

# 7 シリーズ FPGA および Zynq ライブラリ ガイド (HDL 用)

UG768 (V 14.4) 2012 年 12 月 18 日



## Notice of Disclaimer

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

© Copyright 2002–2012 Xilinx Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

本資料は英語版 (v14.4) 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。

- ・ はじめに
- ・ 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

このバージョンのライブラリ ガイドでは、Zynq™ を含む 7 シリーズ アーキテクチャのデザイン 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

---

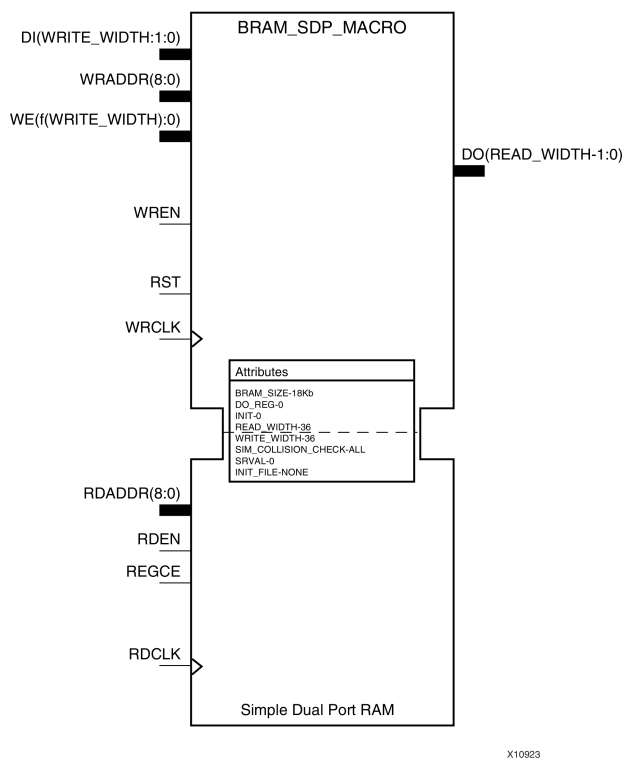
このセクションでは、Zynq™ を含む 7 シリーズ アーキテクチャで利用できる UniMacro について説明します。UniMacro は、アルファベット順に並べられています。

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
- ・ インスタンスエーション 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

7 シリーズ FPGA デバイスにはブロック RAM が数個含まれ、36kb または 18kb の汎用 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。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。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す。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
DO	出力	「ポートのコンフィギュレーション」の表を参照	RDADDR で指定されたデータ出力バス
DI	入力	「ポートのコンフィギュレーション」の表を参照	WRADDR で指定されたデータ入力バス
WRADDR、RDADDR	入力	「ポートのコンフィギュレーション」の表を参照	書き込み/読み出しアドレス入力バス
WE	入力	「ポートのコンフィギュレーション」の表を参照	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

## ポートのコンフィギュレーション

この UniMacro は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、インスタンス化のみが可能です。デザイン要件を満たすには、この表を参照して正しくコンフィギュレーションしてください。

DATA_WIDTH	BRAM_SIZE	ADDR	WE
72 ~ 37	36Kb	9	8
36 ~ 19	36Kb	10	4
	18Kb	9	
18 ~ 10	36Kb	11	2
	18Kb	10	
9 ~ 5	36Kb	12	1
	18Kb	11	
4 ~ 3	36Kb	13	1
	18Kb	12	
2	36Kb	14	1
	18Kb	13	
1	36Kb	15	1
	18Kb	14	

## デザインの入力方法

この UniMacro は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、インスタンス化のみが可能です。デザイン要件を満たすには、「ポートのコンフィギュレーション」の表を参照して正しくコンフィギュレーションしてください。

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	"36Kb"、"18Kb"	"18Kb"	RAM を 36Kb または 18Kb メモリとしてコンフィギュレーションします。
DEVICE	文字列	"7SERIES"	"7SERIES"	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。

属性	データ型	値	デフォルト	説明
DO_REG	整数	0、1	0	1 に設定すると、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。0 に設定すると、読み出しを 1 クロック サイクルで実行できますが、clock-to-out タイムが長くなります。
INIT	16 進数	72 ビット値	すべてゼロ	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。



属性	データ型	値	デフォルト	説明
SRVAL	16 進数	72 ビット値	すべてゼロ	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_7F	16 進数	256 ビット値	すべてゼロ	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	すべてゼロ	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。

## 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;

-- BRAM_SDP_MACRO: Simple Dual Port RAM
--               7 Series
-- Xilinx HDL Libraries Guide, version 14.4

-- 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.
```

```
-----
-- READ_WIDTH | BRAM_SIZE | READ Depth | RDADDR Width | WE Width --
-- WRITE_WIDTH |          | WRITE Depth | WRADDR Width |          --
-- ===== | ===== | ===== | ===== | =====
-- 37-72      | "36Kb"   | 512      | 9-bit      | 8-bit --
-- 19-36      | "36Kb"   | 1024     | 10-bit     | 4-bit --
-- 19-36      | "18Kb"   | 512      | 9-bit      | 4-bit --
-- 10-18      | "36Kb"   | 2048     | 11-bit     | 2-bit --
-- 10-18      | "18Kb"   | 1024     | 10-bit     | 2-bit --
-- 5-9        | "36Kb"   | 4096     | 12-bit     | 1-bit --
-- 5-9        | "18Kb"   | 2048     | 11-bit     | 1-bit --
-- 3-4        | "36Kb"   | 8192     | 13-bit     | 1-bit --
-- 3-4        | "18Kb"   | 4096     | 12-bit     | 1-bit --
-- 2          | "36Kb"   | 16384    | 14-bit     | 1-bit --
-- 2          | "18Kb"   | 8192     | 13-bit     | 1-bit --
-- 1          | "36Kb"   | 32768    | 15-bit     | 1-bit --
-- 1          | "18Kb"   | 16384    | 14-bit     | 1-bit --
-----
```

```
BRAM_SDP_MACRO_inst : BRAM_SDP_MACRO
generic map (
    BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
    DEVICE => "7SERIES", -- Target device: "VIRTEX5", "VIRTEX6", "7SERIES", "SPARTAN6"
    WRITE_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
    READ_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
    DO_REG => 0, -- Optional output register (0 or 1)
    INIT_FILE => "NONE",
    SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
    -- "GENERATE_X_ONLY" or "NONE"
    SRVAL => X"00000000000000000000", -- Set/Reset value for port output
    WRITE_MODE => "WRITE_FIRST", -- Specify "READ_FIRST" for same clock or synchronous clocks
    -- Specify "WRITE_FIRST" for asynchronous clocks on ports
    INIT => X"00000000000000000000", -- Initial values on output port
    -- The following INIT_xx declarations specify the initial contents of the RAM
    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",
```

```
-- The next set of INIT_xx are valid when configured as 36Kb
```

```
-- The next set of INITP xx are for the parity bits
```

[illegible]

```
-- The next set of INIT xx are valid when configured as 36Kb
```

[illegible]

```

port map (
    DO => DO,          -- Output read data port, width defined by READ_WIDTH parameter
    DI => DI,          -- Input write data port, width defined by WRITE_WIDTH parameter

```

## Verilog 記述 (インスタンス化)

[illegible]

