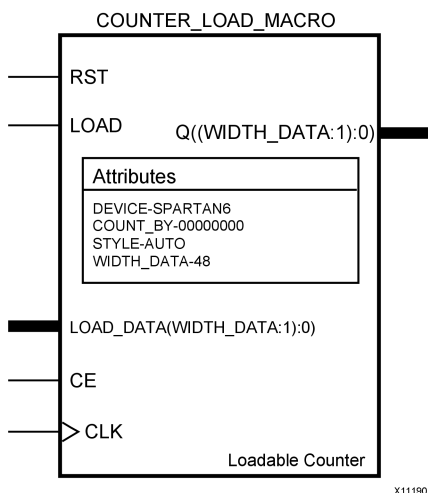
// End of ADDSUB_MACRO_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## COUNTER\_LOAD\_MACRO

### マクロ : Loadable Counter



### 概要

COUNTER\_LOAD\_MACRO を使用すると、DSP48 ブロックをダイナミック ロード アップ/ダウン カウンターとして使用する場合のインスタンス化が簡単になります。出力幅およびカウント値を指定可能であり、DSP48 ブロックを HDL に統合しやすくなっています。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
Q	出力	可変 (WIDTH_DATA 属性を参照)	カウンター出力
入力ポート			
CE	入力	1	クロック イネーブル
CLK	入力	1	クロック
LOAD	入力	可変 (WIDTH_DATA 属性を参照)	アサートすると、カウンターに LOAD_DATA の値が読み込まれます (2 クロックのレイテンシ)。
LOAD_DATA	入力	可変 (WIDTH_DATA 属性を参照)	DSP スライスでは、LOAD ピンをアサートすると、このデータが P レジスタに入力されます (2 クロックのレイテンシ)。
DIRECTION	入力	1	カウントの方向を指定します。High の場合はアップ、Low の場合はダウンです (2 クロックのレイテンシ)。
RST	入力	1	同期リセット

### デザインの入力方法

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメーターを設定できるようにしたものです。

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	推奨

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DEVICE	文字列	"VIRTEX6"、 "SPARTAN6"	"VIRTEX6"	ターゲットのハードウェア アーキテクチャを指定します。
COUNT_BY	16 進数	48 ビット値	000000000001	N ごとにカウントします。WIDTH_DATA より優先されます。
WIDTH_DATA	整数	1 ~ 48	48	カウンターの幅を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- COUNTER_LOAD_MACRO: Loadable variable counter implemented in a DSP48E
--                               Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

COUNTER_LOAD_MACRO_inst : COUNTER_LOAD_MACRO
generic map (
    COUNT_BY => X"00000000000001", -- Count by value
    DEVICE => "SPARTAN6",          -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    WIDTH_DATA => 48)              -- Counter output bus width, 1-48
port map (
    Q => Q,                        -- Counter output, width determined by WIDTH_DATA generic
    CLK => CLK,                    -- 1-bit clock input
    CE => CE,                      -- 1-bit clock enable input
    DIRECTION => DIRECTION,        -- 1-bit up/down count direction input, high is count up
    LOAD => LOAD,                  -- 1-bit active high load input
    LOAD_DATA => LOAD_DATA,        -- Counter load data, width determined by WIDTH_DATA generic
    RST => RST                     -- 1-bit active high synchronous reset
);
-- End of COUNTER_LOAD_MACRO_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// COUNTER_LOAD_MACRO: Loadable variable counter implemented in a DSP48E
//                               Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

COUNTER_LOAD_MACRO #(
    .COUNT_BY(48'h00000000000001), // Count by value
    .DEVICE("SPARTAN6"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    .WIDTH_DATA(48) // Counter output bus width, 1-48
) COUNTER_LOAD_MACRO_inst (
    .Q(Q), // Counter output, width determined by WIDTH_DATA parameter
    .CLK(CLK), // 1-bit clock input
    .CE(CE), // 1-bit clock enable input
    .DIRECTION(DIRECTION), // 1-bit up/down count direction input, high is count up
    .LOAD(LOAD), // 1-bit active high load input
    .LOAD_DATA(LOAD_DATA), // Counter load data, width determined by WIDTH_DATA parameter

```

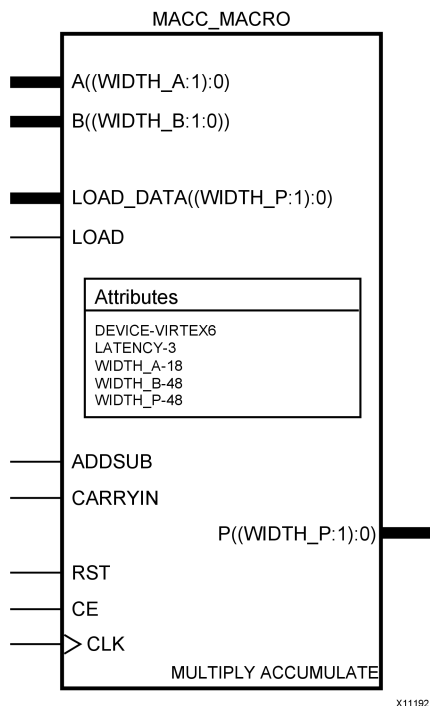
```
.RST(RST)           // 1-bit active high synchronous reset
);
// End of COUNTER_LOAD_MACRO_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MACC\_MACRO

マクロ : Multiplier/Accumulator



### 概要

MACC\_MACRO を使用すると、DSP48 ブロックを単純な符号付き乗算器/アキュムレータとして使用する場合はインスタネーションが簡単になります。入力幅、出力幅、レイテンシを指定可能であり、DSP48 ブロックを HDL に統合しやすくなっています。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
P	出力	可変 (WIDTH_A 属性値 + WIDTH_B 属性値)	プライマリ データ出力
入力ポート			
A	入力	可変 (WIDTH_A 属性を参照)	乗算データ入力
B	入力	可変 (WIDTH_B 属性を参照)	乗算データ入力
CARRYIN	入力	1	キャリー入力
CE	入力	1	クロック イネーブル
CLK	入力	1	クロック
LOAD	入力	1	ロード
LOAD_DATA	入力	可変 (WIDTH_A 属性値 + WIDTH_B 属性値)	DSP スライスでは、LOAD がアサートされると P に A*B+LOAD_DATA が読み込まれます。

ポート名	方向	幅	機能
RST	入力	1	同期リセット
ADDSUB	入力	1	High の場合はアキュムレータを加算モードに、Low の場合は減算モードに設定します。

## デザインの入力方法

この UniMacro はインスタンスエーションのみが可能で、プリミティブにパラメーターを設定できるようにしたものです。

インスタンスエーション	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	推奨

## 使用可能な属性

属性	データ型	値	デフォルト	説明
WIDTH_A	整数	1 ~ 18	18	A 入力の幅を指定します。
WIDTH_B	整数	1 ~ 18	18	B 入力の幅を指定します。
LATENCY	整数	0、1、2、3、4	3	パイプライン レジスタの数を指定します。 <ul style="list-style-type: none"> <li>1 : MREG == 1</li> <li>2 : AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1</li> <li>3 : AREG == BREG == 1 および MREG == 1 および PREG == 1</li> <li>4 : AREG == BREG == 2 および MREG == 1 および PREG == 1</li> </ul>
DEVICE	文字列	"VIRTEX5"、 "VIRTEX6"、 "SPARTAN6"	"VIRTEX6"	ターゲットのハードウェア アーキテクチャを指定します。

## VHDL 記述 (インスタンスエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MACC_MACRO: Multiple Accumulate Function implemented in a DSP48E
--             Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

MACC_MACRO_inst : MACC_MACRO
generic map (
  DEVICE => "SPARTAN6", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  LATENCY => 3,          -- Desired clock cycle latency, 1-4
  WIDTH_A => 25,         -- Multiplier A-input bus width, 1-25
  WIDTH_B => 18,         -- Multiplier B-input bus width, 1-18

```

```

    WIDTH_P => 48)          -- Accumulator output bus width, 1-48
port map (
    P => P,                -- MACC output bus, width determined by WIDTH_P generic
    A => A,                -- MACC input A bus, width determined by WIDTH_A generic
    ADDSUB => ADDSUB,      -- 1-bit add/sub input, high selects add, low selects subtract
    B => B,                -- MACC input B bus, width determined by WIDTH_B generic
    CARRYIN => CARRYIN,    -- 1-bit carry-in input to accumulator
    CE => CE,              -- 1-bit active high input clock enable
    CLK => CLK,            -- 1-bit positive edge clock input
    LOAD => LOAD,          -- 1-bit active high input load accumulator enable
    LOAD_DATA => LOAD_DATA, -- Load accumulator input data,
                          -- width determined by WIDTH_P generic
    RST => RST             -- 1-bit input active high reset
);

-- End of MACC_MACRO_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// MACC_MACRO: Multiply Accumulate Function implemented in a DSP48E
//              Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

MACC_MACRO #(
    .DEVICE("SPARTAN6"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    .LATENCY(3),         // Desired clock cycle latency, 1-4
    .WIDTH_A(18),        // Multiplier A-input bus width, 1-18
    .WIDTH_B(18),        // Multiplier B-input bus width, 1-18
    .WIDTH_P(48)         // Accumulator output bus width, 1-48
) MACC_MACRO_inst (
    .P(P),               // MACC output bus, width determined by WIDTH_P parameter
    .A(A),               // MACC input A bus, width determined by WIDTH_A parameter
    .ADDSUB(ADDSUB),     // 1-bit add/sub input, high selects add, low selects subtract
    .B(B),               // MACC input B bus, width determined by WIDTH_B parameter
    .CARRYIN(CARRYIN),   // 1-bit carry-in input to accumulator
    .CE(CE),             // 1-bit active high input clock enable
    .CLK(CLK),           // 1-bit positive edge clock input
    .LOAD(LOAD),         // 1-bit active high input load accumulator enable
    .LOAD_DATA(LOAD_DATA), // Load accumulator input data, width determined by WIDTH_P parameter
    .RST(RST)            // 1-bit input active high reset
);

// End of MACC_MACRO_inst instantiation

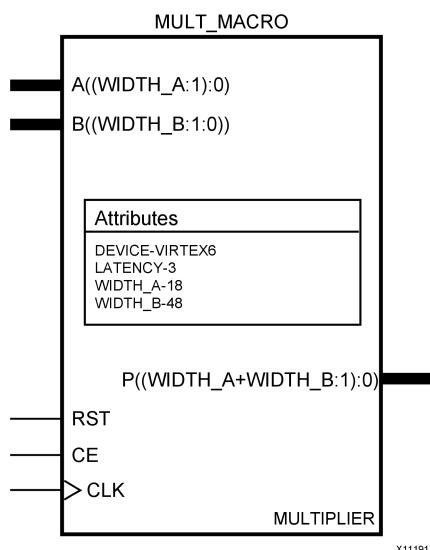
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MULT\_MACRO

### マクロ : Multiplier



### 概要

MULT\_MACRO を使用すると、DSP48 ブロックを単純な符号付き乗算器として使用する場合のインスタンス化が簡単になります。入力幅、出力幅、およびレイテンシを指定可能であり、DSP48 ブロックを HDL に統合しやすくなっています。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
P	出力	可変 (WIDTH_A 属性値 + WIDTH_B 属性値)	プライマリ データ出力
入力ポート			
A	入力	可変 (WIDTH_A 属性を参照)	乗算データ入力
B	入力	可変 (WIDTH_B 属性を参照)	乗算データ入力
CE	入力	1	クロック イネーブル
CLK	入力	1	クロック
RST	入力	1	同期リセット

### デザインの入力方法

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメーターを設定できるようにしたものです。



インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	推奨

## 使用可能な属性

属性	データ型	値	デフォルト	説明
WIDTH_A	整数	1 ~ 18	18	A 入力の幅を指定します。
WIDTH_B	整数	1 ~ 18	18	B 入力の幅を指定します。
LATENCY	整数	0、1、2、3、4	3	パイプライン レジスタの数を指定します。 <ul style="list-style-type: none"> <li>1 : MREG == 1</li> <li>2 : AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1</li> <li>3 : AREG == BREG == 1 および MREG == 1 および PREG == 1</li> <li>4 : AREG == BREG == 2 および MREG == 1 および PREG == 1</li> </ul>
DEVICE	文字列	"VIRTEX5"、 "VIRTEX6"、 "SPARTAN6"	"VIRTEX6"	ターゲットのハードウェア アーキテクチャを指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- MULT_MACRO: Multiply Function implemented in a DSP48E
--               Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

MULT_MACRO_inst : MULT_MACRO
generic map (
  DEVICE => "SPARTAN6",    -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  LATENCY => 3,             -- Desired clock cycle latency, 0-4
  WIDTH_A => 18,            -- Multiplier A-input bus width, 1-25
  WIDTH_B => 18)            -- Multiplier B-input bus width, 1-18
port map (
  P => P,                   -- Multiplier output bus, width determined by WIDTH_P generic
  A => A,                   -- Multiplier input A bus, width determined by WIDTH_A generic
  B => B,                   -- Multiplier input B bus, width determined by WIDTH_B generic
  CE => CE,                -- 1-bit active high input clock enable
  CLK => CLK,              -- 1-bit positive edge clock input
  RST => RST               -- 1-bit input active high reset
);
-- End of MULT_MACRO_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// MULT_MACRO: Multiply Function implemented in a DSP48E
//               Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

MULT_MACRO #(
    .DEVICE("SPARTAN6"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    .LATENCY(3),          // Desired clock cycle latency, 0-4
    .WIDTH_A(18),         // Multiplier A-input bus width, 1-18
    .WIDTH_B(18)          // Multiplier B-input bus width, 1-18
) MULT_MACRO_inst (
    .P(P),                // Multiplier output bus, width determined by WIDTH_P parameter
    .A(A),                // Multiplier input A bus, width determined by WIDTH_A parameter
    .B(B),                // Multiplier input B bus, width determined by WIDTH_B parameter
    .CE(CE),              // 1-bit active high input clock enable
    .CLK(CLK),            // 1-bit positive edge clock input
    .RST(RST)             // 1-bit input active high reset
);

// End of MULT_MACRO_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ファンクション別分類

このセクションでは、デバイスに含まれるデザイン エLEMENTをファンクション別に分類して示します。ELEMENT (プリミティブおよびマクロのインプリメンテーション) は、各カテゴリでアルファベット順にリストしています。

アドバンス	コンフィギュレーション/BSCAN	レジスタおよびラッチ
	コンポーネント	
演算ファンクション	I/O コンポーネント	スライス/CLB プリミティブ
クロック コンポーネント	RAM/ROM	

### アドバンス

デザイン エLEMENT	説明
MCB	プリミティブ : Memory Control Block
PCIE_A1	プリミティブ : PCI Express

### 演算ファンクション

デザイン エLEMENT	説明
DSP48A1	プリミティブ : Multi-Functional, Cascadable, 48-bit Output, Arithmetic Block

### クロック コンポーネント

デザイン エLEMENT	説明
BUFG	プリミティブ : Global Clock Buffer
BUFGCE	プリミティブ : Global Clock Buffer with Clock Enable
BUFGCE_1	プリミティブ : Global Clock Buffer with Clock Enable and Output State 1
BUFGMUX	プリミティブ : Global Clock MUX Buffer
BUFGMUX_1	プリミティブ : Global Clock MUX Buffer with Output State 1
BUFH	プリミティブ : Clock buffer for a single clocking region

デザイン エLEMENT	説明
BUFIO2	プリミティブ : Dual Clock Buffer and Strobe Pulse
BUFIO2_2CLK	プリミティブ : Dual Clock Buffer and Strobe Pulse with Differential Input
BUFIO2FB	プリミティブ : Feedback Clock Buffer.
BUFPLL	プリミティブ : PLL Buffer
BUFPLL_MCB	プリミティブ : PLL Buffer for the Memory Controller Block
DCM_CLKGEN	プリミティブ : Digital Clock Manager.
DCM_SP	プリミティブ : Digital Clock Manager
PLL_BASE	プリミティブ : Basic Phase Locked Loop Clock Circuit

### コンフィギュレーション/BSCAN コンポーネント

デザイン エLEMENT	説明
BSCAN_SPARTAN6	プリミティブ : Spartan®-6 JTAG Boundary Scan Logic Control Circuit
DNA_PORT	プリミティブ : Device DNA Data Access Port
ICAP_SPARTAN6	プリミティブ : Internal Configuration Access Port
JTAG_SIM_SPARTAN6	シミュレーション : JTAG TAP Controller Simulation Model
POST_CRC_INTERNAL	プリミティブ : Post-configuration CRC error detection
SIM_CONFIG_S6	シミュレーション : Configuration Simulation Model
SIM_CONFIG_S6_SERIAL	シミュレーション : Serial Configuration Simulation Model
STARTUP_SPARTAN6	プリミティブ : Spartan®-6 Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface
SUSPEND_SYNC	プリミティブ : Suspend Mode Access

### I/O コンポーネント

デザイン エLEMENT	説明
GTPA1_DUAL	プリミティブ : Dual Gigabit Transceiver
IBUF	プリミティブ : Input Buffer
IBUFDS	プリミティブ : Differential Signaling Input Buffer
IBUFDS_DIFF_OUT	プリミティブ : Signaling Input Buffer with Differential Output
IBUFG	プリミティブ : Dedicated Input Clock Buffer
IBUFGDS	プリミティブ : Differential Signaling Dedicated Input Clock Buffer and Optional Delay
IBUFGDS_DIFF_OUT	プリミティブ : Differential Signaling Input Buffer with Differential Output
IOBUF	プリミティブ : Bi-Directional Buffer
IOBUFDS	プリミティブ : 3-State Differential Signaling I/O Buffer with Active Low Output Enable
IDELAY2	プリミティブ : Input and Output Fixed or Variable Delay Element

デザイン エLEMENT	説明
IODRP2	プリミティブ : I/O Control Port
IODRP2_MCB	プリミティブ : I/O Control Port for the Memory Controller Block
ISERDES2	プリミティブ : Input SERIAL/DESerializer.
KEEPER	プリミティブ : KEEPER Symbol
OBUF	プリミティブ : Output Buffer
OBUFDS	プリミティブ : Differential Signaling Output Buffer
OBUFT	プリミティブ : 3-State Output Buffer with Active Low Output Enable
OBUFTDS	プリミティブ : 3-State Output Buffer with Differential Signaling, Active-Low Output Enable
OSERDES2	プリミティブ : Dedicated IOB Output Serializer
PULLDOWN	プリミティブ : Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs
PULLUP	プリミティブ : Resistor to VCC for Input PADS, Open-Drain, and 3-State Outputs

## RAM/ROM

デザイン エLEMENT	説明
RAM128X1D	プリミティブ : 128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)
RAM256X1S	プリミティブ : 256-Deep by 1-Wide Random Access Memory (Select RAM)
RAM32M	プリミティブ : 32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)
RAM32X1D	プリミティブ : 32-Deep by 1-Wide Static Dual Port Synchronous RAM
RAM32X1S	プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM
RAM32X1S_1	プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAM32X2S	プリミティブ : 32-Deep by 2-Wide Static Synchronous RAM
RAM64M	プリミティブ : 64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)
RAM64X1D	プリミティブ : 64-Deep by 1-Wide Dual Port Static Synchronous RAM
RAM64X1S	プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM
RAM64X1S_1	プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAMB16BWER	プリミティブ : 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers

デザイン エLEMENT	説明
RAMB8BWER	プリミティブ : 8K-bit Data and 1K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers
ROM128X1	プリミティブ : 128-Deep by 1-Wide ROM
ROM256X1	プリミティブ : 256-Deep by 1-Wide ROM
ROM32X1	プリミティブ : 32-Deep by 1-Wide ROM
ROM64X1	プリミティブ : 64-Deep by 1-Wide ROM

## レジスタおよびラッチ

デザイン エLEMENT	説明
FDCE	プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Clear
FDPE	プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Preset
FDRE	プリミティブ : D Flip-Flop with Clock Enable and Synchronous Reset
FDSE	プリミティブ : D Flip-Flop with Clock Enable and Synchronous Set
IDDR2	プリミティブ : Double Data Rate Input D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset
LDCE	プリミティブ : Transparent Data Latch with Asynchronous Clear and Gate Enable
LDPE	プリミティブ : Transparent Data Latch with Asynchronous Preset and Gate Enable
ODDR2	プリミティブ : Dual Data Rate Output D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset

## スライス/CLB プリミティブ

デザイン エLEMENT	説明
CARRY4	プリミティブ : Fast Carry Logic with Look Ahead
CFGLUT5	プリミティブ : 5-input Dynamically Reconfigurable Look-Up Table (LUT)
LUT1	マクロ : 1-Bit Look-Up Table with General Output
LUT1_D	マクロ : 1-Bit Look-Up Table with Dual Output
LUT1_L	マクロ : 1-Bit Look-Up Table with Local Output
LUT2	マクロ : 2-Bit Look-Up Table with General Output
LUT2_D	マクロ : 2-Bit Look-Up Table with Dual Output
LUT2_L	マクロ : 2-Bit Look-Up Table with Local Output
LUT3	マクロ : 3-Bit Look-Up Table with General Output
LUT3_D	マクロ : 3-Bit Look-Up Table with Dual Output

デザイン エlement	説明
LUT3_L	マクロ : 3-Bit Look-Up Table with Local Output
LUT4	マクロ : 4-Bit Look-Up-Table with General Output
LUT4_D	マクロ : 4-Bit Look-Up Table with Dual Output
LUT4_L	マクロ : 4-Bit Look-Up Table with Local Output
LUT5	プリミティブ : 5-Input Lookup Table with General Output
LUT5_D	プリミティブ : 5-Input Lookup Table with General and Local Outputs
LUT5_L	プリミティブ : 5-Input Lookup Table with Local Output
LUT6	プリミティブ : 6-Input Lookup Table with General Output
LUT6_2	プリミティブ : Six-input, 2-output, Look-Up Table
LUT6_D	プリミティブ : 6-Input Lookup Table with General and Local Outputs
LUT6_L	プリミティブ : 6-Input Lookup Table with Local Output
MUXF7	プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output
MUXF7_D	プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF7_L	プリミティブ : 2-to-1 look-up table Multiplexer with Local Output
MUXF8	プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output
MUXF8_D	プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF8_L	プリミティブ : 2-to-1 Look-Up Table Multiplexer with Local Output
SRL16E	プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Clock Enable
SRLC32E	プリミティブ : 32 Clock Cycle, Variable Length Shift Register Look-Up Table (LUT) with Clock Enable





## デザイン エLEMENT

---

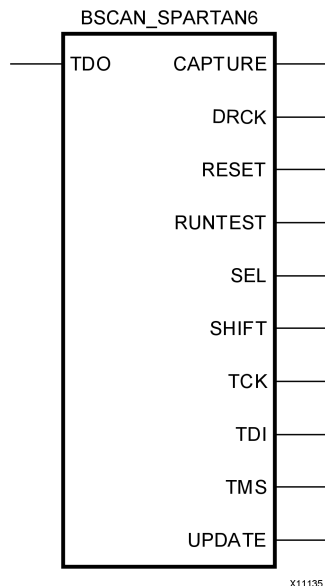
このセクションでは、Spartan®-6 デバイスで利用できるデザイン エLEMENTについて説明します。デザイン エLEMENTは、アルファベット順に並べられています。

各ライブラリ エLEMENTについて、次の情報を示します。

- ・ 名称
- ・ 説明
- ・ 回路図シンボル (該当するELEMENTでのみ)
- ・ 論理表 (該当するELEMENTでのみ)
- ・ ポートの説明
- ・ デザインの入力方法
- ・ 使用可能な属性 (該当するELEMENTでのみ)
- ・ インスタンス化コードの例
- ・ その他のリソース

## BSCAN\_SPARTAN6

### プリミティブ：Spartan®-6 JTAG Boundary Scan Logic Control Circuit



### 概要

このデザイン エLEMENTを使用すると、JTAG バウンダリ スキャン ロジック コントローラーを介して内部ロジックへアクセスにアクセスできるようになり、内部実行デザインと FPGA の専用 JTAG ピン間の通信が可能になります。

このデザイン エLEMENTの各インスタンスでは、JTAG\_CHAIN 属性の設定に従い、JTAG USER 命令 1 つ (USER1 から USER4 まで) が処理されます。4 つの USER 命令すべてを処理するには、ELEMENTを 4 つインスタンス化して JTAG\_CHAIN 属性を設定します。

**注記：** 各アーキテクチャのバウンダリ スキャンの詳細は、『Spartan-6 FPGA コンフィギュレーション ユーザー ガイド』を参照してください。

### ポートの説明

ポート名	方向	幅	機能
CAPTURE	出力	1	TAP コントローラーの CAPTURE 出力
DRCK	出力	1	USER ファンクションのデータレジスタ出力
RESET	出力	1	TAP コントローラーのリセット出力
RUNTEST	出力	1	TAP コントローラーが Run Test Idle ステートのときにアサートされる出力信号
SEL	出力	1	USER アクティブ出力
SHIFT	出力	1	TAP コントローラーの SHIFT 出力
TCK	出力	1	スキャン クロック出力。TAP クロック ピンへのファブリック接続。
TDI	出力	1	TAP コントローラーの TDI 出力
TDO	入力	1	USER ファンクションのデータ入力
TMS	出力	1	テスト モード セレクト出力。TAP へのファブリック接続。
UPDATE	出力	1	TAP コントローラーの UPDATE 出力

## デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
JTAG_CHAIN	整数	1、2、3、4	1	エレメントのインスタンスで処理可能な JTAG USER 命令数を設定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BSCAN_SPARTAN6: JTAG Boundary Scan Logic Control Circuit
--                 Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BSCAN_SPARTAN6_inst : BSCAN_SPARTAN6
generic map (
    JTAG_CHAIN => 1 -- Value for USER command. Possible values: (1,2,3 or 4).
)
port map (
    CAPTURE => CAPTURE, -- 1-bit output: CAPTURE output from TAP controller.
    DRCK => DRCK,       -- 1-bit output: Data register output for USER functions.
    RESET => RESET,     -- 1-bit output: Reset output for TAP controller.
    RUNTEST => RUNTEST, -- 1-bit output: Output signal that gets asserted when TAP controller is in Run Test
                        -- Idle state.

    SEL => SEL,         -- 1-bit output: USER active output.
    SHIFT => SHIFT,     -- 1-bit output: SHIFT output from TAP controller.
    TCK => TCK,         -- 1-bit output: Scan Clock output. Fabric connection to TAP Clock pin.
    TDI => TDI,         -- 1-bit output: TDI output from TAP controller.
    TMS => TMS,         -- 1-bit output: Test Mode Select output. Fabric connection to TAP.
    UPDATE => UPDATE,   -- 1-bit output: UPDATE output from TAP controller
    TDO => TDO          -- 1-bit input: Data input for USER function.
);

-- End of BSCAN_SPARTAN6_inst instantiation
```

## Verilog 記述 (インスタンスーション)

```
// BSCAN_SPARTAN6: JTAG Boundary Scan Logic Control Circuit
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BSCAN_SPARTAN6 #(
    .JTAG_CHAIN(1) // Value for USER command. Possible values: (1,2,3 or 4).
)
BSCAN_SPARTAN6_inst (
    .CAPTURE(CAPTURE), // 1-bit output: CAPTURE output from TAP controller.
    .DRCK(DRCK),       // 1-bit output: Data register output for USER functions.
    .RESET(RESET),     // 1-bit output: Reset output for TAP controller.
    .RUNTEST(RUNTEST), // 1-bit output: Output signal that gets asserted when TAP controller is in Run Test
                      // Idle state.

    .SEL(SEL),         // 1-bit output: USER active output.
    .SHIFT(SHIFT),     // 1-bit output: SHIFT output from TAP controller.
    .TCK(TCK),         // 1-bit output: Scan Clock output. Fabric connection to TAP Clock pin.
    .TDI(TDI),         // 1-bit output: TDI output from TAP controller.
    .TMS(TMS),         // 1-bit output: Test Mode Select output. Fabric connection to TAP.
    .UPDATE(UPDATE),   // 1-bit output: UPDATE output from TAP controller
    .TDO(TDO)          // 1-bit input: Data input for USER function.
);

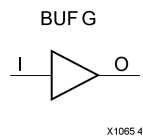
// End of BSCAN_SPARTAN6_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュレーション ユーザー ガイド』\(UG380\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## BUFG

### プリミティブ：Global Clock Buffer



### 概要

このデザイン エLEMENTはファンアウトが大きいバッファで、スキューを抑えて信号を分散するために、グローバル配線リソースへの信号に接続します。BUFG は、通常セット/リセットやクロック イネーブルなどのファンアウトの大きいネットやクロック ネットに使用されます。

### ポートの説明

ポート名	方向	幅	機能
I	入力	1	クロック バッファ入力
O	出力	1	クロック バッファ出力

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFG: Global Clock Buffer
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BUFG_inst : BUFG
port map (
  O => O, -- 1-bit output: Clock buffer output
  I => I  -- 1-bit input: Clock buffer input
);

-- End of BUFG_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFG: Global Clock Buffer
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BUFG BUFG_inst (
    .O(O), // 1-bit output: Clock buffer output
    .I(I)  // 1-bit input: Clock buffer input
);

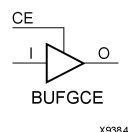
// End of BUFG_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA クロックリソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## BUFGCE

プリミティブ：Global Clock Buffer with Clock Enable



### 概要

このデザイン エLEMENTは、クロック イネーブル付きグローバル クロック バッファです。O 出力は、クロック イネーブル (CE) が Low (非アクティブ) のときに 0 になります。CE が High になると、I 入力の値が O に出力されます。

### 論理表

入力		出力
I	CE	O
X	0	0
I	1	I

### ポートの説明

ポート名	方向	幅	機能
I	入力	1	クロック バッファ入力
CE	入力	1	クロック イネーブル入力
O	出力	1	クロック バッファ出力

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGCE: Global Clock Buffer with Clock Enable
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BUFGCE_inst : BUFGCE
port map (
  O => O,  -- 1-bit output: Clock buffer output
  CE => CE, -- 1-bit input: Clock buffer select
  I => I   -- 1-bit input: Clock buffer input (S=0)
);

-- End of BUFGCE_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGCE: Global Clock Buffer with Clock Enable
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BUFGCE BUFGCE_inst (
  .O(O),    // 1-bit output: Clock buffer output
  .CE(CE),  // 1-bit input: Clock buffer select
  .I(I)     // 1-bit input: Clock buffer input (S=0)
);

// End of BUFGCE_inst instantiation
```

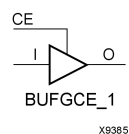
## 詳細情報

- ・ [『Spartan-6 FPGA クロックリソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## BUFGCE\_1

プリミティブ：Global Clock Buffer with Clock Enable and Output State 1



### 概要

このデザイン エLEMENTは、クロック イネーブル付きグローバル クロック バッファです。O 出力は、クロック イネーブル (CE) が Low (非アクティブ) のときに High (1) になります。CE が High になると、I 入力の値が O に出力されます。

### 論理表

入力		出力
I	CE	O
X	0	1
I	1	I

### ポートの説明

ポート名	方向	幅	機能
I	入力	1	クロック バッファ入力
CE	入力	1	クロック イネーブル入力
O	出力	1	クロック バッファ出力

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGCE_1: Global Clock Buffer with Clock Enable and Output State 1
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BUFGCE_1_inst : BUFGCE_1
port map (
    O => O,  -- 1-bit output: Clock buffer output
    CE => CE, -- 1-bit input: Clock buffer select
    I => I   -- 1-bit input: Clock buffer input (S=0)
);

-- End of BUFGCE_1_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFGCE_1: Global Clock Buffer with Clock Enable and Output State 1
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BUFGCE_1 BUFGCE_1_inst (
    .O(O),    // 1-bit output: Clock buffer output
    .CE(CE),  // 1-bit input: Clock buffer select
    .I(I)     // 1-bit input: Clock buffer input (S=0)
);

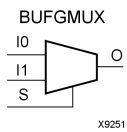
// End of BUFGCE_1_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA クロックリソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## BUFGMUX

### プリミティブ：Global Clock MUX Buffer



### 概要

BUFGMUX はマルチプレクサーの機能を持つグローバル クロック バッファで、2 つの入力クロック (I0 および I1) のいずれかを選択できます。セレクト入力 (S) が Low の場合、I0 の信号が出力 (O) に選択されます。S が High の場合は、I1 の信号が O に選択されます。

BUFGMUX および BUFGMUX\_1 では、S の値が変化した後クロックが切り替わるまでに保持される出力ステートが異なります。BUFGMUX は出力ステートが 0 に、BUFGMUX\_1 は出力ステートが 1 に保持されます。

**注記：** BUFGMUX では、S がトグルされると、次のアクティブ クロック エッジ (I0 または I1) まで、出力のステートが非アクティブのまま保持されます。

### 論理表

入力			出力
I0	I1	S	O
I0	X	0	I0
X	I1	1	I1
X	X	↑	0
X	X	↓	0

### ポートの説明

ポート名	方向	幅	機能
I0	入力	1	クロック 0 入力
I1	入力	1	クロック 1 入力
O	出力	1	クロック MUX 出力
S	入力	1	クロック セレクト入力

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
CLK_SEL_TYPE	文字列	"SYNC"、"ASYNC"	"SYNC"	同期クロックまたは非同期クロックを指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGMUX: Global Clock Mux Buffer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BUFGMUX_inst : BUFGMUX
generic map (
  CLK_SEL_TYPE => "SYNC" -- Glitchless ("SYNC") or fast ("ASYNC") clock switch-over
)
port map (
  O => O, -- 1-bit output: Clock buffer output
  IO => IO, -- 1-bit input: Clock buffer input (S=0)
  I1 => I1, -- 1-bit input: Clock buffer input (S=1)
  S => S -- 1-bit input: Clock buffer select
);

-- End of BUFGMUX_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFGMUX: Global Clock Mux Buffer
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BUFGMUX #(
  .CLK_SEL_TYPE("SYNC") // Glitchless ("SYNC") or fast ("ASYNC") clock switch-over
)
BUFGMUX_inst (
  .O(O), // 1-bit output: Clock buffer output
  .IO(IO), // 1-bit input: Clock buffer input (S=0)
  .I1(I1), // 1-bit input: Clock buffer input (S=1)
  .S(S) // 1-bit input: Clock buffer select
);

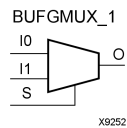
// End of BUFGMUX_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA クロック リソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## BUFGMUX\_1

### プリミティブ：Global Clock MUX Buffer with Output State 1



### 概要

このデザイン エLEMENTは、マルチプレクサーの機能を持つグローバル クロック バッファーで、2 つの入力クロック (I0 および I1) のいずれかを選択できます。セレクト入力 (S) が Low の場合、I0 の信号が出力 (O) に選択されます。S が High の場合は、I1 の信号が O に選択されます。

このデザイン エLEMENTと BUFGMUX では、S の値が変化した後クロックが切り替わるまでに保持される出力ステートが異なります。BUFGMUX は出力ステートが 0 に、BUFGMUX\_1 は出力ステートが 1 に保持されます。

### 論理表

入力			出力
I0	I1	S	O
I0	X	0	I0
X	I1	1	I1
X	X	↑	1
X	X	↓	1

### ポートの説明

ポート名	方向	幅	機能
I0	入力	1	クロック 0 入力
I1	入力	1	クロック 1 入力
O	出力	1	クロック MUX 出力
S	入力	1	クロック セレクト入力

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGMUX_1: Global Clock Mux Buffer with Output State 1
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BUFGMUX_1_inst : BUFGMUX_1
generic map (
    CLK_SEL_TYPE => "SYNC"  -- Glitchles ("SYNC") or fast ("ASYNC") clock switch-over
)
port map (
    O => O,  -- 1-bit output: Clock buffer output
    IO => IO, -- 1-bit input: Clock buffer input
    I1 => I1, -- 1-bit input: Clock buffer input
    S => S   -- 1-bit input: Clock buffer select
);

-- End of BUFGMUX_1_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFGMUX_1: Global Clock Mux Buffer with Output State 1
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BUFGMUX_1 #(
    .CLK_SEL_TYPE("SYNC") // Glitchles ("SYNC") or fast ("ASYNC") clock switch-over
)
BUFGMUX_1_inst (
    .O(O), // 1-bit output: Clock buffer output
    .IO(IO), // 1-bit input: Clock buffer input
    .I1(I1), // 1-bit input: Clock buffer input
    .S(S) // 1-bit input: Clock buffer select
);

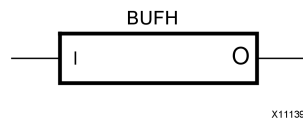
// End of BUFGMUX_1_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA クロックリソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## BUFH

プリミティブ：Clock buffer for a single clocking region



### 概要

BUFH プリミティブは、インスタンス化で HCLK クロック バッファ リソースにアクセスできるようにします。このコンポーネントは手動で配置する必要があり、また特別な考慮が必要なため、アドバンス ユーザー向けです。このコンポーネントの詳細は、『[Spartan-6 FPGA クロック リソース ユーザー ガイド](#)』(UG382) を参照してください。

### ポートの説明

ポート名	方向	幅	機能
I	入力	1	クロック入力
O	出力	1	クロック出力

### デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFH: HROW Clock Buffer for a Single Clocking Region
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BUFH_inst : BUFH
port map (
  O => O, -- 1-bit output: Clock output
  I => I  -- 1-bit input: Clock input
);

-- End of BUFH_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFH: HROW Clock Buffer for a Single Clocking Region
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BUFH BUFH_inst (
    .O(O), // 1-bit output: Clock output
    .I(I)  // 1-bit input: Clock input
);

// End of BUFH_inst instantiation
```

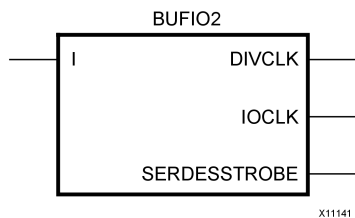
## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## BUFIO2

### プリミティブ：Dual Clock Buffer and Strobe Pulse



### 概要

このプリミティブは、同期 I/O リソース (ISERDES2、OSERDES2) および関連のファブリック リソースをスキューが小さい BUFG を介して駆動するため、オフチップ ソースからの高速 I/O クロック リソースを供給します。このコンポーネントの詳細は、『[Spartan-6 FPGA クロック リソース ユーザー ガイド](#)』(UG382) を参照してください。

### ポートの説明

ポート名	方向	幅	機能
DIVCLK	出力	1	分周されたクロック出力
I	入力	1	クロック入力
IOCLK	出力	1	クロック出力
SERDESSTROBE	出力	1	出力 SERDES ストロブ (ISERDES/OSERDES に接続)

### デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
DIVIDE	10 進数	1、2、3、4、5、6、7、8	1	DIVCLK の分周比を指定します。
DIVIDE_BYPASS	ブール 代数	TRUE、FALSE	TRUE	分周回路をバイパスします。
I_INVERT	ブール 代数	FALSE、TRUE	FALSE	クロックを反転します。
USE_DOUBLER	ブール 代数	FALSE、TRUE	FALSE	2 通倍回路を使用します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFIO2: I/O Clock Buffer
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BUFIO2_inst : BUFIO2
generic map (
    DIVIDE => 1,           -- DIVCLK divider (1-8)
    DIVIDE_BYPASS => TRUE, -- Bypass the divider circuitry (TRUE/FALSE)
    I_INVERT => FALSE,     -- Invert clock (TRUE/FALSE)
    USE_DOUBLER => FALSE   -- Use doubler circuitry (TRUE/FALSE)
)
port map (
    DIVCLK => DIVCLK,      -- 1-bit output: Divided clock output
    IOCLK => IOCLK,        -- 1-bit output: I/O output clock
    SERDESSTROBE => SERDESSTROBE, -- 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
    I => I                 -- 1-bit input: Clock input (connect to IBUFG)
);

-- End of BUFIO2_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFIO2: I/O Clock Buffer
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BUFIO2 #(
    .DIVIDE(1),           // DIVCLK divider (1-8)
    .DIVIDE_BYPASS("TRUE"), // Bypass the divider circuitry (TRUE/FALSE)
    .I_INVERT("FALSE"),   // Invert clock (TRUE/FALSE)
    .USE_DOUBLER("FALSE") // Use doubler circuitry (TRUE/FALSE)
)
BUFIO2_inst (
    .DIVCLK(DIVCLK),      // 1-bit output: Divided clock output
    .IOCLK(IOCLK),        // 1-bit output: I/O output clock
    .SERDESSTROBE(SERDESSTROBE), // 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
    .I(I)                 // 1-bit input: Clock input (connect to IBUFG)
);

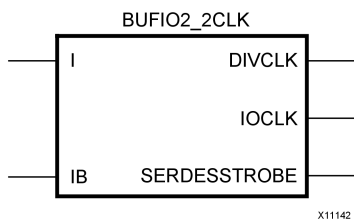
// End of BUFIO2_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA クロックリソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## BUFIO2\_2CLK

プリミティブ：Dual Clock Buffer and Strobe Pulse with Differential Input



### 概要

BUFIO2\_2CLK は、同期 I/O リソース (ISERDES2、OSERDES2) および関連のファブリック リソースをスキューが小さい BUFG を介して駆動するため、オフチップ ソースからの高速 I/O クロック リソースを供給します。このコンポーネントの詳細は、[『Spartan-6 FPGA クロック リソース ユーザー ガイド』\(UG382\)](#) を参照してください。

### デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	可
マクロのサポート	不可

### VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFIO2_2CLK: Dual Input Differential Clock Buffer
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BUFIO2_2CLK_inst : BUFIO2_2CLK
generic map (
  DIVIDE => 2 -- DIVCLK divider (2-8)
)
port map (
  DIVCLK => DIVCLK,      -- 1-bit output: Divided clock output
  IOCLK => IOCLK,        -- 1-bit output: I/O output clock
  SERDESSTROBE => SERDESSTROBE, -- 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
  I => I,                -- 1-bit input: Clock input (connect to IBUFG)
  IB => IB               -- 1-bit input: Secondary clock input
);

-- End of BUFIO2_2CLK_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFIO2_2CLK: Dual Input Differential Clock Buffer
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BUFIO2_2CLK #(
    .DIVIDE(2) // DIVCLK divider (2-8)
)
BUFIO2_2CLK_inst (
    .DIVCLK(DIVCLK),           // 1-bit output: Divided clock output
    .IOCLK(IOCLK),             // 1-bit output: I/O output clock
    .SERDESSTROBE(SERDESSTROBE), // 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
    .I(I),                     // 1-bit input: Clock input (connect to IBUFG)
    .IB(IB)                     // 1-bit input: Secondary clock input
);

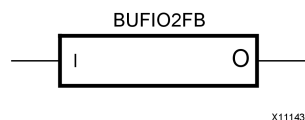
// End of BUFIO2_2CLK_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA クロック リソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## BUFIO2FB

**プリミティブ：Feedback Clock Buffer**



X11143

### 概要

このエレメントは、関連付けられている BUFIO2 と遅延が同じ単純なバッファで、DLL または PLL を使用する際にフィードバックの位相を適切に調整できるようフィードバック パスで使用されます。

### ポートの説明

ポート名	方向	幅	機能
I	入力	1	入力フィードバック クロック
O	出力	1	出力フィードバック クロック (DCM/PLL のフィードバック入力に接続)

### デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
DIVIDE_BYPASS	ブール代数	TRUE、FALSE	TRUE	分周回路をバイパスします。関連付けられている BUFIO2 と同じ値に設定します。

### VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFIO2FB: DCM/PLL Feedback Clock Buffer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BUFIO2FB_inst : BUFIO2FB
generic map (
    DIVIDE_BYPASS => TRUE -- Bypass divider (TRUE/FALSE)
)
port map (
    O => O, -- 1-bit output: Output feedback clock (connect to feedback input of DCM/PLL)
    I => I -- 1-bit input: Feedback clock input (connect to input port)
);

-- End of BUFIO2FB_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFIO2FB: DCM/PLL Feedback Clock Buffer
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BUFIO2FB #(
    .DIVIDE_BYPASS("TRUE") // Bypass divider (TRUE/FALSE)
)
BUFIO2FB_inst (
    .O(0), // 1-bit output: Output feedback clock (connect to feedback input of DCM/PLL)
    .I(I) // 1-bit input: Feedback clock input (connect to input port)
);

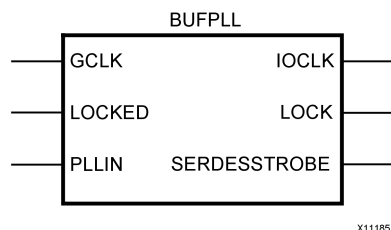
// End of BUFIO2FB_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA クロックリソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## BUFPLL

プリミティブ：PLL Buffer



### 概要

PLL コンポーネントからクロックが供給される高速 I/O クロック バッファです。

### ポートの説明

ポート名	方向	幅	機能
GCLK	入力	1	BUFG クロック入力
IOCLK	出力	1	出力 I/O クロック
LOCK	出力	1	同期化された LOCK 出力
LOCKED	入力	1	PLL からの LOCKED 入力
PLLIN	入力	1	PLL からのクロック入力
SERDESSTROBE	出力	1	SERDES ストロブ (ISERDES/OSERDES に接続)

### デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
DIVIDE	整数	1、2、3、4、5、6、7、8	1	分周比 (1 ~ 8) を指定します。
ENABLE_SYNC	ブール代数	TRUE、FALSE	TRUE	PLL と GCLK 間の同期をイネーブルにします。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFPLL: High-speed I/O PLL clock buffer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

BUFPLL_inst : BUFPLL
generic map (
    DIVIDE => 1,          -- DIVCLK divider (1-8)
    ENABLE_SYNC => TRUE  -- Enable synchrronization between PLL and GCLK (TRUE/FALSE)
)
port map (
    IOCLK => IOCLK,          -- 1-bit output: Output I/O clock
    LOCK => LOCK,           -- 1-bit output: Synchronized LOCK output
    SERDESSTROBE => SERDESSTROBE, -- 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
    GCLK => GCLK,           -- 1-bit input: BUFG clock input
    LOCKED => LOCKED,       -- 1-bit input: LOCKED input from PLL
    PLLIN => PLLIN         -- 1-bit input: Clock input from PLL
);

-- End of BUFPLL_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFPLL: High-speed I/O PLL clock buffer
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

BUFPLL #(
    .DIVIDE(1),          // DIVCLK divider (1-8)
    .ENABLE_SYNC("TRUE") // Enable synchrronization between PLL and GCLK (TRUE/FALSE)
)
BUFPLL_inst (
    .IOCLK(IOCLK),          // 1-bit output: Output I/O clock
    .LOCK(LOCK),           // 1-bit output: Synchronized LOCK output
    .SERDESSTROBE(SERDESSTROBE), // 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
    .GCLK(GCLK),           // 1-bit input: BUFG clock input
    .LOCKED(LOCKED),       // 1-bit input: LOCKED input from PLL
    .PLLIN(PLLIN)          // 1-bit input: Clock input from PLL
);

// End of BUFPLL_inst instantiation
```

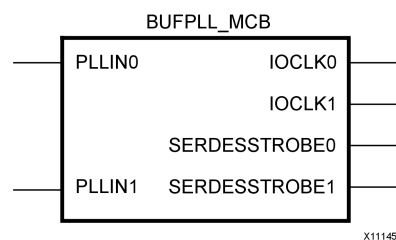
### 詳細情報

- ・ [『Spartan-6 FPGA クロックリソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## BUFPLL\_MCB

プリミティブ：PLL Buffer for the Memory Controller Block



### 概要

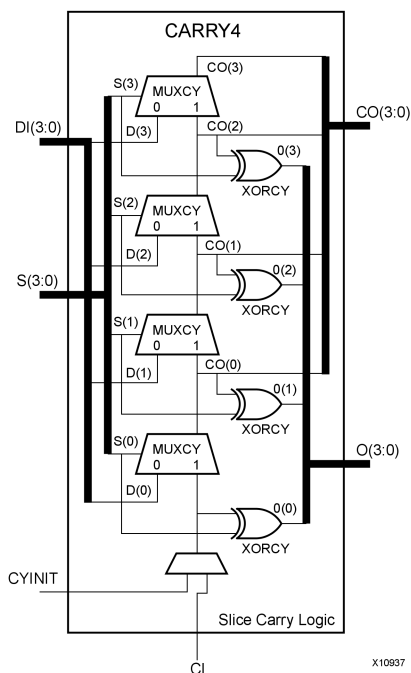
このデザイン エLEMENTは、外部メモリ インターフェイスをインプリメントするために MIG (Memory Interface Generator) コアで MCB ブロックと組み合わせて使用されます。MIG 外での使用はサポートされていません。

### 詳細情報

- ・ [『ザイリンクス メモリ インターフェイス ソリューション ユーザー ガイド』\(UG086\)](#)
- ・ [『Spartan-6 FPGA クロック リソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## CARRY4

プリミティブ：Fast Carry Logic with Look Ahead



### 概要

このデザイン エLEMENTは、スライスの高速キャリー ロジックです。キャリー チェーンには MUX および XOR がそれぞれ 4 個含まれています。これらの MUX および XOR はさらに複雑なファンクションを形成するために、専用配線を介してスライス内のその他のロジック (LUT) に接続されます。高速キャリー ロジックは、加算器、カウンタ、減算器、加減算器などの演算ファンクションの構築に加え、多入力コンパレータ、アドレス デコーダ、ロジック ゲート (AND、OR、XOR など) などのその他のロジック ファンクションに使用できます。

### ポートの説明

ポート名	方向	幅	機能
O	出力	4	キャリー チェーン XOR の通常のデータ出力
CO	出力	4	キャリー チェーンの各段のキャリー出力
DI	入力	4	キャリー MUX のデータ入力
S	入力	4	キャリー MUX のセレクト入力
CYINIT	入力	1	キャリー 初期化入力
CI	入力	1	キャリー カスケード入力

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- CARRY4: Fast Carry Logic Component
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

CARRY4_inst : CARRY4
port map (
    CO => CO,           -- 4-bit carry out
    O => O,             -- 4-bit carry chain XOR data out
    CI => CI,           -- 1-bit carry cascade input
    CYINIT => CYINIT,   -- 1-bit carry initialization
    DI => DI,           -- 4-bit carry-MUX data in
    S => S              -- 4-bit carry-MUX select input
);

-- End of CARRY4_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// CARRY4: Fast Carry Logic Component
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

CARRY4 CARRY4_inst (
    .CO(CO),           // 4-bit carry out
    .O(O),             // 4-bit carry chain XOR data out
    .CI(CI),           // 1-bit carry cascade input
    .CYINIT(CYINIT),   // 1-bit carry initialization
    .DI(DI),           // 4-bit carry-MUX data in
    .S(S)              // 4-bit carry-MUX select input
);

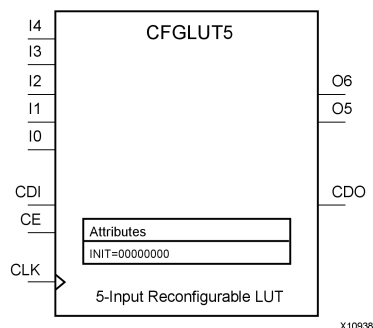
// End of CARRY4_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## CFGLUT5

### プリミティブ：5-input Dynamically Reconfigurable Look-Up Table (LUT)



### 概要

このデザイン エLEMENTは、ランタイムのダイナミック リコンフィギュレーションが可能な 5 入力ルックアップ テーブル (LUT) で、回路の動作中に LUT のロジック ファンクションを変更できます。CDI ピンを使用すると、クロックに同期して新しい INIT 値がシリアルにシフトされ、ロジック ファンクションが変更されます。O6 出力ピンでは、LUT に読み込まれた現在の INIT 値と現在選択されている入力ピン I0 ～ I4 に基づいてロジック ファンクションが生成されます。オプションで O5 出力と O6 出力を使用して、同じ入力を共有する 4 入力ファンクションを 2 つ作成するか、または 5 入力ファンクション 1 つとその 5 入力ロジックのサブセットを使用する 4 入力ファンクションを作成できます (下の表を参照)。このELEMENTは、1 つのスライス M に含まれる 4 個の LUT6 のうちの 1 つを使用します。

このELEMENTをカスケード接続するには、CDO ピンを次のELEMENTの CDI 入力に接続します。これにより、1 つのシリアル チェーンのデータ (LUT につき 32 ビット) で複数の LUT をリコンフィギュレーションできます。

### ポートの説明

ポート名	方向	幅	機能
O6	出力	1	5 入力 LUT 出力
O5	出力	1	4 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力
CDO	出力	1	リコンフィギュレーション データのカスケード出力 (オプションで次の LUT の CDI 入力に接続)
CDI	入力	1	リコンフィギュレーション データ シリアル入力
CLK	入力	1	リコンフィギュレーション クロック
CE	入力	1	アクティブ High リコンフィギュレーション クロック イネーブル

## デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

- ・ CLK 入力をリコンフィギュレーション データを供給するのに使用するクロック ソースに接続します。
- ・ CDI 入力をリコンフィギュレーション データのソースに接続します。
- ・ CE ピンを LUT のリコンフィギュレーションをイネーブルまたはディスエーブルにするには、アクティブ High のロジックに接続します。
- ・ I4 ～ I0 ピンを論理式のソース入力に接続します。ロジック ファンクションは、O6 および O5 から出力されます。
- ・ このエレメントをカスケード接続する場合は、CDO ピンを次のエレメントの CDI ピンに接続し、1 つのシリアルチェーンのデータで複数の LUT をリコンフィギュレーションできるようにします。

INIT 属性をこのデザイン エLEMENT に設定して、LUT の初期ロジック ファンクションを指定する必要があります。新しい INIT 値は、チェーンに含まれる LUT ごとに 32 ビットをシフトインすることで、回路の作動中いつでも読み込むことができます。O6 および O5 の出力値は、新しい 32 ビットの INIT 値がすべて LUT に入力されるまで無視します。新しい INIT 値が LUT にシフトインされると、LUT のロジック ファンクションが変化します。データは MSB (INIT[31]) から順に LSB (INIT[0]) までシフトインされる必要があります。

次の表に示すように、O6 および O5 の論理値は、現在の INIT 値に基づいています。

I4 I3 I2 I1 I0	O6 値	O5 値
1 1 1 1 1	INIT[31]	INIT[15]
1 1 1 1 0	INIT[30]	INIT[14]
...	...	...
1 0 0 0 1	INIT[17]	INIT[1]
1 0 0 0 0	INIT[16]	INIT[0]
0 1 1 1 1	INIT[15]	INIT[15]
0 1 1 1 0	INIT[14]	INIT[14]
...	...	...
0 0 0 0 1	INIT[1]	INIT[1]
0 0 0 0 0	INIT[0]	INIT[0]

たとえば INIT 値が FFFF8000 の場合は、次の論理式を表します。

- ・  $O6 = I4 \text{ or } (I3 \text{ and } I2 \text{ and } I1 \text{ and } I0)$
- ・  $O5 = I3 \text{ and } I2 \text{ and } I1 \text{ and } I0$

入力を共有するが機能は異なる 2 つの 4 入力 LUT として使用するには、I4 信号を論理 1 に接続します。INIT[31:16] が O6 出力の論理値に、INIT[15:0] の値が O5 出力の論理値に適用されます。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	このエレメントの初期値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- CFGLUT5: Reconfigurable 5-input LUT
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

CFGLUT5_inst : CFGLUT5
generic map (
  INT => X"00000000")
port map (
  CDO => CDO, -- Reconfiguration cascade output
  O5 => O5,   -- 4-LUT output
  O6 => O6,   -- 5-LUT output
  CDI => CDI, -- Reconfiguration data input
  CE  => CE,  -- Reconfiguration enable input
  CLK => CLK, -- Clock input
  I0  => I0,  -- Logic data input
  I1  => I1,  -- Logic data input
  I2  => I2,  -- Logic data input
  I3  => I3,  -- Logic data input
  I4  => I4,  -- Logic data input
);

-- End of CFGLUT5_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// CFGLUT5: Reconfigurable 5-input LUT
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

CFGLUT5 #(
  .INIT(32'h00000000) // Specify initial LUT contents
) CFGLUT5_inst (
  .CDO(CDO), // Reconfiguration cascade output
  .O5(O5),   // 4-LUT output
  .O6(O6),   // 5-LUT output
  .CDI(CDI), // Reconfiguration data input
  .CE(CE),   // Reconfiguration enable input
  .CLK(CLK), // Clock input
  .I0(I0),   // Logic data input
  .I1(I1),   // Logic data input
  .I2(I2),   // Logic data input
  .I3(I3),   // Logic data input
  .I4(I4),   // Logic data input
);

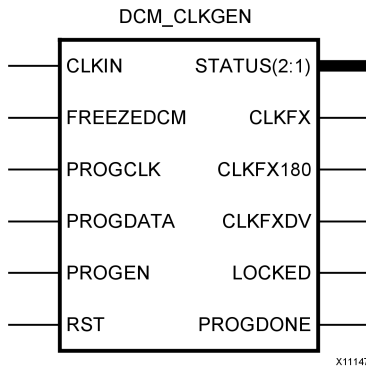
// End of CFGLUT5_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## DCM\_CLKGEN

プリミティブ：Digital Clock Manager.



### 概要

デジタル クロック マネージャー (DCM) が周波数アライメント モードになっており、入力クロックに対して位相が揃っていない (位相関係がない) とき、プログラマブル出力クロック合成、ジッターの削減、スペクトラム拡散、およびフリー ランニング オシレーター モードなどの機能を使用できます。このコンポーネントの使用の詳細は、『Spartan-6 FPGA クロック リソース ユーザー ガイド』(UG382) を参照してください。

### ポートの説明

ポート名	方向	幅	機能
CLKFX	出力	1	CLKFX_MULTIPLY および CLKFX_DIVIDE 属性で制御される同期クロック出力。スタティックに設定するか、または 4 ワイヤ SPI ポート (PROGDATA、PROGCLK、PROGDONE、および PROGEN) を介してダイナミック プログラムできます。デューティ サイクルは常に 50% です。
CLKFXDV	出力	1	CLKFX の分周出力クロック。分周値は CLKFXDV_DIVIDE 属性の値で決まります。CLKFX および CLKFXDV の位相は揃えられません。
CLKFX180	出力	1	180 度位相がシフトされている同期クロック出力 CLKFX (CLKFX の反転バージョン)。デューティ サイクルは常に 50% です。
CLKIN	入力	1	DCM へのクロック入力。常に必要で、CLKIN 周波数およびジッターがデータシートに記載されている範囲内である必要があります。フリーランニング オシレーター モードの場合は、DCM がロックしてフリーズしたら動作クロックの接続を解除できます。その他のモードでは、フリーランニング クロックを供給し続ける必要があります。
FREEZEDCM	入力	1	CLKIN 入力失われたときのタップ調整の変動を防ぎます。これで、DCM がフリー ランニング にモード設定されます。
LOCKED	出力	1	DCM の動作の準備が完了しているかを示す同期出力。 <ul style="list-style-type: none"> <li>0 : DCM クロック出力が無効です。</li> <li>1 : DCM で動作準備が完了しています。</li> <li>1 が 0 になるとき : DCM で LOCK が失われ、DCM がリセットされます。</li> </ul>
PROGCLK	入力	1	M および/または D リコンフィギュレーションのクロック入力

ポート名	方向	幅	機能
PROGDATA	入力	1	DCM の M (通倍) および D (分周) 値の再プログラム用データを供給するシリアル データ入力。この入力は、PROGCLK 入力に同期させる必要があります。
PROGDONE	出力	1	M または D 値の再プログラミングが正しく完了したことを示すアクティブ High 出力
PROGEN	入力	1	M/D 値を再プログラムする アクティブ High のイネーブル入力。この入力は、PROGCLK 入力に同期させる必要があります。
RST	入力	1	DCM 回路をリセットします。RST 信号は、アクティブ High の非同期リセットです。RST 信号をアサートすると、すべての DCM 出力 (LOCKED 信号、ステータス信号、出力クロック) がソースクロックの 4 サイクル以内に Low になります。リセットは非同期であるため、ディアサート中にクロックの最後のサイクルが短いパルスになったり、デューティサイクルが崩れたり、クロック間のスキューが調整される可能性があります。このため、デバイスをリコンフィギュレーションする場合や、入力周波数を変更する場合は、RST ピンを使用する必要があります。RST 信号をディアサートすると、次の CLKIN サイクルに同期してクロックをロック状態にするプロセスが開始します。DCM がリセット後に正しくロックされるようにするには、CLKIN 信号が供給され、3 クロックサイクル以上安定するまで RST 信号をディアサートしておく必要があります。どのデザインでも、クロックが安定するまで DCM をリセットに保持する必要があります。コンフィギュレーションでは、GWE が解除されるまで DCM は自動的にリセット状態に保持されます。GSR が解除されたときにクロックが安定していれば、コンフィギュレーション後に DCM をリセットする必要はありません。
STATUS[2:1]	出力	2	クロック ステータス出力 <ul style="list-style-type: none"> <li>STATUS[1] : CLKIN の停止</li> <li>STATUS[2] : CLKFX または CLKFX180 の停止</li> </ul>

## デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
CLKFX_DIVIDE	整数	1 ~ 256	1	この値と入力周波数および CLKFX_MULTIPLY の値を組み合わせ、CLKFX および CLKFX180 の出力周波数が決定されます。
CLKFXDV_DIVIDE	整数	2、4、8、16、32	2	CLKFXDV の分周値を指定します。
CLKFX_MD_MAX	3 上位ビット 浮動小数点	0.000 ~ 256.000	0.000	DCM_CLKGEN を変数 M および D 値で使用する時、スタティック タイミング解析中に使用する M と D の最大比率を指定します。



属性	データ型	値	デフォルト	説明
CLKFX_MULTIPLY	整数	2 ~ 256	4	この値と入力周波数および CLKFX_DIVIDE の値を組み合わせ、CLKFX および CLKFX180 の出力周波数が決定されます。
CLKIN_PERIOD	浮動小数点	2.000 ~ 1000.00	なし	CLKFX/CLKFX180 出力のために DCM の調整を補助しロックにかかる時間を短縮するために使用するソースクロックの周期を指定します。
DFS_BANDWIDTH	文字列	"OPTIMIZED"、 "HIGH"、 "LOW"	"OPTIMIZED"	プロセス、電圧、温度 (PVT) に対する DCM の周波数調整バンド幅を指定します。
PROG_MD_BANDWIDTH	文字列	"OPTIMIZED"、 "HIGH"、 "LOW"	"OPTIMIZED"	M および D 値のプログラミング変更に対する DCM の周波数調整バンド幅を指定します。
SPREAD_SPECTRUM	文字列	"NONE"、"CENTER_LOW_SPREAD"、 "CENTER_HIGH_SPREAD"、 "VIDEO_LINK_M0"、 "VIDEO_LINK_M1"、 "VIDEO_LINK_M2"	"NONE"	スペクトラム拡散でサポートされるモードを指定します。周波数ホッピングを実現するには、適切な IP と共に使用する必要があります。  固定スペクトラム拡散 ("CENTER_LOW_SPREAD"、 "CENTER_HIGH_SPREAD") またはソフト スペクトラム拡散 ("VIDEO_LINK_M0"、 "VIDEO_LINK_M1"、 "VIDEO_LINK_M2") で使用されます。ソフト スペクトラム拡散は、ソフト スペクトラム拡散リファレンス デザインと共に使用する必要があります。
STARTUP_WAIT	ブール代数	FALSE、TRUE	FALSE	コンフィギュレーション DONE 信号を DCM LOCKED 信号が High になるまで遅らせます。

## VHDL Instantiation Template

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- DCM_CLKGEN: Frequency Aligned Digital Clock Manager
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

DCM_CLKGEN_inst : DCM_CLKGEN
generic map (
    CLKFXDV_DIVIDE => 2,      -- CLKFXDV divide value (2, 4, 8, 16, 32)
    CLKFX_DIVIDE   => 1,      -- Divide value - D - (1-256)
    CLKFX_MD_MAX   => 0.0,    -- Specify maximum M/D ratio for timing analysis
    CLKFX_MULTIPLY => 4,      -- Multiply value - M - (2-256)
    CLKIN_PERIOD   => 0.0,    -- Input clock period specified in nS
    SPREAD_SPECTRUM => "NONE", -- Spread Spectrum mode "NONE", "CENTER_LOW_SPREAD", "CENTER_HIGH_SPREAD",
                                -- "VIDEO_LINK_M0", "VIDEO_LINK_M1" or "VIDEO_LINK_M2"
    STARTUP_WAIT   => FALSE   -- Delay config DONE until DCM_CLKGEN LOCKED (TRUE/FALSE)
)

```

```

port map (
    CLKFX => CLKFX,           -- 1-bit output: Generated clock output
    CLKFX180 => CLKFX180,     -- 1-bit output: Generated clock output 180 degree out of phase from CLKFX.
    CLKFXDV => CLKFXDV,       -- 1-bit output: Divided clock output
    LOCKED => LOCKED,         -- 1-bit output: Locked output
    PROGDONE => PROGDONE,      -- 1-bit output: Active high output to indicate the successful re-programming
    STATUS => STATUS,         -- 2-bit output: DCM_CLKGEN status
    CLKIN => CLKIN,           -- 1-bit input: Input clock
    FREEZEDCM => FREEZEDCM,    -- 1-bit input: Prevents frequency adjustments to input clock
    PROGCLK => PROGCLK,        -- 1-bit input: Clock input for M/D reconfiguration
    PROGDATA => PROGDATA,      -- 1-bit input: Serial data input for M/D reconfiguration
    PROGEN => PROGEN,          -- 1-bit input: Active high program enable
    RST => RST                 -- 1-bit input: Reset input pin
);

-- End of DCM_CLKGEN_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// DCM_CLKGEN: Frequency Aligned Digital Clock Manager
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

DCM_CLKGEN #(
    .CLKFXDV_DIVIDE(2),        // CLKFXDV divide value (2, 4, 8, 16, 32)
    .CLKFX_DIVIDE(1),          // Divide value - D - (1-256)
    .CLKFX_MD_MAX(0.0),        // Specify maximum M/D ratio for timing analysis
    .CLKFX_MULTIPLY(4),        // Multiply value - M - (2-256)
    .CLKIN_PERIOD(0.0),        // Input clock period specified in nS
    .SPREAD_SPECTRUM("NONE"),  // Spread Spectrum mode "NONE", "CENTER_LOW_SPREAD", "CENTER_HIGH_SPREAD",
                                // "VIDEO_LINK_M0", "VIDEO_LINK_M1" or "VIDEO_LINK_M2"
    .STARTUP_WAIT("FALSE")     // Delay config DONE until DCM_CLKGEN LOCKED (TRUE/FALSE)
)
DCM_CLKGEN_inst (
    .CLKFX(CLKFX),             // 1-bit output: Generated clock output
    .CLKFX180(CLKFX180),       // 1-bit output: Generated clock output 180 degree out of phase from CLKFX.
    .CLKFXDV(CLKFXDV),         // 1-bit output: Divided clock output
    .LOCKED(LOCKED),           // 1-bit output: Locked output
    .PROGDONE(PROGDONE),        // 1-bit output: Active high output to indicate the successful re-programming
    .STATUS(STATUS),           // 2-bit output: DCM_CLKGEN status
    .CLKIN(CLKIN),             // 1-bit input: Input clock
    .FREEZEDCM(FREEZEDCM),     // 1-bit input: Prevents frequency adjustments to input clock
    .PROGCLK(PROGCLK),          // 1-bit input: Clock input for M/D reconfiguration
    .PROGDATA(PROGDATA),        // 1-bit input: Serial data input for M/D reconfiguration
    .PROGEN(PROGEN),           // 1-bit input: Active high program enable
    .RST(RST)                  // 1-bit input: Reset input pin
);

// End of DCM_CLKGEN_inst instantiation

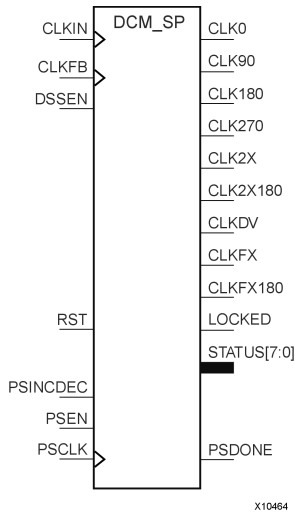
```

## 詳細情報

- ・ [『Spartan-6 FPGA クロックリソース ユーザー ガイド』 \(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』 \(DS162\)](#)

## DCM\_SP

### プリミティブ：Digital Clock Manager



## 概要

このデザイン エLEMENTは、さまざまな機能を備えたデジタル クロック マネージャーで、クロック遅延ロック ループ (DLL)、デジタル周波数合成 (DFS)、デジタル位相シフト (DPS) といった機能をインプリメントできます。DCM\_SP は、オンチップおよびオフチップからのクロック遅延をなくしたり、データ キャプチャを向上するためにクロック位相をシフトしたり、異なる周波数のクロックを生成させる場合などに便利です。

## ポートの説明

ポート名	方向	幅	機能
CLKDV	出力	1	CLKDV_DIVIDE 属性で制御される分周クロック出力。CLKDV_DIVIDE 属性で整数以外の値が設定されていない限り、CLKDV 出力のデューティサイクルは 50% になります。
CLKFB	入力	1	DCM へのクロック フィードバック入力。DFS 出力、CLKFX、または CLKFX180 がスタンドアロンで使用されない限り、フィードバック入力が必要です。CLKFB 入力は、DCM の CLK0 または CLK2X 出力から供給し、それに応じて CLK_FEEDBACK を 1X または 2X に設定する必要があります。NONE に設定すると、CLKFB は使用されず Low に保持する必要があります。フィードバック ポイントには、内部または外部でクロック分配ネットワークに追加される遅延が含まれているのが理想的です。
CLKFX	出力	1	CLKFX_MULTIPLY および CLKFX_DIVIDE 属性で制御される同期クロック出力。デューティサイクルは常に 50% で、位相関係が不要な場合は、クロック フィードバックも不要です。
CLKFX180	出力	1	180 度位相がシフトされている同期クロック出力 CLKFX (CLKFX の反転バージョン)。デューティサイクルは常に 50% で、位相関係が不要な場合は、フィードバック ループも不要です。
CLKIN	入力	1	DCM へのクロック入力。常に必要で、CLKIN 周波数およびジッターがデータシートに記載されている範囲内である必要があります。
CLK0	出力	1	位相シフトが 0 の (位相がシフトされていない) CLKIN と同じ周波数。Spartan®-6 FPGA ではデューティサイクルが 50% になるように調整されます。CLK_FEEDBACK は CLK0 のスキューを調整するように 1X または 2X に設定する必要があります。

ポート名	方向	幅	機能
CLK2X	出力	1	位相シフトが 0 の (位相がシフトされていない) 2 通倍の周波数クロック出力。CLK2X 出力のデューティ サイクルは常に 50% です。DLL 機能のフィードバックソースとして CLK0 または CLK2X のいずれかが必要です。
CLK2X180	出力	1	位相シフトが 180 の (位相がシフトされていない) 2 通倍の周波数クロック出力。CLK2X180 出力のデューティ サイクルは常に 50% です。
CLK90	出力	1	CLKIN と同じ周波数で位相を 90 度 (1/4 周期) シフトしたクロック。Spartan®-6 FPGA ではデューティ サイクルが 50% になるように調整されます。
CLK180	出力	1	CLKIN と同じ周波数で位相を 180 度 (1/2 周期) シフトしたクロック。Spartan®-6 FPGA ではデューティ サイクルが 50% になるように調整されます。
CLK270	出力	1	CLKIN と同じ周波数で位相を 270 度 (3/4 周期) シフトしたクロック。Spartan®-6 FPGA ではデューティ サイクルが 50% になるように調整されます。
LOCKED	出力	1	すべての DCM 機能が CLKIN 周波数にロックされています。クロック出力は有効で、CLKIN が特定の範囲内にあることが想定されます。 <ul style="list-style-type: none"> <li>・ 0 : DCM で CLKIN 周波数にロックが試みられます。DCM クロック出力は無効です。</li> <li>・ 1 : DCM が CLKIN 周波数にロックされています。DCM クロック出力は有効です。</li> <li>・ 1 から 0 になる : DCM で LOCK が失われ、DCM がリセットされます。</li> </ul>
PSCLK	入力	1	立ち上がりエッジでクロックが供給される可変位相シフターへのクロック入力。グローバル クロック バッファを使用している場合、PSCLK を駆動できるのは上部の 8 つの BUFGMUX のみです (BUFGMUX_X2Y1、BUFGMUX_X2Y2、BUFGMUX_X2Y3、BUFGMUX_X2Y4、BUFGMUX_X3Y5、BUFGMUX_X3Y6、BUFGMUX_X3Y7、および BUFGMUX_X3Y8)。
PSDONE	出力	1	可変位相シフトの完了 <ul style="list-style-type: none"> <li>・ 0 : 位相シフト操作が実行されていないか、位相シフト操作を実行中です。</li> <li>・ 1 : 要求された位相シフト操作が完了しています。1 RCLK サイクル間 High になります。次の可変位相シフト操作を開始できます。</li> </ul>
PSEN	入力	1	可変位相シフト イネーブル。DCM ブロック内で反転可能です。次に、反転しない場合の動作を示します。 <ul style="list-style-type: none"> <li>・ 0 : 可変位相シフトをディスエーブルにします。位相シフターへの入力が無視されます。</li> <li>・ 1 : 次の PSCLK クロックの立ち上がりエッジで可変位相シフト操作をイネーブルにします。</li> </ul> <b>注記 :</b> 使用しないときは 0 にします。

ポート名	方向	幅	機能
PSINCDEC	入力	1	<p>可変位相シフトを増分または減分します。DCM ブロック内で反転可能です。次に、反転しない場合の動作を示します。</p> <ul style="list-style-type: none"> <li>・ 0：次にイネーブルにされている PSCLK クロックの立ち上がりエッジで位相シフト値を減分します。</li> <li>・ 1：次にイネーブルにされている PSCLK クロックの立ち上がりエッジで位相シフト値を増分します。</li> </ul>
RST	入力	1	<p>非同期リセット入力。DCM ロジックをコンフィギュレーション後の状態にリセットします。これにより、DCM は CLKIN 入力に再度ロックされます。DCM ブロック内で反転可能です。次に、反転しない場合の動作を示します。</p> <ul style="list-style-type: none"> <li>・ 0：影響なし</li> <li>・ 1：DCM ブロックのリセット。最低 CLKIN 3 サイクル間 RST を High に保持します。</li> </ul>
STATUS[7:0]	出力	8	<p>ステータス出力バスにより DCM のステータスが示されます。</p> <ul style="list-style-type: none"> <li>・ STATUS[0]：可変位相シフト オーバーフロー。可変ファイン位相シフトの制御出力です。可変位相シフターが許容される最低値または最大値に達しています。位相シフトが遅延ラインの最後に到達している場合は、許容範囲は <math>\pm 255</math> 以下です。 <ul style="list-style-type: none"> <li>－ 0：位相シフトが制限値に達成していません。</li> <li>－ 1：位相シフトが制限値に達成しました。</li> </ul> </li> <li>・ STATUS[1]：CLKIN 入力の停止。CLKFB フィードバック入力に接続されているときのみ使用できます。LOCKED 出力がアサートされるまでリセットに保持されます。アクティブになるには最低 CLKIN が 1 サイクル必要です。CLKIN がトグルしない場合はアサートされません。 <ul style="list-style-type: none"> <li>－ 0：CLKIN 入力にトグルしています。</li> <li>－ 1：CLKIN 入力は、LOCKED 出力が High になることが可能なときでもトグルしていません。</li> </ul> </li> <li>・ STATUS[2]：CLKFX または CLKFX180 出力の停止 <ul style="list-style-type: none"> <li>－ 0：CLKFX および CLKFX180 出力にトグルしています。</li> <li>－ 1：CLKFX および CLKFX180 出力は、LOCKED 出力が High になることが可能なときでもトグルしていません。</li> </ul> </li> <li>・ STATUS[4:3]：予約</li> <li>・ STATUS[5]：DCM_SP のロック周期ステータス中の CLKFX のミラーバージョン</li> <li>・ STATUS[6]：予約</li> <li>・ STATUS[7]：DCM_SP ロック周期ステータス中の CLKIN のミラーバージョン</li> </ul>

## デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
CLK_FEEDBACK	文字列	"1X"、"2X"、 "NONE"	"1X"	DCM フィードバック モードを定義します。 <ul style="list-style-type: none"> <li>"1X" : CLK0 をフィードバックに使用します。</li> <li>"2X" : CLK2X をフィードバックに使用します。</li> </ul>
CLKDV_DIVIDE	1 上位ビット 浮動小数点	2.0、1.5、2.5、3.0、 3.5、4.0、4.5、5.0、 5.5、6.0、6.5、7.0、 7.5、8.0、9.0、10.0、 11.0、12.0、13.0、 14.0、15.0、16.0	2.0	CLKDLL、CLKDLLE、CLKDLLHF、 DCM_SP のクロック分周出力 CLKDV の分 周比を指定します。
CLKFX_DIVIDE	整数	1 ~ 32	1	CLKFX 出力の分周比を指定します。
CLKFX_MULTIPLY	整数	2 ~ 32	4	CLKFX 出力の通倍比を指定します。
CLKIN_DIVIDE_BY_2	ブール代数	FALSE、TRUE	FALSE	CLKIN を 2 で分周します。
CLKIN_PERIOD	浮動小数点	2.000 ~ 1000.00	なし	CLKIN の入力周期を ns で指定します。
CLKOUT_PHASE_SHIFT	文字列	"NONE"、 "FIXED"、 "VARIABLE"	"NONE"	位相シフト モードを指定します。 <ul style="list-style-type: none"> <li>"NONE" : 位相シフト機能を使用しません。設定されている値は反映されません。</li> <li>"FIXED" : DCM の出力は CLKIN から決まった位相だけシフトされたものになります。値は PHASE_SHIFT 属性で指定されます。</li> <li>"VARIABLE" : DCM 出力を CLKIN に対して正および負の範囲にシフトできるようにします。開始値は PHASE_SHIFT 属性で指定されます。</li> </ul>
DESKEW_ADJUST	文字列	"SYSTEM_ SYNCHRONOUS"、 "SOURCE_ SYNCHRONOUS"	"SYSTEM_ SYNCHRONOUS"	DCM_SP クロック出力と FPGA のクロック入力ピン間のクロック遅延の分配に影響する コンフィギュレーション ビットを設定します。
DFS_FREQUENCY_MODE	文字列	"LOW"、"HIGH"	"LOW"	この属性はレガシ属性です。DCM は常に 自動周波数検索モードになります。High または Low のどちらに設定しても、影響ありません。

属性	データ型	値	デフォルト	説明
DLL_FREQUENCY_MODE	文字列	"LOW"、"HIGH"	"LOW"	この属性はレガシ属性です。DCM は常に自動周波数検索モードになります。High または Low のどちらに設定しても、影響ありません。
DUTY_CYCLE_CORRECTION	ブール代数	TRUE、FALSE	TRUE	サポートなし
FACTORY_JF	16 進数	16'h8080 ～ 16'hffff	16'hc080	サポートなし
PHASE_SHIFT	整数	-255 ～ 255	0	この属性は、CLKOUT_PHASE_SHIFT 属性が FIXED または VARIABLE に設定されている場合のみ使用できます。コンフィギュレーションでの CLKIN とすべての DCM クロック出力間の立ち上がりエッジ スキューを定義し、DCM クロック出力の位相をシフトします。スキューまたは位相シフト値は、ファイン位相シフトの式で表現されているように、クロック周期の係数を表す整数で指定します。実際に許容される値は、入力クロックの周波数によって異なります。TCLKIN が FINE_SHIFT_RANGE より大きいとき実際の範囲は狭くなります。FINE_SHIFT_RANGE は、遅延ラインのすべてのタップの総遅延を示します。
STARTUP_WAIT	ブール代数	FALSE、TRUE	FALSE	<p>FPGA コンフィギュレーション DONE 信号を High にするのを、DCM の LOCKED 信号がアサートされるまで待つかどうかを指定します。</p> <ul style="list-style-type: none"> <li>FALSE：デフォルト値。DCM の LOCKED 信号がアサートされるのを待たずにコンフィギュレーションの最後にアサートされます。</li> <li>TRUE：DONE 信号は関連する DCM の LOCKED 信号が High になるまで High になりません。</li> </ul> <p>STARTUP_WAIT の指定にかかわらず、LOCKED 信号は High になります。FPGA のスタートアップ シーケンスも変更し、延期サイクルの前に LCK (ロック) サイクルを挿入する必要があります。DONE サイクルまたは GWE サイクルが一般的です。複数の DCM をコンフィギュレーションする場合は、すべての DCM がロックされるまで DONE ピンは High になりません。</p>

## VHDL 記述 (インスタンスエーション)

次の2つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- DCM_SP: Digital Clock Manager
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

DCM_SP_inst : DCM_SP
generic map (
    CLKDV_DIVIDE => 2.0,                -- CLKDV divide value
                                         -- (1.5,2,2.5,3,3.5,4,4.5,5,5.5,6,6.5,7,7.5,8,9,10,11,12,13,14,15,16).
    CLKFX_DIVIDE => 1,                -- Divide value on CLKFX outputs - D - (1-32)
    CLKFX_MULTIPLY => 4,              -- Multiply value on CLKFX outputs - M - (2-32)
    CLKIN_DIVIDE_BY_2 => FALSE,        -- CLKIN divide by two (TRUE/FALSE)
    CLKIN_PERIOD => 10.0,              -- Input clock period specified in ns
    CLKOUT_PHASE_SHIFT => "NONE",      -- Output phase shift (NONE, FIXED, VARIABLE)
    CLK_FEEDBACK => "1X",              -- Feedback source (NONE, 1X, 2X)
    DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS", -- SYSTEM_SYNCHRONOUS or SOURCE_SYNCHRONOUS
    DFS_FREQUENCY_MODE => "LOW",       -- Unsupported - Do not change value
    DLL_FREQUENCY_MODE => "LOW",       -- Unsupported - Do not change value
    DSS_MODE => "NONE",               -- Unsupported - Do not change value
    DUTY_CYCLE_CORRECTION => TRUE,     -- Unsupported - Do not change value
    FACTORY_JF => X"c080",             -- Unsupported - Do not change value
    PHASE_SHIFT => 0,                 -- Amount of fixed phase shift (-255 to 255)
    STARTUP_WAIT => FALSE              -- Delay config DONE until DCM_SP LOCKED (TRUE/FALSE)
)
port map (
    CLK0 => CLK0,                    -- 1-bit output: 0 degree clock output
    CLK180 => CLK180,                -- 1-bit output: 180 degree clock output
    CLK270 => CLK270,                -- 1-bit output: 270 degree clock output
    CLK2X => CLK2X,                  -- 1-bit output: 2X clock frequency clock output
    CLK2X180 => CLK2X180,            -- 1-bit output: 2X clock frequency, 180 degree clock output
    CLK90 => CLK90,                  -- 1-bit output: 90 degree clock output
    CLKDV => CLKDV,                  -- 1-bit output: Divided clock output
    CLKFX => CLKFX,                  -- 1-bit output: Digital Frequency Synthesizer output (DFS)
    CLKFX180 => CLKFX180,            -- 1-bit output: 180 degree CLKFX output
    LOCKED => LOCKED,                -- 1-bit output: DCM_SP Lock Output
    PSDONE => PSDONE,                -- 1-bit output: Phase shift done output
    STATUS => STATUS,                -- 8-bit output: DCM_SP status output
    CLKFB => CLKFB,                  -- 1-bit input: Clock feedback input
    CLKIN => CLKIN,                  -- 1-bit input: Clock input
    DSSEN => DSSEN,                  -- 1-bit input: Unsupported, specify to GND.
    PSCLK => PSCLK,                  -- 1-bit input: Phase shift clock input
    PSEN => PSEN,                    -- 1-bit input: Phase shift enable
    PSINCDEC => PSINCDEC,            -- 1-bit input: Phase shift increment/decrement input
    RST => RST                        -- 1-bit input: Active high reset input
);

-- End of DCM_SP_inst instantiation
```



## Verilog 記述 (インスタンス化)

```
// DCM_SP: Digital Clock Manager
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

DCM_SP #(
    .CLKDV_DIVIDE(2.0),                // CLKDV divide value
                                        // (1.5,2,2.5,3,3.5,4,4.5,5,5.5,6,6.5,7,7.5,8,9,10,11,12,13,14,15,16).
    .CLKFX_DIVIDE(1),                  // Divide value on CLKFX outputs - D - (1-32)
    .CLKFX_MULTIPLY(4),                 // Multiply value on CLKFX outputs - M - (2-32)
    .CLKIN_DIVIDE_BY_2("FALSE"),        // CLKIN divide by two (TRUE/FALSE)
    .CLKIN_PERIOD(10.0),                 // Input clock period specified in nS
    .CLKOUT_PHASE_SHIFT("NONE"),         // Output phase shift (NONE, FIXED, VARIABLE)
    .CLK_FEEDBACK("1X"),                 // Feedback source (NONE, 1X, 2X)
    .DESKEW_ADJUST("SYSTEM_SYNCHRONOUS"), // SYSTEM_SYNCHRONOUS or SOURCE_SYNCHRONOUS
    .DFS_FREQUENCY_MODE("LOW"),           // Unsupported - Do not change value
    .DLL_FREQUENCY_MODE("LOW"),           // Unsupported - Do not change value
    .DSS_MODE("NONE"),                   // Unsupported - Do not change value
    .DUTY_CYCLE_CORRECTION("TRUE"),       // Unsupported - Do not change value
    .FACTORY_JF(16'hc080),                // Unsupported - Do not change value
    .PHASE_SHIFT(0),                      // Amount of fixed phase shift (-255 to 255)
    .STARTUP_WAIT("FALSE")                // Delay config DONE until DCM_SP LOCKED (TRUE/FALSE)
)
DCM_SP_inst (
    .CLK0(CLK0),                        // 1-bit output: 0 degree clock output
    .CLK180(CLK180),                    // 1-bit output: 180 degree clock output
    .CLK270(CLK270),                    // 1-bit output: 270 degree clock output
    .CLK2X(CLK2X),                      // 1-bit output: 2X clock frequency clock output
    .CLK2X180(CLK2X180),                // 1-bit output: 2X clock frequency, 180 degree clock output
    .CLK90(CLK90),                      // 1-bit output: 90 degree clock output
    .CLKDV(CLKDV),                      // 1-bit output: Divided clock output
    .CLKFX(CLKFX),                      // 1-bit output: Digital Frequency Synthesizer output (DFS)
    .CLKFX180(CLKFX180),                 // 1-bit output: 180 degree CLKFX output
    .LOCKED(LOCKED),                    // 1-bit output: DCM_SP Lock Output
    .PSDONE(PSDONE),                   // 1-bit output: Phase shift done output
    .STATUS(STATUS),                   // 8-bit output: DCM_SP status output
    .CLKFB(CLKFB),                      // 1-bit input: Clock feedback input
    .CLKIN(CLKIN),                      // 1-bit input: Clock input
    .DSSSEN(DSSSEN),                    // 1-bit input: Unsupported, specify to GND.
    .PSCLK(PSCLK),                      // 1-bit input: Phase shift clock input
    .PSEN(PSEN),                        // 1-bit input: Phase shift enable
    .PSINCDEC(PSINCDEC),                // 1-bit input: Phase shift increment/decrement input
    .RST(RST)                           // 1-bit input: Active high reset input
);

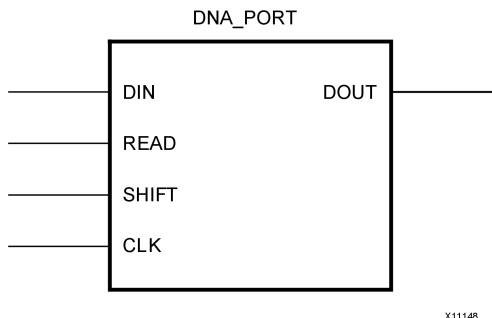
// End of DCM_SP_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA クロック リソース ユーザー ガイド』\(UG382\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## DNA\_PORT

プリミティブ：Device DNA Data Access Port



### 概要

DNA\_PORT を使用すると専用のシフトレジスタにアクセスできます。このシフトレジスタにはデバイスの Device DNA データビット (固有 ID) が読み込まれます。このコンポーネントを使用すると、DNA データビットをシフトアウトできるだけでなく、補足ビットを含めたり、DNA データをロールオーバーする (初期データのシフトアウト後に DNA データを繰り返す) こともできます。このコンポーネントは、主にほかの回路と組み合わせて FPGA ビットストリームの不正コピー防止を構築するのに使用します。正しく動作するように、入力および出力をすべてデザインに接続してください。Device DNA データにアクセスするにはまず、アクティブ High の READ 信号を 1 クロック サイクル間 High にして、シフトレジスタをロードする必要があります。シフトレジスタをロードした後、アクティブ High の SHIFT 入力をイネーブルにして、DOUT 出力ポートのデータを取り込むことで、データをクロックに同期させてシフトアウトできます。追加のデータがある場合は、適切なロジックを DIN ポートに接続すると、57 ビットのシフトレジスタの最後に追加できます。DNA データをロールオーバーする場合は、DOUT ポートを直接 DIN ポートに接続し、57 ビットのシフト操作の後で同じデータがシフトアウトされるようにします。追加データが不要な場合は、DIN ポートを論理 0 に固定できます。SIM\_DNA\_VALUE 属性を設定すると、DNA データシーケンスをシミュレーションできます。デフォルトでは、シミュレーション モデルの Device DNA データビットはすべて 0 です。

### ポートの説明

ポート名	方向	幅	機能
CLK	入力	1	クロック入力
DIN	入力	1	ユーザー データ入力
DOUT	出力	1	DNA 出力データ
READ	入力	1	アクティブ High のロード DNA、アクティブ Low の読み出し入力
SHIFT	入力	1	アクティブ High のシフト イネーブル入力

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

正しく動作するように、入力および出力をすべてデザインに接続します。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
SIM_DNA_VALUE	16 進数	57'h00000000 0000000 ~ 57'h1fffffffff	57'h00000000 0000000	あらかじめプログラムされている工場 ID 値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- DNA_PORT: Device DNA Data Access Port
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

DNA_PORT_inst : DNA_PORT
generic map (
  SIM_DNA_VALUE => X"0000000000000000" -- Specifies the Pre-programmed factory ID value
)
port map (
  DOUT => DOUT,    -- 1-bit output: DNA output data
  CLK  => CLK,      -- 1-bit input: Clock input
  DIN  => DIN,      -- 1-bit input: User data input pin
  READ => READ,     -- 1-bit input: Active high load DNA, active low read input
  SHIFT => SHIFT    -- 1-bit input: Active high shift enable input
);

-- End of DNA_PORT_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// DNA_PORT: Device DNA Data Access Port
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

DNA_PORT #(
  .SIM_DNA_VALUE(57'h0000000000000000) // Specifies the Pre-programmed factory ID value
)
DNA_PORT_inst (
  .DOUT(DOUT),    // 1-bit output: DNA output data
  .CLK(CLK),      // 1-bit input: Clock input
  .DIN(DIN),      // 1-bit input: User data input pin
  .READ(READ),    // 1-bit input: Active high load DNA, active low read input
  .SHIFT(SHIFT)   // 1-bit input: Active high shift enable input
);

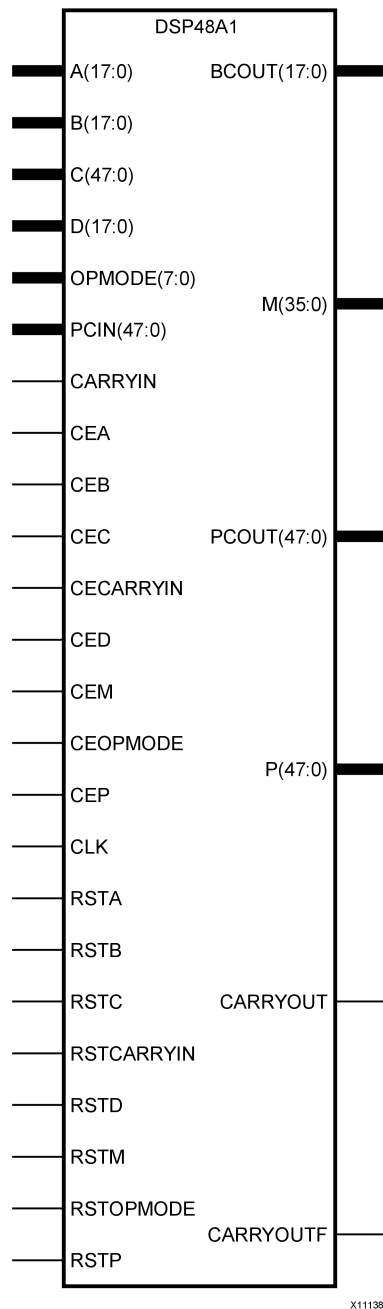
// End of DNA_PORT_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュレーション ユーザー ガイド』\(UG380\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)
- ・ [Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## DSP48A1