```
// The next set of INIT_xx are valid when configured as 36Kb
```

[illegible]

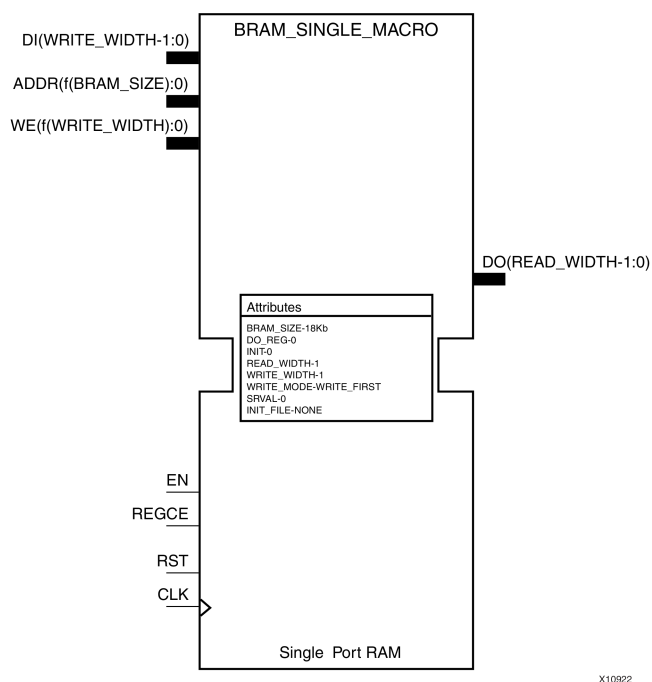
UG768 (V 14.4) 2012 年 12 月 18 日

詳細情報

[7 シリーズ 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

7 シリーズ FPGA デバイスにはブロック RAM が数個含まれ、36kb または 18kb の汎用 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
DO	出力	「ポートのコンフィギュレーション」の表を参照	ADDR で指定されたデータ出力バス
DI	入力	「ポートのコンフィギュレーション」の表を参照	ADDR で指定されたデータ入力バス
ADDR	入力	「ポートのコンフィギュレーション」の表を参照	アドレス入力バス
WE	入力	「ポートのコンフィギュレーション」の表を参照	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



## ポートのコンフィギュレーション

この UniMacro は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、インスタンス化のみが可能です。デザイン要件を満たすには、この表を参照して正しくコンフィギュレーションしてください。

WRITE_WIDTH	READ_WIDTH	BRAM_SIZE	ADDR	WE
72 ～ 37	72 ～ 37	36Kb	9	8
	36 ～ 19		10	
	18 ～ 10		11	
	9 ～ 5		12	
	4 ～ 3		13	
	2		14	
	1		15	
36 ～ 19	36 ～ 19	36Kb	10	4
	18 ～ 10		11	
	9 ～ 5		12	
	4 ～ 3		13	
	2		14	
	1		15	
18 ～ 10	36 ～ 19	36Kb	11	2
	18 ～ 10		11	
	9 ～ 5		12	
	4 ～ 3		13	
	2		14	
	1		15	
9 ～ 5	36 ～ 19	36Kb	12	1
	18 ～ 10		12	
	9 ～ 5		12	
	4 ～ 3		13	
	2		14	
	1		15	
4 ～ 3	36 ～ 19	36Kb	13	1
	18 ～ 10		13	
	9 ～ 5		13	
	4 ～ 3		13	
	2		14	
	1		15	

WRITE_WIDTH	READ_WIDTH	BRAM_SIZE	ADDR	WE
2	36 ~ 19	36Kb	14	1
	18 ~ 10		14	
	9 ~ 5		14	
	4 ~ 3		14	
	2		14	
	1		15	
1	36 ~ 19	36Kb	15	1
	18 ~ 10		15	
	9 ~ 5		15	
	3 ~ 4		15	
	2		15	
	1		15	
18 ~ 10	18 ~ 10	18Kb	10	2
	9 ~ 5		11	
	4 ~ 3		12	
	2		13	
	1		14	
9 ~ 5	18 ~ 10	18Kb	11	1
	9 ~ 5		11	
	4 ~ 3		12	
	2		13	
	1		14	
4 ~ 3	18 ~ 10	18Kb	12	1
	9 ~ 5		12	
	4 ~ 3		12	
	2		13	
	1		14	
2	18 ~ 10	18Kb	13	1
	9 ~ 5		13	
	4 ~ 3		13	
	2		13	
	1		14	

WRITE_WIDTH	READ_WIDTH	BRAM_SIZE	ADDR	WE
1	18 ~ 10	18Kb	14	1
	9 ~ 5		14	
	4 ~ 3		14	
	2		14	
	1		14	

## デザインの入力方法

この UniMacro は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、インスタンスエーションのみが可能です。デザイン要件を満たすには、「ポートのコンフィギュレーション」の表を参照して正しくコンフィギュレーションしてください。

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	"36Kb"、"18Kb"	"18Kb"	RAM を 36Kb または 18Kb メモリとしてコンフィギュレーションします。
DEVICE	文字列	"7SERIES"	"7SERIES"	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。
DO_REG	整数	0、1	0	1 に設定すると、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。0 に設定すると、読み出しを 1 クロック サイクルで実行できますが、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。 次の設定が可能です。 ・ READ_WIDTH = WRITE_WIDTH ・ 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。
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。
INIT	16 進数	72 ビット値	すべてゼロ	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。

属性	データ型	値	デフォルト	説明
SRVAL	16 進数	72 ビット値	すべてゼロ	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	すべてゼロ	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	すべてゼロ	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。

## 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;

-- BRAM_SINGLE_MACRO: Single Port RAM
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

-- 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.
```

```
-----
```

READ_WIDTH	BRAM_SIZE	READ Depth	ADDR Width	WE Width
WRITE_WIDTH		WRITE Depth		
=====	=====	=====	=====	=====
37-72	"36Kb"	512	9-bit	8-bit
19-36	"36Kb"	1024	10-bit	4-bit
19-36	"18Kb"	512	9-bit	4-bit
10-18	"36Kb"	2048	11-bit	2-bit
10-18	"18Kb"	1024	10-bit	2-bit
5-9	"36Kb"	4096	12-bit	1-bit
5-9	"18Kb"	2048	11-bit	1-bit
3-4	"36Kb"	8192	13-bit	1-bit
3-4	"18Kb"	4096	12-bit	1-bit
2	"36Kb"	16384	14-bit	1-bit
2	"18Kb"	8192	13-bit	1-bit
1	"36Kb"	32768	15-bit	1-bit
1	"18Kb"	16384	14-bit	1-bit

```
-----
```

```
BRAM_SINGLE_MACRO_inst : BRAM_SINGLE_MACRO
generic map (
  BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
  DEVICE => "7SERIES", -- Target Device: "VIRTEX5", "7SERIES", "VIRTEX6", "SPARTAN6"
  DO_REG => 0, -- Optional output register (0 or 1)
  INIT => X"00000000", -- Initial values on output port
  INIT_FILE => "NONE",
  WRITE_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  READ_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  SRVAL => X"00000000", -- Set/Reset value for port output
  WRITE_MODE => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  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",
```

[illegible]

```
INIT_54 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_55 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_56 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_57 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_58 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_59 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_60 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_61 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_62 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_63 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_64 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_65 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_66 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_67 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_68 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_69 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_70 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_71 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_72 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_73 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_74 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_75 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_76 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_77 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_78 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_79 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7F => 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",
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

-- The next set of INIT\_xx are valid when configured as 36Kb

```
INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000")
```

port map (

```
DO => DO,          -- Output data, width defined by READ_WIDTH parameter
ADDR => ADDR,       -- Input address, width defined by read/write port depth
CLK => CLK,         -- 1-bit input clock
DI => DI,           -- Input data port, width defined by WRITE_WIDTH parameter
EN => EN,           -- 1-bit input RAM enable
REGCE => REGCE,     -- 1-bit input output register enable
RST => RST,         -- 1-bit input reset
WE => WE            -- Input write enable, width defined by write port depth
```

## Verilog 記述 (インスタンス化)

7 シリーズ FPGA および Zynq ライブラリ ガイド (HDL 用)

```
// The next set of INIT_xx are valid when configured as 36Kb
```

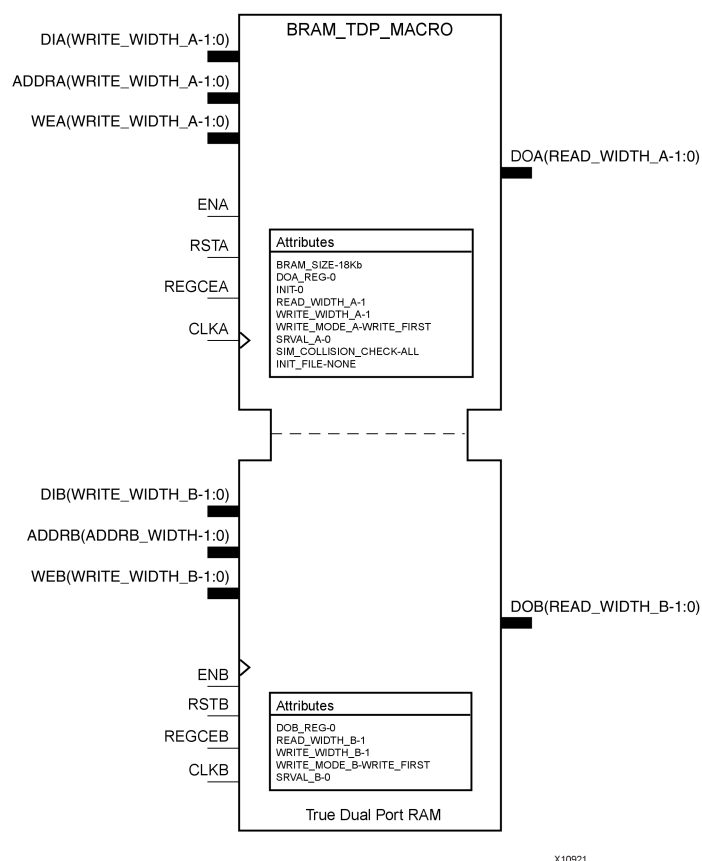
[illegible]



## 7 シリーズ FPGA の資料 (ユーザー ガイドおよびデータシート)

## BRAM\_TDP\_MACRO

### マクロ : True Dual Port RAM



## 概要

7 シリーズ FPGA デバイスにはブロック RAM が数個含まれ、18kb または 36kb の汎用 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。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。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。

## ポートの説明

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

ポート名	方向	幅	機能
ADDRA、ADDRB	入力	「ポートのコンフィギュレーション」の表を参照	ポート A およびポート B のアドレス入力バス
WEA、WEB	入力	「ポートのコンフィギュレーション」の表を参照	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

## ポートのコンフィギュレーション

この UniMacro は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、インスタンスエーションのみが可能です。デザイン要件を満たすには、この表を参照して正しくコンフィギュレーションしてください。

WRITE_WIDTH_A/B-DIA/DIB	READ_WIDTH_A/B-DOA/DOB	BRAM_SIZE	ADDRA/B	WEA/B
36 ~ 19	36 ~ 19	36Kb	10	4
	18 ~ 10		11	
	9 ~ 5		12	
	4 ~ 3		13	
	2		14	
	1		15	
18 ~ 10	36 ~ 19	36Kb	11	2
	18 ~ 10		11	
	9 ~ 5		12	
	4 ~ 3		13	
	2		14	
	1		15	
9 ~ 5	36 ~ 19	36Kb	12	1
	18 ~ 10		12	
	9 ~ 5		12	
	4 ~ 3		13	
	2		14	
	1		15	
4 ~ 3	36 ~ 19	36Kb	13	1
	18 ~ 10		13	
	9 ~ 5		13	
	4 ~ 3		13	
	2		14	
	1		15	

WRITE_WIDTH_A/B-DIA/DIB	READ_WIDTH_A/B-DOA/DOB	BRAM_SIZE	ADDRA/B	WEA/B
2	36 ~ 19	36Kb	14	1
	18 ~ 10		14	
	9 ~ 5		14	
	4 ~ 3		14	
	2		14	
	1		15	
1	36 ~ 19	36Kb	15	1
	18 ~ 10		15	
	9 ~ 5		15	
	4 ~ 3		15	
	2		15	
	1		15	
18 ~ 10	18 ~ 10	18Kb	10	2
	9 ~ 5		11	
	4 ~ 3		12	
	2		13	
	1		14	
9 ~ 5	18 ~ 10	18Kb	11	1
	9 ~ 5		11	
	4 ~ 3		12	
	2		13	
	1		14	
4 ~ 3	18 ~ 10	18Kb	12	1
	9 ~ 5		12	
	4 ~ 3		12	
	2		13	
	1		14	
2	18 ~ 10	18Kb	13	1
	9 ~ 5		13	
	4 ~ 3		13	
	2		13	
	1		14	
1	18 ~ 10	18Kb	14	1
	9 ~ 5		14	
	4 ~ 3		14	
	2		14	
	1		14	

## デザインの入力方法

この UniMacro は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、インスタンス化のみが可能です。デザイン要件を満たすには、「ポートのコンフィギュレーション」の表を参照して正しくコンフィギュレーションしてください。

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	"36Kb"、"18Kb"	"18Kb"	RAM を 36Kb または 18Kb メモリとしてコンフィギュレーションします。
DEVICE	文字列	"7SERIES"	"7SERIES"	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。
DO_REG	整数	0、1	0	1 に設定すると、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。0 に設定すると、読み出しを 1 クロック サイクルで実行できますが、clock-to-out タイムが長くなります。
INIT	16 進数	72 ビット値	すべてゼロ	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。  次の設定が可能です。  ・ READ_WIDTH = WRITE_WIDTH  ・ 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) に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。  ・ "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。  ・ "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。  ・ "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。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> <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。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>
SRVAL_A, SRVAL_B	16 進数	72 ビット値	すべてゼロ	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	すべてゼロ	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	すべてゼロ	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。

## 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;

-- BRAM_TDP_MACRO: True Dual Port RAM
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

-- 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 | ADDRA/B Width | WEA/B Width --
-- -----
-- 19-36            | "36Kb"    | 1024      | 10-bit         | 4-bit        --
-- 10-18            | "36Kb"    | 2048      | 11-bit         | 2-bit        --
-- 10-18            | "18Kb"    | 1024      | 10-bit         | 2-bit        --
-- 5-9              | "36Kb"    | 4096      | 12-bit         | 1-bit        --
-- 5-9              | "18Kb"    | 2048      | 11-bit         | 1-bit        --
-- 3-4              | "36Kb"    | 8192      | 13-bit         | 1-bit        --
-- 3-4              | "18Kb"    | 4096      | 12-bit         | 1-bit        --
-- 2                | "36Kb"    | 16384     | 14-bit         | 1-bit        --
-- 2                | "18Kb"    | 8192      | 13-bit         | 1-bit        --
-- 1                | "36Kb"    | 32768     | 15-bit         | 1-bit        --
-- 1                | "18Kb"    | 16384     | 14-bit         | 1-bit        --
-----

BRAM_TDP_MACRO_inst : BRAM_TDP_MACRO
generic map (
  BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
  DEVICE => "7SERIES", -- Target Device: "VIRTEX5", "VIRTEX6", "7SERIES", "SPARTAN6"
  DOA_REG => 0, -- Optional port A output register (0 or 1)
  DOB_REG => 0, -- Optional port B output register (0 or 1)
  INIT_A => X"000000000", -- Initial values on A output port
  INIT_B => X"000000000", -- Initial values on B output port
  INIT_FILE => "NONE",
  READ_WIDTH_A => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  READ_WIDTH_B => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
  -- "GENERATE_X_ONLY" or "NONE"
  SRVAL_A => X"000000000", -- Set/Reset value for A port output
  SRVAL_B => X"000000000", -- Set/Reset value for B port output
  WRITE_MODE_A => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"