プリミティブ : Multi-Functional, Cascadable, 48-bit Output, Arithmetic Block



## 概要

このデザイン エLEMENTは、柔軟性が高い多用途のハード IP ブロックで、多くの DSP アルゴリズムで見られる小型で高速な演算処理を作成できます。このブロックでは、コンフィギュレーション可能な 18 ビットの加減算器に、18 X 18 符号付き乗算器、48 ビットの加減/アキュムレータが順に付けられています。このブロックには、コンフィギュレーション可能なパイプライン レジスタが数個含まれているため、追加されるレイテンシをトレードオフにする高速クロックを実現できます。OpMode ピンでは、ブロック操作を 1 クロック サイクルから次サイクルに変更でき、デザインに含まれる複数の演算ファンクションに 1 つのブロックを使用できます。また、複数の DSP48A1 ブロックをカスケード接続して、大型の乗算および加算ファンクションを作成できます。このELEMENTの使用法の詳細は、『[Spartan-6 FPGA DSP48A1 スライス ユーザー ガイド](#)』(UG389)を参照してください。

## ポートの説明

ポート名	方向	幅	機能
A[17:0]	入力	18	OPMODE[1:0] の値によって、乗算器または後置加減算器に送られる 18 ビット データ入力
B[17:0]	入力	18	OPMODE[1:0] の値によって、乗算器、前置加減算器、またはオプションで後置加減算器に送られる 18 ビット データ入力
BCOUT[17:0]	出力	18	ポート B のカスケード出力で、カスケード接続されている下位の DSP48A1 の B ポートに接続します。使用しない場合は未接続にします。
C[47:0]	入力	48	後置加減算器への 48 ビット入力
CARRYIN	入力	1	後置加減算器への外部キャリー入力。別の DSP48A1 ブロックの CARRYOUT ピンにのみ接続します。
CARRYOUT	出力	1	後置加減算器の外部キャリー出力信号。別の DSP48A1 ブロックの CARRYIN ピンにのみ接続します。
CARRYOUTF	出力	1	ファブリックに配線可能な後置加減算器の外部キャリー出力信号
CEA	入力	1	A ポートレジスタ (A0REG=1 または A1REG=1) のアクティブ High のクロック イネーブルで、使用しない場合および A0REG=1 または A1REG=1 の場合は論理 1 に、A0REG=0 および A1REG=0 の場合は論理 0 に接続します。
CEB	入力	1	B ポートレジスタ (B0REG=1 または B1REG=1) のアクティブ High のクロック イネーブルで、使用しない場合および B0REG=1 または B1REG=1 の場合は論理 1 に、B0REG=0 および B1REG=0 の場合は論理 0 に接続します。
CEC	入力	1	C ポートレジスタ (CREG=1) のアクティブ High のクロック イネーブルで、使用しない場合および CREG=1 の場合は論理 1 に、CREG=0 の場合は論理 0 に接続します。
CECARRYIN	入力	1	キャリー入力レジスタ (CARRYINREG=1) のアクティブ High のクロック イネーブルで、使用しない場合および CARRYINREG=1 の場合は論理 1 に、CARRYINREG=0 の場合は論理 0 に接続します。
CED	入力	1	D ポートレジスタ (DREG=1) のアクティブ High のクロック イネーブルで、使用しない場合および DREG=1 の場合は論理 1 に、DREG=0 の場合は論理 0 に接続します。
CEM	入力	1	乗算レジスタ (MREG=1) のアクティブ High のクロック イネーブルで、使用しない場合および MREG=1 の場合は論理 1 に、MREG=0 の場合は論理 0 に接続します。
CEOPMODE	入力	1	OPMODE 入力レジスタ (OPMODEREG=1) のアクティブ High のクロック イネーブルで、使用しない場合および OPMODEREG=1 の場合は論理 1 に、OPMODEREG=0 の場合は論理 0 に接続します。

ポート名	方向	幅	機能
CEP	入力	1	出力ポートレジスタ (PREG=1) のアクティブ High のクロック イネーブルで、使用しない場合および PREG=1 の場合は論理 1 に、PREG=0 の場合は論理 0 に接続します。
CLK	入力	1	DSP48A1 クロック
D[17:0]	入力	18	前置加減算器への 18 ビット入力
M[35:0]	出力	36	ファブリックへのダイレクト乗算器データ出力。P を使用する場合は使用しないでください。
OPMODE	入力	8	<p>DSP48A1 の演算処理を選択する制御入力</p> <ul style="list-style-type: none"> <li>OPMODE[1:0]: 後置加減算器への X 入力のソースを指定します。 <ul style="list-style-type: none"> <li>0: すべて 0 を配置します (後置加減算器をディスエーブル)。</li> <li>1: 乗算器の積項を使用します。</li> <li>2: POUT 出力信号を使用します。</li> <li>3: 連結された D、B、A 入力信号を使用します。</li> </ul> </li> <li>OPMODE[3:2]: 後置加減算器への Z 入力のソースを指定します。 <ul style="list-style-type: none"> <li>0: 後置加減算器をディスエーブルにし、乗算器の積項を POUT に伝搬します。</li> <li>1: PCIN を使用します。</li> <li>2: POUT ポート (アキュムレータ) を使用します。</li> <li>3: C ポートを使用します。</li> </ul> </li> <li>OPMODE[4]: 前置加減算器を使用するように指定します。 <ul style="list-style-type: none"> <li>0: 前置加算器をバイパスして、ポート B のデータを直接乗算器に送ります。</li> <li>1: 前置加算器を使用し、乗算器に入力する前に B および D ポートの値を加算または減算します。</li> </ul> </li> <li>OPMODE[5]: キャリー入力の値を後置加算器に送ります。これは CARRYINSEL = OPMODE5 のときにのみ適用されます。</li> <li>OPMODE[6]: 前置加減算器が加算器なのか減算器なのかを指定します。 <ul style="list-style-type: none"> <li>0: 加算を実行します。</li> <li>1: 減算を実行します。</li> </ul> </li> <li>OPMODE[7]: 後置加減算器が加算器なのか減算器なのかを指定します。 <ul style="list-style-type: none"> <li>0: 加算を実行します。</li> <li>1: 減算を実行します。</li> </ul> </li> </ul>
P[47:0]	出力	48	プライマリ データ出力
PCIN[47:0]	入力	48	ポート P のカスケード入力で、カスケード接続されている上位の DSP48A1 の PCOUT に接続します。使用しない場合は、ポートをすべて 0 にします。

ポート名	方向	幅	機能
PCOUT[47:0]	出力	48	ポート P のカスケード出力で、カスケード接続されている下位の DSP48A1 の PCIN に接続します。使用しない場合は未接続にします。
RSTA	入力	1	A ポートレジスタ (A0REG=1 または A1REG=1) のアクティブ High のリセットで、使用しない場合は論理 0 に接続します。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。
RSTB	入力	1	B ポートレジスタ (B0REG=1 または B1REG=1) のアクティブ High のリセットで、使用しない場合は論理 0 に接続します。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。
RSTC	入力	1	C ポートレジスタ (CREG=1) のアクティブ High のリセットで、使用しない場合は論理 0 に接続します。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。
RSTCARRYIN	入力	1	キャリー入力レジスタ (CARRYINREG =1) のアクティブ High のリセットで、使用しない場合は論理 0 に接続します。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。
RSTD	入力	1	D ポートレジスタ (DREG=1) のアクティブ High のリセットで、使用しない場合は論理 0 に接続します。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。
RSTM	入力	1	乗算器レジスタ (MREG=1) のアクティブ High のリセットで、使用しない場合は論理 0 に接続します。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。
RSTOPMODE	入力	1	OPMODE レジスタ (OPMODEREG=1) のアクティブ High のリセットで、使用しない場合は論理 0 に接続します。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。
RSTP	入力	1	P 出力レジスタ (PREG=1) のアクティブ High のリセットで、使用しない場合は論理 0 に接続します。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
A0REG	整数	0、1	0	1 段目の A 入力パイプライン レジスタを使用するかどうかを指定します。使用する場合は 1 に設定します。
A1REG	整数	0、1	1	2 段目の A 入力パイプライン レジスタを使用するかどうかを指定します。使用する場合は 1 に設定します。
B0REG	整数	0、1	0	1 段目の B 入力パイプライン レジスタを使用するかどうかを指定します。使用する場合は 1 に設定します。
B1REG	整数	0、1	1	2 段目の B 入力パイプライン レジスタを使用するかどうかを指定します。使用する場合は 1 に設定します。2 段目の B パイプライン レジスタは前置加算器の後ろに配置されます。
CARRYINREG	整数	0、1	1	CARRYIN 入力パイプライン レジスタを使用するかどうかを指定します。使用する場合は 1 に設定します。
CARRYINSEL	文字列	"CARRYIN"、 "OPMODE5"	"OPMODE5"	後置加減算器のキャリー入力信号を別の DSP48A1 の CARRYOUT ピンに接続されている CARRYIN ピンから送るか、OPMODE[5] 入力を使用して FPGA から直接制御するかを指定します。
CARRYOUTREG	整数	1、0	1	キャリー出力パイプライン レジスタを使用するかどうかを指定します。使用する場合は 1 に設定します。レジスタが付けられた出力には CARRYOUT および CARRYOUTF があります。
CREG	整数	1、0	1	C 入力パイプライン レジスタを使用するかどうかを指定します。使用する場合は 1 に設定します。
DREG	整数	1、0	1	D 前置加算器入力パイプラインレジスタを使用するかどうかを指定します。使用する場合は 1 に設定します。
MREG	整数	1、0	1	M 乗算器出力パイプライン レジスタを使用するかどうかを指定します。使用する場合は 1 に設定します。
OPMODEREG	整数	1、0	1	OPMODE 入力パイプライン レジスタの使用を使用するかどうかを指定します。使用する場合は 1 に設定します。
PREG	整数	1、0	1	P 出力パイプラインレジスタを使用するかどうかを指定します。使用する場合は 1 に設定します。レジスタが付けられた出力が P および PCOUT に含められます。
RSTTYPE	文字列	"SYNC"、 "ASYN"	"SYNC"	リセットすべてを同期リセットにするか非同期リセットにするかを指定します。タイミングの向上と回路の安定性の点から、非同期リセットが必要でない限り "SYNC" に設定してください。

## VHDL 記述 (インスタンスエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;
```



```

-- DSP48A1: 48-bit Multi-Functional Arithmetic Block
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

DSP48A1_inst : DSP48A1
generic map (
  A0REG => 0,          -- First stage A input pipeline register (0/1)
  A1REG => 1,          -- Second stage A input pipeline register (0/1)
  B0REG => 0,          -- First stage B input pipeline register (0/1)
  B1REG => 1,          -- Second stage B input pipeline register (0/1)
  CARRYINREG => 1,     -- CARRYIN input pipeline register (0/1)
  CARRYINSEL => "OPMODE5", -- Specify carry-in source, "CARRYIN" or "OPMODE5"
  CARRYOUTREG => 1,    -- CARRYOUT output pipeline register (0/1)
  CREG => 1,          -- C input pipeline register (0/1)
  DREG => 1,          -- D pre-adder input pipeline register (0/1)
  MREG => 1,          -- M pipeline register (0/1)
  OPMODEREG => 1,     -- Enable=1/disable=0 OPMODE input pipeline registers
  PREG => 1,          -- P output pipeline register (0/1)
  RSTTYPE => "SYNC"   -- Specify reset type, "SYNC" or "ASYN"
)
port map (
  -- Cascade Ports: 18-bit (each) output: Ports to cascade from one DSP48 to another
  BCOUT => BCOUT,      -- 18-bit output: B port cascade output
  PCOUT => PCOUT,      -- 48-bit output: P cascade output (if used, connect to PCIN of another DSP48A1)
  -- Data Ports: 1-bit (each) output: Data input and output ports
  CARRYOUT => CARRYOUT, -- 1-bit output: carry output (if used, connect to CARRYIN pin of another
                        -- DSP48A1)

  CARRYOUTF => CARRYOUTF, -- 1-bit output: fabric carry output
  M => M,                -- 36-bit output: fabric multiplier data output
  P => P,                -- 48-bit output: data output
  -- Cascade Ports: 48-bit (each) input: Ports to cascade from one DSP48 to another
  PCIN => PCIN,          -- 48-bit input: P cascade input (if used, connect to PCOUT of another DSP48A1)
  -- Control Input Ports: 1-bit (each) input: Clocking and operation mode
  CLK => CLK,            -- 1-bit input: clock input
  OPMODE => OPMODE,      -- 8-bit input: operation mode input
  -- Data Ports: 18-bit (each) input: Data input and output ports
  A => A,                -- 18-bit input: A data input
  B => B,                -- 18-bit input: B data input (connected to fabric or BCOUT of adjacent DSP48A1)
  C => C,                -- 48-bit input: C data input
  CARRYIN => CARRYIN,    -- 1-bit input: carry input signal (if used, connect to CARRYOUT pin of another
                        -- DSP48A1)

  D => D,                -- 18-bit input: B pre-adder data input
  -- Reset/Clock Enable Input Ports: 1-bit (each) input: Reset and enable input ports
  CEA => CEA,            -- 1-bit input: active high clock enable input for A registers
  CEB => CEB,            -- 1-bit input: active high clock enable input for B registers
  CEC => CEC,            -- 1-bit input: active high clock enable input for C registers
  CECARRYIN => CECARRYIN, -- 1-bit input: active high clock enable input for CARRYIN registers
  CED => CED,            -- 1-bit input: active high clock enable input for D registers
  CEM => CEM,            -- 1-bit input: active high clock enable input for multiplier registers
  CEOPMODE => CEOPMODE,  -- 1-bit input: active high clock enable input for OPMODE registers
  CEP => CEP,            -- 1-bit input: active high clock enable input for P registers
  RSTA => RSTA,          -- 1-bit input: reset input for A pipeline registers
  RSTB => RSTB,          -- 1-bit input: reset input for B pipeline registers
  RSTC => RSTC,          -- 1-bit input: reset input for C pipeline registers
  RSTCARRYIN => RSTCARRYIN, -- 1-bit input: reset input for CARRYIN pipeline registers
  RSTD => RSTD,          -- 1-bit input: reset input for D pipeline registers
  RSTM => RSTM,          -- 1-bit input: reset input for M pipeline registers
  RSTOPMODE => RSTOPMODE, -- 1-bit input: reset input for OPMODE pipeline registers
  RSTP => RSTP,          -- 1-bit input: reset input for P pipeline registers
);

-- End of DSP48A1_inst instantiation

```

## Verilog 記述 (インスタンス化)

```
// DSP48A1: 48-bit Multi-Functional Arithmetic Block
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

DSP48A1 #(
    .AOREG(0),           // First stage A input pipeline register (0/1)
    .A1REG(1),           // Second stage A input pipeline register (0/1)
    .BOREG(0),           // First stage B input pipeline register (0/1)
    .B1REG(1),           // Second stage B input pipeline register (0/1)
    .CARRYINREG(1),      // CARRYIN input pipeline register (0/1)
    .CARRYINSEL("OPMODE5"), // Specify carry-in source, "CARRYIN" or "OPMODE5"
    .CARRYOUTREG(1),      // CARRYOUT output pipeline register (0/1)
    .CREG(1),            // C input pipeline register (0/1)
    .DREG(1),            // D pre-adder input pipeline register (0/1)
    .MREG(1),            // M pipeline register (0/1)
    .OPMODEREG(1),       // Enable=1/disable=0 OPMODE input pipeline registers
    .PREG(1),            // P output pipeline register (0/1)
    .RSTTYPE("SYNC")     // Specify reset type, "SYNC" or "ASYN"
)
DSP48A1_inst (
    // Cascade Ports: 18-bit (each) output: Ports to cascade from one DSP48 to another
    .BCOUT(BCOUT),       // 18-bit output: B port cascade output
    .PCOUT(PCOUT),       // 48-bit output: P cascade output (if used, connect to PCIN of another DSP48A1)
    // Data Ports: 1-bit (each) output: Data input and output ports
    .CARRYOUT(CARRYOUT), // 1-bit output: carry output (if used, connect to CARRYIN pin of another
                        // DSP48A1)

    .CARRYOUTF(CARRYOUTF), // 1-bit output: fabric carry output
    .M(M),                 // 36-bit output: fabric multiplier data output
    .P(P),                 // 48-bit output: data output
    // Cascade Ports: 48-bit (each) input: Ports to cascade from one DSP48 to another
    .PCIN(PCIN),          // 48-bit input: P cascade input (if used, connect to PCOUT of another DSP48A1)
    // Control Input Ports: 1-bit (each) input: Clocking and operation mode
    .CLK(CLK),            // 1-bit input: clock input
    .OPMODE(OPMODE),      // 8-bit input: operation mode input
    // Data Ports: 18-bit (each) input: Data input and output ports
    .A(A),                // 18-bit input: A data input
    .B(B),                // 18-bit input: B data input (connected to fabric or BCOUT of adjacent DSP48A1)
    .C(C),                // 48-bit input: C data input
    .CARRYIN(CARRYIN),    // 1-bit input: carry input signal (if used, connect to CARRYOUT pin of another
                        // DSP48A1)

    .D(D),                // 18-bit input: B pre-adder data input
    // Reset/Clock Enable Input Ports: 1-bit (each) input: Reset and enable input ports
    .CEA(CEA),            // 1-bit input: active high clock enable input for A registers
    .CEB(CEB),            // 1-bit input: active high clock enable input for B registers
    .CEC(CEC),            // 1-bit input: active high clock enable input for C registers
    .CECARRYIN(CECARRYIN), // 1-bit input: active high clock enable input for CARRYIN registers
    .CED(CED),            // 1-bit input: active high clock enable input for D registers
    .CEM(CEM),            // 1-bit input: active high clock enable input for multiplier registers
    .CEOPMODE(CEOPMODE),  // 1-bit input: active high clock enable input for OPMODE registers
    .CEP(CEP),            // 1-bit input: active high clock enable input for P registers
    .RSTA(RSTA),          // 1-bit input: reset input for A pipeline registers
    .RSTB(RSTB),          // 1-bit input: reset input for B pipeline registers
    .RSTC(RSTC),          // 1-bit input: reset input for C pipeline registers
    .RSTCARRYIN(RSTCARRYIN), // 1-bit input: reset input for CARRYIN pipeline registers
    .RSTD(RSTD),          // 1-bit input: reset input for D pipeline registers
    .RSTM(RSTM),          // 1-bit input: reset input for M pipeline registers
    .RSTOPMODE(RSTOPMODE), // 1-bit input: reset input for OPMODE pipeline registers
    .RSTP(RSTP),          // 1-bit input: reset input for P pipeline registers
);

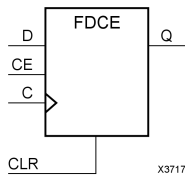
// End of DSP48A1_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)
- ・ [『Spartan-6 FPGA DSP48A1 スライス ユーザー ガイド』\(UG389\)](#)

## FDCE

### プリミティブ：D Flip-Flop with Clock Enable and Asynchronous Clear



### 概要

このデザイン エLEMENTは、クロック イネーブルと非同期クリアがある単一の D タイプ フリップフロップです。クロック イネーブル (CE) が High、非同期クリア (CLR) が Low の場合、クロック (C) が Low から High に切り替わるときにデータ入力 (D) の値がデータ出力 (Q) に送られます。CLR が High になると、ほかのすべての入力は無視され、出力 (Q) の値が Low にリセットされます。CE が Low の場合、クロック遷移は無視されます。

電力を供給すると、このフリップフロップは非同期にクリアされ、出力が Low になります。FPGA では、グローバル セット/リセット (GSR) をアクティブにすると、電源投入時の状態をシミュレーションできます。GSR のデフォルトはアクティブ High ですが、STARTUP\_architecture シンボルの GSR 入力の前にインバーターを追加するとアクティブ Low にできます。

### 論理表

入力				出力
CLR	CE	D	C	Q
1	X	X	X	0
0	0	X	X	変化なし
0	1	D	↑	D

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数	0、1	0	<p>コンフィギュレーション後の Q 出力の初期値を指定します。</p> <p>Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。このELEMENTでは、INIT 値を 0 にする必要があります。1 に設定する場合、この動作を表す非同期回路を作成する必要がありますが、ザイリンクスでは推奨しません。</p>

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
--       Clock Enable (posedge clk).
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

FDCE_inst : FDCE
generic map (
    INIT => '0') -- Initial value of register ('0' or '1')
port map (
    Q => Q,      -- Data output
    C => C,      -- Clock input
    CE => CE,    -- Clock enable input
    CLR => CLR,  -- Asynchronous clear input
    D => D       -- Data input
);

-- End of FDCE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
//       Clock Enable (posedge clk).
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

FDCE FDCE_inst (
    .Q(Q),      // 1-bit Data output
    .C(C),      // 1-bit Clock input
    .CE(CE),    // 1-bit Clock enable input
    .CLR(CLR),  // 1-bit Asynchronous clear input
    .D(D)       // 1-bit Data input
);

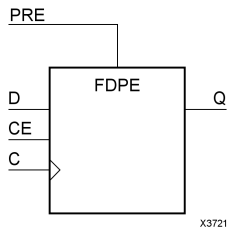
// End of FDCE_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## FDPE

### プリミティブ：D Flip-Flop with Clock Enable and Asynchronous Preset



## 概要

このデザイン エLEMENTは、データ (D)、クロック イネーブル (CE)、非同期プリセット (PRE) の各入力とデータ出力 (Q) がある単一の D フリップフロップです。非同期の PRE が High になると、ほかのすべての入力は無視され、Q 出力が High にセットされます。PRE が Low、CE が High の場合、クロック (C) が Low から High に切り替わる時に D 入力の値がフリップフロップにロードされます。CE が Low の場合、クロック遷移は無視されます。

FPGA では、電力を供給すると、フリップフロップは非同期にプリセットされ、出力が High になります。グローバル セット/リセット (GSR) をアクティブにすると、電源投入時の状態をシミュレーションできます。GSR のデフォルトはアクティブ High ですが、STARTUP\_architecture シンボルの GSR 入力の前にインバーターを追加するとアクティブ Low にできます。

## 論理表

入力				出力
PRE	CE	D	C	Q
1	X	X	X	1
0	0	X	X	変化なし
0	1	D	↑	D

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数	0、1	1	<p>コンフィギュレーション後の Q 出力の初期値を指定します。</p> <p>Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。このELEMENTでは、INIT 値を 1 にする必要があります。0 に設定する場合は、この動作を表す非同期回路が作成されます。</p>

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- FDPE: Single Data Rate D Flip-Flop with Asynchronous Preset and
--       Clock Enable (posedge clk).
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

FDPE_inst : FDPE
generic map (
    INIT => '0') -- Initial value of register ('0' or '1')
port map (
    Q => Q,        -- Data output
    C => C,        -- Clock input
    CE => CE,      -- Clock enable input
    PRE => PRE,    -- Asynchronous preset input
    D => D         -- Data input
);

-- End of FDPE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// FDPE: Single Data Rate D Flip-Flop with Asynchronous Preset and
//       Clock Enable (posedge clk).
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

FDPE FDPE_inst (
    .Q(Q),        // 1-bit Data output
    .C(C),        // 1-bit Clock input
    .CE(CE),      // 1-bit Clock enable input
    .PRE(PRE),    // 1-bit Asynchronous preset input
    .D(D)         // 1-bit Data input
);

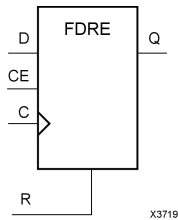
// End of FDPE_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## FDRE

プリミティブ：D Flip-Flop with Clock Enable and Synchronous Reset



### 概要

このデザイン エLEMENTは、データ (D)、クロック イネーブル (CE)、同期リセット (R) の各入力とデータ出力 (Q) がある単一の D タイプ フリップフロップです。同期リセット入力 (R) が High になると、ほかの入力は無視され、クロック (C) が Low から High に切り替わるときに出力 (Q) が Low にリセットされます。R が Low、CE が High の場合、クロックが Low から High に切り替わるときに D 入力の値がフリップフロップにロードされます。

電力を供給すると、このフリップフロップは非同期にクリアされ、出力が Low になります。FPGA では、グローバル セット/リセット (GSR) をアクティブにすると、電源投入時の状態をシミュレーションできます。GSR のデフォルトはアクティブ High ですが、STARTUP\_architecture シンボルの GSR 入力の前にインバーターを追加するとアクティブ Low にできます。

### 論理表

入力				出力
R	CE	D	C	Q
1	X	X	↑	0
0	0	X	X	変化なし
0	1	D	↑	D

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数	0、1	0	<p>コンフィギュレーション後の Q 出力の初期値を指定します。</p> <p>Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。このELEMENTでは、INIT 値を 0 にする必要があります。1 に設定する場合、この動作を表す非同期回路を作成する必要がありますが、ザイリンクスでは推奨しません。</p>

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- FDRE: Single Data Rate D Flip-Flop with Synchronous Reset and
--       Clock Enable (posedge clk).
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

FDRE_inst : FDRE
generic map (
    INIT => '0') -- Initial value of register ('0' or '1')
port map (
    Q => Q,        -- Data output
    C => C,        -- Clock input
    CE => CE,      -- Clock enable input
    R => R,        -- Synchronous reset input
    D => D         -- Data input
);

-- End of FDRE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// FDRE: Single Data Rate D Flip-Flop with Synchronous Reset and
//       Clock Enable (posedge clk).
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

FDRE FDRE_inst (
    .Q(Q),        // 1-bit Data output
    .C(C),        // 1-bit Clock input
    .CE(CE),      // 1-bit Clock enable input
    .R(R),        // 1-bit Synchronous reset input
    .D(D)         // 1-bit Data input
);

// End of FDRE_inst instantiation
```

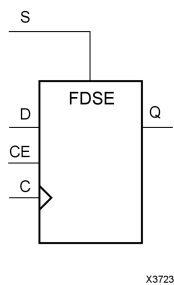
## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## FDSE

### プリミティブ：D Flip-Flop with Clock Enable and Synchronous Set



### 概要

FDSE は、データ (D)、クロック イネーブル (CE)、同期セット (S) の各入力とデータ出力 (Q) がある単一の D タイプ フリップフロップです。同期セット (S) 入力が高になると、クロック イネーブル (CE) 入力は無視され、クロック (C) が Low から High に切り替わる時に Q 出力が高にセットされます。S が Low、CE が High の場合、クロック (C) が Low から High に切り替わる時に D 入力の値がフリップフロップにロードされます。

FPGA では、電力を供給すると、フリップフロップは非同期にプリセットされ、出力が高になります。グローバル セット/リセット (GSR) をアクティブにすると、電源投入時の状態をシミュレーションできます。GSR のデフォルトはアクティブ High ですが、STARTUP\_architecture シンボルの GSR 入力の前にインバーターを追加するとアクティブ Low にできます。

### 論理表

入力				出力
S	CE	D	C	Q
1	X	X	↑	1
0	0	X	X	変化なし
0	1	D	↑	D

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数	0、1	1	<p>コンフィギュレーション後の Q 出力の初期値を指定します。</p> <p>Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。このエレメントでは、INIT 値を 1 にする必要があります。0 に設定する場合は、この動作を表す非同期回路が作成されます。</p>

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- FDSE: Single Data Rate D Flip-Flop with Synchronous Set and
--       Clock Enable (posedge clk).
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

FDSE_inst : FDSE
generic map (
  INIT => '0') -- Initial value of register ('0' or '1')
port map (
  Q => Q,      -- Data output
  C => C,      -- Clock input
  CE => CE,    -- Clock enable input
  S => S,      -- Synchronous Set input
  D => D       -- Data input
);

-- End of FDSE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// FDSE: Single Data Rate D Flip-Flop with Synchronous Set and
//       Clock Enable (posedge clk).
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

FDSE FDSE_inst (
  .Q(Q),      // 1-bit Data output
  .C(C),      // 1-bit Clock input
  .CE(CE),    // 1-bit Clock enable input
  .S(S),      // 1-bit Synchronous set input
  .D(D)       // 1-bit Data input
);

// End of FDSE_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

**プリミティブ** : Dual Gigabit Transceiver

[illegible]

## 概要

このデザイン エLEMENTでは、消費電力を抑え、詳細にコンフィギュレーション可能なトランシーバーである Spartan®-6 FPGA RocketIO™ GTP トランシーバーです。このELEMENTの詳細は、『Spartan-6 FPGA RocketIO GTP トランシーバー ユーザー ガイド』を参照してください。GTPA1\_DUAL プリミティブをインスタンスエートするには、Spartan-6 FPGA RocketIO GTX Transceiver Wizard を使用してラッパーの生成するのが推奨される方法です。このウィザードは、ザイリックス CORE Generator™ ツールに含まれています。

## デザインの入力方法

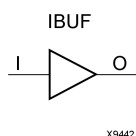
このELEMENTをインスタンスエートするには、Spartan-6 FPGA RocketIO GTX Transceiver Wizard またはこのELEMENTを含む関連コアを使用します。直接インスタンスエートしないでください。

## 詳細情報

- ・ [『Spartan-6 FPGA RocketIO GTP トランシーバー ユーザー ガイド』\(UG386\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## IBUF

プリミティブ：Input Buffer



## 概要

このデザイン エLEMENTは、最上位の入力ポートまたは入出力ポートに接続されている信号に自動的に挿入されます。このバッファは通常、合成ツールで推論されますが、必要に応じてインスタンス化することも可能です。インスタンス化するには、入力ポート (I) を関連する最上位の入力ポートまたは入出力ポートに接続し、出力ポート (O) をそのポートをソースとする FPGA ロジックに接続します。必要なジェネリック マップ (VHDL) またはパラメーター値代入 (Verilog) に変更を加えて、コンポーネントのデフォルトのビヘイビアーを変更します。

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファの出力
I	入力	1	バッファの入力

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

このELEMENTは通常、デザインの最上位入力ポートに対して合成ツールで推論されます。通常はソースコードで指定する必要はありませんが、必要に応じてインスタンス化できます。このコンポーネントをインスタンス化するには、下のインスタンス化コードを最上位エンティティ/モジュールに挿入します。デザイン階層を保つため、すべての I/O コンポーネントをデザインの最上位に配置してください。I ポートをデザインの最上位入力ポートに、O ポートをこの入力に供給されるロジックに直接接続します。generic/defparam 値を設定し、バッファのビヘイビアーを適切に設定してください。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	ELEMENTに I/O 規格を割り当てます。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUF: Single-ended Input Buffer
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

IBUF_inst : IBUF
generic map (
  IBUF_LOW_PWR => TRUE, -- Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
  IOSTANDARD => "DEFAULT")
port map (
  O => O,      -- Buffer output
  I => I       -- Buffer input (connect directly to top-level port)
);

-- End of IBUF_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUF: Single-ended Input Buffer
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

IBUF #(
  .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUF_inst (
  .O(O), // Buffer output
  .I(I)  // Buffer input (connect directly to top-level port)
);

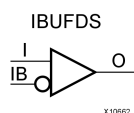
// End of IBUF_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## IBUFDS

### プリミティブ：Differential Signaling Input Buffer



### 概要

このデザイン エLEMENTは、低電圧差動信号を使用する入力バッファです。IBUFDS では、デザイン レベルのインターフェイス信号は、一方がマスターでもう一方がスレーブとなる 2 つの異なるポート (I、IB) で表されます。マスターとスレーブは MYNET\_P と MYNET\_N のように、同じ論理信号の反対の状態を示します。また、オプションの差動終端を使用すると、シグナル インテグリティが向上し、外部コンポーネントの数を削減できます。

### 論理表

入力		出力
I	IB	O
0	0	変化なし
0	1	0
1	0	1
1	1	変化なし

### ポートの説明

ポート名	方向	幅	機能
I	入力	1	Diff_p バッファの入力
IB	入力	1	Diff_p バッファの入力
O	出力	1	バッファの出力

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

デザイン階層を保つため、すべての I/O コンポーネントをデザインの最上位に配置してください。I ポートを直接デザインの最上位のマスターとなる入力ポートに、IB ポートを最上位のスレーブとなる入力ポートに、O ポートをこの入力供給されるロジックに接続します。generic/defparam 値を設定し、バッファのビヘイビアーを適切に設定してください。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	ビルトインの差動終端抵抗をイネーブルにします。
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFDS: Differential Input Buffer
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

IBUFDS_inst : IBUFDS
generic map (
    DIFF_TERM => FALSE, -- Differential Termination
    IBUF_LOW_PWR => TRUE, -- Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
    IOSTANDARD => "DEFAULT")
port map (
    O => O, -- Buffer output
    I => I, -- Diff_p buffer input (connect directly to top-level port)
    IB => IB -- Diff_n buffer input (connect directly to top-level port)
);

-- End of IBUFDS_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IBUFDS: Differential Input Buffer
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

IBUFDS #(
    .DIFF_TERM("FALSE"), // Differential Termination
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFDS_inst (
    .O(O), // Buffer output
    .I(I), // Diff_p buffer input (connect directly to top-level port)
    .IB(IB) // Diff_n buffer input (connect directly to top-level port)
);

// End of IBUFDS_inst instantiation
```

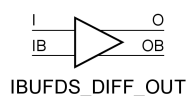
## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## IBUFDS\_DIFF\_OUT

**プリミティブ**：Signaling Input Buffer with Differential Output



X10107

### 概要

このデザイン エLEMENTは、差動信号を使用する入力バッファです。IBUFDS\_DIFF\_OUT では、デザイン レベルのインターフェイス信号は、一方がマスターでもう一方がスレーブとなる 2 つの異なるポート (I、IB) で表されます。マスターとスレーブは MYNET\_P と MYNET\_N のように、同じ論理信号の反対の状態を示します。IBUFDS\_DIFF\_OUT では、差動信号の両方の位相に内部アクセスできる点が IBUFDS と異なります。また、オプションの差動終端を使用すると、シグナル インテグリティが向上し、外部コンポーネントの数を削減できます。

### 論理表

入力		出力	
I	IB	O	OB
0	0	変化なし	変化なし
0	1	0	1
1	0	1	0
1	1	変化なし	変化なし

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

デザイン階層を保つため、すべての I/O コンポーネントをデザインの最上位に配置してください。I ポートを直接デザインの最上位のマスターとなる入力ポートに、IB ポートを最上位のスレーブとなる入力ポートに、O および OB ポートをこの入力が見供給されるロジックに接続します。generic/パラメーター値を設定し、バッファのビヘイビアを適切に設定してください。

### 使用可能な属性

属性	データ型	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	ELEMENTに I/O 規格を割り当てます。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFDS_DIFF_OUT: Differential Input Buffer with Differential Output
--                      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

IBUFDS_DIFF_OUT_inst : IBUFDS_DIFF_OUT
generic map (
    DIFF_TERM => FALSE, -- Differential Termination
    IOSTANDARD => "DEFAULT") -- Specify the input I/O standard
port map (
    O => O,      -- Buffer diff_p output
    OB => OB,     -- Buffer diff_n output
    I => I,      -- Diff_p buffer input (connect directly to top-level port)
    IB => IB     -- Diff_n buffer input (connect directly to top-level port)
);

-- End of IBUFDS_DIFF_OUT_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUFDS_DIFF_OUT: Differential Input Buffer with Differential Output
//                      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

IBUFDS_DIFF_OUT #(
    .DIFF_TERM("FALSE"), // Differential Termination, "TRUE"/"FALSE"
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFDS_DIFF_OUT_inst (
    .O(O), // Buffer diff_p output
    .OB(OB), // Buffer diff_n output
    .I(I), // Diff_p buffer input (connect directly to top-level port)
    .IB(IB) // Diff_n buffer input (connect directly to top-level port)
);

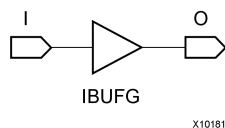
// End of IBUFDS_DIFF_OUT_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA クロック リソース ユーザー ガイド』 \(UG382\)](#)
- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』 \(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』 \(DS162\)](#)

## IBUFG

### プリミティブ：Dedicated Input Clock Buffer



### 概要

IBUFG は、FPGA への入力クロックをグローバル クロック配線リソースに接続するために使用する専用入力です。DCM、PLL、および BUFG への専用接続となり、デバイスのクロック遅延とジッターが最小限に抑えられます。IBUFG の入力は、グローバル クロック (GC) ピンでのみ駆動できます。

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	クロック バッファ出力
I	入力	1	クロック バッファ入力

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。

### VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFG: Single-ended global clock input buffer
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

IBUFG_inst : IBUFG
generic map (
    IBUF_LOW_PWR => TRUE, -- Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
    IOSTANDARD => "DEFAULT")
port map (
    O => O, -- Clock buffer output
    I => I -- Clock buffer input (connect directly to top-level port)
);
```

```
-- End of IBUFG_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IBUFG: Single-ended global clock input buffer
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

IBUFG #(
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFG_inst (
    .O(O), // Clock buffer output
    .I(I) // Clock buffer input (connect directly to top-level port)
);

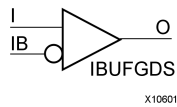
// End of IBUFG_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## IBUFGDS

**プリミティブ**：Differential Signaling Dedicated Input Clock Buffer and Optional Delay



### 概要

このデザイン エLEMENT は、クロック バッファ (BUFG) または MMCM に接続するための専用の差動信号入力バッファです。IBUFGDS では、デザイン レベルのインターフェイス信号は、一方がマスターでもう一方がスレーブとなる 2 つの異なるポート (I、IB) で表されます。マスターとスレーブは MYNET\_P と MYNET\_N のように、同じ論理信号の反対の状態を示します。また、オプションの差動終端を使用すると、シグナル インテグリティが向上し、外部コンポーネントの数を削減できます。デバイスへの入力データの遅延を調整する遅延 ELEMENT も含まれています。

### 論理表

入力		出力
I	IB	O
0	0	変化なし
0	1	0
1	0	1
1	1	変化なし

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	クロック バッファ出力
IB	入力	1	Diff_n クロック バッファの入力
I	入力	1	Diff_p クロック バッファの入力

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

デザイン階層を保つため、すべての I/O コンポーネントを必ずデザインの最上位に配置してください。I ポートを直接デザインの最上位のマスターとなる入力ポートに、IB ポートを最上位のスレーブとなる入力ポートに、O ポートをこの入力をソースとする MMCM、BUFG、またはロジックに接続してください。一部の合成ツールでは、IBUFG を FPGA のクロックリソースに接続すると、必要に応じて BUFG が自動的に推論されます。generic/defparam 値を設定し、バッファのビヘイビアーを適切に設定してください。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFGDS: Differential Global Clock Input Buffer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

IBUFGDS_inst : IBUFGDS
generic map (
    DIFF_TERM => FALSE, -- Differential Termination
    IBUF_LOW_PWR => TRUE, -- Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
    IOSTANDARD => "DEFAULT")
port map (
    O => O, -- Clock buffer output
    I => I, -- Diff_p clock buffer input (connect directly to top-level port)
    IB => IB -- Diff_n clock buffer input (connect directly to top-level port)
);

-- End of IBUFGDS_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUFGDS: Differential Global Clock Input Buffer
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

IBUFGDS #(
    .DIFF_TERM("FALSE"), // Differential Termination
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFGDS_inst (
    .O(O), // Clock buffer output
    .I(I), // Diff_p clock buffer input (connect directly to top-level port)
    .IB(IB) // Diff_n clock buffer input (connect directly to top-level port)
);

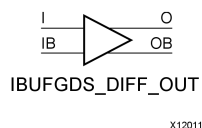
// End of IBUFGDS_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## IBUFGDS\_DIFF\_OUT

**プリミティブ：Differential Signaling Input Buffer with Differential Output**



### 概要

このデザイン エLEMENTは、差動信号を使用する入力バッファです。IBUFGDS\_DIFF\_OUT では、デザイン レベルのインターフェイス信号は、一方がマスターでもう一方がスレーブとなる 2 つの異なるポート (I、IB) で表されます。マスターとスレーブは MYNET\_P と MYNET\_N のように、同じ論理信号の反対の状態を示します。IBUFGDS\_DIFF\_OUT は、差動信号の両方の位相に内部アクセスできる点が IBUFGDS と異なります。また、オプションの差動終端を使用すると、シグナル インテグリティが向上し、外部コンポーネントの数を削減できます。

### 論理表

入力		出力	
I	IB	O	OB
0	0	変化なし	変化なし
0	1	0	1
1	0	1	0
1	1	変化なし	変化なし

### ポートの説明

ポート名	方向	幅	機能
I	入力	1	Diff_p バッファ入力 (デザインの最上位ポートに接続)
IB	入力	1	Diff_n バッファ入力 (デザインの最上位ポートに接続)
O	出力	1	Diff_p バッファ出力
OB	出力	1	Diff_n バッファ出力

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

デザイン階層を保つため、すべての I/O コンポーネントをデザインの最上位に配置してください。I ポートを直接デザインの最上位のマスターとなる入力ポートに、IB ポートを最上位のスレーブとなる入力ポートに、O および OB ポートをこの入力に供給されるロジックに接続します。generic/パラメーター値を設定し、バッファのビヘイビアーを適切に設定してください。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	内部差動終端抵抗を使用するかどうかを指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFGDS_DIFF_OUT: Differential Global Clock Buffer with Differential Output
--                      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

IBUFGDS_DIFF_OUT_inst : IBUFGDS_DIFF_OUT
generic map (
  DIFF_TERM => FALSE, -- Differential Termination
  IOSTANDARD => "DEFAULT") -- Specify the input I/O standard
port map (
  O => O,      -- Buffer diff_p output
  OB => OB,    -- Buffer diff_n output
  I => I,      -- Diff_p buffer input (connect directly to top-level port)
  IB => IB     -- Diff_n buffer input (connect directly to top-level port)
);

-- End of IBUFGDS_DIFF_OUT_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IBUFGDS_DIFF_OUT: Differential Global Clock Buffer with Differential Output
//                      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

IBUFGDS_DIFF_OUT #(
  .DIFF_TERM("FALSE"), // Differential Termination, "TRUE"/"FALSE"
  .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFGDS_DIFF_OUT_inst (
  .O(O), // Buffer diff_p output
  .OB(OB), // Buffer diff_n output
  .I(I), // Diff_p buffer input (connect directly to top-level port)
  .IB(IB) // Diff_n buffer input (connect directly to top-level port)
);

// End of IBUFGDS_DIFF_OUT_inst instantiation
```

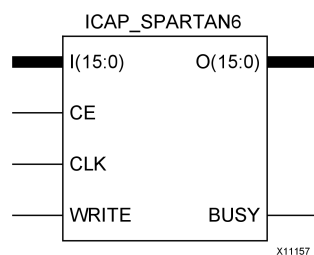
## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## ICAP\_SPARTAN6

### プリミティブ：Internal Configuration Access Port



### 概要

このデザイン エLEMENTを使用すると、FPGA ファブリックから FPGA のコンフィギュレーション機能にアクセスできます。FPGA アレイのコンフィギュレーション ロジックにコマンドおよびデータを書き込んだり、コンフィギュレーション ロジックからデータを読み出したりすることができます。この機能を不正に使用すると FPGA の機能および信頼性に悪影響を与えるため、この機能に精通していない場合はこのELEMENTを使用しないでください。

### ポートの説明

ポート名	方向	幅	機能
BUSY	出力	1	Busy/Ready 出力
CE	入力	1	アクティブ Low の ICAP イネーブル入力
CLK	入力	1	クロック入力
I[15:0]	入力	16	コンフィギュレーション データ入力バス
O[15:0]	出力	16	コンフィギュレーション データ出力バス
WRITE	入力	1	読み出し/書き込みクロック入力

### デザインの入力方法

インスタンシエーション	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DEVICE_ID	16 進数	32'h02000093、 32'h0200E093、 32'h0201D093、 32'h0202E093、 32'h0203D093、 32'h02001093、 32'h02002093、 32'h02004093、 32'h02008093、 32'h02011093、 32'h02024093、 32'h02028093、 32'h02031093	32'h02000093	シミュレーションで使用するあらかじめプログラムされているデバイス ID 値を指定します。
SIM_CFG_FILE_NAME	文字列	ファイルの名前と場所	なし	シミュレーション モデルで解析するロービット ファイル (RBT) を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ICAP_SPARTAN6: Internal Configuration Access Port
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

ICAP_SPARTAN6_inst : ICAP_SPARTAN6
generic map (
    DEVICE_ID => X"4000093",    -- Specifies the pre-programmed Device ID value
    SIM_CFG_FILE_NAME => "NONE" -- Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
)
port map (
    BUSY => BUSY,    -- 1-bit output: Busy/Ready output
    O => O,           -- 16-bit output: Configuration data output bus
    CE => CE,         -- 1-bit input: Active-Low ICAP Enable input
    CLK => CLK,       -- 1-bit input: Clock input
    I => I,           -- 16-bit input: Configuration data input bus
    WRITE => WRITE    -- 1-bit input: Read/Write control input
);

-- End of ICAP_SPARTAN6_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ICAP_SPARTAN6: Internal Configuration Access Port
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

ICAP_SPARTAN6 #(
    .DEVICE_ID(0'h4000093),    // Specifies the pre-programmed Device ID value
    .SIM_CFG_FILE_NAME("NONE") // Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
)
ICAP_SPARTAN6_inst (
    .BUSY(BUSY),    // 1-bit output: Busy/Ready output
    .O(O),          // 16-bit output: Configuration data output bus
    .CE(CE),        // 1-bit input: Active-Low ICAP Enable input
    .CLK(CLK),      // 1-bit input: Clock input

```

```
.I(I),           // 16-bit input: Configuration data input bus
.WRITE(WRITE)    // 1-bit input: Read/Write control input
);

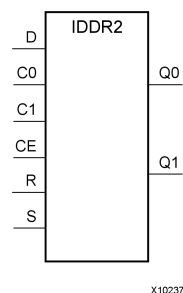
// End of ICAP_SPARTAN6_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュレーション ユーザー ガイド』\(UG380\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## IDDR2

**プリミティブ：Double Data Rate Input D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset**



### 概要

このデザイン エLEMENTは、ザイリンクス FPGA で外部デュアル データ レート (DDR) 信号を受信するための専用入力レジスタです。C0 と C1 の 2 つのクロックを使用してコンポーネントに接続されるので、C0 および C1 の両方の立ち上がりエッジでデータが取り込まれます。IDDR2 は、レジスタの動作を停止するために使用できるアクティブ High のクロック イネーブル (CE) ポート、対応するクロックに同期または非同期になるよう設定できるセット/リセット ポートを備えています。また、オプションの調整機能を使用すると、コンポーネントへの両方の出力データ ポートを 1 つのクロックに揃えることができます。

### 論理表

入力						出力	
S	R	CE	D	C0	C1	Q0	Q1
1	X	X	X	X	X	INIT_Q0	INIT_Q1
0	1	X	X	X	X	not INIT_Q0	not INIT_Q1
0	0	0	X	X	X	変化なし	変化なし
0	0	1	D	↑	X	D	変化なし
0	0	1	D	X	↑	変化なし	D

セット/リセットは SRTYPE 値で同期に設定可能

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

デフォルトの動作を変更するには、ジェネリック マップ (VHDL) またはパラメーター値代入 (Verilog) を使用して、インスタンス化されたコンポーネントの一部として属性を変更します。IDDR2 は、入力バッファが推論されるデザインの最上位入力ポートに接続するか、インスタンス化された IBUF、IOBUF、IBUFDS、または IOBUFDS のいずれかに直接接続できます。このコンポーネントのすべての入力と出力は、接続しておく必要があります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DDR_ALIGNMENT	文字列	"NONE"、"C0"、 "C1"	"NONE"	DDR レジスタの出力アライメントを設定します。 <ul style="list-style-type: none"> <li>"NONE"：対応する C0 または C1 の立ち上がりエッジの直後に Q0 および Q1 にデータが出力されます。</li> <li>"C0"：Q0 と Q1 両方のデータが C0 クロックの立ち上がりエッジに同期します。</li> <li>"C1"：Q0 と Q1 両方のデータが C1 クロックの立ち上がりエッジに同期します。</li> </ul>
INIT_Q0	整数	0、1	0	Q0 出力の初期値を 0 または 1 に設定します。
INIT_Q1	整数	0、1	0	Q1 出力の初期値を 0 または 1 に設定します。
SRTYPE	文字列	"SYNC"、"ASYNC"	"SYNC"	セット/リセットを同期または非同期に設定します。

## VHDL 記述 (インスタンスエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IDDR2: Input Double Data Rate Input Register with Set, Reset
--       and Clock Enable.
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

IDDR2_inst : IDDR2
generic map(
  DDR_ALIGNMENT => "NONE", -- Sets output alignment to "NONE", "C0", "C1"
  INIT_Q0 => '0', -- Sets initial state of the Q0 output to '0' or '1'
  INIT_Q1 => '0', -- Sets initial state of the Q1 output to '0' or '1'
  SRTYPE => "SYNC") -- Specifies "SYNC" or "ASYNC" set/reset
port map (
  Q0 => Q0, -- 1-bit output captured with C0 clock
  Q1 => Q1, -- 1-bit output captured with C1 clock
  C0 => C0, -- 1-bit clock input
  C1 => C1, -- 1-bit clock input
  CE => CE, -- 1-bit clock enable input
  D => D,   -- 1-bit data input
  R => R,   -- 1-bit reset input
  S => S    -- 1-bit set input
);

-- End of IDDR2_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IDDR2: Input Double Data Rate Input Register with Set, Reset
//      and Clock Enable.
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

IDDR2 #(
    .DDR_ALIGNMENT("NONE"), // Sets output alignment to "NONE", "C0" or "C1"
    .INIT_Q0(1'b0), // Sets initial state of the Q0 output to 1'b0 or 1'b1
    .INIT_Q1(1'b0), // Sets initial state of the Q1 output to 1'b0 or 1'b1
    .SRTYPE("SYNC") // Specifies "SYNC" or "ASYN" set/reset
) IDDR2_inst (
    .Q0(Q0), // 1-bit output captured with C0 clock
    .Q1(Q1), // 1-bit output captured with C1 clock
    .C0(C0), // 1-bit clock input
    .C1(C1), // 1-bit clock input
    .CE(CE), // 1-bit clock enable input
    .D(D),   // 1-bit DDR data input
    .R(R),   // 1-bit reset input
    .S(S)    // 1-bit set input
);

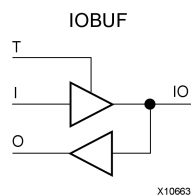
// End of IDDR2_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## IOBUF

プリミティブ：Bi-Directional Buffer



## 概要

このデザイン エLEMENTは双方向でシングルエンドの I/O バッファで、内部ロジックを外部双方向ピンに接続する場合に使用します。

## 論理表

入力		双方向	出力
T	I	I/O	O
1	X	Z	I/O
0	1	1	1
0	0	0	0

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファの出力
IO	入出力	1	バッファの入出力
I	入力	1	バッファの入力
T	入力	1	トライステート イネーブル入力

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	I/O 規格として LVTTTL、LVC MOS12、LVC MOS15、LVC MOS18、LVC MOS25 または LVC MOS33 を使用する SelectIO™ バッファの出力の駆動電流 (mA) を選択します。
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。
SLEW	文字列	"SLOW"、"FAST"、 "QUIETIO"	"SLOW"	出力の立ち上がり時間と立ち下がり時間を設定します。この属性の最適な設定方法は、データシートを参照してください。

## VHDL 記述 (インスタンスエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IOBUF: Single-ended Bi-directional Buffer
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

IOBUF_inst : IOBUF
generic map (
    DRIVE => 12,
    IOSTANDARD => "DEFAULT",
    SLEW => "SLOW")
port map (
    O => O,      -- Buffer output
    IO => IO,    -- Buffer inout port (connect directly to top-level port)
    I => I,      -- Buffer input
    T => T      -- 3-state enable input, high=input, low=output
);

-- End of IOBUF_inst instantiation
```

## Verilog 記述 (インスタンスエーション)

```
// IOBUF: Single-ended Bi-directional Buffer
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

IOBUF #(
    .DRIVE(12), // Specify the output drive strength
    .IOSTANDARD("DEFAULT"), // Specify the I/O standard
    .SLEW("SLOW") // Specify the output slew rate
) IOBUF_inst (
    .O(O),      // Buffer output
    .IO(IO),    // Buffer inout port (connect directly to top-level port)
    .I(I),      // Buffer input
    .T(T)      // 3-state enable input, high=input, low=output
);

// End of IOBUF_inst instantiation
```

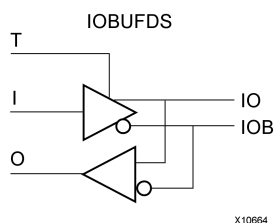


## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## IOBUFDS

プリミティブ：3-State Differential Signaling I/O Buffer with Active Low Output Enable



## 概要

このデザイン エLEMENTは、低電圧差動信号を使用する双方向バッファです。IOBUFDS では、デザインレベルのインターフェイス信号は、一方がマスターでもう一方がスレーブとなる 2 つの異なるポート (IO、IOB) で表されます。マスターとスレーブは MYNET\_P と MYNET\_N のように、同じ論理信号の反対の状態を示します。また、オプションの差動終端を使用すると、シグナル インテグリティが向上し、外部コンポーネントの数を削減できます。デバイスへの入力データの遅延を調整する遅延ELEMENTも含まれています。

## 論理表

入力		双方向		出力
I	T	I/O	IOB	O
X	1	Z	Z	変化なし
0	0	0	1	0
1	0	1	0	1

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファの出力
IO	入出力	1	Diff_p 入出力
IOB	入出力	1	Diff_n 入出力
I	入力	1	バッファの入力
T	入力	1	トリステート イネーブル入力

## デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IOBUFDS: Differential Bi-directional Buffer
--           Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 14.5

IOBUFDS_inst : IOBUFDS
generic map (
    IOSTANDARD => "BLVDS_25")
port map (
    O => O,      -- Buffer output
    IO => IO,    -- Diff_p inout (connect directly to top-level port)
    IOB => IOB,  -- Diff_n inout (connect directly to top-level port)
    I => I,      -- Buffer input
    T => T      -- 3-state enable input, high=input, low=output
);

-- End of IOBUFDS_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IOBUFDS: Differential Bi-directional Buffer
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

IOBUFDS #(
    .IOSTANDARD("BLVDS_25") // Specify the I/O standard
) IOBUFDS_inst (
    .O(O), // Buffer output
    .IO(IO), // Diff_p inout (connect directly to top-level port)
    .IOB(IOB), // Diff_n inout (connect directly to top-level port)
    .I(I), // Buffer input
    .T(T) // 3-state enable input, high=input, low=output
);

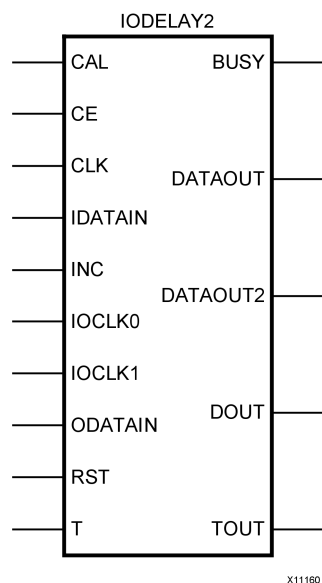
// End of IOBUFDS_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## IODELAY2

### プリミティブ：Input and Output Fixed or Variable Delay Element



### 概要

このデザイン エLEMENTは、Spartan®-6 FPGA の入力パスに固定遅延または可変遅延を、出力パスに固定遅延を追加します。この遅延は、チップへの入力データまたはチップからの出力データのデータ アライメントに便利です。VARIABLE モードでは、入力パスを調整して遅延の量を増分または減分できます。出力遅延パスは、固定遅延でのみ使用できます。IODELAY を使用して、FPGA の内部パスに一定の遅延または可変遅延を追加することも可能です。ただし、IODELAY をこのように使用する場合、入力パス遅延または出力パス遅延に関連する I/O には使用できなくなります。

### ポートの説明

ポート名	方向	幅	機能
BUSY	出力	1	キャリブレーション中
CAL	入力	1	キャリブレーションを開始
CE	入力	1	インクリメント/デクリメントをイネーブル
CLK	入力	1	IODELAY クロック入力
DATAOUT	出力	1	入力ポートからの遅延データ出力 (入力データパス ロジックに接続、ILOGIC 内のレジスタにのみ配線可能)
DATAOUT2	出力	1	入力ポートからの遅延されたデータ出力 (入力データパス ロジックに接続、FPGA に配線可能)
DOUT	出力	1	IOB への遅延データ出力
IDATAIN	入力	1	IOB からのデータ信号
INC	入力	1	インクリメント/デクリメント入力
IOCLK0	入力	1	反転可能な I/O クロック入力 (オプション)
IOCLK1	入力	1	反転可能な I/O クロック入力 (オプション)

ポート名	方向	幅	機能
ODATAIN	入力	1	OLOGIC または OSERDES からの出力データ入力
RST	入力	1	IODELAY2 を 0 または合計周期の 1/2 にリセットします。どちらにリセットするかは、RST_VALUE 属性で指定します。
T	入力	1	トライステート入力制御ピン。入力のみまたは内部遅延の場合は High に、出力のみの場合は Low にします。
TOUT	出力	1	遅延トライステート信号出力。入力のみまたは内部遅延の場合は High に、出力のみの場合は Low にします。

## デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
COUNTER_WRAP AROUND	文字列	"WRAPAROUND"、 "STAY_AT_LIMIT"	"WRAPAROUND"	タップ設定がインクリメントかデクリメントかにより、タップ カウントが最大値または最小値を超えたときのビヘイビアーを設定します。
DATA_RATE	文字列	"SDR"、"DDR"	"SDR"	シングル データレートまたはダブル データレートを指定します。
DELAY_SRC	文字列	"IO"、 "IDATAIN"、 "ODATAIN"	"IO"	<ul style="list-style-type: none"> <li>・ "ODATAIN"：遅延ソースを OSERDES または OLOGIC からの ODATAIN ピンに設定します。</li> <li>・ "IDATAIN"：遅延ソースを IDATAIN ピン（専用 IOB (P/N) パッドの 1 つ）に設定します。</li> <li>・ "IO"：T (トライステート) 入力の極性に基づいて、信号ソースが IDATAIN と ODATAIN の間で切り替わります。</li> </ul>
IDELAY_MODE	文字列	"NORMAL"、"PCI"	"NORMAL"	この属性を指定または変更しないでください。

属性	データ型	値	デフォルト	説明
IDELAY_TYPE	文字列	"DEFAULT"、 "DIFF_PHASE_DETECTOR"、 "FIXED"、 "VARIABLE_FROM_HALF_MAX"、 "VARIABLE_FROM_ZERO"	"DEFAULT"	遅延タイプを指定します。 "VARIABLE" は、ユーザー キャリブレーション遅延モードを示します。  <ul style="list-style-type: none"> <li>"DEFAULT"：ゼロ ホールド タイムのプログラムに最も近い物理チップ設定を使用します。</li> <li>"VARIABLE_FROM_ZERO" および "VARIABLE_FROM_HALF_MAX"：リセット動作を指定します。</li> <li>"DIFF_PHASE_DETECTOR"：マスターおよびスレーブ IODELAY2 がカスケードされた特殊なモードです。</li> </ul>
IDELAY_VALUE	整数	0 ～ 255	0	IDELAY モードでの遅延タップ値を指定します。
IDELAY2_VALUE	整数	0 ～ 255	0	IDELAY モードでの遅延タップ値を指定します。IDELAY_MODE が PCI に設定されている場合にのみ使用されます。
ODELAY_VALUE	整数	0 ～ 255	0	ODELAY モードでの遅延タップ値を指定します。
SERDES_MODE	文字列	"NONE"、 "MASTER"、 "SLAVE"	"NONE"	カスケード接続してデータ幅を拡張する場合に、ISERDES2 をマスター モードにするかスレーブ モードにするかを指定します。
SIM_TAPDELAY_VALUE	整数	10 ～ 90	75	シミュレーションのみの属性で、標準タップ遅延をシミュレーション用に別の設定に変更します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IODELAY2: Input and Output Fixed or Variable Delay Element
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

IODELAY2_inst : IODELAY2
generic map (
  COUNTER_WRAPAROUND => "WRAPAROUND", -- "STAY_AT_LIMIT" or "WRAPAROUND"
  DATA_RATE => "SDR",                -- "SDR" or "DDR"
  DELAY_SRC => "IO",                  -- "IO", "ODATAIN" or "IDATAIN"
  IDELAY2_VALUE => 0,                  -- Delay value when IDELAY_MODE="PCI" (0-255)
  IDELAY_MODE => "NORMAL",             -- "NORMAL" or "PCI"
  IDELAY_TYPE => "DEFAULT",            -- "FIXED", "DEFAULT", "VARIABLE_FROM_ZERO", "VARIABLE_FROM_HALF_MAX"
                                     -- or "DIFF_PHASE_DETECTOR"

```

```

IDELAY_VALUE => 0,          -- Amount of taps for fixed input delay (0-255)
ODELAY_VALUE => 0,          -- Amount of taps fixed output delay (0-255)
SERDES_MODE => "NONE",     -- "NONE", "MASTER" or "SLAVE"
SIM_TAPDELAY_VALUE => 75    -- Per tap delay used for simulation in ps
)
port map (
  BUSY => BUSY,             -- 1-bit output: Busy output after CAL
  DATAOUT => DATAOUT,     -- 1-bit output: Delayed data output to ISERDES/input register
  DATAOUT2 => DATAOUT2,   -- 1-bit output: Delayed data output to general FPGA fabric
  DOUT => DOUT,             -- 1-bit output: Delayed data output
  TOUT => TOUT,             -- 1-bit output: Delayed 3-state output
  CAL => CAL,               -- 1-bit input: Initiate calibration input
  CE => CE,                 -- 1-bit input: Enable INC input
  CLK => CLK,               -- 1-bit input: Clock input
  IDATAIN => IDATAIN,       -- 1-bit input: Data input (connect to top-level port or I/O buffer)
  INC => INC,               -- 1-bit input: Increment / decrement input
  IOCLK0 => IOCLK0,         -- 1-bit input: Input from the I/O clock network
  IOCLK1 => IOCLK1,         -- 1-bit input: Input from the I/O clock network
  ODATAIN => ODATAIN,       -- 1-bit input: Output data input from output register or OSERDES2.
  RST => RST,               -- 1-bit input: Reset to zero or 1/2 of total delay period
  T => T                    -- 1-bit input: 3-state input signal
);

-- End of IDELAY2_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// IDELAY2: Input and Output Fixed or Variable Delay Element
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

IDELAY2 #(
  .COUNTER_WRAPAROUND("WRAPAROUND"), // "STAY_AT_LIMIT" or "WRAPAROUND"
  .DATA_RATE("SDR"),                 // "SDR" or "DDR"
  .DELAY_SRC("IO"),                  // "IO", "ODATAIN" or "IDATAIN"
  .IDELAY2_VALUE(0),                 // Delay value when IDELAY_MODE="PCI" (0-255)
  .IDELAY_MODE("NORMAL"),             // "NORMAL" or "PCI"
  .IDELAY_TYPE("DEFAULT"),            // "FIXED", "DEFAULT", "VARIABLE_FROM_ZERO", "VARIABLE_FROM_HALF_MAX"
                                      // or "DIFF_PHASE_DETECTOR"
  .IDELAY_VALUE(0),                  // Amount of taps for fixed input delay (0-255)
  .ODELAY_VALUE(0),                  // Amount of taps fixed output delay (0-255)
  .SERDES_MODE("NONE"),               // "NONE", "MASTER" or "SLAVE"
  .SIM_TAPDELAY_VALUE(75)             // Per tap delay used for simulation in ps
)
IDELAY2_inst (
  .BUSY(BUSY),                       // 1-bit output: Busy output after CAL
  .DATAOUT(DATAOUT),                 // 1-bit output: Delayed data output to ISERDES/input register
  .DATAOUT2(DATAOUT2),               // 1-bit output: Delayed data output to general FPGA fabric
  .DOUT(DOUT),                       // 1-bit output: Delayed data output
  .TOUT(TOUT),                       // 1-bit output: Delayed 3-state output
  .CAL(CAL),                         // 1-bit input: Initiate calibration input
  .CE(CE),                           // 1-bit input: Enable INC input
  .CLK(CLK),                         // 1-bit input: Clock input
  .IDATAIN(IDATAIN),                 // 1-bit input: Data input (connect to top-level port or I/O buffer)
  .INC(INC),                         // 1-bit input: Increment / decrement input
  .IOCLK0(IOCLK0),                   // 1-bit input: Input from the I/O clock network
  .IOCLK1(IOCLK1),                   // 1-bit input: Input from the I/O clock network
  .ODATAIN(ODATAIN),                 // 1-bit input: Output data input from output register or OSERDES2.
  .RST(RST),                         // 1-bit input: Reset to zero or 1/2 of total delay period
  .T(T)                             // 1-bit input: 3-state input signal
);

// End of IDELAY2_inst instantiation

```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## IODRP2

プリミティブ：I/O Control Port

### 概要

ザイリンクスではこのエレメントの使用をサポートしていません。

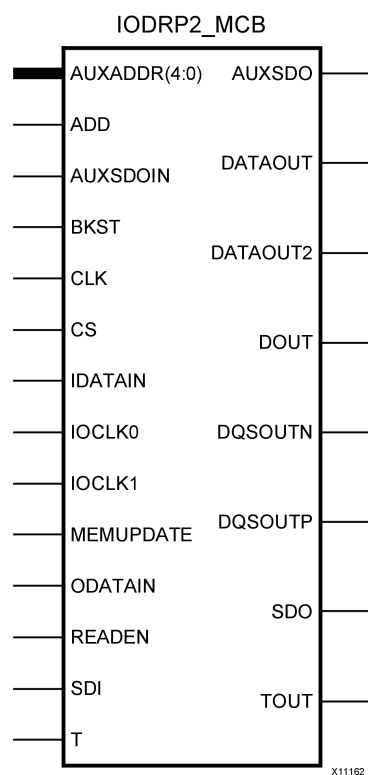
### 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## IODRP2\_MCB

プリミティブ : I/O Control Port for the Memory Controller Block



### 概要

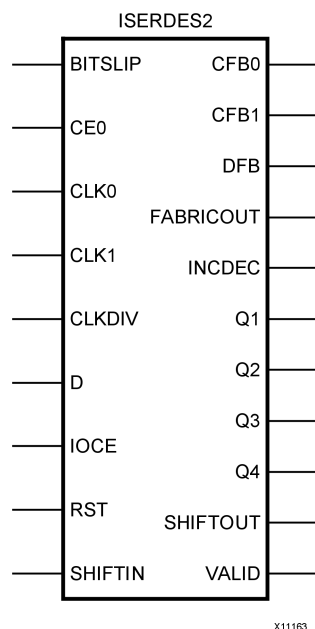
このデザイン エLEMENTは、外部メモリ インターフェイスをインプリメントするために MIG (Memory Interface Generator) コアで MCB ブロックと組み合わせて使用されます。MIG の外部での使用はサポートされていません。

### 詳細情報

- ・ [メモリ インターフェイス ソリューション ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

## ISERDES2

**プリミティブ**：Input SERial/DESerializer.



### 概要

各 IOB には入力デシリアライザー ブロックが含まれており、ISERDES2 プリミティブを使用してデザインにインスタンス化できます。ISERDES2 を使用すると、SerDes 比 1:2、1:3、および 1:4 のシリアル/パラレル変換が可能です。SerDes 比は、データをキャプチャする高速 I/O クロックと、それより低速のパラレル データの処理に使用する内部グローバルクロックとの比です。たとえば、500MHz で動作するシングル レート I/O クロックを使用して 500Mb/s でデータを受信する場合、ISERDES2 により 4 ビットのデータが 1/4 のレート (125MHz) で FPGA ロジックに転送されます。差動入力を使用する場合、2 つの IOB に関連付けられた 2 つの ISERDES2 プリミティブをカスケード接続して、SerDes 比 1:5、1:6、1:7、および 1:8 を達成できます。各 ISERDES2 には、パラレル データをワードで揃えるロジックも含まれています。このロジックは、ビットスリップ処理を実行する場合に必要です。

### ポートの説明

ポート名	方向	幅	機能
BITSLLIP	入力	1	High の場合、ビットスリップが実行されます。ビットスリップ処理は、カスケード接続されているかどうかにかかわらず、任意の DATA_WIDTH で使用できます。ビットスリップの量は、DATA_WIDTH の値によって決まります。
CE0	入力	1	最終 (グローバル クロック ドリブン) レジスタのクロック イネーブル入力
CFB0	出力	1	PLL/DCM で生成されたクロックを BUFIO2FB を介して PLL/DCM にフィードバックします。
CFB1	出力	1	PLL/DCM で生成されたクロックを BUFIO2FB を介して PLL/DCM にフィードバックします (セカンダリ)。
CLKDIV	入力	1	グローバル クロック ネットワーク入力。FPGA ロジックドメインのクロックです。

ポート名	方向	幅	機能
CLK0	入力	1	I/O クロック ネットワークの入力。オプションで反転可能です。クロック 2 通倍回路が使用されていない場合に使用されるプライマリクロック入力です (DATA_RATE 属性を参照)。
CLK1	入力	1	I/O クロック ネットワークの入力。オプションで反転可能です。クロック 2 通倍回路が使用されている場合のみに使用されるセカンダリクロック入力です (DATA_RATE 属性を参照)。
D	入力	1	データ入力。IODELAY2 ブロックによる遅延の後のデータ入力です。
DFB	出力	1	IODELAY2 エLEMENTで遅延した入力クロックを BUFIO2 を介して DCM、PLL、または BUFG に転送します。
FABRICOUT	出力	1	FPGA ロジックで使用される非同期データ。
INCDEC	出力	1	マスター モードでの位相検出器の出力です (スレーブ モードではダミー)。受信データのサンプリングのタイミングが早かったか遅かったかを FPGA ロジックに通知します。
IOCE	入力	1	BUFIO CE から派生するデータ ストローブ信号。選択されている SerDes モードでの I/O およびグローバル クロックに対し、正しいタイミングでデータがキャプチャされるようにします。
Q1 ~ Q4	出力	1	ハードウェアへのレジスタ付き出力信号。
RST	入力	1	非同期リセットのみ。
SHIFTIN	入力	1	マスター/スレーブ I/O のカスケード入力信号。DATA_WIDTH が 5 以上でマスターとスレーブのサイトが一緒に使用される場合に使用します。ブロックがマスターの場合、位相検出器モードで使用されるデータ入力を送信します。スレーブの場合は、パラレル データとなるシリアル データ入力を送信します。
SHIFTOUT	出力	1	マスター/スレーブ I/O のカスケード出力信号。サンプリングされたデータをスレーブから送信するのに使用します。マスターモードの場合、入力シフトレジスタの 4 段目からシリアル データをスレーブに送信します。
VALID	出力	1	マスター モードでの位相検出器の出力です (スレーブ モードではダミー)。入力データにエッジがない場合 (位相検出器で使用できる情報がない場合)、VALID 信号は Low になり、FPGA ロジックで INCDEC 信号が無視されることを示します。

## デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
BITSLIP_ENABLE	ブール代数	FALSE、TRUE	FALSE	BITSLIP 入力ピンで制御するビットスリップ機能のオン/オフを指定します。スリップするビット数は、DATA_WIDTH で選択されている値によって決まります。オフの場合、ビットスリップ CE は、IOCE クロック イネーブルの1 つ前の I/O クロックのデフォルト値になります。
DATA_RATE	文字列	"SDR"、"DDR"	"SDR"	データ レートを設定します。DDR クロックは、個別の複数の I/O クロックまたは 1 つの I/O クロックから供給できます。2 つのクロックが供給される場合、その位相差は約 180 度である必要があります。
DATA_WIDTH	整数	1、2、3、4、5、6、7、8	1	データ幅を指定します。シリアル/パラレル コンバーターのパラレル データ出力幅を定義します。5 以上の値は、2 つの ISERDES2 ブロックをカスケード接続する場合にのみ有効です。この場合、同じ値をマスター ブロックとスレーブ ブロックに適用する必要があります。
INTERFACE_TYPE	文字列	"NETWORKING"、 "NETWORKING_PIPELINED"、 "RETIMED"	"NETWORKING"	操作モードを選択し、どのセットのパラレル データが FPGA ロジックに使用可能であるかを決定します。
SERDES_MODE	文字列	"NONE"、 "MASTER"、 "SLAVE"	"NONE"	ISERDES を単独で使用するか、2 つの ISERDES2 ブロックをカスケード接続している場合はマスターまたはスレーブとして使用するかを指定します。

## VHDL 記述 (インスタンスーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- ISERDES2: Input SERIAL/DESerializer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

ISERDES2_inst : ISERDES2
generic map (
    BITSLIP_ENABLE => FALSE,          -- Enable Bitflip Functionality (TRUE/FALSE)
    DATA_RATE => "SDR",              -- Data-rate ("SDR" or "DDR")
    DATA_WIDTH => 1,                 -- Parallel data width selection (2-8)
    INTERFACE_TYPE => "NETWORKING",  -- "NETWORKING", "NETWORKING_PIPELINED" or "RETIMED"
    SERDES_MODE => "NONE"            -- "NONE", "MASTER" or "SLAVE"
)
port map (
    CFB0 => CFB0,                    -- 1-bit output: Clock feed-through route output
    CFB1 => CFB1,                    -- 1-bit output: Clock feed-through route output
    DFB => DFB,                      -- 1-bit output: Feed-through clock output
    FABRICOUT => FABRICOUT,          -- 1-bit output: Unsynchronized data output
    INCDEC => INCDEC,               -- 1-bit output: Phase detector output
    -- Q1 - Q4: 1-bit (each) output: Registered outputs to FPGA logic
    Q1 => Q1,
    Q2 => Q2,
    Q3 => Q3,
    Q4 => Q4,
    SHIFTOUT => SHIFTOUT,           -- 1-bit output: Cascade output signal for master/slave I/O
    VALID => VALID,                -- 1-bit output: Output status of the phase detector

```

```

BITSLLIP => BITSLLIP,      -- 1-bit input: Bitsllip enable input
CE0 => CE0,                -- 1-bit input: Clock enable input
CLK0 => CLK0,              -- 1-bit input: I/O clock network input
CLK1 => CLK1,              -- 1-bit input: Secondary I/O clock network input
CLKDIV => CLKDIV,          -- 1-bit input: FPGA logic domain clock input
D => D,                    -- 1-bit input: Input data
IOCE => IOCE,              -- 1-bit input: Data strobe input
RST => RST,                -- 1-bit input: Asynchronous reset input
SHIFTIN => SHIFTIN         -- 1-bit input: Cascade input signal for master/slave I/O
);

-- End of ISERDES2_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// ISERDES2: Input SERial/DESerializer
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

ISERDES2 #(
    .BITSLLIP_ENABLE("FALSE"),      // Enable Bitsllip Functionality (TRUE/FALSE)
    .DATA_RATE("SDR"),              // Data-rate ("SDR" or "DDR")
    .DATA_WIDTH(1),                 // Parallel data width selection (2-8)
    .INTERFACE_TYPE("NETWORKING"),  // "NETWORKING", "NETWORKING_PIPELINED" or "RETIMED"
    .SERDES_MODE("NONE")            // "NONE", "MASTER" or "SLAVE"
)
ISERDES2_inst (
    .CFB0(CFB0),                    // 1-bit output: Clock feed-through route output
    .CFB1(CFB1),                    // 1-bit output: Clock feed-through route output
    .DFB(DFB),                      // 1-bit output: Feed-through clock output
    .FABRICOUT(FABRICOUT),          // 1-bit output: Unsynchrnonized data output
    .INCDEC(INCDEC),                // 1-bit output: Phase detector output
    // Q1 - Q4: 1-bit (each) output: Registered outputs to FPGA logic
    .Q1(Q1),
    .Q2(Q2),
    .Q3(Q3),
    .Q4(Q4),
    .SHIFTOUT(SHIFTOUT),            // 1-bit output: Cascade output signal for master/slave I/O
    .VALID(VALID),                  // 1-bit output: Output status of the phase detector
    .BITSLLIP(BITSLLIP),            // 1-bit input: Bitsllip enable input
    .CE0(CE0),                      // 1-bit input: Clock enable input
    .CLK0(CLK0),                    // 1-bit input: I/O clock network input
    .CLK1(CLK1),                    // 1-bit input: Secondary I/O clock network input
    .CLKDIV(CLKDIV),                // 1-bit input: FPGA logic domain clock input
    .D(D),                          // 1-bit input: Input data
    .IOCE(IOCE),                    // 1-bit input: Data strobe input
    .RST(RST),                      // 1-bit input: Asynchronous reset input
    .SHIFTIN(SHIFTIN)               // 1-bit input: Cascade input signal for master/slave I/O
);

// End of ISERDES2_inst instantiation

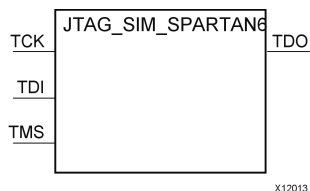
```

## 詳細情報

- [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- [Spartan-6 FPGA データシート：DC 特性およびスイッチ特性](#)

## JTAG\_SIM\_SPARTAN6

### シミュレーション：JTAG TAP Controller Simulation Model



### 概要

このシミュレーション コンポーネントを使用すると、JTAG TAP コントローラー インターフェイス、ファンクション、およびコマンドの論理シミュレーションを実行でき、JTAG およびバウンダリ スキャン動作、USER コマンドおよび BSCAN\_SPARTAN6 コンポーネントに関連する動作をボード レベルで理解し、デバッグするのに役立ちます。このモデルは、FPGA ツールの特定のプリミティブにはマップされず、デザインに直接インスタンス化することはできませんが、テストベンチなどのシミュレーションのみのファイルで指定するなど、合成で除外してデザイン ネットリストに含まれないようにすれば、ソース デザインと共に使用できます。このモデルは、論理 (RTL) シミュレーションおよびタイミング シミュレーションで使用できます。

### ポートの説明

ポート名	方向	幅	機能
TDO	出力	1	<b>テスト データ出力：</b> すべての JTAG 命令およびデータレジスタのシリアル出力です。TAP コントローラーのステートおよび現在の命令により、特定の操作に TDO に命令またはデータを送信するレジスタ (命令またはデータ) が決定します。TDO のステートは TCK の立ち下がりエッジで変化し、デバイス内を命令またはデータがシフトされている間のみアクティブになります。TDO はアクティブ ドライバー出力です。
TCK	入力	1	<b>テスト クロック：</b> JTAG のテスト クロックです。TAP コントローラーおよび JTAG レジスタは TCK に同期して動作します。
TDI	入力	1	<b>テスト データ：</b> すべての JTAG 命令およびデータレジスタのシリアル入力です。TAP コントローラーのステートおよび現在の命令により、特定の操作に TDI から命令またはデータを入力するレジスタ (命令またはデータ) が決定します。TDI には内部プルアップ抵抗が含まれており、駆動されない場合はシステムにロジック High を供給します。TDI からの JTAG レジスタへの命令またはデータ供給は、TCK の立ち上がりエッジに同期します。
TMS	入力	1	<b>テスト モード セレクト：</b> TCK の立ち上がりエッジで TAP コントローラーのステートのシーケンスを選択します。TMS には内部プルアップ抵抗が含まれており、駆動されない場合はロジック High を供給します。

### デザインの入力方法

インスタンス化	テストベンチまたはシミュレーション ファイルでのみ
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

テストベンチ ファイルにインスタンス化し、インプリメンテーション ファイルまたはデザインの合成に使用されるファイルには含めないことをお勧めします。

このコンポーネントの使用方法的詳細およびシミュレーションについては、『合成/シミュレーション デザイン ガイド』を参照してください。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
PART_NAME	文字列	"LX4"、"LX9"、 "LX16"、"LX25"、 "LX25T"、"LX45"、 "LX45T"、"LX75"、 "LX75T"、"LX100"、 "LX100T"、"LX150"、 "LX150T"	"LX4"	IDCODE およびその他のデバイス特定の属性を正しく設定するため、ターゲット デバイスを指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- JTAG_SIM_SPARTAN6: JTAG Interface Simulation Model
--                      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

JTAG_SIM_SPARTAN6_inst : JTAG_SIM_SPARTAN6
generic map (
  PART_NAME => "LX4") -- Specify target S6 device. Possible values are:
                      -- "LX4", "LX150", "LX150T", "LX16", "LX45", "LX45T"
port map (
  TDO => TDO,          -- JTAG data output (1-bit)
  TCK => TCK,          -- Clock input (1-bit)
  TDI => TDI,          -- JTAG data input (1-bit)
  TMS => TMS           -- JTAG command input (1-bit)
);

-- End of JTAG_SIM_SPARTAN6_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// JTAG_SIM_SPARTAN6: JTAG Interface Simulation Model
//                      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

JTAG_SIM_SPARTAN6 #(
  .PART_NAME("LX4") // Specify target S6 device. Possible values are:
                   // "LX4", "LX150", "LX150T", "LX16", "LX45", "LX45T"
) JTAG_SIM_SPARTAN6_inst (
  .TDO(TDO), // 1-bit JTAG data output
  .TCK(TCK), // 1-bit Clock input
  .TDI(TDI), // 1-bit JTAG data input
  .TMS(TMS) // 1-bit JTAG command input
);

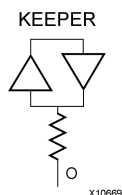
// End of JTAG_SIM_SPARTAN6_inst instantiation
```

## 詳細情報

- ・ [『合成/シミュレーション デザイン ガイド』\(UG626\)](#)
- ・ [『Spartan-6 FPGA コンフィギュレーション ユーザー ガイド』\(UG380\)](#)

## KEEPER

### プリミティブ：KEEPER Symbol



### 概要

このデザイン エLEMENTは、双方向出力ピンに接続されるネットの値を保持するウィークキーパー エLEMENTです。たとえば、ネットに論理 1 を駆動すると、KEEPER はそのネットにウィーク/抵抗値 1 を駆動します。その後、ネットドライバーがトライステートになっても、KEEPER はウィーク/抵抗値 1 を駆動し続けます。

### ポートの説明

ポート名	方向	幅	機能
O	出力	1 ビット	キーパー出力

### デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### VHDL 記述（インスタンス化）

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- KEEPER: I/O Buffer Weak Keeper
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

KEEPER_inst : KEEPER
port map (
  O => O      -- Keeper output (connect directly to top-level port)
);

-- End of KEEPER_inst instantiation
```



## Verilog 記述 (インスタンス化)

```
// KEEPER: I/O Buffer Weak Keeper
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

KEEPER KEEPER_inst (
    .O(0)      // Keeper output (connect directly to top-level port)
);

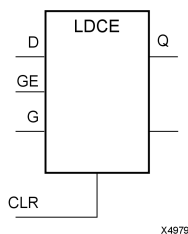
// End of KEEPER_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## LDCE

### プリミティブ：Transparent Data Latch with Asynchronous Clear and Gate Enable



### 概要

このデザイン エLEMENTは、非同期クリア (CLR) およびゲート イネーブル (GE) 付き透過データ ラッチです。非同期クリア入力 (CLR) が High になると、ほかの入力は無視され、データ出力 (Q) が Low にリセットされます。ゲート入力 (G) およびゲート イネーブル (GE) が High で、CLR が Low のとき、Q にはデータ入力 (D) が使用されます。GE が Low の場合、D の値は不定値になります。D 入力の値は、G が High から Low に切り替わるときにラッチ内に格納されます。Q 出力の値は、G または GE が Low の間は変化しません。

電力を供給すると、ラッチは非同期にクリアされ、出力が Low になります。FPGA では、グローバル セット/リセット (GSR) をアクティブにすると、電源投入時の状態をシミュレーションできます。GSR のデフォルトはアクティブ High ですが、STARTUP\_architecture シンボルの GSR 入力の前にインバーターを追加するとアクティブ Low にできます。

### 論理表

入力				出力
CLR	GE	G	D	Q
1	X	X	X	0
0	0	X	X	変化なし
0	1	1	D	D
0	1	0	X	変化なし
0	1	↓	D	D

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数	0、1	0	コンフィギュレーション後の Q 出力の初期値を指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LDCE: Transparent latch with Asynchronous Reset and
--       Gate Enable.
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LDCE_inst : LDCE
generic map (
  INIT => '0') -- Initial value of latch ('0' or '1')
port map (
  Q => Q,        -- Data output
  CLR => CLR,     -- Asynchronous clear/reset input
  D => D,        -- Data input
  G => G,        -- Gate input
  GE => GE       -- Gate enable input
);

-- End of LDCE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LDCE: Transparent latch with Asynchronous Reset and Gate Enable.
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LDCE LDCE_inst (
  .Q(Q),        // Data output
  .CLR(CLR),    // Asynchronous clear/reset input
  .D(D),        // Data input
  .G(G),        // Gate input
  .GE(GE)       // Gate enable input
);

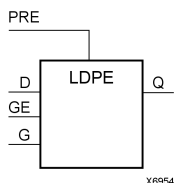
// End of LDCE_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LDPE

### プリミティブ : Transparent Data Latch with Asynchronous Preset and Gate Enable



## 概要

このデザイン エLEMENTは、非同期プリセット (PRE) およびゲート イネーブル (GE) 付き透過データ ラッチです。PRE が High になると、ほかの入力は無視され、データ出力 (Q) が High にプリセットされます。ゲート入力 (G) およびゲート イネーブル (GE) が High の場合、Q にデータ入力 (D) の値が出力されます。D 入力の値は、G が High から Low に切り替わる時にラッチ内に格納されます。Q 出力の値は、G または GE が Low の間は変化しません。

電力が供給されると、ラッチは非同期にプリセットされ、出力が High になります。FPGA では、グローバル セット/リセット (GSR) をアクティブにすると、電源投入時の状態をシミュレーションできます。GSR のデフォルトはアクティブ High ですが、STARTUP\_architecture シンボルの GSR 入力の前にインバーターを追加するとアクティブ Low にできます。

## 論理表

入力				出力
PRE	GE	G	D	Q
1	X	X	X	1
0	0	X	X	変化なし
0	1	1	D	D
0	1	0	X	変化なし
0	1	↓	D	D

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数	0、1	1	電源投入時または GSR のアサート時の Q ポートの初期値を指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LDPE: Transparent latch with Asynchronous Set and
--       Gate Enable.
--       Spartan-6
--       Xilinx HDL Libraries Guide, version 14.5

LDPE_inst : LDPE
generic map (
    INIT => '0') -- Initial value of latch ('0' or '1')
port map (
    Q => Q,        -- Data output
    CLR => CLR,    -- Asynchronous preset/set input
    D => D,        -- Data input
    G => G,        -- Gate input
    GE => GE       -- Gate enable input
);

-- End of LDPE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LDPE: Transparent latch with Asynchronous Preset and Gate Enable.
//       Spartan-6
//       Xilinx HDL Libraries Guide, version 14.5

LDPE LDPE_inst (
    .Q(Q),          // Data output
    .PRE(PRE),      // Asynchronous preset/set input
    .D(D),          // Data input
    .G(G),          // Gate input
    .GE(GE)         // Gate enable input
);

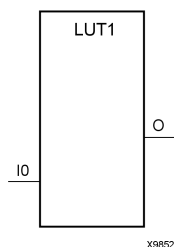
// End of LDPE_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT1

### マクロ : 1-Bit Look-Up Table with General Output



### 概要

このデザイン エレメントは汎用出力 (O) を持つ 1 ビットのルックアップ テーブル (LUT) です。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。このエレメントは、バッファまたはインバーターの機能を果たします。これらのエレメントは基本ブロックで、各 CLB スライスに 2 つ、各 CLB に 4 つずつあります。LUT には複数のバリエーションがあり、異なるタイミング モデルでレイアウト前のタイミングをより正確に予測する必要がある場合に使用できます。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法 :** LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法 :** リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力	出力
I0	O
0	INIT[0]
1	INIT[1]
INIT = INIT 属性に割り当てられた 2 進数値	

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	2 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT1: 1-input Look-Up Table with general output
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT1_inst : LUT1
generic map (
    INIT => "00")
port map (
    O => O,    -- LUT general output
    I0 => I0   -- LUT input
);

-- End of LUT1_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT1: 1-input Look-Up Table with general output
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT1 #(
    .INIT(2'b00) // Specify LUT Contents
) LUT1_inst (
    .O(O),       // LUT general output
    .I0(I0)     // LUT input
);

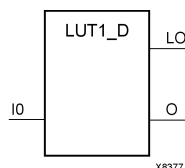
// End of LUT1_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT1\_D

### マクロ：1-Bit Look-Up Table with Dual Output



### 概要

このデザイン エLEMENTは 1 ビットのルックアップ テーブル (LUT) で、同じ機能を持つ O および LO という 2 つの出力があります。このELEMENTはバッファまたはインバータの機能を果たします。

出力 O は汎用インターコネクトです。LO 出力は同じ CLB スライス内の別の入力または高速バッファに接続します。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT の値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

- ・ **論理表を使用する方法**：LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。
- ・ **論理式を使用する方法**：リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力	出力	
IO	O	LO
0	INIT[0]	INIT[0]
1	INIT[1]	INIT[1]
INIT = INIT 属性に割り当てられた 2 進数値		

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	2 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。



## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT1_D: 1-input Look-Up Table with general and local outputs
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT1_D_inst : LUT1_D
generic map (
    INIT => "00")
port map (
    LO => LO, -- LUT local output
    O => O,   -- LUT general output
    IO => IO  -- LUT input
);

-- End of LUT1_D_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT1_D: 1-input Look-Up Table with general and local outputs
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT1_D #(
    .INIT(2'b00) // Specify LUT Contents
) LUT1_D_inst (
    .LO(LO), // LUT local output
    .O(O),  // LUT general output
    .IO(IO) // LUT input
);

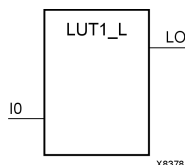
// End of LUT1_D_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT1\_L

### マクロ：1-Bit Look-Up Table with Local Output



### 概要

このデザイン エLEMENT は、1 ビットのルックアップ テーブル (LUT) で、同じ CLB スライス内にある別の出力および高速バッファへの接続に使用するローカル出力 (LO) があります。このELEMENTはバッファまたはインバーターの機能を果たします。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法：**リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力	出力
I0	LO
0	INIT[0]
1	INIT[1]
INIT = INIT 属性に割り当てられた 2 進数値	

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	2 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT1_L: 1-input Look-Up Table with local output
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT1_L_inst : LUT1_L
generic map (
    INIT => "00")
port map (
    LO => LO, -- LUT local output
    IO => IO  -- LUT input
);

-- End of LUT1_L_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT1_L: 1-input Look-Up Table with local output
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT1_L #(
    .INIT(2'b00) // Specify LUT Contents
) LUT1_L_inst (
    .LO(LO), // LUT local output
    .IO(IO) // LUT input
);

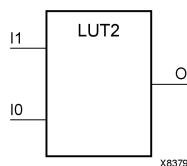
// End of LUT1_L_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT2

### マクロ：2-Bit Look-Up Table with General Output



### 概要

このデザイン エLEMENTは、汎用出力 (O) を持つ 2 ビットのルックアップ テーブル (LUT) です。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。このELEMENTは、バッファまたはインバーターの機能を果たします。これらのELEMENTは基本ブロックで、各 CLB スライスに 2 つ、各 CLB に 4 つずつあります。LUT には複数のバリエーションがあり、異なるタイミング モデルでレイアウト前のタイミングをより正確に予測する必要がある場合に使用できます。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法：**リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力		出力
I1	I0	O
0	0	INIT[0]
0	1	INIT[1]
1	0	INIT[2]
1	1	INIT[3]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値		

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	4 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT2: 2-input Look-Up Table with general output
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT2_inst : LUT2
generic map (
  INIT => X"0")
port map (
  O => O,    -- LUT general output
  I0 => I0,  -- LUT input
  I1 => I1   -- LUT input
);

-- End of LUT2_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT2: 2-input Look-Up Table with general output
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT2 #(
  .INIT(4'h0) // Specify LUT Contents
) LUT2_inst (
  .O(O),      // LUT general output
  .I0(I0),    // LUT input
  .I1(I1)     // LUT input
);

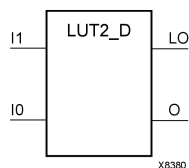
// End of LUT2_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT2\_D

### マクロ：2-Bit Look-Up Table with Dual Output



### 概要

このデザイン エレメントは 2 ビットのルックアップ テーブル (LUT) で、同じ機能を持つ O および LO という 2 つの出力があります。

出力 O は汎用インターコネクトです。LO 出力は同じ CLB スライス内の別の入力または高速バッファに接続します。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT の値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

- ・ **論理表を使用する方法**：LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。
- ・ **論理式を使用する方法**：リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力		出力	
I1	I0	O	LO
0	0	INIT[0]	INIT[0]
0	1	INIT[1]	INIT[1]
1	0	INIT[2]	INIT[2]
1	1	INIT[3]	INIT[3]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値			

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	4 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT2_D: 2-input Look-Up Table with general and local outputs
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT2_D_inst : LUT2_D
generic map (
  INIT => X"0")
port map (
  LO => LO, -- LUT local output
  O  => O,  -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1  -- LUT input
);

-- End of LUT2_D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT2_D: 2-input Look-Up Table with general and local outputs
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT2_D #(
  .INIT(4'h0) // Specify LUT Contents
) LUT2_D_inst (
  .LO(LO), // LUT local output
  .O(O),   // LUT general output
  .I0(I0), // LUT input
  .I1(I1)  // LUT input
);

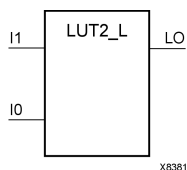
// End of LUT2_D_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT2\_L

### マクロ：2-Bit Look-Up Table with Local Output



### 概要

このデザイン エLEMENT は 2 ビットのルックアップ テーブル (LUT) で、同じ CLB スライス内にある別の出力および高速バッファへの接続に使用するローカル出力 (LO) があります。このELEMENTはバッファまたはインバーターの機能を果たします。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法：**リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力		出力
I1	I0	LO
0	0	INIT[0]
0	1	INIT[1]
1	0	INIT[2]
1	1	INIT[3]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値		

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可



## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	4 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT2_L: 2-input Look-Up Table with local output
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT2_L_inst : LUT2_L
generic map (
  INIT => X"0")
port map (
  LO => LO, -- LUT local output
  IO => IO, -- LUT input
  I1 => I1 -- LUT input
);

-- End of LUT2_L_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT2_L: 2-input Look-Up Table with local output
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT2_L #(
  .INIT(4'h0) // Specify LUT Contents
) LUT2_L_inst (
  .LO(IO), // LUT local output
  .IO(IO), // LUT input
  .I1(I1) // LUT input
);

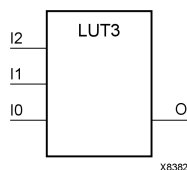
// End of LUT2_L_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT3

### マクロ : 3-Bit Look-Up Table with General Output



## 概要

このデザイン エレメントは、汎用出力 (O) を持つ 3 ビットのルックアップ テーブル (LUT) です。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。このエレメントは、バッファまたはインバーターの機能を果たします。これらのエレメントは基本ブロックで、各 CLB スライスに 2 つ、各 CLB に 4 つずつあります。LUT には複数のバリエーションがあり、異なるタイミング モデルでレイアウト前のタイミングをより正確に予測する必要がある場合に使用できます。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法 :** LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法 :** リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

## 論理表

入力			出力
I2	I1	I0	O
0	0	0	INIT[0]
0	0	1	INIT[1]
0	1	0	INIT[2]
0	1	1	INIT[3]
1	0	0	INIT[4]
1	0	1	INIT[5]
1	1	0	INIT[6]
1	1	1	INIT[7]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値			

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	8 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT3: 3-input Look-Up Table with general output
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT3_inst : LUT3
generic map (
  INIT => X"00")
port map (
  O => O,    -- LUT general output
  I0 => I0,  -- LUT input
  I1 => I1,  -- LUT input
  I2 => I2   -- LUT input
);

-- End of LUT3_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT3: 3-input Look-Up Table with general output
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT3 #(
  .INIT(8'h00) // Specify LUT Contents
) LUT3_inst (
  .O(O),      // LUT general output
  .I0(I0),    // LUT input
  .I1(I1),    // LUT input
  .I2(I2)     // LUT input
);

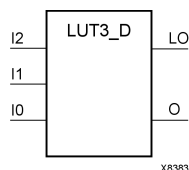
// End of LUT3_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT3\_D

### マクロ：3-Bit Look-Up Table with Dual Output



### 概要

このデザイン エLEMENTは 3 ビットのルックアップ テーブル (LUT) で、同じ機能を持つ O および LO という 2 つの出力があります。

出力 O は汎用インターコネクトです。LO 出力は同じ CLB スライス内の別の入力または高速バッファに接続します。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT の値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

- ・ **論理表を使用する方法**：LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。
- ・ **論理式を使用する方法**：リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力			出力	
I2	I1	I0	O	LO
0	0	0	INIT[0]	INIT[0]
0	0	1	INIT[1]	INIT[1]
0	1	0	INIT[2]	INIT[2]
0	1	1	INIT[3]	INIT[3]
1	0	0	INIT[4]	INIT[4]
1	0	1	INIT[5]	INIT[5]
1	1	0	INIT[6]	INIT[6]
1	1	1	INIT[7]	INIT[7]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値				

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	8 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT3_D: 3-input Look-Up Table with general and local outputs
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT3_D_inst : LUT3_D
generic map (
  INIT => X"00")
port map (
  LO => LO, -- LUT local output
  O  => O,  -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2  -- LUT input
);

-- End of LUT3_D_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT3_D: 3-input Look-Up Table with general and local outputs
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT3_D #(
  .INIT(8'h00) // Specify LUT Contents
) LUT3_D_inst (
  .LO(LO), // LUT local output
  .O(O),  // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2)  // LUT input
);

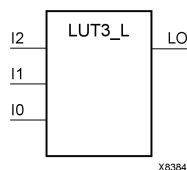
// End of LUT3_D_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT3\_L

### マクロ : 3-Bit Look-Up Table with Local Output



### 概要

このデザイン エLEMENT は 3 ビットのルックアップ テーブル (LUT) で、同じ CLB スライス内にある別の出力および高速バッファへの接続に使用するローカル出力 (LO) があります。このELEMENTはバッファまたはインバーターの機能を果たします。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法 :** LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法 :** リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力			出力
I2	I1	I0	LO
0	0	0	INIT[0]
0	0	1	INIT[1]
0	1	0	INIT[2]
0	1	1	INIT[3]
1	0	0	INIT[4]
1	0	1	INIT[5]
1	1	0	INIT[6]
1	1	1	INIT[7]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値			

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	8 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT3_L: 3-input Look-Up Table with local output
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT3_L_inst : LUT3_L
generic map (
  INIT => X"00")
port map (
  LO => LO,    -- LUT local output
  I0 => I0,    -- LUT input
  I1 => I1,    -- LUT input
  I2 => I2     -- LUT input
);

-- End of LUT3_L_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT3_L: 3-input Look-Up Table with local output
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT3_L #(
  .INIT(8'h00) // Specify LUT Contents
) LUT3_L_inst (
  .LO(LO), // LUT local output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2)  // LUT input
);

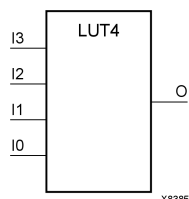
// End of LUT3_L_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT4

### マクロ：4-Bit Look-Up-Table with General Output



### 概要

このデザイン エレメントは、汎用出力 (O) を持つ 4 ビットのルックアップ テーブル (LUT) です。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。このエレメントは、バッファまたはインバーターの機能を果たします。これらのエレメントは基本ブロックで、各 CLB スライスに 2 つ、各 CLB に 4 つずつあります。LUT には複数のバリエーションがあり、異なるタイミング モデルでレイアウト前のタイミングをより正確に予測する必要がある場合に使用できます。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法：**リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力				出力
I3	I2	I1	I0	O
0	0	0	0	INIT[0]
0	0	0	1	INIT[1]
0	0	1	0	INIT[2]
0	0	1	1	INIT[3]
0	1	0	0	INIT[4]
0	1	0	1	INIT[5]
0	1	1	0	INIT[6]
0	1	1	1	INIT[7]
1	0	0	0	INIT[8]
1	0	0	1	INIT[9]
1	0	1	0	INIT[10]
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]



入力				出力
I3	I2	I1	I0	O
1	1	0	1	INIT[13]
1	1	1	0	INIT[14]
1	1	1	1	INIT[15]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値				

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4: 4-input Look-Up Table with general output
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT4_inst : LUT4
generic map (
  INIT => X"0000")
port map (
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3 -- LUT input
);

-- End of LUT4_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT4: 4-input Look-Up Table with general output
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT4 #(
    .INIT(16'h0000) // Specify LUT Contents
) LUT4_inst (
    .O(O), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3) // LUT input
);

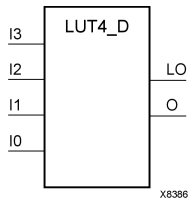
// End of LUT4_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT4\_D

### マクロ：4-Bit Look-Up Table with Dual Output



### 概要

このデザイン エLEMENTは 4 ビットのルックアップ テーブル (LUT) で、同じ機能を持つ O および LO という 2 つの出力があります。

出力 O は汎用インターコネクトです。LO 出力は同じ CLB スライス内の別の入力または高速バッファに接続します。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT の値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

- ・ **論理表を使用する方法：** LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。
- ・ **論理式を使用する方法：** リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力				出力	
I3	I2	I1	I0	O	LO
0	0	0	0	INIT[0]	INIT[0]
0	0	0	1	INIT[1]	INIT[1]
0	0	1	0	INIT[2]	INIT[2]
0	0	1	1	INIT[3]	INIT[3]
0	1	0	0	INIT[4]	INIT[4]
0	1	0	1	INIT[5]	INIT[5]
0	1	1	0	INIT[6]	INIT[6]
0	1	1	1	INIT[7]	INIT[7]
1	0	0	0	INIT[8]	INIT[8]
1	0	0	1	INIT[9]	INIT[9]
1	0	1	0	INIT[10]	INIT[10]
1	0	1	1	INIT[11]	INIT[11]
1	1	0	0	INIT[12]	INIT[12]

入力				出力	
I3	I2	I1	I0	O	LO
1	1	0	1	INIT[13]	INIT[13]
1	1	1	0	INIT[14]	INIT[14]
1	1	1	1	INIT[15]	INIT[15]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値					

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_D: 4-input Look-Up Table with general and local outputs
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT4_D_inst : LUT4_D
generic map (
  INIT => X"0000")
port map (
  LO => LO, -- LUT local output
  O  => O,  -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3  -- LUT input
);

-- End of LUT4_D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT4_D: 4-input Look-Up Table with general and local outputs
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT4_D #(
    .INIT(16'h0000) // Specify LUT Contents
) LUT4_D_inst (
    .LO(LO), // LUT local output
    .O(O), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3) // LUT input
);

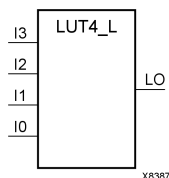
// End of LUT4_D_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT4\_L

### マクロ : 4-Bit Look-Up Table with Local Output



### 概要

このデザイン エLEMENT は 4 ビットのルックアップ テーブル (LUT) で、同じ CLB スライス内にある別の出力および高速バッファへの接続に使用するローカル出力 (LO) があります。このELEMENTはバッファまたはインバーターの機能を果たします。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法 :** LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法 :** リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力				出力
I3	I2	I1	I0	LO
0	0	0	0	INIT[0]
0	0	0	1	INIT[1]
0	0	1	0	INIT[2]
0	0	1	1	INIT[3]
0	1	0	0	INIT[4]
0	1	0	1	INIT[5]
0	1	1	0	INIT[6]
0	1	1	1	INIT[7]
1	0	0	0	INIT[8]
1	0	0	1	INIT[9]
1	0	1	0	INIT[10]
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]

入力				出力
I3	I2	I1	I0	LO
1	1	0	1	INIT[13]
1	1	1	0	INIT[14]
1	1	1	1	INIT[15]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値				

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_L: 4-input Look-Up Table with local output
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT4_L_inst : LUT4_L
generic map (
  INIT => X"0000")
port map (
  LO => LO, -- LUT local output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3  -- LUT input
);

-- End of LUT4_L_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// LUT4_L: 4-input Look-Up Table with local output
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT4_L #(
    .INIT(16'h0000) // Specify LUT Contents
) LUT4_L_inst (
    .LO(LO), // LUT local output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3)  // LUT input
);

// End of LUT4_L_inst instantiation
```