```

7 シリーズ FPGA および Zynq ライブラリ ガイド (HDL 用)

```
-- The next set of INIT_xx are valid when configured as 36Kb
```



```

INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Output port-A data, width defined by READ_WIDTH_A parameter
  DOB => DOB,      -- Output port-B data, width defined by READ_WIDTH_B parameter
  ADDRA => ADDRA,   -- Input port-A address, width defined by Port A depth
  ADDR_B => ADDR_B, -- Input port-B address, width defined by Port B depth
  CLKA => CLKA,     -- 1-bit input port-A clock
  CLKB => CLKB,     -- 1-bit input port-B clock
  DIA => DIA,      -- Input port-A data, width defined by WRITE_WIDTH_A parameter
  DIB => DIB,      -- Input port-B data, width defined by WRITE_WIDTH_B parameter
  ENA => ENA,      -- 1-bit input port-A enable
  ENB => ENB,      -- 1-bit input port-B enable
  REGCEA => REGCEA, -- 1-bit input port-A output register enable
  REGCEB => REGCEB, -- 1-bit input port-B output register enable
  RSTA => RSTA,    -- 1-bit input port-A reset
  RSTB => RSTB,    -- 1-bit input port-B reset
  WEA => WEA,      -- Input port-A write enable, width defined by Port A depth
  WEB => WEB       -- Input port-B write enable, width defined by Port B depth
);

-- End of BRAM_TDP_MACRO_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// BRAM_TDP_MACRO: True Dual Port RAM
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// DATA_WIDTH_A/B | BRAM_SIZE | RAM Depth | ADDRA/B Width | WEA/B Width //
// ===== | ===== | ===== | ===== | ===== //
// 19-36 | "36Kb" | 1024 | 10-bit | 4-bit //
// 10-18 | "36Kb" | 2048 | 11-bit | 2-bit //
// 10-18 | "18Kb" | 1024 | 10-bit | 2-bit //
// 5-9 | "36Kb" | 4096 | 12-bit | 1-bit //
// 5-9 | "18Kb" | 2048 | 11-bit | 1-bit //
// 3-4 | "36Kb" | 8192 | 13-bit | 1-bit //
// 3-4 | "18Kb" | 4096 | 12-bit | 1-bit //
// 2 | "36Kb" | 16384 | 14-bit | 1-bit //
// 2 | "18Kb" | 8192 | 13-bit | 1-bit //
// 1 | "36Kb" | 32768 | 15-bit | 1-bit //
// 1 | "18Kb" | 16384 | 14-bit | 1-bit //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

BRAM_TDP_MACRO #(
  .BRAM_SIZE("18Kb"), // Target BRAM: "18Kb" or "36Kb"
  .DEVICE("7SERIES"), // Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6", "7SERIES"
  .DOA_REG(0), // Optional port A output register (0 or 1)
  .DOB_REG(0), // Optional port B output register (0 or 1)
  .INIT_A(36'h00000000), // Initial values on port A output port
  .INIT_B(36'h00000000), // Initial values on port B output port
  .INIT_FILE ("NONE"),
  .READ_WIDTH_A (0), // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  .READ_WIDTH_B (0), // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  .SIM_COLLISION_CHECK ("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
  // "GENERATE_X_ONLY" or "NONE"
  .SRVAL_A(36'h00000000), // Set/Reset value for port A output
  .SRVAL_B(36'h00000000), // Set/Reset value for port B output
  .WRITE_MODE_A("WRITE_FIRST"), // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
  .WRITE_MODE_B("WRITE_FIRST"), // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
  .WRITE_WIDTH_A(0), // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  .WRITE_WIDTH_B(0), // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

```
// The next set of INIT_xx are valid when configured as 36Kb
```

35

```
.INITP_OD(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_OE(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_OF(256'h0000000000000000000000000000000000000000000000000000000000000000)
) BRAM_TDP_MACRO_inst (
.DOA(DOA),           // Output port-A data, width defined by READ_WIDTH_A parameter
.DOB DOB),           // Output port-B data, width defined by READ_WIDTH_B parameter
.ADDRA(ADDRA),        // Input port-A address, width defined by Port A depth
.ADDRB(ADDRB),        // Input port-B address, width defined by Port B depth
.CLKA(CLKA),          // 1-bit input port-A clock
.CLKB(CLKB),          // 1-bit input port-B clock
.DIA(DIA),            // Input port-A data, width defined by WRITE_WIDTH_A parameter
.DIB(DIB),            // Input port-B data, width defined by WRITE_WIDTH_B parameter
.ENA(ENA),             // 1-bit input port-A enable
.ENB(ENB),             // 1-bit input port-B enable
.REGCEA(REGCEA),      // 1-bit input port-A output register enable
.REGCEB(REGCEB),      // 1-bit input port-B output register enable
.RSTA(RSTA),          // 1-bit input port-A reset
.RSTB(RSTB),          // 1-bit input port-B reset
.WEA(WEA),             // Input port-A write enable, width defined by Port A depth
.WEB(WEB),             // Input port-B write enable, width defined by Port B depth
);

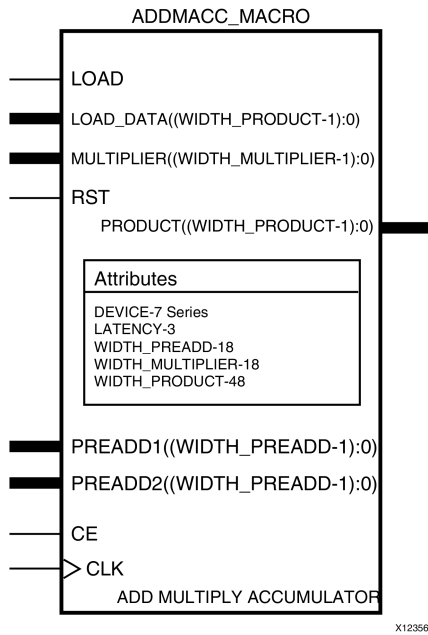
// End of BRAM_TDP_MACRO_inst instantiation
```

詳細情報

## 7 シリーズ 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
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
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
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、インスタンス化のみが可能です。

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_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	"7SERIES"	"7SERIES"	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;

-- ADDMACC_MACRO: Add and Multiple Accumulate Function implemented in a DSP48E
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

ADDMACC_MACRO_inst : ADDMACC_MACRO
generic map (
    DEVICE => "7SERIES", -- Target Device: "7SERIES", "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
//                7 Series
// Xilinx HDL Libraries Guide, version 14.4

ADDMACC_MACRO #(
    .DEVICE("7SERIES"),    // Target Device: "VIRTEX6", "SPARTAN6", "7SERIES"
    .LATENCY(4),           // Desired clock cycle latency, 0-4
    .WIDTH_PREADD(25),     // Pre-adder input width, 1-25
    .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

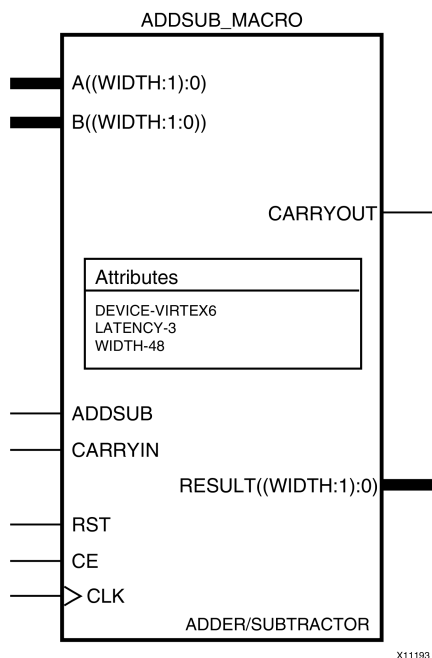
```

### 詳細情報

[7 シリーズ 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
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
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、インスタンス化のみが可能です。



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	"7SERIES"	"7SERIES"	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 のポート幅を指定します。B および RESULT ポート幅は、ほかのパラメーター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
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

ADDSUB_MACRO_inst : ADDSUB_MACRO
generic map (
    DEVICE => "7SERIES", -- Target Device: "VIRTEX5", "7SERIES", "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
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

ADDSUB_MACRO #(
    .DEVICE("7SERIES"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6", "7SERIES"
    .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
);

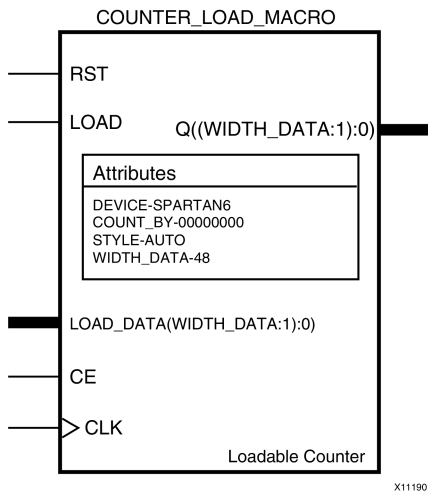
// End of ADDSUB_MACRO_inst instantiation
```

詳細情報

[7 シリーズ 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
Q	出力	可変 (WIDTH_DATA 属性を参照)	カウンタ出力
CE	入力	1	クロック イネーブル
CLK	入力	1	クロック
LOAD	入力	可変 (WIDTH_DATA 属性を参照)	アサートすると、カウンタに 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、インスタンス化のみが可能です。

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	"7SERIES"	"7SERIES"	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
--                      7 Series
-- Xilinx HDL Libraries Guide, version 14.4

COUNTER_LOAD_MACRO_inst : COUNTER_LOAD_MACRO
generic map (
    COUNT_BY => X"000000000001", -- Count by value
    DEVICE => "7SERIES",          -- Target Device: "VIRTEX5", "7SERIES", "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
//                      7 Series
// Xilinx HDL Libraries Guide, version 14.4

COUNTER_LOAD_MACRO #(
    .COUNT_BY(48'h000000000001), // Count by value
    .DEVICE("7SERIES"),           // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6", "7SERIES"
    .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
);

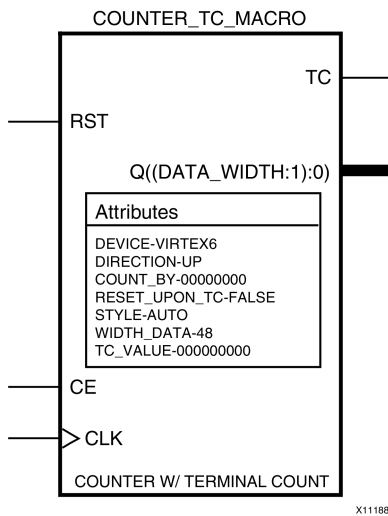
// End of COUNTER_LOAD_MACRO_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## COUNTER\_TC\_MACRO

### マクロ : Counter with Terminal Count



### 概要

COUNTER\_TC\_MACRO を使用すると、DSP48 ブロックをターミナル カウント アップ/ダウン カウンター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。出力幅、ターミナル カウント値、カウント設定、カウント方向を指定可能であり、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
TC	出力	1	TC_VALUE に達すると High になります。
Q	出力	可変 (WIDTH_DATA 属性を参照)	カウンタ出力
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、インスタネーションのみが可能です。

インスタネーション	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
RESET_UPON_TC	ブール代数	TRUE、FALSE	FALSE	ターミナル カウントに達したときにカウンタをリセットするかどうかを指定します。
DEVICE	文字列	"7SERIES"	"7SERIES"	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。
DIRECTION	文字列	"UP"、"DOWN"	"UP"	カウンタの方向を指定します。
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。
TC_VALUE	16 進数	48 ビット値	すべてゼロ	ターミナル カウント値を指定します。
WIDTH_DATA	整数	1 ~ 48	48	カウンタの幅を指定します。

## 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;

-- COUNTER_TC_MACRO: Counter with terminal count implemented in a DSP48E
--                      7 Series
-- Xilinx HDL Libraries Guide, version 14.4

COUNTER_TC_MACRO_inst : COUNTER_TC_MACRO
generic map (
    COUNT_BY => X"000000000001", -- Count by value
    DEVICE => "7SERIES",         -- Target Device: "VIRTEX5", "7SERIES"
    DIRECTION => "UP",           -- Counter direction "UP" or "DOWN"
    RESET_UPON_TC => "FALSE",    -- Reset counter upon terminal count, TRUE or FALSE
    TC_VALUE => X"000000000000", -- Terminal count value
    WIDTH_DATA => 48)           -- Counter output bus width, 1-48
port map (
    Q => Q,                      -- Counter output, width determined by WIDTH_DATA generic
    TC => TC,                    -- 1-bit terminal count output, high = terminal count is reached
    CLK => CLK,                  -- 1-bit clock input
    CE => CE,                    -- 1-bit clock enable input
    RST => RST                   -- 1-bit active high synchronous reset
);
-- End of COUNTER_TC_MACRO_inst instantiation
```

## Verilog 記述 (インスタンスーション)

```
// COUNTER_TC_MACRO: Counter with terminal count implemented in a DSP48E
//                      7 Series
// Xilinx HDL Libraries Guide, version 14.4

COUNTER_TC_MACRO #(
    .COUNT_BY(48'h000000000001), // Count by value
    .DEVICE("7SERIES"),           // Target Device: "VIRTEX5", "VIRTEX6", "7SERIES"
    .DIRECTION("UP"),             // Counter direction, "UP" or "DOWN"
    .RESET_UPON_TC("FALSE"),      // Reset counter upon terminal count, "TRUE" or "FALSE"
    .TC_VALUE(48'h000000000000), // Terminal count value
    .WIDTH_DATA(48)              // Counter output bus width, 1-48
) COUNTER_TC_MACRO_inst (
    .Q(Q),                        // Counter output bus, width determined by WIDTH_DATA parameter
    .TC(TC),                     // 1-bit terminal count output, high = terminal count is reached
    .CLK(CLK),                   // 1-bit positive edge clock input
    .CE(CE),                     // 1-bit active high clock enable input
```

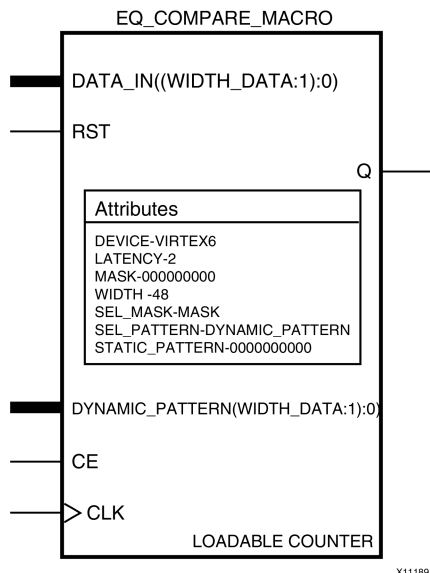
```
.RST(RST) // 1-bit active high synchronous reset  
);  
  
// End of COUNTER_TC_MACRO_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## EQ\_COMPARE\_MACRO

### マクロ : Equality Comparator



### 概要

EQ\_COMPARE\_MACRO を使用すると、DSP48 ブロックを等価コンパレータ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、マスク、および入力ソースを指定可能であり、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
Q	出力	1	アクティブ High のパターン検出で、MASK 属性でマスクした DYNAMIC_PATTERN の値と DATA_IN の値が一致したときに High になります。結果は P と同じクロック サイクルで出力されます。
DATA_IN	入力	可変 (WIDTH 属性で指定した値)	比較する入力データ
DYNAMIC_PATTERN	入力	可変 (WIDTH 属性で指定した値)	DATA_IN と比較するダイナミック データ
CLK	入力	1	クロック
CE	入力	1	クロック イネーブル
RST	入力	1	同期リセット

### デザインの入力方法

この UniMacro は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、インスタンス化のみが可能です。

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	"7SERIES"	"7SERIES"	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。
SEL_PATTERN	整数	1 ~ 24	24	PREADD1 および PREADD2 入力の幅を指定します。
MASK	16 進数	48 ビット値	すべてゼロ	パターン検出器で使用するマスクを指定します。
STATIC_PATTERN	16 進数	48 ビット値	すべてゼロ	パターン検出器で使用するパターンを指定します。
SEL_MASK	文字列	"MASK"、 "DYNAMIC_PATTERN"	"MASK"	パターン検出器のマスクに MASK を使用するか、C 入力を使用するかを指定します。
WIDTH	整数	1 ~ 48	48	DATA_IN および DYNAMIC_PATTERN の幅を指定します。
LATENCY	整数	0、1、2、3	2	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 : QREG == 1</li> <li>・ 2 : AREG == BREG == CREG == QREG == 1</li> <li>・ 3 : AREG == BREG == 2 and CREG == QREG == 1</li> </ul>

## 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;

-- EQ_COMPARE_MACRO: Equality Comparator implemented in a DSP48E
--                      7 Series
-- Xilinx HDL Libraries Guide, version 14.4

EQ_COMPARE_MACRO_inst : EQ_COMPARE_MACRO
generic map (
    DEVICE => "7SERIES",           -- Target Device: "VIRTEX5", "7SERIES"
    LATENCY => 2,                   -- Desired clock cycle latency, 0-2
    MASK => X"000000000000",       -- Select bits to be masked, must set
    SEL_MASK => "MASK",            -- SEL_MASK = "MASK"
    SEL_MASK => "MASK",            -- "MASK" = use MASK generic,
    SEL_PATTERN => "DYNAMIC_PATTERN", -- "DYNAMIC_PATTERN" = use DYNAMIC_PATTERN input bus
    STATIC_PATTERN => X"000000000000", -- Specify static pattern,
    WIDTH => 48)                   -- must set SEL_PATTERN = "STATIC_PATTERN"
    -- Comparator output bus width, 1-48
port map (
    Q => Q,                         -- 1-bit output indicating a match
    CE => CE,                       -- 1-bit active high input clock enable input
    CLK => CLK,                     -- 1-bit positive edge clock input
    DATA_IN => DATA_IN,           -- Input Data Bus, width determined by WIDTH generic
    DYNAMIC_PATTERN => DYNAMIC_PATTERN, -- Input Dynamic Match/Mask Bus, width determined by WIDTH generic
    RST => RST                      -- 1-bit input active high reset
);