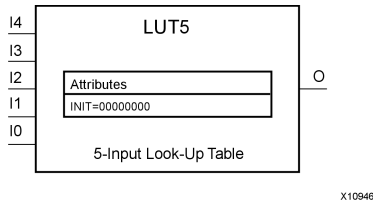
## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## LUT5

### プリミティブ：5-Input Lookup Table with General Output



### 概要

このデザイン エLEMENTは、入力 5 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 32 ビット ROM (5 ビットのアドレス指定) または 5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT5 が 1 つの場合はスライス内の LUT6 に、2 つの場合は多少の制限はありますが 1 つの LUT6 にパックできます。LUT5、LUT5\_L、および LUT5\_D の機能は同じですが、LUT5\_L および LUT5\_D では、LUT5 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT5\_L では LUT5 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT5\_D では LUT5 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT5 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 32 ビットの 16 進数を設定する必要があります。入力が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を 32'h80000000 (VHDL では X"80000000") に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (5 入力の AND ゲート)。また、Verilog で INIT 値を 32'hffffff (VHDL では X"FFFFFFFE") に設定すると、入力がすべて 0 の場合以外は出力は 1 になります (5 入力 OR ゲート)。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法：**リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力					出力
I4	I3	I2	I1	I0	LO
0	0	0	0	0	INIT[0]
0	0	0	0	1	INIT[1]
0	0	0	1	0	INIT[2]
0	0	0	1	1	INIT[3]
0	0	1	0	0	INIT[4]
0	0	1	0	1	INIT[5]

入力					出力
I4	I3	I2	I1	I0	LO
0	0	1	1	0	INIT[6]
0	0	1	1	1	INIT[7]
0	1	0	0	0	INIT[8]
0	1	0	0	1	INIT[9]
0	1	0	1	0	INIT[10]
0	1	0	1	1	INIT[11]
0	1	1	0	0	INIT[12]
0	1	1	0	1	INIT[13]
0	1	1	1	0	INIT[14]
0	1	1	1	1	INIT[15]
1	0	0	0	0	INIT[16]
1	0	0	0	1	INIT[17]
1	0	0	1	0	INIT[18]
1	0	0	1	1	INIT[19]
1	0	1	0	0	INIT[20]
1	0	1	0	1	INIT[21]
1	0	1	1	0	INIT[22]
1	0	1	1	1	INIT[23]
1	1	0	0	0	INIT[24]
1	1	0	0	1	INIT[25]
1	1	0	1	0	INIT[26]
1	1	0	1	1	INIT[27]
1	1	1	0	0	INIT[28]
1	1	1	0	1	INIT[29]
1	1	1	1	0	INIT[30]
1	1	1	1	1	INIT[31]

INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT5: 5-input Look-Up Table with general output
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT5_inst : LUT5
generic map (
  INIT => X"00000000") -- Specify LUT Contents
port map (
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4 -- LUT input
);

-- End of LUT5_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT5: 5-input Look-Up Table with general output
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT5 #(
  .INIT(32'h00000000) // Specify LUT Contents
) LUT5_inst (
  .O(O), // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4) // LUT input
);

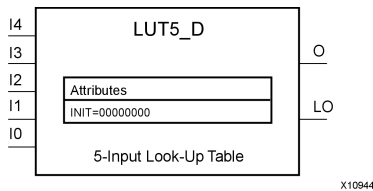
// End of LUT5_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## LUT5\_D

### プリミティブ：5-Input Lookup Table with General and Local Outputs



### 概要

このデザイン エLEMENTは、入力 5 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 32 ビット ROM (5 ビットのアドレス指定) または 5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT5 が 1 つの場合はスライス内の LUT6 に、2 つの場合は多少の制限はありますが 1 つの LUT6 にパックできます。LUT5、LUT5\_L、および LUT5\_D の機能は同じですが、LUT5\_L および LUT5\_D では、LUT5 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT5\_L では LUT5 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT5\_D では LUT5 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT5 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 32 ビットの 16 進数を設定する必要があります。入力が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を 32'h80000000 (VHDL では X"80000000") に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (5 入力の AND ゲート)。また、Verilog で INIT 値を 32'hffffff (VHDL では X"FFFFFFFE") に設定すると、入力がすべて 0 の場合以外は出力は 1 になります (5 入力 OR ゲート)。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT の値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

- ・ **論理表を使用する方法**：LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。
- ・ **論理式を使用する方法**：リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力					出力	
I4	I3	I2	I1	I0	O	LO
0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	1	1	INIT[3]	INIT[3]
0	0	1	0	0	INIT[4]	INIT[4]
0	0	1	0	1	INIT[5]	INIT[5]

入力					出力	
I4	I3	I2	I1	I0	O	LO
0	0	1	1	0	INIT[6]	INIT[6]
0	0	1	1	1	INIT[7]	INIT[7]
0	1	0	0	0	INIT[8]	INIT[8]
0	1	0	0	1	INIT[9]	INIT[9]
0	1	0	1	0	INIT[10]	INIT[10]
0	1	0	1	1	INIT[11]	INIT[11]
0	1	1	0	0	INIT[12]	INIT[12]
0	1	1	0	1	INIT[13]	INIT[13]
0	1	1	1	0	INIT[14]	INIT[14]
0	1	1	1	1	INIT[15]	INIT[15]
1	0	0	0	0	INIT[16]	INIT[16]
1	0	0	0	1	INIT[17]	INIT[17]
1	0	0	1	0	INIT[18]	INIT[18]
1	0	0	1	1	INIT[19]	INIT[19]
1	0	1	0	0	INIT[20]	INIT[20]
1	0	1	0	1	INIT[21]	INIT[21]
1	0	1	1	0	INIT[22]	INIT[22]
1	0	1	1	1	INIT[23]	INIT[23]
1	1	0	0	0	INIT[24]	INIT[24]
1	1	0	0	1	INIT[25]	INIT[25]
1	1	0	1	0	INIT[26]	INIT[26]
1	1	0	1	1	INIT[27]	INIT[27]
1	1	1	0	0	INIT[28]	INIT[28]
1	1	1	0	1	INIT[29]	INIT[29]
1	1	1	1	0	INIT[30]	INIT[30]
1	1	1	1	1	INIT[31]	INIT[31]

INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	5 入力 LUT 出力
LO	出力	1	内部 CLB 接続用の 5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT5_D: 5-input Look-Up Table with general and local outputs
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT5_D_inst : LUT5_D
generic map (
  INIT => X"00000000") -- Specify LUT contents
port map (
  LO => LO, -- LUT local output
  O  => O,  -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4  -- LUT input
);

-- End of LUT5_D_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT5_D: 5-input Look-Up Table with general and local outputs
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT5_D #(
  .INIT(32'h00000000) // Specify LUT Contents
) LUT5_D_inst (
  .LO(LO), // LUT local output
  .O(O),   // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4)  // LUT input
);

// End of LUT5_D_inst instantiation
```

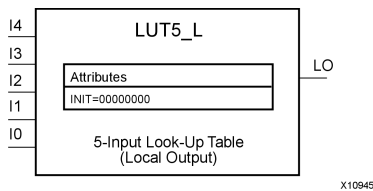
## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## LUT5\_L

### プリミティブ：5-Input Lookup Table with Local Output



### 概要

このデザイン エLEMENTは、入力 5 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 32 ビット ROM (5 ビットのアドレス指定) または 5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT5 が 1 つの場合はスライス内の LUT6 に、2 つの場合は多少の制限はありますが 1 つの LUT6 にパックできます。LUT5、LUT5\_L、および LUT5\_D の機能は同じですが、LUT5\_L および LUT5\_D では、LUT5 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT5\_L では LUT5 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT5\_D では LUT5 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT5 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 32 ビットの 16 進数を設定する必要があります。入力が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を 32'h80000000 (VHDL では X"80000000") に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (5 入力の AND ゲート)。また、Verilog で INIT 値を 32'hffffff (VHDL では X"FFFFFFFE") に設定すると、入力がすべて 0 の場合以外は出力は 1 になります (5 入力 OR ゲート)。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法：**リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力					出力
I4	I3	I2	I1	I0	LO
0	0	0	0	0	INIT[0]
0	0	0	0	1	INIT[1]
0	0	0	1	0	INIT[2]
0	0	0	1	1	INIT[3]
0	0	1	0	0	INIT[4]
0	0	1	0	1	INIT[5]

入力					出力
I4	I3	I2	I1	I0	LO
0	0	1	1	0	INIT[6]
0	0	1	1	1	INIT[7]
0	1	0	0	0	INIT[8]
0	1	0	0	1	INIT[9]
0	1	0	1	0	INIT[10]
0	1	0	1	1	INIT[11]
0	1	1	0	0	INIT[12]
0	1	1	0	1	INIT[13]
0	1	1	1	0	INIT[14]
0	1	1	1	1	INIT[15]
1	0	0	0	0	INIT[16]
1	0	0	0	1	INIT[17]
1	0	0	1	0	INIT[18]
1	0	0	1	1	INIT[19]
1	0	1	0	0	INIT[20]
1	0	1	0	1	INIT[21]
1	0	1	1	0	INIT[22]
1	0	1	1	1	INIT[23]
1	1	0	0	0	INIT[24]
1	1	0	0	1	INIT[25]
1	1	0	1	0	INIT[26]
1	1	0	1	1	INIT[27]
1	1	1	0	0	INIT[28]
1	1	1	0	1	INIT[29]
1	1	1	1	0	INIT[30]
1	1	1	1	1	INIT[31]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値					

## ポートの説明

ポート名	方向	幅	機能
L0	出力	1	内部 CLB 接続用の 6/5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT5_L: 5-input Look-Up Table with local output
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT5_L_inst : LUT5_L
generic map (
  INIT => X"00000000") -- Specify LUT Contents
port map (
  LO => LO, -- LUT local output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4  -- LUT input
);

-- End of LUT5_L_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT5_L: 5-input Look-Up Table with local output
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT5_L #(
  .INIT(32'h00000000) // Specify LUT Contents
) LUT5_L_inst (
  .LO(LO), // LUT local output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4)  // LUT input
);

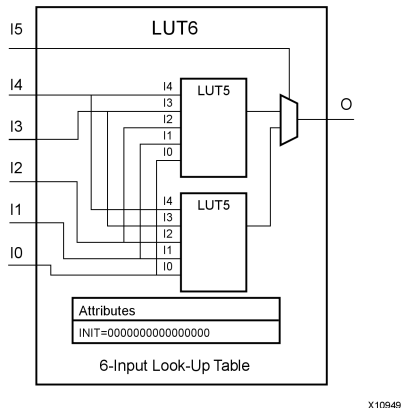
// End of LUT5_L_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## LUT6

### プリミティブ：6-Input Lookup Table with General Output



### 概要

このデザイン エLEMENTは、入力 6 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 64 ビット ROM (6 ビットのアドレス指定) または 6 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT6 は、4 個のルックアップ テーブル (LUT) のいずれかにマップされます。LUT6、LUT6\_L、および LUT6\_D の機能は同じですが、LUT6\_L および LUT6\_D では、LUT6 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT6\_L では LUT6 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT6\_D では LUT6 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数値を設定する必要があります。入力 が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を `64'h8000000000000000` (VHDL では `X"8000000000000000"`) に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値を `64'hffffffffffffff` (VHDL では `X"FFFFFFFFFFFFFFFF"`) に設定すると、入力すべてが 0 の場合以外は出力は 1 になります (6 入力 OR ゲート)。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：** LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法：** リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力						出力
I5	I4	I3	I2	I1	I0	O
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]
0	0	0	0	1	0	INIT[2]

入力						出力
I5	I4	I3	I2	I1	I0	O
0	0	0	0	1	1	INIT[3]
0	0	0	1	0	0	INIT[4]
0	0	0	1	0	1	INIT[5]
0	0	0	1	1	0	INIT[6]
0	0	0	1	1	1	INIT[7]
0	0	1	0	0	0	INIT[8]
0	0	1	0	0	1	INIT[9]
0	0	1	0	1	0	INIT[10]
0	0	1	0	1	1	INIT[11]
0	0	1	1	0	0	INIT[12]
0	0	1	1	0	1	INIT[13]
0	0	1	1	1	0	INIT[14]
0	0	1	1	1	1	INIT[15]
0	1	0	0	0	0	INIT[16]
0	1	0	0	0	1	INIT[17]
0	1	0	0	1	0	INIT[18]
0	1	0	0	1	1	INIT[19]
0	1	0	1	0	0	INIT[20]
0	1	0	1	0	1	INIT[21]
0	1	0	1	1	0	INIT[22]
0	1	0	1	1	1	INIT[23]
0	1	1	0	0	0	INIT[24]
0	1	1	0	0	1	INIT[25]
0	1	1	0	1	0	INIT[26]
0	1	1	0	1	1	INIT[27]
0	1	1	1	0	0	INIT[28]
0	1	1	1	0	1	INIT[29]
0	1	1	1	1	0	INIT[30]
0	1	1	1	1	1	INIT[31]
1	0	0	0	0	0	INIT[32]
1	0	0	0	0	1	INIT[33]
1	0	0	0	1	0	INIT[34]
1	0	0	0	1	1	INIT[35]
1	0	0	1	0	0	INIT[36]
1	0	0	1	0	1	INIT[37]

入力						出力
I5	I4	I3	I2	I1	I0	O
1	0	0	1	1	0	INIT[38]
1	0	0	1	1	1	INIT[39]
1	0	1	0	0	0	INIT[40]
1	0	1	0	0	1	INIT[41]
1	0	1	0	1	0	INIT[42]
1	0	1	0	1	1	INIT[43]
1	0	1	1	0	0	INIT[44]
1	0	1	1	0	1	INIT[45]
1	0	1	1	1	0	INIT[46]
1	0	1	1	1	1	INIT[47]
1	1	0	0	0	0	INIT[48]
1	1	0	0	0	1	INIT[49]
1	1	0	0	1	0	INIT[50]
1	1	0	0	1	1	INIT[51]
1	1	0	1	0	0	INIT[52]
1	1	0	1	0	1	INIT[53]
1	1	0	1	1	0	INIT[54]
1	1	0	1	1	1	INIT[55]
1	1	1	0	0	0	INIT[56]
1	1	1	0	0	1	INIT[57]
1	1	1	0	1	0	INIT[58]
1	1	1	0	1	1	INIT[59]
1	1	1	1	0	0	INIT[60]
1	1	1	1	0	1	INIT[61]
1	1	1	1	1	0	INIT[62]
1	1	1	1	1	1	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値						

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	6/5 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT6: 6-input Look-Up Table with general output
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT6_inst : LUT6
generic map (
  INIT => X"0000000000000000") -- Specify LUT Contents
port map (
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4, -- LUT input
  I5 => I5  -- LUT input
);

-- End of LUT6_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT6: 6-input Look-Up Table with general output
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT6 #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_inst (
  .O(O), // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4), // LUT input
  .I5(I5) // LUT input
);

// End of LUT6_inst instantiation
```

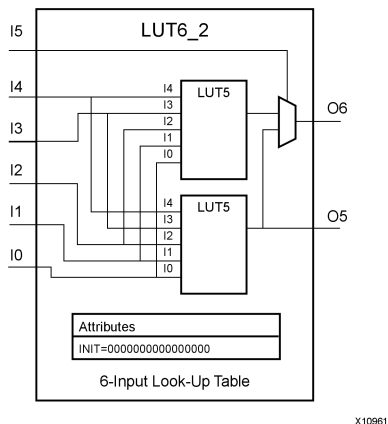


## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## LUT6\_2

プリミティブ：Six-input, 2-output, Look-Up Table



### 概要

このデザイン エレメントは、入力 6 個、出力 2 個を持つルックアップ テーブル (LUT) で、非同期 32 ビット デュアル ROM (5 ビットのアドレス指定)、入力を共有する 5 入力のロジック ファンクション 2 つ、または入力と論理値を共有する 6 入力および 5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT6\_2 は、スライスに含まれる 4 個のルックアップ テーブル (LUT) のいずれかにマップされます。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数を設定する必要があります。入力が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば、Verilog で INIT 値を `64'hfffffffffe` (VHDL では `X"FFFFFFFFFFFFFFFE"`) に設定すると、入力がすべて 0 の場合以外は O6 出力は 1 になり、I[4:0] がすべて 0 の場合以外は O5 出力は 1 になります (5 または 6 入力の OR ゲート)。INIT 値の下位半分 (ビット 31:0) は O5 出力のロジック ファンクションに適用されます。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT の値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

- ・ **論理表を使用する方法**：LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。
- ・ **論理式を使用する方法**：リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力						出力	
I5	I4	I3	I2	I1	I0	O5	O6
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	0	1	1	INIT[3]	INIT[3]
0	0	0	1	0	0	INIT[4]	INIT[4]

入力						出力	
0	0	0	1	0	1	INIT[5]	INIT[5]
0	0	0	1	1	0	INIT[6]	INIT[6]
0	0	0	1	1	1	INIT[7]	INIT[7]
0	0	1	0	0	0	INIT[8]	INIT[8]
0	0	1	0	0	1	INIT[9]	INIT[9]
0	0	1	0	1	0	INIT[10]	INIT[10]
0	0	1	0	1	1	INIT[11]	INIT[11]
0	0	1	1	0	0	INIT[12]	INIT[12]
0	0	1	1	0	1	INIT[13]	INIT[13]
0	0	1	1	1	0	INIT[14]	INIT[14]
0	0	1	1	1	1	INIT[15]	INIT[15]
0	1	0	0	0	0	INIT[16]	INIT[16]
0	1	0	0	0	1	INIT[17]	INIT[17]
0	1	0	0	1	0	INIT[18]	INIT[18]
0	1	0	0	1	1	INIT[19]	INIT[19]
0	1	0	1	0	0	INIT[20]	INIT[20]
0	1	0	1	0	1	INIT[21]	INIT[21]
0	1	0	1	1	0	INIT[22]	INIT[22]
0	1	0	1	1	1	INIT[23]	INIT[23]
0	1	1	0	0	0	INIT[24]	INIT[24]
0	1	1	0	0	1	INIT[25]	INIT[25]
0	1	1	0	1	0	INIT[26]	INIT[26]
0	1	1	0	1	1	INIT[27]	INIT[27]
0	1	1	1	0	0	INIT[28]	INIT[28]
0	1	1	1	0	1	INIT[29]	INIT[29]
0	1	1	1	1	0	INIT[30]	INIT[30]
0	1	1	1	1	1	INIT[31]	INIT[31]
1	0	0	0	0	0	INIT[0]	INIT[32]
1	0	0	0	0	1	INIT[1]	INIT[33]
1	0	0	0	1	0	INIT[2]	INIT[34]
1	0	0	0	1	1	INIT[3]	INIT[35]
1	0	0	1	0	0	INIT[4]	INIT[36]
1	0	0	1	0	1	INIT[5]	INIT[37]
1	0	0	1	1	0	INIT[6]	INIT[38]
1	0	0	1	1	1	INIT[7]	INIT[39]
1	0	1	0	0	0	INIT[8]	INIT[40]

入力						出力	
1	0	1	0	0	1	INIT[9]	INIT[41]
1	0	1	0	1	0	INIT[10]	INIT[42]
1	0	1	0	1	1	INIT[11]	INIT[43]
1	0	1	1	0	0	INIT[12]	INIT[44]
1	0	1	1	0	1	INIT[13]	INIT[45]
1	0	1	1	1	0	INIT[14]	INIT[46]
1	0	1	1	1	1	INIT[15]	INIT[47]
1	1	0	0	0	0	INIT[16]	INIT[48]
1	1	0	0	0	1	INIT[17]	INIT[49]
1	1	0	0	1	0	INIT[18]	INIT[50]
1	1	0	0	1	1	INIT[19]	INIT[51]
1	1	0	1	0	0	INIT[20]	INIT[52]
1	1	0	1	0	1	INIT[21]	INIT[53]
1	1	0	1	1	0	INIT[22]	INIT[54]
1	1	0	1	1	1	INIT[23]	INIT[55]
1	1	1	0	0	0	INIT[24]	INIT[56]
1	1	1	0	0	1	INIT[25]	INIT[57]
1	1	1	0	1	0	INIT[26]	INIT[58]
1	1	1	0	1	1	INIT[27]	INIT[59]
1	1	1	1	0	0	INIT[28]	INIT[60]
1	1	1	1	0	1	INIT[29]	INIT[61]
1	1	1	1	1	0	INIT[30]	INIT[62]
1	1	1	1	1	1	INIT[31]	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値							

## ポートの説明

ポート名	方向	幅	機能
O6	出力	1	6/5 LUT 出力
O5	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

## デザインの入力方法

インスタンスエーション	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	LUT5/6 の出力ファンクションを指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT6_2: 6-input 2 output Look-Up Table
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT6_2_inst : LUT6_2
generic map (
  INIT => X"0000000000000000" -- Specify LUT Contents
port map (
  O6 => O6, -- 6/5-LUT output (1-bit)
  O5 => O5, -- 5-LUT output (1-bit)
  I0 => I0, -- LUT input (1-bit)
  I1 => I1, -- LUT input (1-bit)
  I2 => I2, -- LUT input (1-bit)
  I3 => I3, -- LUT input (1-bit)
  I4 => I4, -- LUT input (1-bit)
  I5 => I5  -- LUT input (1-bit)
);

-- End of LUT6_2_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT6_2: 6-input, 2 output Look-Up Table
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT6_2 #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_2_inst (
  .O6(O6), // 1-bit LUT6 output
  .O5(O5), // 1-bit lower LUT5 output
  .I0(I0), // 1-bit LUT input
  .I1(I1), // 1-bit LUT input
  .I2(I2), // 1-bit LUT input
  .I3(I3), // 1-bit LUT input
  .I4(I4), // 1-bit LUT input
  .I5(I5)  // 1-bit LUT input (fast MUX select only available to O6 output)
);

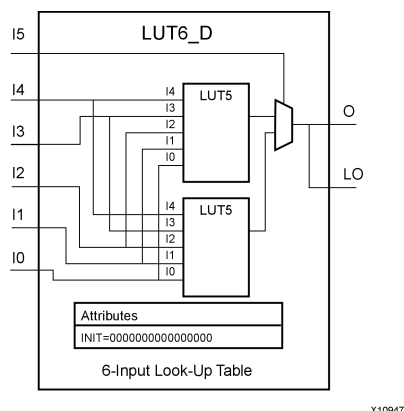
// End of LUT6_2_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## LUT6\_D

**プリミティブ：6-Input Lookup Table with General and Local Outputs**



### 概要

このデザイン エLEMENTは、入力 6 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 64 ビット ROM (6 ビットのアドレス指定) または 6 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT6 は、4 個のルックアップ テーブル (LUT) のいずれかにマップされます。LUT6、LUT6\_L、および LUT6\_D の機能は同じですが、LUT6\_L および LUT6\_D では、LUT6 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT6\_L では LUT6 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT6\_D では LUT6 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数値を設定する必要があります。入力 が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を 64'h8000000000000000 (VHDL では X"8000000000000000") に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値を 64'hffffffff (VHDL では X"FFFFFFFFFFFFFFFF") に設定すると、入力すべてが 0 の場合以外は出力は 1 になります (6 入力 OR ゲート)。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法：**リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	0	1	0	INIT[2]	INIT[2]

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
0	0	0	0	1	1	INIT[3]	INIT[3]
0	0	0	1	0	0	INIT[4]	INIT[4]
0	0	0	1	0	1	INIT[5]	INIT[5]
0	0	0	1	1	0	INIT[6]	INIT[6]
0	0	0	1	1	1	INIT[7]	INIT[7]
0	0	1	0	0	0	INIT[8]	INIT[8]
0	0	1	0	0	1	INIT[9]	INIT[9]
0	0	1	0	1	0	INIT[10]	INIT[10]
0	0	1	0	1	1	INIT[11]	INIT[11]
0	0	1	1	0	0	INIT[12]	INIT[12]
0	0	1	1	0	1	INIT[13]	INIT[13]
0	0	1	1	1	0	INIT[14]	INIT[14]
0	0	1	1	1	1	INIT[15]	INIT[15]
0	1	0	0	0	0	INIT[16]	INIT[16]
0	1	0	0	0	1	INIT[17]	INIT[17]
0	1	0	0	1	0	INIT[18]	INIT[18]
0	1	0	0	1	1	INIT[19]	INIT[19]
0	1	0	1	0	0	INIT[20]	INIT[20]
0	1	0	1	0	1	INIT[21]	INIT[21]
0	1	0	1	1	0	INIT[22]	INIT[22]
0	1	0	1	1	1	INIT[23]	INIT[23]
0	1	1	0	0	0	INIT[24]	INIT[24]
0	1	1	0	0	1	INIT[25]	INIT[25]
0	1	1	0	1	0	INIT[26]	INIT[26]
0	1	1	0	1	1	INIT[27]	INIT[27]
0	1	1	1	0	0	INIT[28]	INIT[28]
0	1	1	1	0	1	INIT[29]	INIT[29]
0	1	1	1	1	0	INIT[30]	INIT[30]
0	1	1	1	1	1	INIT[31]	INIT[31]
1	0	0	0	0	0	INIT[32]	INIT[32]
1	0	0	0	0	1	INIT[33]	INIT[33]
1	0	0	0	1	0	INIT[34]	INIT[34]
1	0	0	0	1	1	INIT[35]	INIT[35]
1	0	0	1	0	0	INIT[36]	INIT[36]
1	0	0	1	0	1	INIT[37]	INIT[37]

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
1	0	0	1	1	0	INIT[38]	INIT[38]
1	0	0	1	1	1	INIT[39]	INIT[39]
1	0	1	0	0	0	INIT[40]	INIT[40]
1	0	1	0	0	1	INIT[41]	INIT[41]
1	0	1	0	1	0	INIT[42]	INIT[42]
1	0	1	0	1	1	INIT[43]	INIT[43]
1	0	1	1	0	0	INIT[44]	INIT[44]
1	0	1	1	0	1	INIT[45]	INIT[45]
1	0	1	1	1	0	INIT[46]	INIT[46]
1	0	1	1	1	1	INIT[47]	INIT[47]
1	1	0	0	0	0	INIT[48]	INIT[48]
1	1	0	0	0	1	INIT[49]	INIT[49]
1	1	0	0	1	0	INIT[50]	INIT[50]
1	1	0	0	1	1	INIT[51]	INIT[51]
1	1	0	1	0	0	INIT[52]	INIT[52]
1	1	0	1	0	1	INIT[53]	INIT[53]
1	1	0	1	1	0	INIT[54]	INIT[54]
1	1	0	1	1	1	INIT[55]	INIT[55]
1	1	1	0	0	0	INIT[56]	INIT[56]
1	1	1	0	0	1	INIT[57]	INIT[57]
1	1	1	0	1	0	INIT[58]	INIT[58]
1	1	1	0	1	1	INIT[59]	INIT[59]
1	1	1	1	0	0	INIT[60]	INIT[60]
1	1	1	1	0	1	INIT[61]	INIT[61]
1	1	1	1	1	0	INIT[62]	INIT[62]
1	1	1	1	1	1	INIT[63]	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値							

## ポートの説明

ポート名	方向	幅	機能
O6	出力	1	6/5 LUT 出力
O5	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力



## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT6_D: 6-input Look-Up Table with general and local outputs
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT6_D_inst : LUT6_D
generic map (
  INIT => X"0000000000000000") -- Specify LUT contents
port map (
  LO => LO, -- LUT local output
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4, -- LUT input
  I5 => I5 -- LUT input
);

-- End of LUT6_D_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT6_D: 6-input Look-Up Table with general and local outputs
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT6_D #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_D_inst (
  .LO(LO), // LUT local output
  .O(O), // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4), // LUT input
  .I5(I5) // LUT input
);

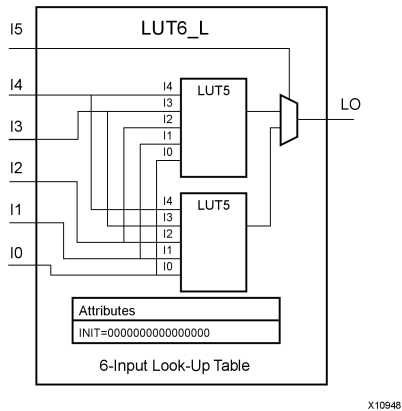
// End of LUT6_D_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## LUT6\_L

**プリミティブ：6-Input Lookup Table with Local Output**



### 概要

このデザイン エLEMENTは、入力 6 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 64 ビット ROM (6 ビットのアドレス指定) または 6 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT6 は、4 個のルックアップ テーブル (LUT) のいずれかにマップされます。LUT6、LUT6\_L、および LUT6\_D の機能は同じですが、LUT6\_L および LUT6\_D では、LUT6 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT6\_L では LUT6 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT6\_D では LUT6 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数値を設定する必要があります。入力 が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を `64'h8000000000000000` (VHDL では `X"8000000000000000"`) に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値を `64'hfffffffffffffffe` (VHDL では `X"FFFFFFFFFFFFFFFFFfe"`) に設定すると、入力すべてが 0 の場合以外は出力は 1 になります (6 入力 OR ゲート)。

FPGA LUT プリミティブでは、INIT パラメーターで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するため、INIT 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：** LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。

**論理式を使用する方法：** リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、コードで最初に適切なパラメーターを指定する必要があります。

### 論理表

入力						出力
I5	I4	I3	I2	I1	I0	LO
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]
0	0	0	0	1	0	INIT[2]

入力						出力
I5	I4	I3	I2	I1	I0	LO
0	0	0	0	1	1	INIT[3]
0	0	0	1	0	0	INIT[4]
0	0	0	1	0	1	INIT[5]
0	0	0	1	1	0	INIT[6]
0	0	0	1	1	1	INIT[7]
0	0	1	0	0	0	INIT[8]
0	0	1	0	0	1	INIT[9]
0	0	1	0	1	0	INIT[10]
0	0	1	0	1	1	INIT[11]
0	0	1	1	0	0	INIT[12]
0	0	1	1	0	1	INIT[13]
0	0	1	1	1	0	INIT[14]
0	0	1	1	1	1	INIT[15]
0	1	0	0	0	0	INIT[16]
0	1	0	0	0	1	INIT[17]
0	1	0	0	1	0	INIT[18]
0	1	0	0	1	1	INIT[19]
0	1	0	1	0	0	INIT[20]
0	1	0	1	0	1	INIT[21]
0	1	0	1	1	0	INIT[22]
0	1	0	1	1	1	INIT[23]
0	1	1	0	0	0	INIT[24]
0	1	1	0	0	1	INIT[25]
0	1	1	0	1	0	INIT[26]
0	1	1	0	1	1	INIT[27]
0	1	1	1	0	0	INIT[28]
0	1	1	1	0	1	INIT[29]
0	1	1	1	1	0	INIT[30]
0	1	1	1	1	1	INIT[31]
1	0	0	0	0	0	INIT[32]
1	0	0	0	0	1	INIT[33]
1	0	0	0	1	0	INIT[34]
1	0	0	0	1	1	INIT[35]
1	0	0	1	0	0	INIT[36]
1	0	0	1	0	1	INIT[37]

入力						出力
I5	I4	I3	I2	I1	I0	LO
1	0	0	1	1	0	INIT[38]
1	0	0	1	1	1	INIT[39]
1	0	1	0	0	0	INIT[40]
1	0	1	0	0	1	INIT[41]
1	0	1	0	1	0	INIT[42]
1	0	1	0	1	1	INIT[43]
1	0	1	1	0	0	INIT[44]
1	0	1	1	0	1	INIT[45]
1	0	1	1	1	0	INIT[46]
1	0	1	1	1	1	INIT[47]
1	1	0	0	0	0	INIT[48]
1	1	0	0	0	1	INIT[49]
1	1	0	0	1	0	INIT[50]
1	1	0	0	1	1	INIT[51]
1	1	0	1	0	0	INIT[52]
1	1	0	1	0	1	INIT[53]
1	1	0	1	1	0	INIT[54]
1	1	0	1	1	1	INIT[55]
1	1	1	0	0	0	INIT[56]
1	1	1	0	0	1	INIT[57]
1	1	1	0	1	0	INIT[58]
1	1	1	0	1	1	INIT[59]
1	1	1	1	0	0	INIT[60]
1	1	1	1	0	1	INIT[61]
1	1	1	1	1	0	INIT[62]
1	1	1	1	1	1	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値						

## ポートの説明

ポート名	方向	幅	機能
LO	出力	1	6/5 入力 LUT 出力または内部 CLB 接続
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT6_L: 6-input Look-Up Table with local output
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

LUT6_L_inst : LUT6_L
generic map (
  INIT => X"0000000000000000") -- Specify LUT Contents
port map (
  LO => LO, -- LUT local output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4, -- LUT input
  I5 => I5 -- LUT input
);

-- End of LUT6_L_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT6_L: 6-input Look-Up Table with local output
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

LUT6_L #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_L_inst (
  .LO(LO), // LUT local output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4), // LUT input
  .I5(I5) // LUT input
);

// End of LUT6_L_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## MCB

プリミティブ：Memory Control Block

### 概要

いくつかのよく使用されているメモリ インターフェイスをサポートするメモリ コントローラーです。MCB は、MIG (Memory Interface Generator) ツールでの使用のみでサポートされています。MCB の使用方法および機能の詳細は、『Spartan-6 FPGA メモリ コントローラー ユーザー ガイド』(UG388) を参照してください。

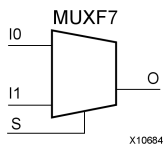
### 詳細情報

- ・ [『Spartan-6 FPGA メモリ コントローラー ユーザー ガイド』\(UG388\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)
- ・ [Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## MUXF7

### プリミティブ：2-to-1 Look-Up Table Multiplexer with General Output



## 概要

このデザイン エLEMENTは、2 つの LUT6 ルックアップ テーブルを組み合わせ、7 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、LUT6 の 2 つのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O は汎用インターコネクトです。

このほか、ローカル出力を持つ MUXF7\_D および MUXF7\_L があり、異なるタイミング モデルでレイアウト前のタイミングをより正確に予測する必要がある場合に使用できます。

## 論理表

入力			出力
S	I0	I1	O
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
I0	入力	1	入力 (MUXF6 LO 出力に接続)
I1	入力	1	入力 (MUXF6 LO 出力に接続)
S	入力	1	MUX への入力セレクト

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF7: CLB MUX to tie two LUT6's together with general output
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

MUXF7_inst : MUXF7
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,    -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S       -- Input select to MUX
);

-- End of MUXF7_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF7: CLB MUX to tie two LUT6's together with general output
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

MUXF7 MUXF7_inst (
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie to LUT6 O6 pin)
    .I1(I1),    // Input (tie to LUT6 O6 pin)
    .S(S)       // Input select to MUX
);

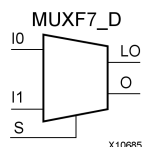
// End of MUXF7_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## MUXF7\_D

プリミティブ：2-to-1 Look-Up Table Multiplexer with Dual Output



### 概要

このデザイン エLEMENTは、2 つの LUT6 ルックアップ テーブルを組み合わせて、7 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、LUT6 の 2 つのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は汎用インターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

「MUXF7」および「MUXF7\_L」も参照してください。

### 論理表

入力			出力	
S	I0	I1	O	LO
0	I0	X	I0	I0
1	X	I1	I1	I1
X	0	0	0	0
X	1	1	1	1

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF6 LO 出力に接続)
I1	入力	1	入力 (MUXF6 LO 出力に接続)
S	入力	1	MUX への入力セレクト

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF7_D: CLB MUX to tie two LUT6's together with general and local outputs
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

MUXF7_D_inst : MUXF7_D
port map (
    LO => LO,  -- Output of MUX to local routing
    O  => O,   -- Output of MUX to general routing
    IO => IO,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S  => S    -- Input select to MUX
);

-- End of MUXF7_D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF7_D: CLB MUX to tie two LUT6's together with general and local outputs
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

MUXF7_D MUXF7_D_inst (
    .LO(LO),  // Output of MUX to local routing
    .O(O),   // Output of MUX to general routing
    .IO(IO), // Input (tie to LUT6 O6 pin)
    .I1(I1), // Input (tie to LUT6 O6 pin)
    .S(S)    // Input select to MUX
);

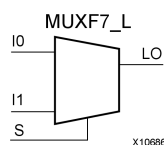
// End of MUXF7_D_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## MUXF7\_L

プリミティブ：2-to-1 look-up table Multiplexer with Local Output



### 概要

このデザイン エLEMENTは、2 つの LUT6 ルックアップ テーブルを組み合わせ、7 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、LUT6 の 2 つのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

「MUXF7」および「MUXF7\_D」も参照してください。

### 論理表

入力			出力
S	I0	I1	LO
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

### ポートの説明

ポート名	方向	幅	機能
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力
I1	入力	1	入力
S	入力	1	MUX への入力セレクト

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF7_L: CLB MUX to tie two LUT6's together with local output
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

MUXF7_L_inst : MUXF7_L
port map (
    LO => LO,  -- Output of MUX to local routing
    IO => IO,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S     -- Input select to MUX
);

-- End of MUXF7_L_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF7_L: CLB MUX to tie two LUT6's together with local output
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

MUXF7_L MUXF7_L_inst (
    .LO(IO), // Output of MUX to local routing
    .IO(IO), // Input (tie to LUT6 O6 pin)
    .I1(I1), // Input (tie to LUT6 O6 pin)
    .S(S)    // Input select to MUX
);

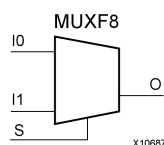
// End of MUXF7_L_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## MUXF8

プリミティブ：2-to-1 Look-Up Table Multiplexer with General Output



### 概要

このデザイン エLEMENTは、ルックアップ テーブル、MUXF5、MUXF6、および MUXF7 と組み合わせて、スライス 8 個に 8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサーを作成するためのマルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

### 論理表

入力			出力
S	I0	I1	O
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF8: CLB MUX to tie two MUXF7's together with general output
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

MUXF8_inst : MUXF8
port map (
    O => O,      -- Output of MUX to general routing
    IO => IO,    -- Input (tie to MUXF7 LO out)
    I1 => I1,    -- Input (tie to MUXF7 LO out)
    S => S       -- Input select to MUX
);

-- End of MUXF8_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF8: CLB MUX to tie two MUXF7's together with general output
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

MUXF8 MUXF8_inst (
    .O(O),      // Output of MUX to general routing
    .IO(IO),    // Input (tie to MUXF7 LO out)
    .I1(I1),    // Input (tie to MUXF7 LO out)
    .S(S)       // Input select to MUX
);

// End of MUXF8_inst instantiation
```

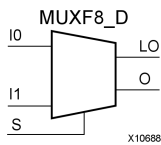
## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## MUXF8\_D

プリミティブ：2-to-1 Look-Up Table Multiplexer with Dual Output



### 概要

このデザイン エLEMENTは、ルックアップ テーブル、MUXF5、MUXF6、および MUXF7 を組み合わせて、スライス 8 個に 8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサーを作成するためのマルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は汎用インターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

### 論理表

入力			出力	
S	I0	I1	O	LO
0	I0	X	I0	I0
1	X	I1	I1	I1
X	0	0	0	0
X	1	1	1	1

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

MUXF8_D_inst : MUXF8_D
port map (
    LO => LO,  -- Output of MUX to local routing
    O  => O,   -- Output of MUX to general routing
    IO => IO,  -- Input (tie to MUXF7 LO out)
    I1 => I1,  -- Input (tie to MUXF7 LO out)
    S  => S    -- Input select to MUX
);

-- End of MUXF8_D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

MUXF8_D MUXF8_D_inst (
    .LO(LO),  // Output of MUX to local routing
    .O(O),   // Output of MUX to general routing
    .IO(IO), // Input (tie to MUXF7 LO out)
    .I1(I1), // Input (tie to MUXF7 LO out)
    .S(S)    // Input select to MUX
);

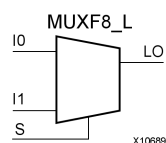
// End of MUXF8_D_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## MUXF8\_L

プリミティブ：2-to-1 Look-Up Table Multiplexer with Local Output



### 概要

このデザイン エLEMENTは、ルックアップ テーブル、MUXF5、MUXF6、および MUXF7 を組み合わせて、スライス 8 個に 8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサーを作成するためのマルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

### 論理表

入力			出力
S	I0	I1	LO
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

### ポートの説明

ポート名	方向	幅	機能
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF8_L: CLB MUX to tie two MUXF7's together with local output
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

MUXF8_L_inst : MUXF8_L
port map (
    LO => LO,  -- Output of MUX to local routing
    IO => IO,  -- Input (tie to MUXF7 LO out)
    I1 => I1,  -- Input (tie to MUXF7 LO out)
    S => S     -- Input select to MUX
);

-- End of MUXF8_L_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF8_L: CLB MUX to tie two MUXF7's together with local output
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

MUXF8_L MUXF8_L_inst (
    .LO(IO), // Output of MUX to local routing
    .IO(IO), // Input (tie to MUXF7 LO out)
    .I1(I1), // Input (tie to MUXF7 LO out)
    .S(S)    // Input select to MUX
);

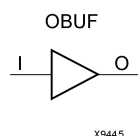
// End of MUXF8_L_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## OBUF

### プリミティブ：Output Buffer



### 概要

このデザイン エLEMENTは単純な出力バッファで、トライステートにならない (常に駆動される) FPGA デバイス ピンへの出力信号を駆動するために使用します。デザインのすべての出力ポートに OBUF、OBUFT、OBUFDS、OBUFTDS のいずれかを接続する必要があります。

このELEMENTは内部回路を外部から分離し、チップから出力する信号の駆動電流を供給します。I/O ブロック (IOB) 内にあります。出力 (O) は、OPAD または IOPAD に接続されます。このELEMENTでは、LVTTL 規格が使用され、DRIVE 制約と SLOW または FAST 制約を使用して駆動電流とスルー レートを選択できます。デフォルトでは、DRIVE は 12mA、スルー レートは SLOW に設定されています。

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	最上位出力ポートに直接接続される OBUF の出力
I	入力	1	OBUF の入力。出力ポートを駆動するロジックに接続

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	ELEMENTに I/O 規格を割り当てます。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- OBUF: Single-ended Output Buffer
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

OBUF_inst : OBUF
generic map (
    DRIVE => 12,
    IOSTANDARD => "DEFAULT",
    SLEW => "SLOW")
port map (
    O => O,      -- Buffer output (connect directly to top-level port)
    I => I       -- Buffer input
);

-- End of OBUF_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// OBUF: Single-ended Output Buffer
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

OBUF #(
    .DRIVE(12),    // Specify the output drive strength
    .IOSTANDARD("DEFAULT"), // Specify the output I/O standard
    .SLEW("SLOW") // Specify the output slew rate
) OBUF_inst (
    .O(O),        // Buffer output (connect directly to top-level port)
    .I(I)         // Buffer input
);

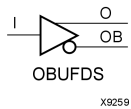
// End of OBUF_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## OBUFDS

### プリミティブ：Differential Signaling Output Buffer



### 概要

このデザイン エLEMENTは、低電圧の差動信号 (1.8V CMOS) をサポートする単一の出力バッファです。内部回路を外部から分離し、チップから出力する信号の駆動電流を供給します。出力には 2 つの異なるポート (O および OB) があり、これらのポートをそれぞれ「マスター」および「スレーブ」と呼びます。マスターとスレーブは MYNET と MYNETB のように、同じ論理信号の反対の状態を示します。

### 論理表

入力	出力	
I	O	OB
0	0	1
1	1	0

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	Diff_p 出力 (最上位ポートに直接接続)
OB	出力	1	Diff_n 出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	ELEMENTに I/O 規格を割り当てます。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- OBUFDS: Differential Output Buffer
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

OBUFDS_inst : OBUFDS
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Diff_p output (connect directly to top-level port)
    OB => OB,    -- Diff_n output (connect directly to top-level port)
    I => I       -- Buffer input
);

-- End of OBUFDS_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// OBUFDS: Differential Output Buffer
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

OBUFDS #(
    .IOSTANDARD("DEFAULT") // Specify the output I/O standard
) OBUFDS_inst (
    .O(O),      // Diff_p output (connect directly to top-level port)
    .OB(OB),    // Diff_n output (connect directly to top-level port)
    .I(I)       // Buffer input
);

// End of OBUFDS_inst instantiation
```

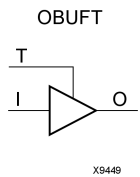
## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## OBUFT

### プリミティブ：3-State Output Buffer with Active Low Output Enable



## 概要

このデザイン エLEMENTは、入力 (I)、出力 (O)、アクティブ Low 出力イネーブル (T) を持つ単一のトライステート出力バッファです。このELEMENTでは、LVTTTL 規格が使用され、DRIVE 制約と SLOW または FAST 制約を使用して駆動電流とスルー レートを選択できます。デフォルトでは、DRIVE=12mA、スルー レートは SLOW に設定されています。

T が Low の場合、バッファに入力された値が対応する出力に送られます。T が High の場合は、出力がハイインピーダンス (オフまたは Z ステート) になります。OBUFT は、双方向 I/O を作成するなど、トライステート機能にシングルエンド出力を使用する必要がある場合に使用します。

## 論理表

入力		出力
T	I	O
1	X	Z
0	1	1
0	0	0

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファ出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力
T	入力	1	トライステート イネーブル入力

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	出力の駆動電流を指定します。デザインの駆動電流およびタイミング要件で許容される最も低い値を使用してください。
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。
SLEW	文字列	"SLOW"、"FAST"	"SLOW"	出力ドライバーのスルー レートを指定します。この属性の最適な設定方法は、データシートを参照してください。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- OBUFT: Single-ended 3-state Output Buffer
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

OBUFT_inst : OBUFT
generic map (
    DRIVE => 12,
    IOSTANDARD => "DEFAULT",
    SLEW => "SLOW")
port map (
    O => O,      -- Buffer output (connect directly to top-level port)
    I => I,      -- Buffer input
    T => T       -- 3-state enable input
);

-- End of OBUFT_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// OBUFT: Single-ended 3-state Output Buffer
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

OBUFT #(
    .DRIVE(12),    // Specify the output drive strength
    .IOSTANDARD("DEFAULT"), // Specify the output I/O standard
    .SLEW("SLOW") // Specify the output slew rate
) OBUFT_inst (
    .O(O),        // Buffer output (connect directly to top-level port)
    .I(I),        // Buffer input
    .T(T)         // 3-state enable input
);

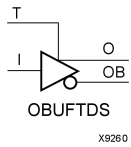
// End of OBUFT_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## OBUFTDS

プリミティブ：3-State Output Buffer with Differential Signaling, Active-Low Output Enable



### 概要

このデザイン エLEMENTは、低電圧差動信号をサポートする出力バッファです。OBUFTDS では、デザイン レベルのインターフェイス信号は、一方がマスターで、もう一方がスレーブとなる 2 つの異なるポート (O、OB) で表されます。マスターとスレーブは、MYNET\_P と MYNET\_N のように、同じ論理信号の反対の状態を示します。

### 論理表

入力		出力	
I	T	O	OB
X	1	Z	Z
0	0	0	1
1	0	1	0

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	Diff_p 出力 (最上位ポートに直接接続)
OB	出力	1	Diff_n 出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力
T	入力	1	トライステート イネーブル入力

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- OBUFTDS: Differential 3-state Output Buffer
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

OBUFTDS_inst : OBUFTDS
generic map (
  IOSTANDARD => "DEFAULT")
port map (
  O => O,      -- Diff_p output (connect directly to top-level port)
  OB => OB,    -- Diff_n output (connect directly to top-level port)
  I => I,      -- Buffer input
  T => T       -- 3-state enable input
);

-- End of OBUFTDS_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// OBUFTDS: Differential 3-state Output Buffer
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

OBUFTDS #(
  .IOSTANDARD("DEFAULT") // Specify the output I/O standard
) OBUFTDS_inst (
  .O(O),      // Diff_p output (connect directly to top-level port)
  .OB(OB),    // Diff_n output (connect directly to top-level port)
  .I(I),      // Buffer input
  .T(T)       // 3-state enable input
);

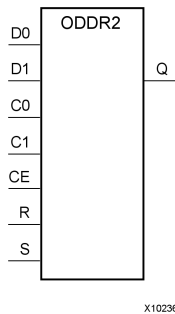
// End of OBUFTDS_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## ODDR2

**プリミティブ : Dual Data Rate Output D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset**



### 概要

このデザイン エLEMENTは、出力ダブル データ レート (DDR) レジスタで、FPGA から出力されるダブル データ レート 信号を生成するために使用します。ODDR2 は、C0 と C1 の 2 つのクロックを使用してコンポーネントに接続されるので、C0 および C1 の両方の立ち上がりエッジでデータが出力されます。ODDR2 は、レジスタの動作を停止するために使用できるアクティブ High のクロック イネーブル (CE) ポート、対応するクロックに同期または非同期になるよう設定できるセット/リセット ポートを備えています。ODDR2 には、1 クロックで取り込まれたデータを 2 クロックで出力するオプションの調整機能があります。

### 論理表

入力							出力
S	R	CE	D0	D1	C0	C1	O
1	X	X	X	X	X	X	1
0	1	X	X	X	X	X	0
0	0	0	X	X	X	X	変化なし
0	0	1	D0	X	↑	X	D0
0	0	1	X	D1	X	↑	D1

セット/リセットは SRTYPE 値で同期に設定可能

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DDR_ALIGNMENT	文字列	"NONE"、"C0"、 "C1"	"NONE"	DDR レジスタの入力キャプチャビヘイビアを設定します。  ・ "NONE" : C0 クロックが Low から High に切り替わる時に D0 入力に、C1 クロックが Low から High に切り替わる時に D1 にデータが入力されます。  ・ "C0" : D0 と D1 両方への入力が C0 クロックの立ち上がりエッジに同期します。  ・ "C1" : D0 と D1 両方への入力が C1 クロックの立ち上がりエッジに同期します。
INIT	2 進数	0, 1	0	Q 出力の初期値を 0 または 1 に設定します。
SRTYPE	文字列	"SYNC"、"ASYN"	"SYNC"	セット/リセットを同期または非同期に設定します。

## VHDL 記述 (インスタンスエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- ODDR2: Output Double Data Rate Output Register with Set, Reset
--      and Clock Enable.
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

ODDR2_inst : ODDR2
generic map(
  DDR_ALIGNMENT => "NONE", -- Sets output alignment to "NONE", "C0", "C1"
  INIT => '0', -- Sets initial state of the Q output to '0' or '1'
  SRTYPE => "SYNC") -- Specifies "SYNC" or "ASYN" set/reset
port map (
  Q => Q, -- 1-bit output data
  C0 => C0, -- 1-bit clock input
  C1 => C1, -- 1-bit clock input
  CE => CE, -- 1-bit clock enable input
  D0 => D0, -- 1-bit data input (associated with C0)
  D1 => D1, -- 1-bit data input (associated with C1)
  R => R, -- 1-bit reset input
  S => S -- 1-bit set input
);

-- End of ODDR2_inst instantiation

```

## Verilog 記述 (インスタンス化)

```
// ODDR2: Output Double Data Rate Output Register with Set, Reset
//      and Clock Enable.
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

ODDR2 #(
    .DDR_ALIGNMENT("NONE"), // Sets output alignment to "NONE", "C0" or "C1"
    .INIT(1'b0),             // Sets initial state of the Q output to 1'b0 or 1'b1
    .SRTYPE("SYNC") // Specifies "SYNC" or "ASYN" set/reset
) ODDR2_inst (
    .Q(Q), // 1-bit DDR output data
    .C0(C0), // 1-bit clock input
    .C1(C1), // 1-bit clock input
    .CE(CE), // 1-bit clock enable input
    .D0(D0), // 1-bit data input (associated with C0)
    .D1(D1), // 1-bit data input (associated with C1)
    .R(R), // 1-bit reset input
    .S(S) // 1-bit set input
);

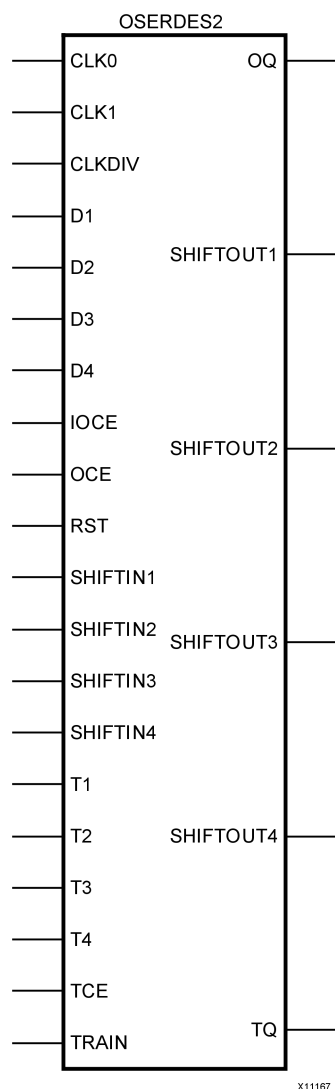
// End of ODDR2_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## OSERDES2

プリミティブ：Dedicated IOB Output Serializer



### 概要

各 IOB には出力シリアライザー ブロックが含まれており、OSERDES2 プリミティブを使用してデザインにインスタンスエートできます。OSERDES2 を使用すると、SerDes 比 2:1、3:1、および 4:1 のパラレル/シリアル変換が可能です。SerDes 比は、データを送信する高速 I/O クロックと、それより低速のパラレル データの処理に使用する内部グローバル クロックとの比です。たとえば、500MHz で動作する I/O クロックを使用して 500Mb/s でデータを送信する場合、OSERDES2 により 4 ビットのデータが 1/4 のレート (125MHz) で FPGA ロジックから転送されます。差動出力を使用する場合、2 つの IOB に関連付けられた 2 つの OSERDES2 プリミティブをカスケード接続して、SerDes 比 5:1、6:1、7:1、および 8:1 を達成できます。



## ポートの説明

ポート名	方向	幅	機能
CLKDIV	入力	1	グローバル クロック ネットワーク入力。ハードウェアのクロックです。
CLK0	入力	1	I/O クロック ネットワークの入力。オプションで反転可能です。クロック 2 通倍回路が使用されていない場合に使用されるプライマリクロック入力です。詳細は、DATA_RATE を参照してください。
CLK1	入力	1	I/O クロック ネットワーク入力。オプションで反転可能です。 クロック 2 通倍回路が使用されている場合のみに使用されるセカンダリクロック入力です。
D1 ~ D4	入力	1	パラレル データ入力
IOCE	入力	1	BUFIO2 CE から生成されたデータ ストローブ信号。選択されている SerDes モードでの I/O およびグローバル クロックに対し、正しいタイミングでデータがキャプチャされるようにします。
OCE	入力	1	データ入力のクロック イネーブル
OQ	出力	1	パッドまたは IODELAY2 へのデータ パス出力
RST	入力	1	共有データ、トライステートリセットピン。非同期のみ。
SHIFTIN1	入力	1	カスケード データ入力信号 (マスターではダミー)。DATA_WIDTH が 4 を超える場合に使用します。
SHIFTIN2	入力	1	カスケード トライステート入力信号 (マスターではダミー)。DATA_WIDTH が 4 を超える場合に使用します。
SHIFTIN3	入力	1	差動データ入力信号 (スレーブではダミー)
SHIFTIN4	入力	1	差動トライステート入力信号 (スレーブではダミー)
SHIFTOUT1	出力	1	カスケード データ出力信号 (スレーブではダミー)。DATA_WIDTH が 4 を超える場合に使用します。
SHIFTOUT2	出力	1	カスケード トライステート出力信号 (スレーブではダミー)。DATA_WIDTH が 4 を超える場合に使用します。
SHIFTOUT3	出力	1	差動データ入力信号 (マスターではダミー)
SHIFTOUT4	出力	1	差動トライステート出力信号 (マスターではダミー)
TCE	入力	1	トライステート入力のクロック イネーブル
TQ	出力	1	パッドまたは IODELAY2 へのトライステート パス出力
TRAIN	入力	1	トレーニング機能は、信号のレシーバーを補正するために使用する固定出力パターンを指定します。このピンを使用して、FPGA ロジックで出力が固定パターンであるか、ピンからの入力データであるかを制御できます。
T1 ~ T4	入力	1	トライステート制御入力

## デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
BYPASS_GCLK_FF	ブール代数	FALSE、TRUE	FALSE	CLKDIV 同期化レジスタをバイパスします。
DATA_RATE_OQ	文字列	"DDR"、"SDR"	"DDR"	データ レートを設定します。DDR クロックは、個別の複数の I/O クロックまたは 1 つの I/O クロックから供給できます。2 つのクロックが供給される場合、その位相差は約 180 度である必要があります。
DATA_RATE_OT	文字列	"DDR"、"BUF"、 "SDR"	"DDR"	トライステート データ レートを設定します。DDR クロックは、個別の複数の I/O クロックまたは 1 つの I/O クロックから供給できます。2 つのクロックが供給される場合、その位相差は約 180 度である必要があります。
DATA_WIDTH	整数	2、1、3、4、5、6、7、8	2	データ幅を指定します。パラレル/シリアル コンバーターのパラレル データ出力幅を定義します。5 以上の値は、2 つの OSERDES2 ブロックをカスケード接続する場合にのみ有効です。この場合、マスター ブロックとスレーブ ブロックに同じ値を適用する必要があります。
OUTPUT_MODE	文字列	"SINGLE_ENDED"、 "DIFFERENTIAL"	"SINGLE_ENDED"	出力モードを指定します。
SERDES_MODE	文字列	"MASTER"、 "SLAVE"	"MASTER"	OSERDES2 ブロックを 1 つのみ使用するか、2 つの OSERDES2 ブロックをカスケード接続する場合はマスターまたはスレーブのどちらとして使用するかを示します。
TRAIN_PATTERN	整数	0、1、2、3、4、5、6、 7、8、9、10、11、12、 13、14、15	0	TRAIN ポートがアクティブな場合に送信するトレーニング パターンを定義します。

## VHDL 記述 (インスタンスエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- OSERDES2: Output SERial/DESerializer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

OSERDES2_inst : OSERDES2
generic map (
    BYPASS_GCLK_FF => FALSE,           -- Bypass CLKDIV synchronization registers (TRUE/FALSE)
    DATA_RATE_OQ => "DDR",            -- Output Data Rate ("SDR" or "DDR")
    DATA_RATE_OT => "DDR",            -- 3-state Data Rate ("SDR" or "DDR")
    DATA_WIDTH => 2,                  -- Parallel data width (2-8)
    OUTPUT_MODE => "SINGLE_ENDED",      -- "SINGLE_ENDED" or "DIFFERENTIAL"
    SERDES_MODE => "NONE",             -- "NONE", "MASTER" or "SLAVE"
    TRAIN_PATTERN => 0                 -- Training Pattern (0-15)
)
port map (
    OQ => OQ,                          -- 1-bit output: Data output to pad or IODELAY2
    SHIFTOUT1 => SHIFTOUT1,            -- 1-bit output: Cascade data output
    SHIFTOUT2 => SHIFTOUT2,            -- 1-bit output: Cascade 3-state output
    SHIFTOUT3 => SHIFTOUT3,            -- 1-bit output: Cascade differential data output
    SHIFTOUT4 => SHIFTOUT4,            -- 1-bit output: Cascade differential 3-state output
    TQ => TQ,                          -- 1-bit output: 3-state output to pad or IODELAY2
    CLK0 => CLK0,                      -- 1-bit input: I/O clock input
    CLK1 => CLK1,                      -- 1-bit input: Secondary I/O clock input

```

```
CLKDIV => CLKDIV,          -- 1-bit input: Logic domain clock input
-- D1 - D4: 1-bit (each) input: Parallel data inputs
D1 => D1,
D2 => D2,
D3 => D3,
D4 => D4,
IOCE => IOCE,              -- 1-bit input: Data strobe input
OCE => OCE,                -- 1-bit input: Clock enable input
RST => RST,                 -- 1-bit input: Asynchronous reset input
SHIFTIN1 => SHIFTIN1,      -- 1-bit input: Cascade data input
SHIFTIN2 => SHIFTIN2,      -- 1-bit input: Cascade 3-state input
SHIFTIN3 => SHIFTIN3,      -- 1-bit input: Cascade differential data input
SHIFTIN4 => SHIFTIN4,      -- 1-bit input: Cascade differential 3-state input
-- T1 - T4: 1-bit (each) input: 3-state control inputs
T1 => T1,
T2 => T2,
T3 => T3,
T4 => T4,
TCE => TCE,                -- 1-bit input: 3-state clock enable input
TRAIN => TRAIN             -- 1-bit input: Training pattern enable input
);

-- End of OSERDES2_inst instantiation
```

## Verilog 記述 (インスタンスエーション)

```
// OSERDES2: Output SERIAL/DESerializer
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

OSERDES2 #(
    .BYPASS_CLK_FF("FALSE"), // Bypass CLKDIV synchronization registers (TRUE/FALSE)
    .DATA_RATE_OQ("DDR"),    // Output Data Rate ("SDR" or "DDR")
    .DATA_RATE_OT("DDR"),    // 3-state Data Rate ("SDR" or "DDR")
    .DATA_WIDTH(2),          // Parallel data width (2-8)
    .OUTPUT_MODE("SINGLE_ENDED"), // "SINGLE_ENDED" or "DIFFERENTIAL"
    .SERDES_MODE("NONE"),    // "NONE", "MASTER" or "SLAVE"
    .TRAIN_PATTERN(0)        // Training Pattern (0-15)
)
OSERDES2_inst (
    .OQ(OQ), // 1-bit output: Data output to pad or IODELAY2
    .SHIFTOUT1(SHIFTOUT1), // 1-bit output: Cascade data output
    .SHIFTOUT2(SHIFTOUT2), // 1-bit output: Cascade 3-state output
    .SHIFTOUT3(SHIFTOUT3), // 1-bit output: Cascade differential data output
    .SHIFTOUT4(SHIFTOUT4), // 1-bit output: Cascade differential 3-state output
    .TQ(TQ), // 1-bit output: 3-state output to pad or IODELAY2
    .CLK0(CLK0), // 1-bit input: I/O clock input
    .CLK1(CLK1), // 1-bit input: Secondary I/O clock input
    .CLKDIV(CLKDIV), // 1-bit input: Logic domain clock input
    // D1 - D4: 1-bit (each) input: Parallel data inputs
    .D1(D1),
    .D2(D2),
    .D3(D3),
    .D4(D4),
    .IOCE(IOCE), // 1-bit input: Data strobe input
    .OCE(OCE), // 1-bit input: Clock enable input
    .RST(RST), // 1-bit input: Asynchronous reset input
    .SHIFTIN1(SHIFTIN1), // 1-bit input: Cascade data input
    .SHIFTIN2(SHIFTIN2), // 1-bit input: Cascade 3-state input
    .SHIFTIN3(SHIFTIN3), // 1-bit input: Cascade differential data input
    .SHIFTIN4(SHIFTIN4), // 1-bit input: Cascade differential 3-state input
    // T1 - T4: 1-bit (each) input: 3-state control inputs
    .T1(T1),
    .T2(T2),
    .T3(T3),
    .T4(T4),
    .TCE(TCE), // 1-bit input: 3-state clock enable input
    .TRAIN(TRAIN) // 1-bit input: Training pattern enable input
);

// End of OSERDES2_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

プリミティブ : PCI Express

X11170

## 概要

このELEMENTは、RocketIO™ トランシーバー、ブロック RAM、さまざまなクロック リソースなど、FPGA のほかのリソースと併用します。PCI EXPRESS® デザインを PCIE\_A1 を使用してインプリメントするには、必ず ISE® Design Suite に含まれる CORE Generator™ ツールを使用して PCI EXPRESS デザイン用の LogiCORE™ IP コアを作成してください。LogiCORE は、PCIE\_A1 ソフトウェア プリミティブをインスタンス化し、インターフェイスを FPGA リソースに接続し、すべての属性を設定して、シンプルでユーザーにとって使いやすいインターフェイスを提供します。

## デザインの入力方法

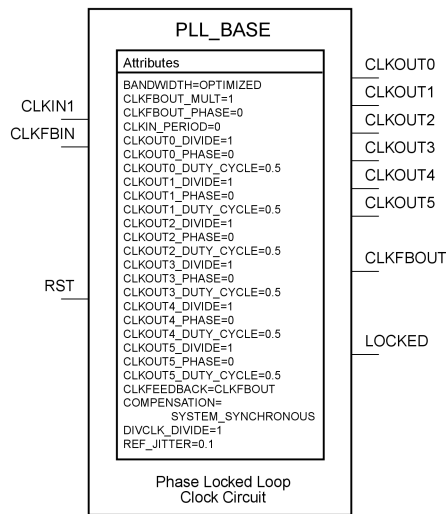
このELEMENTをインスタンス化するには、PCI EXPRESS® コアまたはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンス化しないでください。

## 詳細情報

- ・ [『Spartan-6 FPGA RocketIO GTP トランシーバー ユーザー ガイド』\(UG386\)](#)
- ・ [『PCI EXPRESS® 用 LogiCORE™ IP Spartan-6 FPGA 統合エンドポイント ブロック v1.1 ユーザー ガイド』\(UG654\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## PLL\_BASE

### プリミティブ：Basic Phase Locked Loop Clock Circuit



X10951

## 概要

このデザイン エLEMENTは、FPGA 内部および外部回路の両方でクロック合成およびクロック管理機能を提供するためのエンベデッド位相ロック ループ クロック回路で、PLL\_ADV デザイン エLEMENTのサブセットです。PLL\_BASE を使用すると、ほとんどの PLL クロック回路を簡単に統合できます。PLL のすべての機能は備わっていませんが、入力クロックを位相シフト、通倍、分周でき、またデューティ サイクルを変更したり、ジッターをフィルター処理する機能などがあります。

## ポートの説明

ポート名	方向	幅	機能
クロック出力/入力			
CLKOUT0 ~ 5	出力	1	位相シフトされる 6 個の出力クロック
CLKFBOUT	出力	1	クロック ネットワークの遅延調整方法を指定するために使用する専用 PLL フィードバック出力。調整方法によって、この出力を接続する場合と接続する必要のない場合があります。
CLKIN	入力	1	PLL のクロック ソース入力。FPGA の専用クロックピン、DCM 出力クロックピン、または BUFG 出力ピンで駆動できます。
CLKFBIN	入力	1	クロック フィードバック入力。CLKFBOUT ポートでのみ駆動できます。
ステータス出力/制御入力			
LOCKED	出力	1	位相アライメントが完了し、操作が開始可能であることを示す非同期出力
RST	入力	1	非同期リセット

## デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
COMPENSATION	文字列	"SYSTEM_SYNCHRONOUS"、 "SOURCE_SYNCHRONOUS"	"SYSTEM_SYNCHRONOUS"	入力クロックの PLL 位相調整を指定します。すべてのクロック遅延を調整する場合は "SYSTEM_SYNCHRONOUS" を、クロックがデータと共に供給されて位相が揃っているときには "SOURCE_SYNCHRONOUS" を使用します。
BANDWIDTH	文字列	"HIGH"、"LOW"、 "OPTIMIZED"	"OPTIMIZED"	ジッター、位相マージンなどの PLL 特性に影響する PLL プログラム アルゴリズムを指定します。
CLKOUT0_DIVIDE、 CLKOUT1_DIVIDE、 CLKOUT2_DIVIDE、 CLKOUT3_DIVIDE、 CLKOUT4_DIVIDE、 CLKOUT5_DIVIDE	整数	1 ～ 128	1	CLKOUT クロック出力を分周する値を指定します。この値と FBCLKOUT_MULT 値により出力周波数が決まります。
CLKOUT0_PHASE、 CLKOUT1_PHASE、 CLKOUT2_PHASE、 CLKOUT3_PHASE、 CLKOUT4_PHASE、 CLKOUT5_PHASE	実数	0.0 ～ 360.0	0.0	CLKOUT クロック出力との位相オフセットを度数で指定します。90 は 90 度 (4 分の 1 サイクル) の位相オフセット、180 は 180 度 (2 分の 1 サイクル) の位相オフセットを示します。
CLKOUT0_DUTY_CYCLE、 CLKOUT1_DUTY_CYCLE、 CLKOUT2_DUTY_CYCLE、 CLKOUT3_DUTY_CYCLE、 CLKOUT4_DUTY_CYCLE、 CLKOUT5_DUTY_CYCLE	実数	0.01 ～ 0.99	0.50	CLKOUT クロック出力のデューティサイクルをパーセントで指定します。0.50 に設定すると、デューティサイクルは 50% になります。
CLKFBOUT_MULT	整数	1 ～ 64	1	すべての CLKOUT クロック出力を逡倍する値を指定します。この値と CLKOUT#_DIVIDE 値により出力周波数が決まります。
DIVCLK_DIVIDE	整数	1 ～ 52	1	すべての出力クロックの分周比を指定します。
CLKFBOUT_PHASE	実数	0.0 ～ 360	0.0	クロック フィードバック出力の位相オフセットを度数で指定します。



属性	データ型	値	デフォルト	説明
CLK_FEEDBACK	文字列	"CLKFBOUT"、 "CLKOUT0"	"CLKFBOUT"	CLKFBIN を駆動するクロックソースを指定します。フィードバックリソースの正しい使用法および VCO 周波数の算出方法は、『Spartan®-6 FPGA クロックリソース ユーザー ガイド』(UG382) を参照してください。
REF_JITTER	実数	0 ~ 0.999	0.100	基準クロック ジッターを基準クロックの割合を示す UI (ユニット インターバル) で指定します。この値は、入力クロックの最大ピークトゥピーク値にします。
CLKIN_PERIOD	実数	1.000 ~ 52.630	なし	CLKIN の入力周期を ns で指定します。

## VHDL 記述 (インスタンスエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- PLL_BASE: Phase Locked Loop (PLL) Clock Management Component
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

PLL_BASE_inst : PLL_BASE
generic map (
    BANDWIDTH => "OPTIMIZED",           -- "HIGH", "LOW" or "OPTIMIZED"
    CLKFBOUT_MULT => 1,                  -- Multiply value for all CLKOUT clock outputs (1-64)
    CLKFBOUT_PHASE => 0.0,               -- Phase offset in degrees of the clock feedback output
                                         -- (0.0-360.0).
    CLKIN_PERIOD => 0.0,                 -- Input clock period in ns to ps resolution (i.e. 33.333 is 30
                                         -- MHz).
    -- CLKOUT0_DIVIDE - CLKOUT5_DIVIDE: Divide amount for CLKOUT# clock output (1-128)
    CLKOUT0_DIVIDE => 1,
    CLKOUT1_DIVIDE => 1,
    CLKOUT2_DIVIDE => 1,
    CLKOUT3_DIVIDE => 1,
    CLKOUT4_DIVIDE => 1,
    CLKOUT5_DIVIDE => 1,
    -- CLKOUT0_DUTY_CYCLE - CLKOUT5_DUTY_CYCLE: Duty cycle for CLKOUT# clock output (0.01-0.99).
    CLKOUT0_DUTY_CYCLE => 0.5,
    CLKOUT1_DUTY_CYCLE => 0.5,
    CLKOUT2_DUTY_CYCLE => 0.5,
    CLKOUT3_DUTY_CYCLE => 0.5,
    CLKOUT4_DUTY_CYCLE => 0.5,
    CLKOUT5_DUTY_CYCLE => 0.5,
    -- CLKOUT0_PHASE - CLKOUT5_PHASE: Output phase relationship for CLKOUT# clock output (-360.0-360.0).
    CLKOUT0_PHASE => 0.0,
    CLKOUT1_PHASE => 0.0,
    CLKOUT2_PHASE => 0.0,
    CLKOUT3_PHASE => 0.0,
    CLKOUT4_PHASE => 0.0,
    CLKOUT5_PHASE => 0.0,
    CLK_FEEDBACK => "CLKFBOUT",          -- Clock source to drive CLKFBIN ("CLKFBOUT" or "CLKOUT0")
    COMPENSATION => "SYSTEM_SYNCHRONOUS", -- "SYSTEM_SYNCHRONOUS", "SOURCE_SYNCHRONOUS", "EXTERNAL"
    DIVCLK_DIVIDE => 1,                  -- Division value for all output clocks (1-52)
    REF_JITTER => 0.1,                  -- Reference Clock Jitter in UI (0.000-0.999).
    RESET_ON_LOSS_OF_LOCK => FALSE      -- Must be set to FALSE
)
port map (
    CLKFBOUT => CLKFBOUT, -- 1-bit output: PLL_BASE feedback output
    -- CLKOUT0 - CLKOUT5: 1-bit (each) output: Clock outputs
    CLKOUT0 => CLKOUT0,
    CLKOUT1 => CLKOUT1,
    CLKOUT2 => CLKOUT2,

```

```

CLKOUT3 => CLKOUT3,
CLKOUT4 => CLKOUT4,
CLKOUT5 => CLKOUT5,
LOCKED => LOCKED,      -- 1-bit output: PLL_BASE lock status output
CLKFBIN => CLKFBIN,     -- 1-bit input: Feedback clock input
CLKIN => CLKIN,         -- 1-bit input: Clock input
RST => RST              -- 1-bit input: Reset input
);

-- End of PLL_BASE_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// PLL_BASE: Phase Locked Loop (PLL) Clock Management Component
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

PLL_BASE #(
    .BANDWIDTH("OPTIMIZED"),           // "HIGH", "LOW" or "OPTIMIZED"
    .CLKFBOUT_MULT(1),                 // Multiply value for all CLKOUT clock outputs (1-64)
    .CLKFBOUT_PHASE(0.0),              // Phase offset in degrees of the clock feedback output (0.0-360.0).
    .CLKIN_PERIOD(0.0),               // Input clock period in ns to ps resolution (i.e. 33.333 is 30
                                     // MHz).
    // CLKOUT0_DIVIDE - CLKOUT5_DIVIDE: Divide amount for CLKOUT# clock output (1-128)
    .CLKOUT0_DIVIDE(1),
    .CLKOUT1_DIVIDE(1),
    .CLKOUT2_DIVIDE(1),
    .CLKOUT3_DIVIDE(1),
    .CLKOUT4_DIVIDE(1),
    .CLKOUT5_DIVIDE(1),
    // CLKOUT0_DUTY_CYCLE - CLKOUT5_DUTY_CYCLE: Duty cycle for CLKOUT# clock output (0.01-0.99).
    .CLKOUT0_DUTY_CYCLE(0.5),
    .CLKOUT1_DUTY_CYCLE(0.5),
    .CLKOUT2_DUTY_CYCLE(0.5),
    .CLKOUT3_DUTY_CYCLE(0.5),
    .CLKOUT4_DUTY_CYCLE(0.5),
    .CLKOUT5_DUTY_CYCLE(0.5),
    // CLKOUT0_PHASE - CLKOUT5_PHASE: Output phase relationship for CLKOUT# clock output (-360.0-360.0).
    .CLKOUT0_PHASE(0.0),
    .CLKOUT1_PHASE(0.0),
    .CLKOUT2_PHASE(0.0),
    .CLKOUT3_PHASE(0.0),
    .CLKOUT4_PHASE(0.0),
    .CLKOUT5_PHASE(0.0),
    .CLK_FEEDBACK("CLKFBOUT"),         // Clock source to drive CLKFBIN ("CLKFBOUT" or "CLKOUT0")
    .COMPENSATION("SYSTEM_SYNCHRONOUS"), // "SYSTEM_SYNCHRONOUS", "SOURCE_SYNCHRONOUS", "EXTERNAL"
    .DIVCLK_DIVIDE(1),                 // Division value for all output clocks (1-52)
    .REF_JITTER(0.1),                 // Reference Clock Jitter in UI (0.000-0.999).
    .RESET_ON_LOSS_OF_LOCK("FALSE")    // Must be set to FALSE
)
PLL_BASE_inst (
    .CLKFBOUT(CLKFBOUT), // 1-bit output: PLL_BASE feedback output
    // CLKOUT0 - CLKOUT5: 1-bit (each) output: Clock outputs
    .CLKOUT0(CLKOUT0),
    .CLKOUT1(CLKOUT1),
    .CLKOUT2(CLKOUT2),
    .CLKOUT3(CLKOUT3),
    .CLKOUT4(CLKOUT4),
    .CLKOUT5(CLKOUT5),
    .LOCKED(LOCKED),      // 1-bit output: PLL_BASE lock status output
    .CLKFBIN(CLKFBIN),    // 1-bit input: Feedback clock input
    .CLKIN(CLKIN),        // 1-bit input: Clock input
    .RST(RST)             // 1-bit input: Reset input
);

// End of PLL_BASE_inst instantiation

```

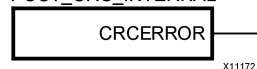
## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## POST\_CRC\_INTERNAL

**プリミティブ**：Post-configuration CRC error detection

POST\_CRC\_INTERNAL



### 概要

このプリミティブを使用しハードウェアから CRC エラーを報告します。この新しいプリミティブは POST\_CRC を拡張するために追加されています。また、CRC\_EXTSTAT\_DISABLE がアクティブな場合の POST CRC ステータスへの唯一のアクセスでもあります。

### ポートの説明

ポート名	方向	幅	機能
CRCERROR	出力	1	コンフィギュレーション後の CRC エラー

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- POST_CRC_INTERNAL: Post-configuration CRC error detection
--                      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

POST_CRC_INTERNAL_inst : POST_CRC_INTERNAL
port map (
  CRCERROR => CRCERROR -- 1-bit output: Post-configuration CRC error output
);

-- End of POST_CRC_INTERNAL_inst instantiation
  
```

### Verilog 記述 (インスタンス化)

```

// POST_CRC_INTERNAL: Post-configuration CRC error detection
//                      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

POST_CRC_INTERNAL POST_CRC_INTERNAL_inst (
  .CRCERROR(CRCERROR) // 1-bit output: Post-configuration CRC error output
);

// End of POST_CRC_INTERNAL_inst instantiation
  
```

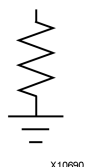
## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュレーション ユーザー ガイド』\(UG380\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## PULLDOWN

### プリミティブ：Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs

PULLDOWN



X10690

## 概要

この抵抗エレメントは、入力、出力、双方向のパッドに接続し、フロートする可能性のあるノードのロジックレベルを Low にします。

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	プルダウン出力 (最上位ポートに直接接続)

## デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- PULLDOWN: I/O Buffer Weak Pull-down
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

PULLDOWN_inst : PULLDOWN
port map (
  O => O      -- Pulldown output (connect directly to top-level port)
);

-- End of PULLDOWN_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// PULLDOWN: I/O Buffer Weak Pull-down
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

PULLDOWN PULLDOWN_inst (
    .O(0)      // Pulldown output (connect directly to top-level port)
);

// End of PULLDOWN_inst instantiation
```

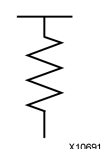
## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## PULLUP

**プリミティブ : Resistor to VCC for Input PADs, Open-Drain, and 3-State Outputs**

PULLUP



### 概要

このデザイン エLEMENTは、入力、トライステート出力、または双方向ポートが内部または外部ソースで駆動されないときに、弱い High で駆動します。すべてのドライバーがオフのときに、オープンドレイン エLEMENTおよびマクロのロジックレベルを High にします。

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	プルアップ出力 (最上位ポートに直接接続)

### デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- PULLUP: I/O Buffer Weak Pull-up
--          Spartan-6
--          Xilinx HDL Libraries Guide, version 14.5

PULLUP_inst : PULLUP
port map (
  O => O      -- Pullup output (connect directly to top-level port)
);

-- End of PULLUP_inst instantiation
```