```

```
-- End of EQ_COMPARE_MACRO_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// EQ_COMPARE_MACRO: Equality Comparator implemented in a DSP48E
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

EQ_COMPARE_MACRO #(
    .DEVICE("7SERIES"),           // Target Device: "VIRTEX5", "VIRTEX6", "7SERIES"
    .LATENCY(2),                  // Desired clock cycle latency, 0-2
    .MASK(48'h000000000000),      // Select bits to be masked, must set SEL_MASK="MASK"
    .SEL_MASK("MASK"),            // "MASK" = use MASK parameter,
                                // "DYNAMIC_PATTERN" = use DYNAMIC_PATTERN input bus
    .SEL_PATTERN("STATIC_PATTERN"), // "STATIC_PATTERN" = use STATIC_PATTERN parameter,
                                // "DYNAMIC_PATTERN" = use DYNAMIC_PATTERN input bus
    .STATIC_PATTERN(48'h000000000000), // Specify static pattern, must set SEL_PATTERN = "STATIC_PATTERN"
    .WIDTH(48)                    // Comparator output bus width, 1-48
) EQ_COMPARE_MACRO_inst (
    .Q(Q),                        // 1-bit output indicating a match
    .CE(CE),                      // 1-bit active high input clock enable
    .CLK(CLK),                    // 1-bit positive edge clock input
    .DATA_IN(DATA_IN),            // Input Data Bus, width determined by WIDTH parameter
    .DYNAMIC_PATTERN(DYNAMIC_PATTERN), // Input Dynamic Match/Mask Bus, width determined by WIDTH parameter
    .RST(RST)                     // 1-bit input active high reset
);

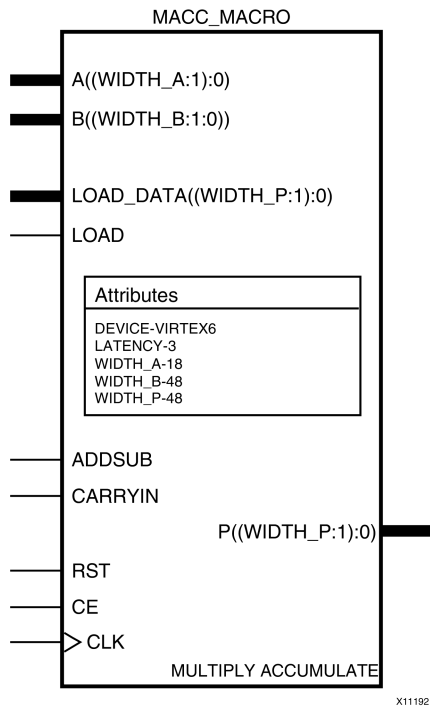
// End of EQ_COMPARE_MACRO_inst instantiation
```

詳細情報

[7 シリーズ 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
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
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。
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、インスタンス化のみが可能です。

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	"7SERIES"	"7SERIES"	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。
WIDTH_A	整数	1 ~ 25	25	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>

## 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;

-- MACC_MACRO: Multiple Accumulate Function implemented in a DSP48E
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

MACC_MACRO_inst : MACC_MACRO
generic map (
  DEVICE => "7SERIES", -- Target Device: "VIRTEX5", "7SERIES", "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 (インスタンスエーション)

```
// MACC_MACRO: Multiply Accumulate Function implemented in a DSP48E
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

MACC_MACRO #(
    .DEVICE("7SERIES"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6", "7SERIES"
    .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
) 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
);

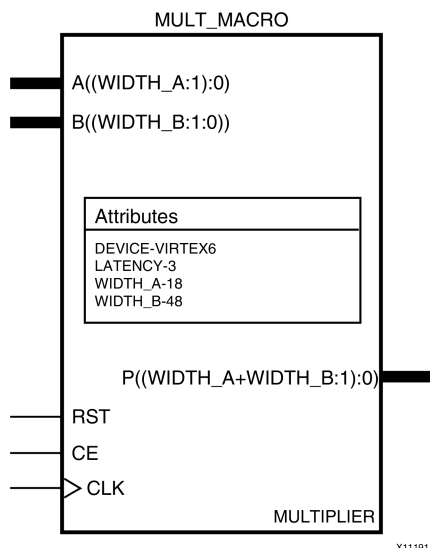
// End of MACC_MACRO_inst instantiation
```

### 詳細情報

[7 シリーズ 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、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
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
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、インスタンス化のみが可能です。

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	"7SERIES"	"7SERIES"	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。
WIDTH_A	整数	1 ~ 25	25	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>

## 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;

-- MULT_MACRO: Multiply Function implemented in a DSP48E
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

MULT_MACRO_inst : MULT_MACRO
generic map (
  DEVICE => "7SERIES",    -- Target Device: "VIRTEX5", "7SERIES", "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
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