## Verilog 記述 (インスタンス化)

```
// PULLUP: I/O Buffer Weak Pull-up
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

PULLUP PULLUP_inst (
    .O(0)      // Pullup output (connect directly to top-level port)
);

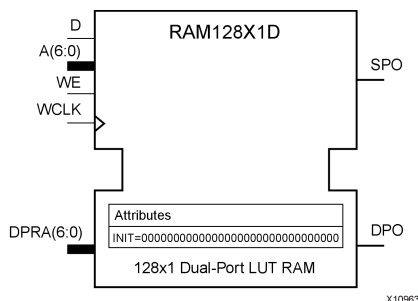
// End of PULLUP_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA SelectIO リソース ユーザー ガイド』\(UG381\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## RAM128X1D

### プリミティブ：128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)



### 概要

このデザイン エLEMENTは 128 ワード X 1 ビットの RAM で読み出し/書き込みポートがあり、ライト イネーブル (WE) が High のときにアドレス バス A で指定されたロケーションに D 入力データピンの値が書き込まれます。この書き込みは WCLK の立ち上がりエッジの直後に実行され、同じ値が SPO に出力されます。WE が Low のときは非同期読み出しが実行され、アドレス バス A で指定されたメモリ ロケーションの値が SPO に非同期で出力されます。アドレス バス DPRA の値を変更することにより、読み出しポートでは非同期読み出しを実行できます。DPO にその値が出力されます。

### ポートの説明

ポート名	方向	幅	機能
SPO	出力	1	アドレス バス A で指定される読み出し/書き込みポートのデータ出力
DPO	出力	1	アドレス バス DPRA で指定される読み出しポートのデータ出力
D	入力	1	アドレス バス A で指定される書き込みデータ入力
A	入力	7	読み出し/書き込みポートのアドレス バス
DPRA	入力	7	読み出しポートのアドレス バス
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

インスタンスエートする場合は、このコンポーネントを次のように接続します。

- WCLK 入力をクロック ソースに、D 入力を格納するデータ ソースに、DPO 出力を FDCE の D 入力などの適切なデスティネーションに接続します。
- オプションで、SPO 出力を適切なデスティネーションに接続するか、または未接続にすることもできます。
- クロック イネーブル ピン (WE) は、適切なライト イネーブル ソースに接続します。
- 7 ビット バス A は読み出し/書き込みアドレスに、7 ビット バス DPRA は読み出しアドレスに接続する必要があります。
- 128 ビットの 16 進数で構成される INIT 属性で、RAM の初期値を指定できます。

指定しない場合は、初期値はすべて 0 になります。

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	128 ビット値	すべて 0	RAM の初期値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM128X1D: 128-deep by 1-wide positive edge write, asynchronous read
--             dual-port distributed LUT RAM
--             Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM128X1D_inst : RAM128X1D
generic map (
  INIT => X"00000000000000000000000000000000"
)
port map (
  DPO => DPO,      -- Read/Write port 1-bit output
  SPO => SPO,      -- Read port 1-bit output
  A => A,          -- Read/Write port 7-bit address input
  D => D,          -- RAM data input
  DPRA => DPRA,    -- Read port 7-bit address input
  WCLK => WCLK,    -- Write clock input
  WE => WE         -- RAM data input
);

-- End of RAM128X1D_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAM128X1D: 128-deep by 1-wide positive edge write, asynchronous read
//             dual-port distributed LUT RAM
//             Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM128X1D #(
  .INIT(128'h00000000000000000000000000000000)
) RAM128X1D_inst (
  .DPO(DPO),      // Read port 1-bit output
  .SPO(SPO),      // Read/Write port 1-bit output
  .A(A),          // Read/Write port 7-bit address input
  .D(D),          // RAM data input
  .DPRA(DPRA),    // Read port 7-bit address input
  .WCLK(WCLK),    // Write clock input
  .WE(WE)         // Write enable input
);

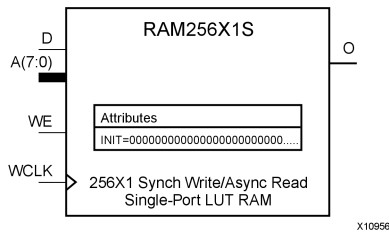
// End of RAM128X1D_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## RAM256X1S

### プリミティブ：256-Deep by 1-Wide Random Access Memory (Select RAM)



## 概要

このデザイン エLEMENTは、256 ワード X 1 ビットの RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM と呼ばれる) を使用してインプリメントされるため、ブロック RAM リソースは使用しません。同期読み出しを行う場合は、出力にレジスタを付けて同じスライスに配置できます。この場合、RAM とレジスタで同じクロックを使用する必要があります。アクティブ High のライト イネーブル WE が High になると、WCLK ピンの立ち上がりエッジで D 入力データピンの値がメモリ アレイに書き込まれます。出力 O は、WE の値にかかわらず、アドレス バス A で指定されたメモリ ロケーションの値を出力します。書き込みが実行されると、その直後に出力の値が新しい値に更新されます。

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	アドレス バス A で指定される読み出し/書き込みポートのデータ出力
D	入力	1	アドレス バス A で指定される書き込みデータ入力
A	入力	8	読み出し/書き込みポートのアドレス バス
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

インスタンス化する場合、このコンポーネントを次のように接続します。

- WCLK 入力をクロック ソースに、D 入力を格納するデータ ソースに、O 出力を FDCE の D 入力などの適切なデスティネーションに接続します。
- クロック イネーブル ピン WE は、適切なライト イネーブル ソースに接続します。
- 8 ビット バス A は、読み出し/書き込みのソースに接続します。
- 256 ビットの 16 進数で構成される INIT 属性で、RAM の初期値を指定できます。

指定しない場合は、初期値はすべて 0 になります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	256 ビット値	すべて 0	RAM の初期値を指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
--             single-port distributed LUT RAM
--             Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM256X1S_inst : RAM256X1S
generic map (
  INIT => X"0000000000000000000000000000000000000000000000000000000000000000"
)
port map (
  O => O, -- Read/Write port 1-bit output
  A => A, -- Read/Write port 8-bit address input
  D => D, -- RAM data input
  WCLK => WCLK, -- Write clock input
  WE => WE -- Write enable input
);

-- End of RAM256X1S_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
//             single-port distributed LUT RAM
//             Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM256X1S #(
  .INIT(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAM256X1S_inst (
  .O(O), // Readw/rite port 1-bit output
  .A(A), // Readw/rite port 8-bit address input
  .WE(WE), // Write enable input
  .WCLK(WCLK), // Write clock input
  .D(D) // RAM data input
);

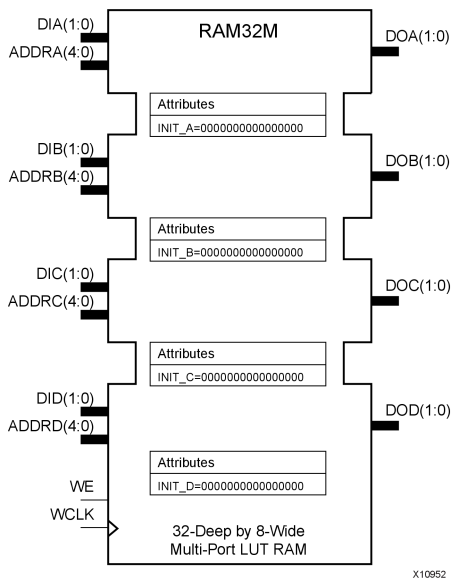
// End of RAM256X1S_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## RAM32M

**プリミティブ：32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)**



## 概要

このデザイン エLEMENTは、32 ワード X 8 ビットのマルチポート RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM™) を使用してインプリメントされるため、デバイスのブロック RAM リソースを使用しません。RAM32M コンポーネントは 1 つのスライスにインプリメントされ、8 ビット書き込みポート 1 つ、2 ビット読み出しポート 1 つ、および同じメモリからの 2 ビット読み出しポート 3 つから構成されています。これにより、RAM のバイト幅の書き込みと独立した 2 ビットの読み出しが可能です。DIA、DIB、DIC、および DID 入力をすべて同じデータ入力に接続すると、読み出し/書き込みポート 1 つ、独立した読み出しポート 3 つの 32x2 クワッドポートメモリになります。DID をグランドに接続した場合、DOD は使用されません。ADDRA、ADDRb、ADDRc を同じアドレスに接続すると、32x6 のシンプルデュアルポート RAM になります。ADDRd を ADDRA、ADDRb、ADDRc に接続すると、32x8 のシングルポート RAM になります。この RAM には、ほかにも可能なコンフィギュレーションがあります。

## ポートの説明

ポート名	方向	幅	機能
DOA	出力	2	アドレス バス ADDRA で指定された読み出しポートのデータ出力
DOB	出力	2	アドレス バス ADDRb で指定された読み出しポートのデータ出力
DOC	出力	2	アドレス バス ADDRc で指定された読み出しポートのデータ出力
DOD	出力	2	アドレス バス ADDRd で指定された読み出し/書き込みポートのデータ出力
DIA	入力	2	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRA で指定)
DIB	入力	2	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRb で指定)

ポート名	方向	幅	機能
DIC	入力	2	ADDRD で指定された書き込みデータ入力 (読み出し出力は ADDRRC で指定)
DID	入力	2	アドレス バス ADDRD で指定された書き込みデータ入力
ADDRA	入力	5	読み出しアドレス バス A
ADDRB	入力	5	読み出しアドレス バス B
ADDRC	入力	5	読み出しアドレス バス C
ADDRD	入力	5	8 ビットのデータ書き込みポート、2 ビットのデータ読み出しポートのアドレス バス D
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

このエレメントは、同期書き込みと非同期読み出し機能を備えた RAM を記述することにより、一部の合成ツールで推論できます。RAM の推論およびコード例の詳細は、合成ツールのマニュアルを参照してください。RAM32M のインスタンス化は、RAM ファンクションを暗示的に指定する必要がある場合、コンポーネントを手動でまたは相対的に配置する必要がある場合に実行することをお勧めします。同期読み出しが必要な場合は、RAM32M の出力を FDRSE (非同期セット/リセットが必要な場合は FDCPE) に接続してファンクションの出力タイミングを向上させることも可能ですが、RAM の正しい動作には不要です。

インバーターをこのコンポーネントのクロック入力に追加すると、クロックの立ち下がりエッジでデータを入力できます。このインバーターはブロック内に組み込まれ、クロックの立ち下がりエッジで RAM への書き込みを実行できます。

インスタンス化する場合、このコンポーネントは、次のように接続します。WCLK 入力をクロックソースに、DIA、DIB、DIC、DID 入力を格納するデータソースに、DOA、DOB、DOC、DOD 出力を FDCE の D 入力などの適切なデスティネーションに接続するか、使用しない場合は未接続のままにします。クロック イネーブルピン WE は、適切なライト イネーブルソースに接続します。5 ビット バス ADDRD は読み出し/書き込みアドレスに、5 ビット バス ADDRA、ADDRB、ADDRC は読み出しアドレスに接続する必要があります。オプションで INIT\_A、INIT\_B、INIT\_C、INIT\_D 属性を使用すると、各ポートの初期メモリ内容を 64 ビット (16 進数) で指定できます。RAM の INIT 値は、 $ADDRy[z] = INIT_y[2*z+1:2*z]$  で計算されます。たとえば、RAM の ADDRRC ポートが 00001 の場合、INIT\_C[3:2] 値がそのアドレスで最初の書き込みが行われる前の DOC ポートの初期値になります。指定しない場合は、初期値はすべて 0 になります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_A	16 進数	64 ビット値	すべて 0	A ポートの RAM の初期値を指定します。
INIT_B	16 進数	64 ビット値	すべて 0	B ポートの RAM の初期値を指定します。
INIT_C	16 進数	64 ビット値	すべて 0	C ポートの RAM の初期値を指定します。
INIT_D	16 進数	64 ビット値	すべて 0	D ポートの RAM の初期値を指定します。



## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM32M: 32-deep by 8-wide Multi Port LUT RAM
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM32M_inst : RAM32M
generic map (
    INIT_A => X"0000000000000000",    -- Initial contents of A port
    INIT_B => X"0000000000000000",    -- Initial contents of B port
    INIT_C => X"0000000000000000",    -- Initial contents of C port
    INIT_D => X"0000000000000000")    -- Initial contents of D port
port map (
    DOA => DOA, -- Read port A 2-bit output
    DOB => DOB, -- Read port B 2-bit output
    DOC => DOC, -- Read port C 2-bit output
    DOD => DOD, -- Read/Write port D 2-bit output
    ADDRA => ADDRA, -- Read port A 5-bit address input
    ADDR_B => ADDR_B, -- Read port B 5-bit address input
    ADDR_C => ADDR_C, -- Read port C 5-bit address input
    ADDR_D => ADDR_D, -- Read/Write port D 5-bit address input
    DIA => DIA, -- RAM 2-bit data write input addressed by ADDR_D,
                -- read addressed by ADDRA
    DIB => DIB, -- RAM 2-bit data write input addressed by ADDR_D,
                -- read addressed by ADDR_B
    DIC => DIC, -- RAM 2-bit data write input addressed by ADDR_D,
                -- read addressed by ADDR_C
    DID => DID, -- RAM 2-bit data write input addressed by ADDR_D,
                -- read addressed by ADDR_D
    WCLK => WCLK, -- Write clock input
    WE => WE      -- Write enable input
);
-- End of RAM32M_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAM32M: 32-deep by 8-wide Multi Port LUT RAM
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM32M # (
    .INIT_A(64'h0000000000000000), // Initial contents of A Port
    .INIT_B(64'h0000000000000000), // Initial contents of B Port
    .INIT_C(64'h0000000000000000), // Initial contents of C Port
    .INIT_D(64'h0000000000000000) // Initial contents of D Port
) RAM32M_inst (
    .DOA(DOA), // Read port A 2-bit output
    .DOB(DOB), // Read port B 2-bit output
    .DOC(DOC), // Read port C 2-bit output
    .DOD(DOD), // Readw/rite port D 2-bit output
    .ADDRA(ADDRA), // Read port A 5-bit address input
    .ADDRB(ADDRB), // Read port B 5-bit address input
    .ADDRC(ADDRC), // Read port C 5-bit address input
    .ADDRD(ADDRD), // Readw/rite port D 5-bit address input
    .DIA(DIA), // RAM 2-bit data write input addressed by ADDRd,
                // read addressed by ADDRd
    .DIB(DIB), // RAM 2-bit data write input addressed by ADDRd,
                // read addressed by ADDRb
    .DIC(DIC), // RAM 2-bit data write input addressed by ADDRd,
                // read addressed by ADDRc
    .DID(DID), // RAM 2-bit data write input addressed by ADDRd,
                // read addressed by ADDRd
    .WCLK(WCLK), // Write clock input
    .WE(WE) // Write enable input
);

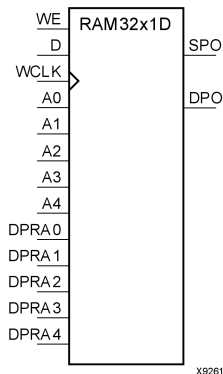
// End of RAM32M_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## RAM32X1D

### プリミティブ：32-Deep by 1-Wide Static Dual Port Synchronous RAM



### 概要

このデザイン エLEMENTは 32 ワード X 1 ビットのデュアル ポート SRAM で、同期書き込み機能を備えています。デバイスには、読み出しアドレス (DPRA4 ~ DPRA0) と書き込みアドレス (A4 ~ A0) の独立した 2 種類のアドレス ポートがあります。この 2 種類のアドレス ポートは完全に非同期です。読み出しアドレスによって出力ピン (DPO) に出力される値が指定され、書き込みアドレスによって書き込みを行う位置が指定されます。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High の場合、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 5 ビットの書き込みアドレスで選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。INIT 属性を使用すると、コンフィギュレーション中に RAM32X1D を初期化できます。モード選択を次の論理表に示します。

SPO 出力には、A4 ~ A0 で指定されたメモリ セルの値が出力されます。DPO 出力には、DPRA4 ~ DPRA0 で指定されたメモリ セルの値が出力されます。書き込み処理は、読み出しアドレス ポートのアドレスには影響されません。

### 論理表

入力			出力	
WE (モード)	WCLK	D	SPO	DPO
0 (読み出し)	X	X	data_a	data_d
1 (読み出し)	0	X	data_a	data_d
1 (読み出し)	1	X	data_a	data_d
1 (書き込み)	↑	D	D	data_d
1 (読み出し)	↓	X	data_a	data_d

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	ROM、RAM、レジスタ、LUT の初期値を指定します。

## VHDL 記述 (インスタンスエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM32X1D: 32 x 1 positive edge write, asynchronous read
--           dual-port distributed RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM32X1D_inst : RAM32X1D
generic map (
  INIT => X"00000000") -- Initial contents of RAM
port map (
  DPO => DPO,          -- Read-only 1-bit data output
  SPO => SPO,          -- R/W 1-bit data output
  A0 => A0,            -- R/W address[0] input bit
  A1 => A1,            -- R/W address[1] input bit
  A2 => A2,            -- R/W address[2] input bit
  A3 => A3,            -- R/W address[3] input bit
  A4 => A4,            -- R/W address[4] input bit
  D => D,              -- Write 1-bit data input
  DPRA0 => DPRA0,      -- Read-only address[0] input bit
  DPRA1 => DPRA1,      -- Read-only address[1] input bit
  DPRA2 => DPRA2,      -- Read-only address[2] input bit
  DPRA3 => DPRA3,      -- Read-only address[3] input bit
  DPRA4 => DPRA4,      -- Read-only address[4] input bit
  WCLK => WCLK,        -- Write clock input
  WE => WE             -- Write enable input
);

-- End of RAM32X1D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM32X1D: 32 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM32X1D #(
    .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1D_inst (
    .DPO(DPO),          // Read-only 1-bit data output
    .SPO(SPO),          // Rw/ 1-bit data output
    .A0(A0),            // Rw/ address[0] input bit
    .A1(A1),            // Rw/ address[1] input bit
    .A2(A2),            // Rw/ address[2] input bit
    .A3(A3),            // Rw/ address[3] input bit
    .A4(A4),            // Rw/ address[4] input bit
    .D(D),              // Write 1-bit data input
    .DPRA0(DPRA0),      // Read-only address[0] input bit
    .DPRA1(DPRA1),      // Read-only address[1] input bit
    .DPRA2(DPRA2),      // Read-only address[2] input bit
    .DPRA3(DPRA3),      // Read-only address[3] input bit
    .DPRA4(DPRA4),      // Read-only address[4] input bit
    .WCLK(WCLK),        // Write clock input
    .WE(WE)             // Write enable input
);

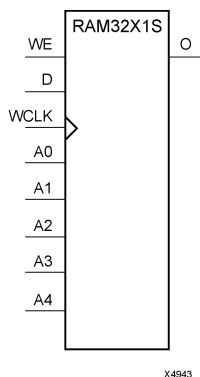
// End of RAM32X1D_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## RAM32X1S

プリミティブ：32-Deep by 1-Wide Static Synchronous RAM



### 概要

このデザイン エLEMENTは 32 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。INIT 属性を使用すると、コンフィギュレーション中に RAM32X1S を初期化できます。

### 論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D	D
1 (読み出し)	↓	X	データ

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	RAM の初期値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM32X1S_inst : RAM32X1S
generic map (
  INIT => X"00000000")
port map (
  O => O,      -- RAM output
  A0 => A0,     -- RAM address[0] input
  A1 => A1,     -- RAM address[1] input
  A2 => A2,     -- RAM address[2] input
  A3 => A3,     -- RAM address[3] input
  A4 => A4,     -- RAM address[4] input
  D => D,       -- RAM data input
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);

-- End of RAM32X1S_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM32X1S #(
  .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1S_inst (
  .O(O),              // RAM output
  .A0(A0),            // RAM address[0] input
  .A1(A1),            // RAM address[1] input
  .A2(A2),            // RAM address[2] input
  .A3(A3),            // RAM address[3] input
  .A4(A4),            // RAM address[4] input
  .D(D),              // RAM data input
  .WCLK(WCLK),        // Write clock input
  .WE(WE)             // Write enable input
);

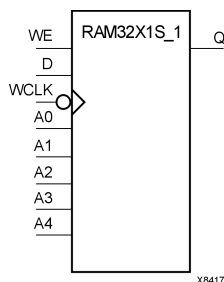
// End of RAM32X1S_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## RAM32X1S\_1

プリミティブ：32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



### 概要

このデザイン エLEMENTは 32 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が High から Low に切り替わる時に、データ入力 (D) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が High から Low に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ Low ですが、インバーターを使用してアクティブ High にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。INIT 属性を使用すると、コンフィギュレーション中に RAM32X1S\_1 を初期化できます。

### 論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↓	D	D
1 (読み出し)	↑	X	データ
データ = A4 ~ A0 で指定されたワード			

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可



## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	0	RAM、レジスタ、LUT の初期値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM32X1S_1_inst : RAM32X1S_1
generic map (
  INIT => X"00000000")
port map (
  O => O,      -- RAM output
  A0 => A0,     -- RAM address[0] input
  A1 => A1,     -- RAM address[1] input
  A2 => A2,     -- RAM address[2] input
  A3 => A3,     -- RAM address[3] input
  A4 => A4,     -- RAM address[4] input
  D => D,      -- RAM data input
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);

-- End of RAM32X1S_1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM32X1S_1 #(
  .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1S_1_inst (
  .O(O),              // RAM output
  .A0(A0),            // RAM address[0] input
  .A1(A1),            // RAM address[1] input
  .A2(A2),            // RAM address[2] input
  .A3(A3),            // RAM address[3] input
  .A4(A4),            // RAM address[4] input
  .D(D),              // RAM data input
  .WCLK(WCLK),        // Write clock input
  .WE(WE)             // Write enable input
);

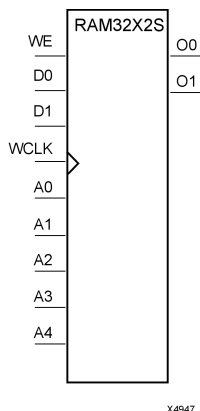
// End of RAM32X1S_1_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## RAM32X2S

プリミティブ：32-Deep by 2-Wide Static Synchronous RAM



### 概要

このデザイン エLEMENTは 32 ワード X 2 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、入力 (D1 ~ D0) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。出力ピン (O1 ~ O0) に出力される値は、アドレスピンで指定された RAM 内の位置に格納されている値です。

INIT\_00 および INIT\_01 属性を使用して RAM32X2S の初期値を指定できます。

### 論理表

入力			出力
WE (モード)	WCLK	D	O0 ~ O1
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D1 ~ D0	D1 ~ D0
1 (読み出し)	↓	X	データ
データ = A4 ~ A0 で指定されたワード			

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT_00	16 進数	32 ビット値	すべて 0	RAM のビット 0 の初期値を指定します。
INIT_01	16 進数	32 ビット値	すべて 0	RAM のビット 1 の初期値を指定します。

## VHDL 記述（インスタンス化）

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM32X2S_inst : RAM32X2S
generic map (
    INIT_00 => X"00000000", -- INIT for bit 0 of RAM
    INIT_01 => X"00000000") -- INIT for bit 1 of RAM
port map (
    O0 => O0,      -- RAM data[0] output
    O1 => O1,      -- RAM data[1] output
    A0 => A0,      -- RAM address[0] input
    A1 => A1,      -- RAM address[1] input
    A2 => A2,      -- RAM address[2] input
    A3 => A3,      -- RAM address[3] input
    A4 => A4,      -- RAM address[4] input
    D0 => D0,      -- RAM data[0] input
    D1 => D1,      -- RAM data[1] input
    WCLK => WCLK,  -- Write clock input
    WE => WE       -- Write enable input
);

-- End of RAM32X2S_inst instantiation
```

## Verilog 記述（インスタンス化）

```
// RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM32X2S #(
    .INIT_00(32'h00000000), // INIT for bit 0 of RAM
    .INIT_01(32'h00000000) // INIT for bit 1 of RAM
) RAM32X2S_inst (
    .O0(O0),      // RAM data[0] output
    .O1(O1),      // RAM data[1] output
    .A0(A0),      // RAM address[0] input
    .A1(A1),      // RAM address[1] input
    .A2(A2),      // RAM address[2] input
    .A3(A3),      // RAM address[3] input
    .A4(A4),      // RAM address[4] input
    .D0(D0),      // RAM data[0] input
    .D1(D1),      // RAM data[1] input
    .WCLK(WCLK),  // Write clock input
    .WE(WE)       // Write enable input
);

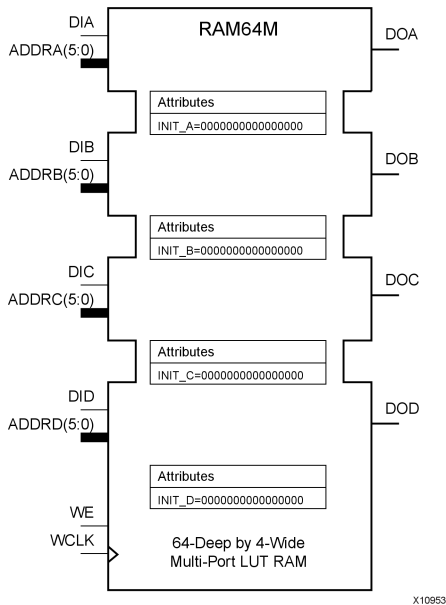
// End of RAM32X2S_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## RAM64M

**プリミティブ：64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)**



### 概要

このデザイン エLEMENTは、64 ワード X 4 ビットのマルチポート RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM™ と呼ばれる) を使用してインプリメントされるため、ブロック RAM リソースを使用しません。RAM64M コンポーネントは 1 つのスライスにインプリメントされます。4 ビット書き込みポート 1 つ、1 ビット読み出しのポート 1 つ、および同じメモリからの 1 ビット読み出しポート 3 つから構成されており、RAM の 4 ビット書き込みおよび個別ビット読み出しが可能です。DIA、DIB、DIC、および DID 入力をすべて同じデータ入力に接続すると、読み出し/書き込みポート 1 つ、独立した読み出しポート 3 つの 64x1 クワッド ポートメモリになります。DID をグラウンドに接続した場合、DOD は使用されません。ADDRA、ADDRb、ADDRc を同じアドレスに接続すると、64x3 のシングル デュアル ポート RAM になります。ADDRd を ADDRA、ADDRb、ADDRc に接続すると、64x4 のシングル ポート RAM になります。この RAM には、ほかにも可能なコンフィギュレーションがあります。

### ポートの説明

ポート名	方向	幅	機能
DOA	出力	1	アドレス バス ADDRA で指定される読み出しポートのデータ出力
DOB	出力	1	アドレス バス ADDRb で指定される読み出しポートのデータ出力
DOC	出力	1	アドレス バス ADDRc で指定される読み出しポートのデータ出力
DOD	出力	1	アドレス バス ADDRd で指定される読み出し/書き込みポートのデータ出力
DIA	入力	1	ADDRd で指定される書き込みデータ入力 (読み出し出力は ADDRA で指定)
DIB	入力	1	ADDRd で指定される書き込みデータ入力 (読み出し出力は ADDRb で指定)

ポート名	方向	幅	機能
DIC	入力	1	ADDRD で指定される書き込みデータ入力 (読み出し出力は ADDRRC で指定)
DID	入力	1	アドレス バス ADDRD で指定される書き込みデータ入力
ADDRA	入力	6	読み出しアドレス バス A
ADDRB	入力	6	読み出しアドレス バス B
ADDRC	入力	6	読み出しアドレス バス C
ADDRD	入力	6	4 ビットのデータ書き込みポート、1 ビットのデータ読み出しポートのアドレス バス D
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

このエレメントは、同期書き込みと非同期読み出し機能を備えた RAM を記述することにより、一部の合成ツールで推論できます。RAM の推論およびコード例の詳細は、合成ツールのマニュアルを参照してください。RAM64M のインスタンス化は、RAM ファンクションを暗示的に指定する必要がある場合、コンポーネントを手動でまたは相対的に配置する必要がある場合に実行することをお勧めします。同期読み出しが必要な場合は、RAM64M の出力を FDRSE (非同期セット/リセットが必要な場合は FDCPE) に接続してファンクションの出力タイミングを向上させることも可能ですが、RAM の正しい動作には不要です。インバーターをこのコンポーネントのクロック入力に追加すると、クロックの立ち下がりエッジでデータを入力できます。このインバーターはブロック内に組み込まれ、クロックの立ち下がりエッジで RAM への書き込みを実行できます。

インスタンス化する場合、このコンポーネントは、次のように接続します。

- WCLK 入力をクロック ソースに、DIA、DIB、DIC、DID 入力を格納するデータ ソースに、DOA、DOB、DOC、DOD 出力を FDCE の D 入力などの適切なデスティネーションに接続するか、使用しない場合は未接続のままにします。
- クロック イネーブル ピン (WE) は、適切なライト イネーブル ソースに接続します。
- 5 ビット バス ADDRD は読み出し/書き込みアドレスに、5 ビット バス ADDRA、ADDRB、ADDRC は読み出しアドレスに接続する必要があります。
- オプションで INIT\_A、INIT\_B、INIT\_C、INIT\_D 属性を使用すると、各ポートの初期メモリ内容を 64 ビット (16 進数) で指定できます。RAM の INIT 値は、 $\text{ADDRy}[z] = \text{INIT}_y[z]$  で計算されます。

たとえば、RAM の ADDRRC ポートが 00001 の場合、INIT\_C[1] 値がそのアドレスで最初の書き込みが行われる前の DOC ポートの初期値になります。指定しない場合は、初期値はすべて 0 になります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_A	16 進数	64 ビット値	すべて 0	A ポートの RAM の初期値を指定します。
INIT_B	16 進数	64 ビット値	すべて 0	B ポートの RAM の初期値を指定します。
INIT_C	16 進数	64 ビット値	すべて 0	C ポートの RAM の初期値を指定します。
INIT_D	16 進数	64 ビット値	すべて 0	D ポートの RAM の初期値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM64M: 64-deep by 4-wide Multi Port LUT RAM
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM64M_inst : RAM64M
generic map (
  INIT_A => X"0000000000000000", -- Initial contents of A port
  INIT_B => X"0000000000000000", -- Initial contents of B port
  INIT_C => X"0000000000000000", -- Initial contents of C port
  INIT_D => X"0000000000000000"  -- Initial contents of D port
) port map (
  DOA => DOA, -- Read port A 1-bit output
  DOB => DOB, -- Read port B 1-bit output
  DOC => DOC, -- Read port C 1-bit output
  DOD => DOD, -- Read/Write port D 1-bit output
  ADDRA => ADDRA, -- Read port A 6-bit address input
  ADDR_B => ADDR_B, -- Read port B 6-bit address input
  ADDR_C => ADDR_C, -- Read port C 6-bit address input
  ADDR_D => ADDR_D, -- Read/Write port D 6-bit address input
  DIA => DIA, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDRA
  DIB => DIB, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_B
  DIC => DIC, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_C
  DID => DID, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_D
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);
-- End of RAM64M_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAM64M: 64-deep by 4-wide Multi Port LUT RAM
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM64M #(
  .INIT_A(64'h0000000000000000), // Initial contents of A Port
  .INIT_B(64'h0000000000000000), // Initial contents of B Port
  .INIT_C(64'h0000000000000000), // Initial contents of C Port
  .INIT_D(64'h0000000000000000) // Initial contents of D Port
) RAM64M_inst (
  .DOA(DOA), // Read port A 1-bit output
  .DOB(DOB), // Read port B 1-bit output
  .DOC(DOC), // Read port C 1-bit output
  .DOD(DOD), // Read/Write port D 1-bit output
```

```
.DIA(DIA),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRA  
.DIB(DIB),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRb  
.DIC(DIC),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRc  
.DID(DID),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRd  
.ADDRA(ADDRA), // Read port A 6-bit address input  
.ADDRB(ADDRB), // Read port B 6-bit address input  
.ADDRC(ADDRC), // Read port C 6-bit address input  
.ADDRD(ADDRD), // Readw/rite port D 6-bit address input  
.WE(WE),       // Write enable input  
.WCLK(WCLK)    // Write clock input  
);  
  
// End of RAM64M_inst instantiation
```

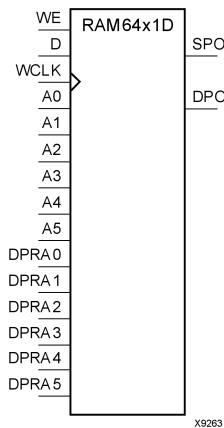
## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## RAM64X1D

プリミティブ：64-Deep by 1-Wide Dual Port Static Synchronous RAM



### 概要

このデザイン エLEMENTは 64 ワード X 1 ビットのデュアル ポート SRAM で、同期書き込み機能を備えています。デバイスには、読み出しアドレス (DPRA5 ～ DPRA0) と書き込みアドレス (A5 ～ A0) の独立した 2 種類のアドレスポートがあります。この 2 種類のアドレスポートは完全に非同期です。読み出しアドレスによって出力ピン (DPO) に出力される値が指定され、書き込みアドレスによって書き込みを行う位置が指定されます。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。

WE が High になると、WCLK が Low から High に切り替わる時に、データ入力 (D) の値が 6 ビットの書き込みアドレス (A0 ～ A5) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

SPO 出力には、A5 ～ A0 で指定されたメモリ セルの値が出力されます。DPO 出力には、DPRA5 ～ DPRA0 で指定されたメモリ セルの値が出力されます。

**注記：** 書き込み処理は、読み出しアドレスポートのアドレスには影響されません。

### 論理表

入力			出力	
WE (モード)	WCLK	D	SPO	DPO
0 (読み出し)	X	X	data_a	data_d
1 (読み出し)	0	X	data_a	data_d
1 (読み出し)	1	X	data_a	data_d
1 (書き込み)	↑	D	D	data_d
1 (読み出し)	↓	X	data_a	data_d
data_a = A5 ～ A0 で指定されたワード				
data_d = DPRA5 ～ DPRA0 で指定されたワード				

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	RAM の初期値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM64X1D: 64 x 1 negative edge write, asynchronous read
--           dual-port distributed RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM64X1D_1_inst : RAM64X1D_1
generic map (
  INIT => X"0000000000000000") -- Initial contents of RAM
port map (
  DPO => DPO,      -- Read-only 1-bit data output
  SPO => SPO,      -- R/W 1-bit data output
  A0 => A0,        -- R/W address[0] input bit
  A1 => A1,        -- R/W address[1] input bit
  A2 => A2,        -- R/W address[2] input bit
  A3 => A3,        -- R/W address[3] input bit
  A4 => A4,        -- R/W address[4] input bit
  A5 => A5,        -- R/W address[5] input bit
  D => D,          -- Write 1-bit data input
  DPRA0 => DPRA0,  -- Read-only address[0] input bit
  DPRA1 => DPRA1,  -- Read-only address[1] input bit
  DPRA2 => DPRA2,  -- Read-only address[2] input bit
  DPRA3 => DPRA3,  -- Read-only address[3] input bit
  DPRA4 => DPRA4,  -- Read-only address[4] input bit
  DPRA5 => DPRA5,  -- Read-only address[5] input bit
  WCLK => WCLK,    -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM64X1D_1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAM64X1D: 64 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM64X1D #(
    .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1D_inst (
    .DPO(DPO),           // Read-only 1-bit data output
    .SPO(SPO),           // Rw/ 1-bit data output
    .A0(A0),             // Rw/ address[0] input bit
    .A1(A1),             // Rw/ address[1] input bit
    .A2(A2),             // Rw/ address[2] input bit
    .A3(A3),             // Rw/ address[3] input bit
    .A4(A4),             // Rw/ address[4] input bit
    .A5(A5),             // Rw/ address[5] input bit
    .D(D),               // Write 1-bit data input
    .DPRA0(DPRA0),       // Read-only address[0] input bit
    .DPRA1(DPRA1),       // Read-only address[1] input bit
    .DPRA2(DPRA2),       // Read-only address[2] input bit
    .DPRA3(DPRA3),       // Read-only address[3] input bit
    .DPRA4(DPRA4),       // Read-only address[4] input bit
    .DPRA5(DPRA5),       // Read-only address[5] input bit
    .WCLK(WCLK),         // Write clock input
    .WE(WE)              // Write enable input
);

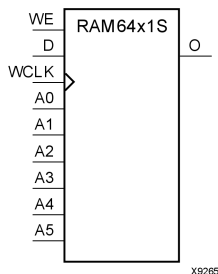
// End of RAM64X1D_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## RAM64X1S

**プリミティブ：64-Deep by 1-Wide Static Synchronous RAM**



### 概要

このデザイン エレメントは 64 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 6 ビットのアドレス (A5 ～ A0) で選択されたワードに書き込まれます。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。

INIT 属性を使用すると、コンフィギュレーション中にこのエレメントを初期化できます。

### 論理表

モード選択を次の論理表に示します。

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D	D
1 (読み出し)	↓	X	データ
データ = A5 ～ A0 で指定されたワード			

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ROM、RAM、レジスタ、LUT の初期値を指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM64X1S_inst : RAM64X1S
generic map (
  INIT => X"0000000000000000")
port map (
  O => O,          -- 1-bit data output
  A0 => A0,         -- Address[0] input bit
  A1 => A1,         -- Address[1] input bit
  A2 => A2,         -- Address[2] input bit
  A3 => A3,         -- Address[3] input bit
  A4 => A4,         -- Address[4] input bit
  A5 => A5,         -- Address[5] input bit
  D => D,          -- 1-bit data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM64X1S_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM64X1S #(
  .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_inst (
  .O(O),           // 1-bit data output
  .A0(A0),         // Address[0] input bit
  .A1(A1),         // Address[1] input bit
  .A2(A2),         // Address[2] input bit
  .A3(A3),         // Address[3] input bit
  .A4(A4),         // Address[4] input bit
  .A5(A5),         // Address[5] input bit
  .D(D),           // 1-bit data input
  .WCLK(WCLK),     // Write clock input
  .WE(WE)          // Write enable input
);

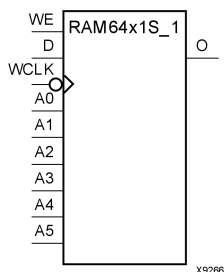
// End of RAM64X1S_inst instantiation
```

## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)

## RAM64X1S\_1

プリミティブ：64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



### 概要

このデザイン エLEMENTは 64 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が High から Low に切り替わるときに、データ入力 (D) の値が 6 ビットのアドレス (A5 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が High から Low に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ Low ですが、インバーターを使用してアクティブ High にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。

INIT 属性を使用すると、コンフィギュレーション中にこのELEMENTを初期化できます。

### 論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↓	D	D
1 (読み出し)	↑	X	データ
データ = A5 ~ A0 で指定されたワード			

### デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ROM、RAM、レジスタ、LUT の初期値を指定します。

## VHDL 記述 (インスタンスレーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAM64X1S_1_inst : RAM64X1S_1
generic map (
  INIT => X"0000000000000000")
port map (
  O => O,          -- 1-bit data output
  A0 => A0,         -- Address[0] input bit
  A1 => A1,         -- Address[1] input bit
  A2 => A2,         -- Address[2] input bit
  A3 => A3,         -- Address[3] input bit
  A4 => A4,         -- Address[4] input bit
  A5 => A5,         -- Address[5] input bit
  D => D,          -- 1-bit data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM64X1S_1_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

RAM64X1S_1 #(
  .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_1_inst (
  .O(O),          // 1-bit data output
  .A0(A0),        // Address[0] input bit
  .A1(A1),        // Address[1] input bit
  .A2(A2),        // Address[2] input bit
  .A3(A3),        // Address[3] input bit
  .A4(A4),        // Address[4] input bit
  .A5(A5),        // Address[5] input bit
  .D(D),          // 1-bit data input
  .WCLK(WCLK),    // Write clock input
  .WE(WE)         // Write enable input
);

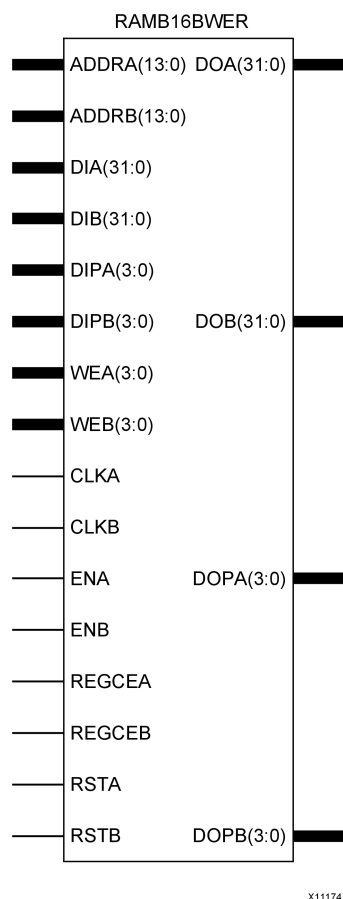
// End of RAM64X1S_1_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## RAMB16BWER

**プリミティブ：16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers**



### 概要

このエレメントには複数のブロックRAMメモリが含まれており、汎用 16kb データ + 2kb パリティ RAM/ROM メモリとしてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。このコンポーネントは、1 ビット X 16K ワード ~ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。読み出しと書き込みは、コンポーネントに供給されるクロックに同期して実行されますが、ポート A とポート B の動作は完全に独立しており、お互いに非同期で、同じメモリ アレイにアクセスします。広いデータ幅でコンフィギュレーションすると、バイト イネーブル書き込み操作が可能になります。この RAM には、コンフィギュレーション可能な出力レジスタもあり、読み出し操作中 1 クロック サイクルのレイテンシが発生するときに RAM の clock-to-out タイムを向上させることができます。

### ポートの説明

次の表に、ポート A またはポート B の DATA\_WIDTH の値と、それに必要な入力接続と出力接続を示します。



DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続
1	DI[0]	ADDR[13:0]	WE[3:0] をシングル ユーザー WE 信号に接続
2	DI[1:0]	ADDR[13:1]	WE[3:0] をシングル ユーザー WE 信号に接続
4	DI[3:0]	ADDR[13:2]	WE[3:0] をシングル ユーザー WE 信号に接続
9	DI[7:0]、DIP[0]	ADDR[13:3]	WE[3:0] をシングル ユーザー WE 信号に接続
18	DI[15:0]、DIP[1:0]	ADDR[13:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続
36	DI[31:0]、DIP[3:0]	ADDR[13:5]	各 WE[3:0] 信号を関連したバイト ライト イネーブルに接続

出力レジスタが不要な場合は、代わりに RAMB16\_Sm\_Sn および RAMB16BWER\_Sm\_Sn エLEMENTをインスタンスエートできます。これらのコンポーネントのいずれかが使用されていれば、ツールで自動的に、適切にコンフィギュレーションされた RAMB16BWE エLEMENTに変換されます。

ポート名	方向	幅	機能
ADDRA[13:0]	入力	14	ポート A のアドレス入力バス。MSB は常に ADDRA[13] ですが、LSB は DATA_WIDTH_A の設定によって決まります。
ADDRB[13:0]	入力	14	ポート B のアドレス入力バス。MSB は常に ADDR[13] ですが、LSB は DATA_WIDTH_B の設定によって決まります。
CLKA	入力	1	ポート A のクロック入力
CLKB	入力	1	ポート B のクロック入力
DIA[31:0]	入力	32	ポート A のデータ入力バス
DIB[31:0]	入力	32	ポート B のデータ入力バス
DIPA[3:0]	入力	4	ポート A のパリティ入力バス
DIPB[3:0]	入力	4	ポート B のパリティ入力バス
DOA[31:0]	出力	32	ポート A のデータ出力バス
DOB[31:0]	出力	32	ポート B のデータ出力バス
DOPA[3:0]	出力	4	ポート A のパリティ出力バス
DOPB[3:0]	出力	4	ポート B のパリティ出力バス
ENA	入力	1	ポート A のイネーブル
ENB	入力	1	ポート B のイネーブル
REGCEA	入力	1	出力レジスタ クロック イネーブル
REGCEB	入力	1	出力レジスタ クロック イネーブル
RSTA	入力	1	ポート A の出力レジスタのセット/リセット。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。
RSTB	入力	1	ポート B の出力レジスタのセット/リセット。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。
WEA[3:0]	入力	4	ポート A のバイト幅ライト イネーブル
WEB[3:0]	入力	4	ポート B のバイト幅ライト イネーブル

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	可

必要な入力はすべて適切な信号に接続してください。CLKA/CLKB クロック信号をアクティブ クロックに、RSTA/RSTB リセット信号を論理 0 または適切なリセット信号に接続する必要があります。ENA/ENB は、論理 1 または適切な RAM ポート イネーブル信号のいずれかに接続します。REGCEA および REGCEB は、対応する DOA\_REG または DOB\_REG 属性が 1 に設定されている場合、適切な出力レジスタのクロック イネーブルまたは論理 1 に接続する必要があります。DOA\_REG が 0 に設定されている場合は、REGCEA および REGCEB を論理 0 に設定する必要があります。

これらの信号に必要な接続は DATA\_WIDTH の設定により変わるため、上記のポートの表で、必要なデータ入力、データ出力、ライト イネーブル、アドレスの接続情報を確認してください。ほかの出力信号はすべて未接続のままでかまいません。使用されていない入力信号は論理 0 に接続してください。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DATA_WIDTH_A	整数	0、1、2、4、9、18、36	0	ポート A のコンフィギュレーション可能なデータ幅を指定します。ポート B の幅と同じにする必要はありません。
DATA_WIDTH_B	整数	0、1、2、4、9、18、36	0	ポート B のコンフィギュレーション可能なデータ幅を指定します。ポート A の幅と同じにする必要はありません。
DOA_REG	整数	0、1	0	ポート A の出力レジスタを使用する場合は 1 に設定します。
DOB_REG	整数	0、1	0	ポート B の出力レジスタを使用する場合は 1 に設定します。
EN_RSTRAM_A	文字列	"TRUE"、"FALSE"	"TRUE"	"FALSE" に設定するとポート A の RST 機能がディスエーブルになり、"TRUE" に設定するとイネーブルになります。
EN_RSTRAM_B	文字列	"TRUE"、"FALSE"	"TRUE"	"FALSE" に設定するとポート B の RST 機能がディスエーブルになり、"TRUE" に設定するとイネーブルになります。
INIT_A	16 進数	36'h000000000 ~ 36'hffffff	すべて 0	コンフィギュレーション後のポート A の出力の初期値を指定します。
INIT_B	16 進数	36'h000000000 ~ 36'hffffff	すべて 0	コンフィギュレーション後のポート B の出力の初期値を指定します。
INIT_FILE	文字列	ファイルの名前と場所	なし	初期 RAM の内容を指定するファイル名
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべて 0	16kb のデータメモリ アレイの初期内容を指定します。
INITP_01 ~ INITP_07	16 進数	256 ビット値	すべて 0	2kb のパリティメモリ アレイの初期内容を指定します。
RST_PRIORITY_A	文字列	"CE"、"SR"	"CE"	DOA_REG=0 の場合はポート A の RAM の EN と RST ピンの間の優先順位を決定し、DOA_REG=1 (オプションの出力レジスタを使用) の場合は REGCE と RST の間の優先順位を決定します。

属性	データ型	値	デフォルト	説明
RST_PRIORITY_B	文字列	"CE"、"SR"	"CE"	DOB.REG=0 の場合はポート B の RAM EN と RST ピンの間の優先順位を決定し、DOB.REG=1 (オプションの出力レジスタを使用) の場合は REGCE と RST の間の優先順位を決定します。
RSTTYPE	文字列	"SYNC"、"ASYNC"	"SYNC"	RAM の出力のリセット機能を同期または非同期に設定します。タイミングの向上と回路の安定性の点から、非同期リセットが必要でない限り "SYNC" に設定してください。
SIM_COLLISION_CHECK	文字列	"ALL"、 "GENERATE_X_ONLY"、 "WARNING_ONLY"、 "NONE"	"ALL"	<p>メモリの競合が発生した場合のシミュレーションの動作を指定します。</p> <ul style="list-style-type: none"> <li>・ "ALL": 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "WARNING_ONLY": 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。</li> <li>・ "GENERATE_X_ONLY": 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "NONE": 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。</li> </ul> <p>注記: "ALL" 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。</p>
SRVAL_A	16 進数	36'h000000000 ~ 36'hffffff	すべて 0	リセット信号 (RSTA) がアサートされたときのポート A の出力値を指定します。
SRVAL_B	16 進数	36'h000000000 ~ 36'hffffff	すべて 0	リセット信号 (RSTB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> <li>・ "WRITE_FIRST": 書き込まれた値が出力ポートに出力されます。</li> <li>・ "READ_FIRST": 書き込み前にそのメモリロケーションに格納されていた値が出力ポートに出力されます。</li> <li>・ "NO_CHANGE": 出力ポートの以前の値が保持されます。</li> </ul>
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> <li>・ "WRITE_FIRST": 書き込まれた値が出力ポートに出力されます。</li> </ul>

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> <li>・ "READ_FIRST": 書き込み前にそのメモリロケーションに格納されていた値が出力ポートに出力されます。</li> <li>・ "NO_CHANGE": 出力ポートの以前の値が保持されます。</li> </ul>

## VHDL 記述 (インスタンスエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16BWER: 16k-bit Data and 2k-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAMB16BWER_inst : RAMB16BWER
generic map (
  -- DATA_WIDTH_A/DATA_WIDTH_B: 0, 1, 2, 4, 9, 18, or 36
  DATA_WIDTH_A => 0,
  DATA_WIDTH_B => 0,
  -- DOA_REG/DOB_REG: Optional output register (0 or 1)
  DOA_REG => 0,
  DOB_REG => 0,
  -- EN_RSTRAM_A/EN_RSTRAM_B: Enable/disable RST
  EN_RSTRAM_A => TRUE,
  EN_RSTRAM_B => TRUE,
  -- INITP_00 to INITP_07: Initial memory contents.
  INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  -- INIT_00 to INIT_3F: Initial memory contents.
  INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",

```

```
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- INIT_A/INIT_B: Initial values on output port
INIT_A => X"0000000000",
INIT_B => X"0000000000",
-- INIT_FILE: Optional file used to specify initial RAM contents
INIT_FILE => "NONE",
-- RSTTYPE: "SYNC" or "ASYN"
RSTTYPE => "SYNC",
-- RST_PRIORITY_A/RST_PRIORITY_B: "CE" or "SR"
RST_PRIORITY_A => "CE",
RST_PRIORITY_B => "CE",
-- SIM_COLLISION_CHECK: Collision check enable "ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"
SIM_COLLISION_CHECK => "ALL",
-- SIM_DEVICE: Must be set to "SPARTAN6" for proper simulation behavior
SIM_DEVICE => "SPARTAN3ADSP",
-- SRVAL_A/SRVAL_B: Set/Reset value for RAM output
SRVAL_A => X"0000000000",
SRVAL_B => X"0000000000",
-- WRITE_MODE_A/WRITE_MODE_B: "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
WRITE_MODE_A => "WRITE_FIRST",
WRITE_MODE_B => "WRITE_FIRST"
)
port map (
-- Port A Data: 32-bit (each) output: Port A data
DOA => DOA, -- 32-bit output: A port data output
DOPA => DOPA, -- 4-bit output: A port parity output
-- Port B Data: 32-bit (each) output: Port B data
DOB => DOB, -- 32-bit output: B port data output
DOPB => DOPB, -- 4-bit output: B port parity output
-- Port A Address/Control Signals: 14-bit (each) input: Port A address and control signals
ADDRA => ADDRA, -- 14-bit input: A port address input
CLKA => CLKA, -- 1-bit input: A port clock input
ENA => ENA, -- 1-bit input: A port enable input
REGCEA => REGCEA, -- 1-bit input: A port register clock enable input
RSTA => RSTA, -- 1-bit input: A port register set/reset input
WEA => WEA, -- 4-bit input: Port A byte-wide write enable input
```

## Verilog 記述 (インスタンスーション)

[illegible]



```
.DIPA(DIPA),      // 4-bit input: A port parity input
// Port B Address/Control Signals: 14-bit (each) input: Port B address and control signals
.ADDRB(ADDRB),    // 14-bit input: B port address input
.CLKB(CLKB),      // 1-bit input: B port clock input
.ENB(ENB),        // 1-bit input: B port enable input
.REGCEB(REGCEB),  // 1-bit input: B port register clock enable input
.RSTB(RSTB),      // 1-bit input: B port register set/reset input
.WEB(WEB),        // 4-bit input: Port B byte-wide write enable input
// Port B Data: 32-bit (each) input: Port B data
.DIB(DIB),        // 32-bit input: B port data input
.DIPB(DIPB)       // 4-bit input: B port parity input
);

// End of RAMB16BWER_inst instantiation
```

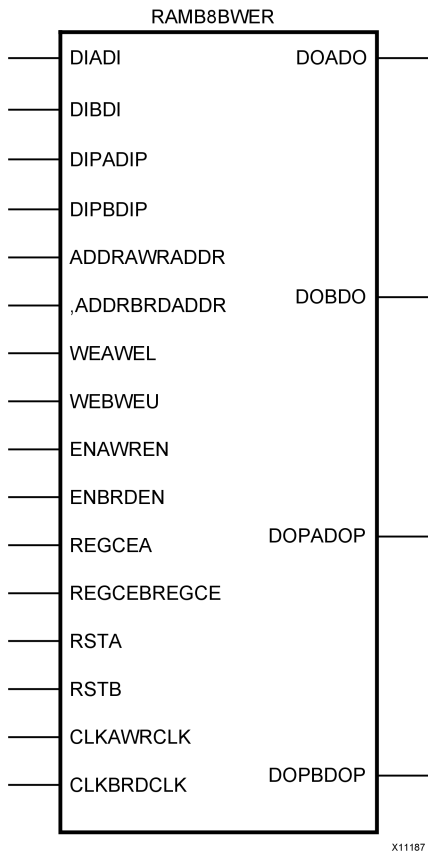
## 詳細情報

- ・ [『Spartan-6 FPGA ブロック RAM ユーザー ガイド』\(UG383\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## RAMB8BWER

**プリミティブ：8K-bit Data and 1K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers**



### 概要

Spartan®-6 デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。RAMB18E1 を使用すると、8KB データと 1KB パリティのコンフィギュレーションでブロック RAM にアクセスできます。このエレメントは、1 ビット X 8K ワード ~ 18 ビット X 512 ワードの完全なデュアルポート RAM としてコンフィギュレーションできます。また 36 ビット幅 X 246 ワードのシングルデュアルポート RAM にコンフィギュレーションすることもできます。読み出しと書き込みは、コンポーネントに供給されるクロックに同期して実行されますが、読み出しポートと書き込みポートは完全に独立しており、お互いに非同期で、同じメモリ アレイにアクセスします。広いデータ幅でコンフィギュレーションすると、バイト イネーブル書き込み操作が可能になります。この RAM には、コンフィギュレーション可能な出力レジスタもあり、読み出し操作中 1 クロック サイクルのレイテンシが発生するときに RAM の clock-to-out タイムを向上させることができます。

## ポートの説明

ポート名	方向	幅	機能
ADDRAWRADDR[12:0]	入力	13	RAM_MODE="TDP" の場合のポート A のアドレス入力バス。MSB は常に ADDRAWRADDR[12] ですが、LSB は DATA_WIDTH_A の設定によって決まります。RAM_MODE="SDP" の場合は書き込みアドレス入力バスです。
ADDRBRDADDR[12:0]	入力	13	RAM_MODE="TDP" の場合のポート B のアドレス入力バス。MSB は常に ADDRBRWADDR[12] ですが、LSB は DATA_WIDTH_B の設定によって決まります。RAM_MODE="SDP" の場合は書き込みアドレス入力バスです。
CLKAWRCLK	入力	1	ポート B クロック入力/書き込みクロック入力
CLKBRDCLK	入力	1	ポート B クロック入力/読み出しクロック入力
DIADI[15:0]	入力	16	RAM_MODE="TDP" の場合はポート A のデータ入力バスで、RAM_MODE="SDP" の場合は WRADDR で指定されたデータ入力バスです。SDP モードでは、DIADI は論理 DI[15:0] です。
DIBDI[15:0]	入力	16	RAM_MODE="TDP" の場合はポート B のデータ入力バスで、RAM_MODE="SDP" の場合は WRADDR で指定されたデータ入力バスです。SDP モードでは、DIBDI は論理 DI[31:16] です。
DIPADIP[1:0]	入力	2	RAM_MODE="TDP" の場合はポート A のパリティデータ入力バスで、RAM_MODE="SDP" の場合は WRADDR で指定されたデータパリティ入力バスです。SDP モードでは、DIPADIP は論理 DIP[1:0] です。
DIPBDIP[1:0]	入力	2	RAM_MODE="TDP" の場合はポート B のデータパリティ入力バスで、RAM_MODE="SDP" の場合は WRADDR で指定されたデータパリティ入力バスです。SDP モードでは、DIPBDIP は論理 DIP[3:2] です。
DOADO[15:0]	出力	16	ポート A データ出力バス/RDADDR でアドレス指定されるデータ出力バス。RAM_MODE="SDP" の場合、DOADO は論理 DO[15:0] です。
DOBDO[15:0]	出力	16	ポート B データ出力バス/RDADDR でアドレス指定されるデータ出力バス。RAM_MODE="SDP" の場合、DOBDO は論理 DO[31:16] です。
DOPADOP[1:0]	出力	2	ポート A パリティデータ出力バス/RDADDR でアドレス指定されるパリティデータ出力バス。RAM_MODE="SDP" の場合、DOPADOP は論理 DOP[1:0] です。
DOPBDOP[1:0]	出力	2	ポート B パリティデータ出力バス/RDADDR でアドレス指定されるパリティデータ出力バス。RAM_MODE="SDP" の場合、DOPBDOP は論理 DOP[3:2] です。
ENAWREN	入力	1	ポート A の RAM イネーブル/ライト イネーブル
ENBRDEN	入力	1	ポート B の RAM イネーブル/リード イネーブル
REGCEA	入力	1	ポート A の出力レジスタのクロック イネーブル入力 (DOA.REG=1 の場合にのみ有効)。RAM_MODE="SDP" の場合は使用されません。
REGCEBREGCE	入力	1	ポート B の出力レジスタのクロック イネーブル入力 (DOB.REG=1 の場合にのみ有効)。RAM_MODE="SDP" の場合は、出力レジスタのクロック イネーブル入力です。

ポート名	方向	幅	機能
RSTA	入力	1	ポート A を SRVAL_A で指定される値にセット/リセットします。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。出力レジスタ (DOA_REG=1) および出力ラッチの出力値に影響します。RAM_MODE="SDP" の場合は使用されません。
RSTBRST	入力	1	ポート B を SRVAL_B で指定される値にセット/リセットします。このリセットは、RSTTYPE 属性を使用して同期または非同期にコンフィギュレーションできます。出力レジスタ (DOB_REG=1) および出力ラッチの出力値に影響します。RAM_MODE="SDP" の場合は、リセット入力になります。
WEAWEL	入力	2	TDP モードではポート A のバイト幅ライト イネーブル、SDP モードでは論理 WE[1:0] です。
WEBWEU	入力	2	TDP モードではポート B のバイト幅ライト イネーブル、SDP モードでは論理 WE[3:2] です。

## デザインの入力方法

インスタンスエーション	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DATA_WIDTH_A	整数	0、1、2、4、9、18、36	0	ポート A のコンフィギュレーション可能なデータ幅を指定します。ポート B の幅と同じにする必要はありません。幅 36 は、SDP モードでのみ有効です。
DATA_WIDTH_B	整数	0、1、2、4、9、18、36	0	ポート B のコンフィギュレーション可能なデータ幅を指定します。ポート A の幅と同じにする必要はありません。幅 36 は、SDP モードでのみ有効です。
DOA_REG	整数	0、1	0	ポート A の出力レジスタを使用する場合は 1 に設定します。TDP モードではポート A に、SDP モードでは下位 18 ビット (パリティビットを含む) に適用されます。
DOB_REG	整数	0、1	0	ポート B の出力レジスタを使用する場合は 1 に設定します。TDP モードではポート B に、SDP モードでは上位 18 ビット (パリティビットを含む) に適用されます。
EN_RSTRAM_A	文字列	"TRUE"、"FALSE"	"TRUE"	"FALSE" に設定するとポート A の RST 機能がディスエーブルになり、"TRUE" に設定するとイネーブルになります。
EN_RSTRAM_B	文字列	"TRUE"、"FALSE"	"TRUE"	"FALSE" に設定するとポート B の RST 機能がディスエーブルになり、"TRUE" に設定するとイネーブルになります。

属性	データ型	値	デフォルト	説明
INIT_A	16 進数	18'h00000 ~ 18'h3fff	すべて 0	コンフィギュレーション後のポート A の出力の初期値を指定します。TDP モードではポート A に、SDP モードでは下位 18 ビット (パリティビットを含む) に適用されます。
INIT_B	16 進数	18'h00000 ~ 18'h3fff	すべて 0	コンフィギュレーション後のポート B の出力の初期値を指定します。TDP モードではポート B に、SDP モードでは上位 18 ビット (パリティビットを含む) に適用されます。
INIT_FILE	文字列	ファイルの名前と場所	なし	RAM の初期内容を記述するファイルの名前を指定します。
INIT_00 ~ INIT_1F	16 進数	256 ビット値	すべて 0	8KB のデータ メモリ アレイの初期値を指定します。
INITP_01 ~ INITP_03	16 進数	256 ビット値	すべて 0	1KB のパリティ データ メモリ アレイの初期値を指定します。
RAM_MODE	文字列	"TDP"、"SDP"	"TDP"	"SDP" に設定するとこのエレメントがシングル デュアル ポート RAM (1 つのポートは書き込みのみ、もう 1 つのポートは読み出しのみ) としてコンフィギュレーションされ、"TDP" に設定すると完全なデュアル ポート RAM (一方または両方のポートで読み出し/書き込み可能) としてコンフィギュレーションされます。
RST_PRIORITY_A	文字列	"CE"、"SR"	"CE"	DOA_REG=0 の場合はポート A の RAM EN と RST ピンの間の優先順位を決定し、DOA_REG=1 (オプションの出力レジスタを使用) の場合は REGCE と RST の間の優先順位を決定します。
RST_PRIORITY_B	文字列	"CE"、"SR"	"CE"	DOB_REG=0 の場合はポート B の RAM EN と RST ピンの間の優先順位を決定し、DOB_REG=1 (オプションの出力レジスタを使用) の場合は REGCE と RST の間の優先順位を決定します。
RSTTYPE	文字列	"SYNC"、"ASYNC"	"SYNC"	RAM の出力のリセット機能を同期または非同期に設定します。タイミングの向上と回路の安定性の点から、非同期リセットが必要でない限り "SYNC" に設定してください。
SIM_COLLISION_CHECK	文字列	"ALL"、 "GENERATE_X_ONLY"、 "WARNING_ONLY"、 "NONE"	"ALL"	メモリの競合が発生した場合のシミュレーションの動作を指定します。 <ul style="list-style-type: none"> <li>"ALL" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。</li> <li>"WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。</li> <li>"GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。</li> <li>"NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。</li> </ul>

属性	データ型	値	デフォルト	説明
				<p><b>注記：</b>“ALL” 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。</p>
SRVAL_A	16 進数	18'h00000 ~ 18'h3fff	すべて 0	リセット信号 (RSTA) がアサートされたときのポート A の出力値を指定します。TDP モードではポート A に適用されます。SDP モードでは、ポート幅が 18 ビット以下の場合は SRVAL_A のみを使用し、ポート幅が 18 ビットより広い場合は SRVAL_A は下位 18 ビット (パリティビットを含む) に適用されます。
SRVAL_B	16 進数	18'h00000 ~ 18'h3fff	すべて 0	リセット信号 (RSTB) がアサートされたときのポート B の出力値を指定します。TDP モードではポート B に適用されます。SDP モードでは、ポート幅が 18 ビット以下の場合は SRVAL_A のみを使用し、ポート幅が 18 ビットより広い場合は SRVAL_B は上位 18 ビット (パリティビットを含む) に適用されます。
WRITE_MODE_A	文字列	“WRITE_FIRST”、 “READ_FIRST”、 “NO_CHANGE”	“WRITE_FIRST”	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> <li>“WRITE_FIRST”：書き込まれた値が出力ポートに出力されます。</li> <li>“READ_FIRST”：書き込み前にそのメモリロケーションに格納されていた値が出力ポートに出力されます。</li> <li>“NO_CHANGE”：出力ポートの以前の値が保持されます。</li> </ul> <p>RAM_MODE が “SDP” に設定されている場合、WRITE_MODE_A を “READ_FIRST” (両方のポートで共通クロックを使用する場合) または “WRITE_FIRST” (両方のポートで異なるクロックを使用する場合) に設定する必要があります。</p>
WRITE_MODE_B	文字列	“WRITE_FIRST”、 “READ_FIRST”、 “NO_CHANGE”	“WRITE_FIRST”	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> <li>“WRITE_FIRST”：書き込まれた値が出力ポートに出力されます。</li> <li>“READ_FIRST”：書き込み前にそのメモリロケーションに格納されていた値が出力ポートに出力されます。</li> <li>“NO_CHANGE”：出力ポートの以前の値が保持されます。</li> </ul> <p>RAM_MODE が “SDP” に設定されている場合、WRITE_MODE_B を “READ_FIRST” (両方のポートで共通クロックを使用する場合) または “WRITE_FIRST” (両方のポートで異なるクロックを使用する場合) に設定する必要があります。</p>

属性	データ型	値	デフォルト	説明
				なるクロックを使用する場合) に設定する必要がある。

## VHDL 記述 (インスタンスエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB8BWER: 8k-bit Data and 1k-bit Parity Configurable Synchronous Block RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

RAMB8BWER_inst : RAMB8BWER
generic map (
  -- DATA_WIDTH_A/DATA_WIDTH_B: 'If RAM_MODE="TDP": 0, 1, 2, 4, 9 or 18; If RAM_MODE="SDP": 36'
  DATA_WIDTH_A => 0,
  DATA_WIDTH_B => 0,
  -- DOA_REG/DOB_REG: Optional output register (0 or 1)
  DOA_REG => 0,
  DOB_REG => 0,
  -- EN_RSTRAM_A/EN_RSTRAM_B: Enable/disable RST
  EN_RSTRAM_A => TRUE,
  EN_RSTRAM_B => TRUE,
  -- INITP_00 to INITP_03: Initial memory contents.
  INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  -- INIT_00 to INIT_1F: Initial memory contents.
  INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
  -- INIT_A/INIT_B: Initial values on output port
  INIT_A => X"00000",
  INIT_B => X"00000",
  -- INIT_FILE: Not Supported

```







```
// when RAM_MODE="SDP")
.ADDRBRDADDR(ADDRBRDADDR), // 13-bit input: B port address/Read address input
.CLKBRDCLK(CLKBRDCLK), // 1-bit input: B port clock/Read clock input
.ENBRDEN(ENBRDEN), // 1-bit input: B port enable/Read enable input
.REGCEBREGCE(REGCEBREGCE), // 1-bit input: B port register enable/Register enable input
.RSTBRST(RSTBRST), // 1-bit input: B port set/reset input
.WEBWEU(WEBWEU), // 2-bit input: B port write enable input
// Port B Data: 16-bit (each) input: Port B data
.DIBDI(DIBDI), // 16-bit input: B port data/MSB data input
.DIPBDIP(DIPBDIP) // 2-bit input: B port parity/MSB parity input
);

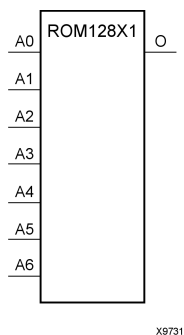
// End of RAMB8BWER_inst instantiation
```

## 詳細情報

- ・ [Spartan-6 FPGA ブロック RAM ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート：DC 特性およびスイッチ特性](#)

## ROM128X1

### プリミティブ：128-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 128 ワード X 1 ビットの ROM です。データ出力 (O) には、7 ビットのアドレス (A6 ~ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 32 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	128 ビット値	すべて 0	ROM の値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

ROM128X1_inst : ROM128X1
generic map (
  INIT => X"00000000000000000000000000000000")
port map (
  O => O,    -- ROM output
  A0 => A0,  -- ROM address[0]
  A1 => A1,  -- ROM address[1]
  A2 => A2,  -- ROM address[2]
  A3 => A3,  -- ROM address[3]
  A4 => A4,  -- ROM address[4]
  A5 => A5,  -- ROM address[5]
  A6 => A6,  -- ROM address[6]
);

-- End of ROM128X1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

ROM128X1 #(
  .INIT(128'h00000000000000000000000000000000) // Contents of ROM
) ROM128X1_inst (
  .O(O),    // ROM output
  .A0(A0),  // ROM address[0]
  .A1(A1),  // ROM address[1]
  .A2(A2),  // ROM address[2]
  .A3(A3),  // ROM address[3]
  .A4(A4),  // ROM address[4]
  .A5(A5),  // ROM address[5]
  .A6(A6)   // ROM address[6]
);

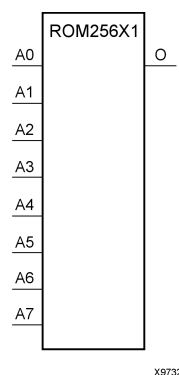
// End of ROM128X1_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ROM256X1

プリミティブ：256-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 256 ワード X 1 ビットの ROM です。データ出力 (O) には、8 ビットのアドレス (A7 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 64 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。

INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	256 ビット値	すべて 0	ROM の値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

ROM256X1_inst : ROM256X1
generic map (
  INIT => X"0000000000000000000000000000000000000000000000000000000000000000"
)
port map (
  O => O,    -- ROM output
  A0 => A0,   -- ROM address[0]
  A1 => A1,   -- ROM address[1]
  A2 => A2,   -- ROM address[2]
  A3 => A3,   -- ROM address[3]
  A4 => A4,   -- ROM address[4]
  A5 => A5,   -- ROM address[5]
  A6 => A6,   -- ROM address[6]
  A7 => A7,   -- ROM address[7]
);

-- End of ROM256X1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

ROM256X1 #(
  .INIT(256'h0000000000000000000000000000000000000000000000000000000000000000) // Contents of ROM
) ROM256X1_inst (
  .O(O),    // ROM output
  .A0(A0),  // ROM address[0]
  .A1(A1),  // ROM address[1]
  .A2(A2),  // ROM address[2]
  .A3(A3),  // ROM address[3]
  .A4(A4),  // ROM address[4]
  .A5(A5),  // ROM address[5]
  .A6(A6),  // ROM address[6]
  .A7(A7)   // ROM address[7]
);

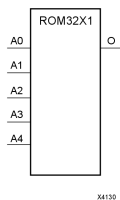
// End of ROM256X1_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ROM32X1

プリミティブ：32-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 32 ワード X 1 ビットの ROM です。データ出力 (O) には、5 ビットのアドレス (A4 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 8 桁の 16 進数で、ROM には最上位ビット A=1FH から最下位ビット A=00H の順に書き込まれます。

たとえば INIT=10A78F39 と指定すると、「0001 0000 1010 0111 1000 1111 0011」というデータストリームが生成されます。INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)



## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	ROM の値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

ROM32X1_inst : ROM32X1
generic map (
  INIT => X"00000000")
port map (
  O => O,    -- ROM output
  A0 => A0,  -- ROM address[0]
  A1 => A1,  -- ROM address[1]
  A2 => A2,  -- ROM address[2]
  A3 => A3,  -- ROM address[3]
  A4 => A4   -- ROM address[4]
);
-- End of ROM32X1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

ROM32X1 #(
  .INIT(32'h00000000) // Contents of ROM
) ROM32X1_inst (
  .O(O),              // ROM output
  .A0(A0),            // ROM address[0]
  .A1(A1),            // ROM address[1]
  .A2(A2),            // ROM address[2]
  .A3(A3),            // ROM address[3]
  .A4(A4)             // ROM address[4]
);

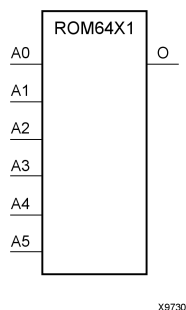
// End of ROM32X1_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ROM64X1

プリミティブ：64-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 64 ワード X 1 ビットの ROM です。データ出力 (O) には、6 ビットのアドレス (A5 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 16 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ROM の値を指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

ROM64X1_inst : ROM64X1
generic map (
  INIT => X"0000000000000000")
port map (
  O => O,    -- ROM output
  A0 => A0,  -- ROM address[0]
  A1 => A1,  -- ROM address[1]
  A2 => A2,  -- ROM address[2]
  A3 => A3,  -- ROM address[3]
  A4 => A4,  -- ROM address[4]
  A5 => A5   -- ROM address[5]
);

-- End of ROM64X1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

ROM64X1 #(
  .INIT(64'h0000000000000000) // Contents of ROM
) ROM64X1_inst (
  .O(O),    // ROM output
  .A0(A0),  // ROM address[0]
  .A1(A1),  // ROM address[1]
  .A2(A2),  // ROM address[2]
  .A3(A3),  // ROM address[3]
  .A4(A4),  // ROM address[4]
  .A5(A5)   // ROM address[5]
);

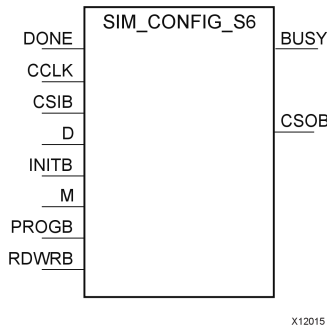
// End of ROM64X1_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## SIM\_CONFIG\_S6

### シミュレーション：Configuration Simulation Model



### 概要

このシミュレーション コンポーネントを使用すると、多数の一般的なコンフィギュレーション インターフェイス、ファンクション、およびコマンドの論理シミュレーションを実行でき、コンフィギュレーション動作をボード レベルで理解し、デバッグするのに役立ちます。また、デザインのグローバル セット/リセット (GSR) やグローバル トライステート (GTS) などの一部のスタートアップ動作もシミュレーションできます。このモデルは、FPGA ツールの特定のプリミティブにはマップされず、デザインに直接インスタンス化することはできませんが、テストベンチなどのシミュレーションのみのファイルで指定するなど、合成で除外してデザイン ネットリストに含まれないようにすれば、ソース デザインと共に使用できます。このモデルは、論理 (RTL) シミュレーションおよびタイミング シミュレーションで使用できます。また、ICAP\_SPARTAN6 をインスタンス化し、そのコンフィギュレーション アクセスをシミュレーションする際にも間接的に使用されます。

### ポートの説明

ポート名	方向	幅	機能
BUSY	出力	1	リードバック中に使用されます。
CSOB	出力	1	アクティブ Low のパラレル デイジー チェーン チップ セレクト出力。FPGA が 1 つのみのアプリケーションでは使用されません。
DONE	入出力	1	コンフィギュレーションの完了を示すアクティブ High の信号 <ul style="list-style-type: none"> <li>0 = FPGA のコンフィギュレーションは完了していない</li> <li>1 = FPGA のコンフィギュレーション完了</li> </ul>
CCLK	入力	1	JTAG を除くすべてのコンフィギュレーション モードのコンフィギュレーション クロック ソース
CSIB	入力	1	SelectMAP データ バスをイネーブルにするアクティブ Low のチップ セレクト <ul style="list-style-type: none"> <li>0 = SelectMAP データ バスをイネーブル</li> <li>1 = SelectMAP データ バスをディスエーブル</li> </ul>
D	入力	32	コンフィギュレーションおよびリードバック データ バス。CCLK の立ち上がりエッジで供給されます。
INITB	入力	1	モード ピンが読み込まれる前は、Low に保持することでコンフィギュレーションを遅延できます。モード ピンが読み込まれた後は、オープン ドレインのアクティブ Low 出力となり、コンフィギュレーション中の CRC エラーの有無を示します。 <ul style="list-style-type: none"> <li>0 = CRC エラー</li> </ul>

ポート名	方向	幅	機能
			<ul style="list-style-type: none"> <li>1 = CRC エラーなし</li> </ul> SEU 検出ファンクションが有効の場合、リードバック CRC エラーが検出されると Low に駆動されます (オプション)。
M	入力	2	モード ピン。コンフィギュレーション モードを指定します。
PROGB	入力	1	アクティブ Low の非同期フルチップ リセット
RDWRB	入力	1	D[x:0] データ バスの方向を指定します。 <ul style="list-style-type: none"> <li>0 = 入力</li> <li>1 = 出力</li> </ul> RDWRB 入力は、CSLB がディアサートの場合にのみ変更可能です。CSLB がディアサートされていない場合は、ABORT が発生します。

## デザインの入力方法

インスタンス化	テストベンチまたはシミュレーション ファイルでのみ
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

テストベンチ ファイルにインスタンス化し、インプリメンテーション ファイルまたはデザインの合成に使用されるファイルには含めないことをお勧めします。コンフィギュレーションの読み込みとデバイスのスタートアップの関係およびスタートアップ シーケンスを決定するために使用できます。通常このモデルは、コンフィギュレーションの動作を観察するためコンフィギュレーション ビットストリーム ファイルで使います。

このコンポーネントの使用法の詳細およびシミュレーションについては、『合成/シミュレーション デザイン ガイド』を参照してください。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DEVICE_ID	32 ビットの 16 進数	有効なデバイス ID コード	32'h00000000	ターゲット デバイスのデバイス ID コードを指定します。ビットストリームの処理およびデバイスの識別読み出しで使用されます。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- SIM_CONFIG_S6: Behavioral Simulation-only Model of FPGA SelectMap Configuration
--               Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

SIM_CONFIG_S6_inst : SIM_CONFIG_S6
generic map (
  DEVICE_ID => X"00000000") -- Specifies the Pre-programmed Device ID value
port map (

```

```
BUSY => BUSY,    -- 1-bit output Busy pin
CSOB => CSOB,    -- 1-bit output chip select pin
DONE => DONE,    -- 1-bit bi-directional Done pin
CCLK => CCLK,    -- 1-bit input configuration clock
D => D,          -- 8-bit bi-directional configuration data
INITB => INITB,  -- 1-bit bi-directional INIT status pin
M => M,          -- 3-bit input Mode pins
PROGB => PROGB,  -- 1-bit input Program pin
RDWRB => RDWRB  -- 1-bit input Read/Write pin
);

-- End of SIM_CONFIG_S6_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// SIM_CONFIG_S6: Behavioral Simulation-only Model of FPGA SelectMap Configuration
//                      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

SIM_CONFIG_S6 #(
    .DEVICE_ID(32'h00000000) // Specify DEVICE_ID
) SIM_CONFIG_S6_inst (
    .BUSY(BUSY), // 1-bit output Busy pin
    .CSOB(CSOB), // 1-bit output chip select pin
    .DONE(DONE), // 1-bit bi-directional Done pin
    .CCLK(CCLK), // 1-bit input configuration clock
    .CSIB(CSIB), // 1-bit input chip select
    .D(D),       // 16-bit bi-directional configuration data
    .INITB(INITB), // 1-bit bi-directional INIT status pin
    .M(M),       // 2-bit input Mode pins
    .PROGB(PROGB), // 1-bit input Program pin
    .RDWRB(RDWRB) // 1-bit input Read/write pin
);

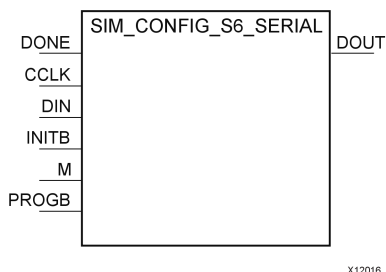
// End of SIM_CONFIG_S6_inst instantiation
```

## 詳細情報

- ・ [『合成/シミュレーション デザイン ガイド』\(UG626\)](#)
- ・ [『Spartan-6 FPGA コンフィギュレーション ユーザー ガイド』\(UG380\)](#)

## SIM\_CONFIG\_S6\_SERIAL

## シミュレーション：Serial Configuration Simulation Model



## 概要

このシミュレーション コンポーネントを使用すると、多数の一般的なシリアル コンフィギュレーション インターフェイス、ファンクション、およびコマンドの論理シミュレーションを実行でき、コンフィギュレーション動作をボード レベルで理解し、デバッグするのに役立ちます。また、デザインのグローバル セット/リセット (GSR) やグローバル トライステート (GTS) などの一部のスタートアップ動作もシミュレーションできます。このモデルは、FPGA ツールの特定のプリミティブにはマップされず、デザインに直接インスタンスエートすることはできませんが、テストベンチなどのシミュレーションのみのファイルで指定するなど、合成で除外してデザイン ネットリストに含まれないようにすれば、ソース デザインと共に使用できます。このモデルは、論理 (RTL) シミュレーションおよびタイミング シミュレーションで使用できます。

## ポートの説明

ポート名	方向	幅	機能
DONE	入出力	1	コンフィギュレーションの完了を示すアクティブ High の信号 <ul style="list-style-type: none"> <li>0 = FPGA のコンフィギュレーションは完了していない</li> <li>1 = FPGA のコンフィギュレーション完了</li> </ul>
DOUT	出力	1	デジタイズ チェーンのダウンストリーム デバイス用のシリアル データ出力。データは CCLK の立ち下がりエッジで供給されます。
CCLK	入力	1	JTAG を除くすべてのコンフィギュレーション モードのコンフィギュレーション クロック ソース
DIN	入力	1	シリアル コンフィギュレーション データ入力 (CCLK の立ち上がりエッジに同期)
INITB	入力	1	モード ピンが読み込まれる前は、Low に保持することでコンフィギュレーションを遅延できます。モード ピンが読み込まれた後は、オープンドレインのアクティブ Low 出力となり、コンフィギュレーション中の CRC エラーの有無を示します。 <ul style="list-style-type: none"> <li>0 = CRC エラー</li> <li>1 = CRC エラーなし</li> </ul> SEU 検出ファンクションが有効の場合、リードバック CRC エラーが検出されると Low に駆動されます (オプション)。
M	入力	2	モード ピン。コンフィギュレーション モードを指定します。
PROGB	入力	1	アクティブ Low の非同期フルチップ リセット



## デザインの入力方法

インスタンス化	テストベンチまたはシミュレーション ファイルでのみ
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

テストベンチ ファイルにインスタンス化し、インプリメンテーション ファイルまたはデザインの合成に使用されるファイルには含めないことをお勧めします。コンフィギュレーションの読み込みとデバイスのスタートアップの関係およびスタートアップ シーケンスを決定するために使用できます。通常このモデルは、コンフィギュレーションの動作を観察するためコンフィギュレーション ビットストリーム ファイルで使われます。

このコンポーネントの使用方法的詳細およびシミュレーションについては、『合成/シミュレーション デザイン ガイド』を参照してください。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DEVICE_ID	32 ビットの 16 進数	有効なデバイス ID コード	32'h00000000	ターゲット デバイスのデバイス ID コードを指定します。ビットストリームの処理およびデバイスの識別読み出しで使用されます。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- SIM_CONFIG_S6_SERIAL: Behavioral Simulation-only Model of FPGA Serial Configuration
--                               Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

SIM_CONFIG_S6_SERIAL_inst : SIM_CONFIG_S6_SERIAL
generic map (
  DEVICE_ID => X"00000000") -- Specifies the Pre-programmed Device ID value
port map (
  DONE => DONE,    -- 1-bit bi-directional Done pin
  CCLK => CCLK,    -- 1-bit input configuration clock
  DIN => DIN,      -- 1-bit input configuration data
  INITB => INITB,  -- 1-bit bi-directional INIT status pin
  M => M,          -- 3-bit input Mode pins
  PROGB => PROGB  -- 1-bit input Program pin
);

-- End of SIM_CONFIG_S6_SERIAL_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// SIM_CONFIG_S6_SERIAL: Behavioral Simulation-only Model of FPGA Serial Configuration
//                               Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

SIM_CONFIG_S6_SERIAL #(
    .DEVICE_ID(32'h00000000) // Specify DEVICE_ID
) SIM_CONFIG_S6_SERIAL_inst (
    .DONE(DONE),           // 1-bit bi-directional Done pin
    .CCLK(CCLK),           // 1-bit input configuration clock
    .DIN(DIN),             // 1-bit input configuration data
    .INITB(INITB),         // 1-bit bi-directional INIT status pin
    .M(M),                 // 2-bit input Mode pins
    .PROGB(PROGB)          // 1-bit input Program pin
);

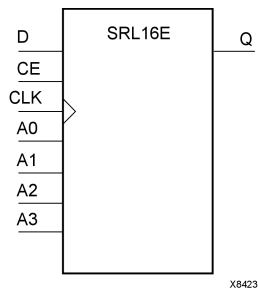
// End of SIM_CONFIG_S6_SERIAL_inst instantiation
```

## 詳細情報

- ・ [『合成/シミュレーション デザイン ガイド』\(UG626\)](#)
- ・ [『Spartan-6 FPGA コンフィギュレーション ユーザー ガイド』\(UG380\)](#)

## SRL16E

## プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Clock Enable



## 概要

このデザイン エLEMENTは、シフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには：** 入力 A3 ～ A0 の値を一定にします。シフトレジスタは 1 ～ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$  という式で算出できます。A3、A2、A1、A0 がすべて 0 の場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには：** 入力 A3 ～ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ～ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中に 0000 にクリアされます。

CE が High の場合、クロック (CLK) が Low から High に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが Low から High に切り替わるときに CE が High の場合、シフトレジスタの値は次の高位ビットにシフトされ、新しい値がロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。CE が Low の場合、クロック遷移は無視されます。

## 論理表

入力				出力
Am	CE	CLK	D	Q
Am	0	X	X	Q(Am)
Am	1	↑	D	Q(Am - 1)
m = 0、1、2、3				

## ポートの説明

ポート名	方向	幅	機能
Q	出力	1	シフトレジスタ データ出力
D	入力	1	シフトレジスタ データ入力
CLK	入力	1	クロック
CE	入力	1	アクティブ High のクロック イネーブル
A	入力	4	SRL のワード数のダイナミック選択 ・ A=0000 ==> 1 ビットシフト長 ・ A=1111 ==> 16 ビットシフト長

## デザインの入力方法

インスタンシエーション	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべて 0	コンフィギュレーション後のシフトレジスタと出力の初期値を指定します。

## VHDL 記述 (インスタンシエーション)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

SRL16E_inst : SRL16E
generic map (
  INIT => X"0000")
port map (
  Q => Q,          -- SRL data output
  A0 => A0,         -- Select[0] input
  A1 => A1,         -- Select[1] input
  A2 => A2,         -- Select[2] input
  A3 => A3,         -- Select[3] input
  CE => CE,         -- Clock enable input
  CLK => CLK,       -- Clock input
  D => D           -- SRL data input
);

-- End of SRL16E_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

SRL16E #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRL16E_inst (
    .Q(Q),           // SRL data output
    .A0(A0),         // Select[0] input
    .A1(A1),         // Select[1] input
    .A2(A2),         // Select[2] input
    .A3(A3),         // Select[3] input
    .CE(CE),         // Clock enable input
    .CLK(CLK),       // Clock input
    .D(D)            // SRL data input
);

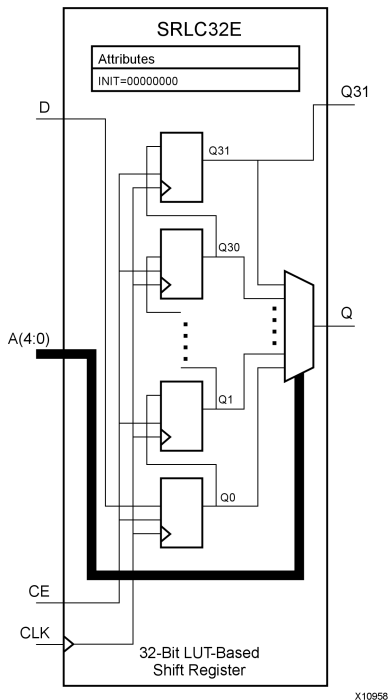
// End of SRL16E_inst instantiation
```

## 詳細情報

[Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## SRLC32E

**プリミティブ：32 Clock Cycle, Variable Length Shift Register Look-Up Table (LUT) with Clock Enable**



### 概要

このデザイン エLEMENTは、1 つのルックアップ テーブル (LUT) にインプリメントされる、可変長で 1 ～ 32 クロック サイクルのシフトレジスタです。シフトレジスタの長さは、固定することも、変動させることもできます。このELEMENTは、アクティブ High のクロック イネーブルおよびカスケード機能も備えているため、複数の SRLC32E をカスケード接続でき、より大型のシフトレジスタを作成できます。

### ポートの説明

ポート名	方向	幅	機能
Q	出力	1	シフトレジスタ データ出力
Q31	出力	1	シフトレジスタ カスケード出力 (後続 SRLC32E の D 入力に接続)
D	入力	1	シフトレジスタ データ入力
CLK	入力	1	クロック
CE	入力	1	アクティブ High のクロック イネーブル
A	入力	5	SRL の長さのダイナミック選択 A=00000 ==> 1 ビット A=11111 ==> 32 ビット

## デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

インスタンス化する場合、このコンポーネントを次のように接続します。

- ・ CLK 入力を適切なクロック ソースに、D 入力をシフト/格納するデータ ソースに、Q 出力を FDCPE 入力または FDRSE 入力などの適切なデスティネーションに接続します。
- ・ クロック イネーブル ピン (CE) はクロック イネーブル信号に接続するか、使用しない場合は論理 1 にします。
- ・ 5 ビット バス A は、一定の値 (0 ~ 31) にしてシフトレジスタの長さを 1 ~ 32 ビットに固定するか、または適切な論理値にしてシフトレジスタの長さを 1 ~ 32 ビットの範囲で変更できます。
- ・ シフトレジスタの長さを 32 ビットより長くする場合は、Q31 出力ピンを後続の SRLC32E の D 入力に接続します。
- ・ Q31 出力は、別の SRLC32E 以外には接続できません。
- ・ Q 出力は、カスケード モードでも使用できます。
- ・ 32 ビットの 16 進数の INIT 属性で、シフトレジスタの初期シフトパターンを指定できます。
- ・ シフトアウトされる最初の値は INIT[0] です。

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	SRLC32E の初期シフト パターンを指定します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- SRLC32E: 32-bit variable length shift register LUT
--           with clock enable
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

SRLC32E_inst : SRLC32E
generic map (
  INIT => X"00000000")
port map (
  Q => Q,           -- SRL data output
  Q31 => Q31,       -- SRL cascade output pin
  A => A,           -- 5-bit shift depth select input
  CE => CE,         -- Clock enable input
  CLK => CLK,       -- Clock input
  D => D           -- SRL data input
);

-- End of SRLC32E_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// SRLC32E: 32-bit variable length cascadable shift register LUT
//           with clock enable
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

SRLC32E #(
    .INIT(32'h00000000) // Initial Value of Shift Register
) SRLC32E_inst (
    .Q(Q),           // SRL data output
    .Q31(Q31),      // SRL cascade output pin
    .A(A),           // 5-bit shift depth select input
    .CE(CE),         // Clock enable input
    .CLK(CLK),       // Clock input
    .D(D)            // SRL data input
);

// End of SRLC32E_inst instantiation
```

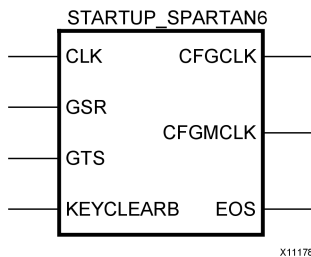
## 詳細情報

- ・ [『Spartan-6 FPGA コンフィギャラブル ロジック ブロック ユーザー ガイド』\(UG384\)](#)
- ・ [『Spartan-6 FPGA データシート：DC 特性およびスイッチ特性』\(DS162\)](#)



## STARTUP\_SPARTAN6

**プリミティブ：Spartan®-6 Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface**



### 概要

このデザイン エLEMENTは、グローバル非同期セット/リセット (GSR) 信号、グローバルトライステート (GTS) 専用配線、内部コンフィギュレーション信号、SPI PROM が使用される場合は SPI PROM の入力ピンなどへのロジックとデバイスピンの接続に使用されます。デバイスのコンフィギュレーションの終わりにスタートアップ シーケンスで別のクロックを使用するのを指定したり、コンフィギュレーション クロックを内部ロジックにアクセスさせるのにも使用されます。

### ポートの説明

ポート名	方向	幅	機能
CFGCLK	出力	1	コンフィギュレーション ロジックのメイン クロック出力
CFGMCLK	出力	1	コンフィギュレーションの内部オシレーターのクロック出力
CLK	入力	1	ユーザー スタートアップ クロック入力
EOS	出力	1	コンフィギュレーションの終了を示すアクティブ High の出力信号
GSR	入力	1	グローバル セット/リセット (GSR) 入力 (ポート名に GSR は使用不可)
GTS	入力	1	グローバルトライステート (GTS) 入力 (ポート名に GTS は使用不可)
KEYCLEARB	入力	1	バッテリー充電 RAM (BDRAM) からのクリア AES デクリプタ キー入力

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

専用の GSR 回路はソース ピンまたはロジックを GSR ピンに接続すれば使用できますが、このコンポーネントの GSR 回路の使用には、特別の注意が必要です。GSR ネットのスキューは確定できないので、セット/リセット信号の一般配線を使用して、配線遅延とスキューをデザインのタイミング解析の一部として計算できるようにするか、クロック サイクルのリリース時にスキューが回路の動作の妨げとならないような回避策をとってください。

同様に、専用のグローバルトライステートが使用される場合は、適切なソース ピンまたはロジックをこのプリミティブの GTS 入力ピンに接続します。コンフィギュレーションのスタートアップ シーケンスのクロックを指定するには、デザインからのクロックをこのデザイン エLEMENTの CLK ピンに接続します。

## VHDL 記述 (インスタンス化)

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- STARTUP_SPARTAN6: STARTUP Block
--                      Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

STARTUP_SPARTAN6_inst : STARTUP_SPARTAN6
port map (
  CFGCLK => CFGCLK,      -- 1-bit output: Configuration logic main clock output.
  CFGMCLK => CFGMCLK,    -- 1-bit output: Configuration internal oscillator clock output.
  EOS => EOS,            -- 1-bit output: Active high output signal indicates the End Of Configuration.
  CLK => CLK,            -- 1-bit input: User startup-clock input
  GSR => GSR,            -- 1-bit input: Global Set/Reset input (GSR cannot be used for the port name)
  GTS => GTS,            -- 1-bit input: Global 3-state input (GTS cannot be used for the port name)
  KEYCLEARB => KEYCLEARB -- 1-bit input: Clear AES Decrypter Key input from Battery-Backed RAM (BBRAM)
);

-- End of STARTUP_SPARTAN6_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// STARTUP_SPARTAN6: STARTUP Block
//                      Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

STARTUP_SPARTAN6 STARTUP_SPARTAN6_inst (
  .CFGCLK(CFGCLK),      // 1-bit output: Configuration logic main clock output.
  .CFGMCLK(CFGMCLK),    // 1-bit output: Configuration internal oscillator clock output.
  .EOS(EOS),            // 1-bit output: Active high output signal indicates the End Of Configuration.
  .CLK(CLK),            // 1-bit input: User startup-clock input
  .GSR(GSR),            // 1-bit input: Global Set/Reset input (GSR cannot be used for the port name)
  .GTS(GTS),            // 1-bit input: Global 3-state input (GTS cannot be used for the port name)
  .KEYCLEARB(KEYCLEARB) // 1-bit input: Clear AES Decrypter Key input from Battery-Backed RAM (BBRAM)
);

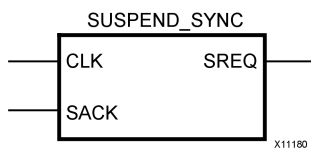
// End of STARTUP_SPARTAN6_inst instantiation
```

## 詳細情報

- ・ [Spartan-6 FPGA コンフィギュレーション ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート：DC 特性およびスイッチ特性](#)

## SUSPEND\_SYNC

### プリミティブ：Suspend Mode Access



### 概要

このデザイン エLEMENTは、一時停止モードを使用しているアプリケーションのデザインを同期化する機能を拡張します。同期化が必要なクロックドメインが数個ある場合でも、一時停止モードを開始するトリガーの同期化に 3 ピン インターフェイスが使用されます。

### ポートの説明

ポート名	方向	幅	機能
CLK	入力	1	ユーザー クロック入力
SACK	入力	1	SUSPEND の確認出力
SREQ	出力	1	SUSPEND の要求出力

### デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### VHDL 記述（インスタンス化）

次の 2 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- SUSPEND_SYNC: Suspend Mode Access
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.5

SUSPEND_SYNC_inst : SUSPEND_SYNC
port map (
    SREQ => SREQ, -- 1-bit output: Suspend request output
    CLK  => CLK,  -- 1-bit input: User clock input
    SACK => SACK  -- 1-bit input: SUSPEND acknowledgement output
);

-- End of SUSPEND_SYNC_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// SUSPEND_SYNC: Suspend Mode Access
//                               Spartan-6
// Xilinx HDL Libraries Guide, version 14.5

SUSPEND_SYNC SUSPEND_SYNC_inst (
    .SREQ(SREQ), // 1-bit output: Suspend request output
    .CLK(CLK),   // 1-bit input: User clock input
    .SACK(SACK)  // 1-bit input: SUSPEND acknowledgement output
);

// End of SUSPEND_SYNC_inst instantiation
```

## 詳細情報

- ・ [Spartan-6 FPGA コンフィギュレーション ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート：DC 特性およびスイッチ特性](#)