MULT_MACRO #(
    .DEVICE("7SERIES"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6", "7SERIES"
    .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
) 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
```

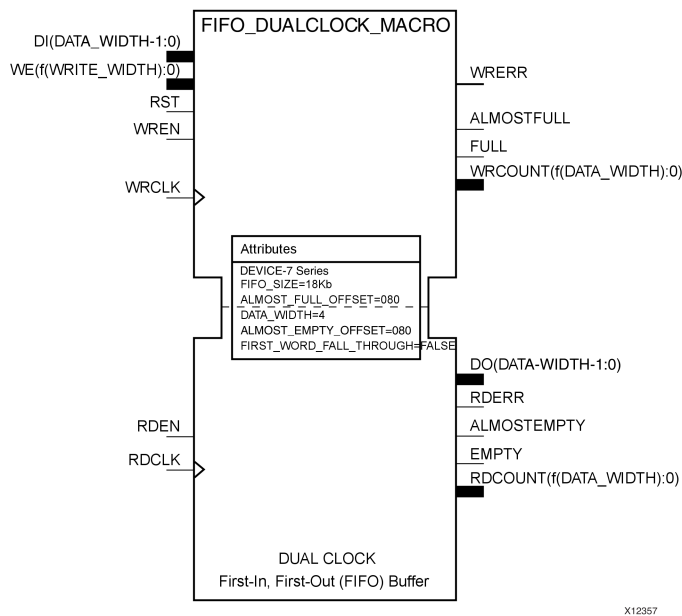
詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## FIFO\_DUALCLOCK\_MACRO

マクロ : Dual Clock First-In, First-Out (FIFO) RAM Buffer



X12357

### 概要

FPGA デバイスにはブロック RAM が数個含まれ、36kb または 18kb の汎用 RAM/ROM としてコンフィギュレーションできます。ブロック RAM に含まれる専用ロジックで FIFO を簡単にインプリメントできます。FIFO は 18Kb または 36Kb メモリとしてコンフィギュレーションできます。この UniMacro を使用すると、読み出しと書き込みに独立したクロックを使用するように FIFO がコンフィギュレーションされます。データは、読み出しクロックの立ち上がりエッジで FIFO から読み出され、書き込みクロックの立ち上がりエッジで FIFO に書き込まれます。

読み出しクロック エッジと書き込みクロック エッジ間のオフセットによっては、EMPTY、ALMOSTEMPTY、FULL、ALMOSTFULL フラグが 1 サイクル後にディアサートされることがあります。クロックが非同期のため、シミュレーションモデルではユーザー ガイドに示されているディアサートレイテンシ サイクルのみが反映されます。

### ポートの説明

ポート名	方向	幅	機能
ALMOSTEMPTY	出力	1	FIFO のほぼすべての有効エントリが読み出されていることを示します。
ALMOSTFULL	出力	1	FIFO メモリのほぼすべてのエントリがフルであることを示します。
DO	出力	「ポートのコンフィギュレーション」の表を参照	ADDR で指定されたデータ出力バス
EMPTY	出力	1	FIFO が空であることを示します。
FULL	出力	1	FIFO メモリのすべてのエントリがフルであることを示します。
RDCOUNT	出力	「ポートのコンフィギュレーション」の表を参照	FIFO データ読み出しポインター

ポート名	方向	幅	機能
RDERR	出力	1	FIFO が空のときに読み出しを行うとアサートされます。
WRCOUNT	出力	「ポートのコンフィギュレーション」の表を参照	FIFO データ書き込みポインター
WRERR	出力	1	FIFO がフルのときに書き込みを行うとアサートされます。
DI	入力	「ポートのコンフィギュレーション」の表を参照	ADDR で指定されたデータ入力バス
RDCLK	入力	1	読み出しクロック
RDEN	入力	1	リード イネーブル
RST	入力	1	非同期リセット
WRCLK	入力	1	書き込みクロック
WREN	入力	1	ライト イネーブル

## ポートのコンフィギュレーション

この UniMacro は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、インスタンスレーションのみが可能です。デザイン要件を満たすには、この表を参照して正しくコンフィギュレーションしてください。

DATA_WIDTH	FIFO_SIZE	WRCOUNT	RDCOUNT
72 ~ 37	36Kb	9	9
36 ~ 19	36Kb	10	10
	18Kb	9	9
18 ~ 10	36Kb	11	11
	18Kb	10	10
9 ~ 5	36Kb	12	12
	18Kb	11	11
1 ~ 4	36Kb	13	13
	18Kb	12	12

## デザインの入力方法

この UniMacro は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、インスタンスレーションのみが可能です。デザイン要件を満たすには、「ポートのコンフィギュレーション」の表を参照して正しくコンフィギュレーションしてください。

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
ALMOST_EMPTY_OFFSET	16 進数	13 ビット値	すべてゼロ	EMPTY と ALMOSTEMPTY の差を設定します。16 進数で指定します。
ALMOST_FULL_OFFSET	16 進数	13 ビット値	すべてゼロ	FULL と ALMOSTFULL の差を設定します。16 進数で指定します。
DATA_WIDTH	整数	1 ~ 72	4	DI/DO バスの幅を指定します。
DEVICE	文字列	"7SERIES"	"7SERIES"	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。
FIFO_SIZE	文字列	"18Kb"、 "36Kb"	"18Kb"	FIFO を 18Kb または 36Kb メモリとしてコンフィギュレーションします。
FIRST_WORD_FALL_THROUGH	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、RDEN をアサートしなくても、空の FIFO に書き込まれた最初のワードが出力されます。

## 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;

-- FIFO_DUALCLOCK_MACRO: Dual-Clock First-In, First-Out (FIFO) RAM Buffer
--                        7 Series
-- Xilinx HDL Libraries Guide, version 14.4

-- 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 | FIFO_SIZE | FIFO Depth | RDCOUNT/WRCOUNT Width --
-- =====
-- 37-72       | "36Kb"   | 512        | 9-bit                 --
-- 19-36       | "36Kb"   | 1024       | 10-bit                --
-- 19-36       | "18Kb"   | 512        | 9-bit                 --
-- 10-18       | "36Kb"   | 2048       | 11-bit                --
-- 10-18       | "18Kb"   | 1024       | 10-bit                --
-- 5-9         | "36Kb"   | 4096       | 12-bit                --
-- 5-9         | "18Kb"   | 2048       | 11-bit                --
-- 1-4         | "36Kb"   | 8192       | 13-bit                --
-- 1-4         | "18Kb"   | 4096       | 12-bit                --
-----

FIFO_DUALCLOCK_MACRO_inst : FIFO_DUALCLOCK_MACRO
generic map (
    DEVICE => "7SERIES",           -- Target Device: "VIRTEX5", "VIRTEX6", "7SERIES"
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DATA_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
    FIFO_SIZE => "18Kb",           -- Target BRAM, "18Kb" or "36Kb"
    FIRST_WORD_FALL_THROUGH => FALSE) -- Sets the FIFO FWFT to TRUE or FALSE
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit output almost empty
    ALMOSTFULL => ALMOSTFULL,   -- 1-bit output almost full
    DO => DO,                   -- Output data, width defined by DATA_WIDTH parameter
    EMPTY => EMPTY,             -- 1-bit output empty
    FULL => FULL,               -- 1-bit output full
    RDCOUNT => RDCOUNT,          -- Output read count, width determined by FIFO depth
    RDERR => RDERR,             -- 1-bit output read error
    WRCOUNT => WRCOUNT,         -- Output write count, width determined by FIFO depth
    WRERR => WRERR,            -- 1-bit output write error

```

```

DI => DI,                -- Input data, width defined by DATA_WIDTH parameter
RDCLK => RDCLK,          -- 1-bit input read clock
RDEN => RDEN,            -- 1-bit input read enable
RST => RST,              -- 1-bit input reset
WRCLK => WRCLK,          -- 1-bit input write clock
WREN => WREN,            -- 1-bit input write enable
);
-- End of FIFO_DUALCLOCK_MACRO_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```

// FIFO_DUALCLOCK_MACRO: Dual Clock First-In, First-Out (Fifor) RAM Buffer
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

////////////////////////////////////
// DATA_WIDTH | FIFO_SIZE | Fifor Depth | RDCOUNT/WRCOUNT Width //
// =====|=====|=====|=====//
// 37-72      | "36Kb"    | 512      | 9-bit      //
// 19-36      | "36Kb"    | 1024     | 10-bit     //
// 19-36      | "18Kb"    | 512      | 9-bit      //
// 10-18      | "36Kb"    | 2048     | 11-bit     //
// 10-18      | "18Kb"    | 1024     | 10-bit     //
// 5-9        | "36Kb"    | 4096     | 12-bit     //
// 5-9        | "18Kb"    | 2048     | 11-bit     //
// 1-4        | "36Kb"    | 8192     | 13-bit     //
// 1-4        | "18Kb"    | 4096     | 12-bit     //
////////////////////////////////////

FIFO_DUALCLOCK_MACRO #(
    .ALMOST_EMPTY_OFFSET(9'h080), // Sets the almost empty threshold
    .ALMOST_FULL_OFFSET(9'h080),  // Sets almost full threshold
    .DATA_WIDTH(0),               // Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
    .DEVICE("7SERIES"),           // Target device: "VIRTEX5", "VIRTEX6", "7SERIES"
    .FIFO_SIZE("18Kb"),           // Target BRAM: "18Kb" or "36Kb"
    .FIRST_WORD_FALL_THROUGH("FALSE") // Sets the Fifor FWFT to "TRUE" or "FALSE"
) FIFO_DUALCLOCK_MACRO_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit output almost empty
    .ALMOSTFULL(ALMOSTFULL),   // 1-bit output almost full
    .DO(DO),                   // Output data, width defined by DATA_WIDTH parameter
    .EMPTY(EMPTY),             // 1-bit output empty
    .FULL(FULL),               // 1-bit output full
    .RDCOUNT(RDCOUNT),         // Output read count, width determined by Fifor depth
    .RDERR(RDERR),             // 1-bit output read error
    .WRCOUNT(WRCOUNT),         // Output write count, width determined by Fifor depth
    .WRERR(WRERR),             // 1-bit output write error
    .DI(DI),                   // Input data, width defined by DATA_WIDTH parameter
    .RDCLK(RDCLK),             // 1-bit input read clock
    .RDEN(RDEN),               // 1-bit input read enable
    .RST(RST),                 // 1-bit input reset
    .WRCLK(WRCLK),             // 1-bit input write clock
    .WREN(WREN),               // 1-bit input write enable
);

// End of FIFO_DUALCLOCK_MACRO_inst instantiation

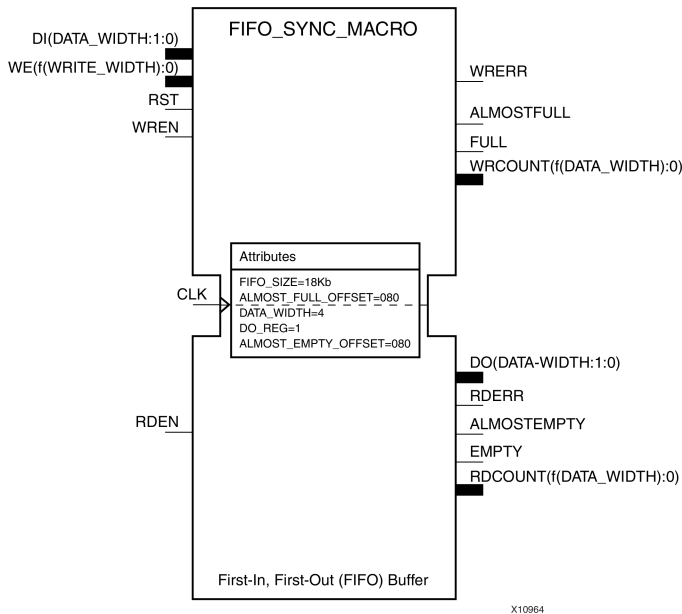
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## FIFO\_SYNC\_MACRO

### マクロ : Synchronous First-In, First-Out (FIFO) RAM Buffer



### 概要

FPGA デバイスにはブロック RAM が数個含まれ、36kb または 18kb の汎用 RAM/ROM としてコンフィギュレーションできます。ブロック RAM に含まれる専用ロジックで FIFO を簡単にインプリメントできます。FIFO は 18Kb または 36Kb メモリとしてコンフィギュレーションできます。この UniMacro を使用すると、読み出しおよび書き込みに 1 つのクロックを使用するように FIFO がコンフィギュレーションされます。

### ポートの説明

ポート名	方向	幅	機能
ALMOSTEMPTY	出力	1	FIFO のほぼすべての有効エントリが読み出されていることを示します。
ALMOSTFULL	出力	1	FIFO メモリのほぼすべてのエントリがフルであることを示します。
DO	出力	「ポートのコンフィギュレーション」の表を参照	ADDR で指定されたデータ出力バス
EMPTY	出力	1	FIFO が空であることを示します。
FULL	出力	1	FIFO メモリのすべてのエントリがフルであることを示します。
RDCOUNT	出力	「ポートのコンフィギュレーション」の表を参照	FIFO データ読み出しポインタ
RDERR	出力	1	FIFO が空のときに読み出しを行うとアサートされます。
WRCOUNT	出力	「ポートのコンフィギュレーション」の表を参照	FIFO データ書き込みポインタ
WRERR	出力	1	FIFO がフルのときに書き込みを行うとアサートされます。

ポート名	方向	幅	機能
CLK	入力	1	読み出し/書き込みクロック
DI	入力	「ポートのコンフィギュレーション」の表を参照	ADDR で指定されたデータ入力バス
RDEN	入力	1	リード イネーブル
RST	入力	1	非同期リセット
WREN	入力	1	ライト イネーブル

## ポートのコンフィギュレーション

この UniMacro は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、インスタンス化のみが可能です。デザイン要件を満たすには、この表を参照して正しくコンフィギュレーションしてください。

DATA_WIDTH	FIFO_SIZE	WRCOUNT	RDCOUNT
72 ~ 37	36Kb	9	9
36 ~ 19	36Kb	10	10
	18Kb	9	9
18 ~ 10	36Kb	11	11
	18Kb	10	10
9 ~ 5	36Kb	12	12
	18Kb	11	11
1 ~ 4	36Kb	13	13
	18Kb	12	12

## デザインの入力方法

この UniMacro は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、インスタンス化のみが可能です。デザイン要件を満たすには、「ポートのコンフィギュレーション」の表を参照して正しくコンフィギュレーションしてください。

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
ALMOST_EMPTY_OFFSET	16 進数	13 ビット値	すべてゼロ	EMPTY と ALMOSTEMPTY の差を設定します。16 進数で指定します。
ALMOST_FULL_OFFSET	16 進数	13 ビット値	すべてゼロ	FULL と ALMOSTFULL の差を設定します。16 進数で指定します。
DATA_WIDTH	整数	1 ~ 72	4	DI/DO バスの幅を指定します。
DEVICE	文字列	"7SERIES"	"7SERIES"	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。
DO_REG	2 進数	0、1	1	同期 FIFO の標準操作を実行する場合は、0 に設定します。  1 に設定すると、同期 FIFO の出力にパイプラインレジスタが追加されます。このためデータに 1 クロックサイクルのレイテンシが発生しますが、clock-to-out タイミングは改善されます。
FIFO_SIZE	文字列	"18Kb"、 "36Kb"	"18Kb"	FIFO を 18Kb または 36Kb メモリとしてコンフィギュレーションします。

## 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;

-- FIFO_SYNC_MACRO: Synchronous First-In, First-Out (FIFO) RAM Buffer
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

-- 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 | FIFO_SIZE | FIFO Depth | RDCOUNT/WRCOUNT Width --
-- -----
-- 37-72      | "36Kb"   | 512        | 9-bit                  --
-- 19-36      | "36Kb"   | 1024       | 10-bit                 --
-- 19-36      | "18Kb"   | 512        | 9-bit                  --
-- 10-18      | "36Kb"   | 2048       | 11-bit                 --
-- 10-18      | "18Kb"   | 1024       | 10-bit                 --
-- 5-9        | "36Kb"   | 4096       | 12-bit                 --
-- 5-9        | "18Kb"   | 2048       | 11-bit                 --
-- 1-4        | "36Kb"   | 8192       | 13-bit                 --
-- 1-4        | "18Kb"   | 4096       | 12-bit                 --
-----

FIFO_SYNC_MACRO_inst : FIFO_SYNC_MACRO
generic map (
    DEVICE => "7SERIES",          -- Target Device: "VIRTEX5", "VIRTEX6", "7SERIES"
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DATA_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
    FIFO_SIZE => "18Kb"          -- Target BRAM, "18Kb" or "36Kb"
)
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit output almost empty
    ALMOSTFULL => ALMOSTFULL,   -- 1-bit output almost full
    DO => DO,                    -- Output data, width defined by DATA_WIDTH parameter
    EMPTY => EMPTY,              -- 1-bit output empty

```

```

FULL => FULL,           -- 1-bit output full
RDCOUNT => RDCOUNT,      -- Output read count, width determined by FIFO depth
RDERR => RDERR,          -- 1-bit output read error
WRCOUNT => WRCOUNT,      -- Output write count, width determined by FIFO depth
WRERR => WRERR,          -- 1-bit output write error
CLK => CLK,              -- 1-bit input clock
DI => DI,                 -- Input data, width defined by DATA_WIDTH parameter
RDEN => RDEN,             -- 1-bit input read enable
RST => RST,               -- 1-bit input reset
WREN => WREN             -- 1-bit input write enable
);
-- End of FIFO_SYNC_MACRO_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```

// FIFO_SYNC_MACRO: Synchronous First-In, First-Out (Fifor) RAM Buffer
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

////////////////////////////////////
// DATA_WIDTH | FIFO_SIZE | Fifor Depth | RDCOUNT/WRCOUNT Width //
// =====|=====|=====|=====//
// 37-72 | "36Kb" | 512 | 9-bit //
// 19-36 | "36Kb" | 1024 | 10-bit //
// 19-36 | "18Kb" | 512 | 9-bit //
// 10-18 | "36Kb" | 2048 | 11-bit //
// 10-18 | "18Kb" | 1024 | 10-bit //
// 5-9 | "36Kb" | 4096 | 12-bit //
// 5-9 | "18Kb" | 2048 | 11-bit //
// 1-4 | "36Kb" | 8192 | 13-bit //
// 1-4 | "18Kb" | 4096 | 12-bit //
////////////////////////////////////

FIFO_SYNC_MACRO #(
    .DEVICE("7SERIES"), // Target Device: "VIRTEX5", "VIRTEX6", "7SERIES"
    .ALMOST_EMPTY_OFFSET(9'h080), // Sets the almost empty threshold
    .ALMOST_FULL_OFFSET(9'h080), // Sets almost full threshold
    .DATA_WIDTH(0), // Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
    .DO_REG(0), // Optional output register (0 or 1)
    .FIFO_SIZE ("18Kb") // Target BRAM: "18Kb" or "36Kb"
) FIFO_SYNC_MACRO_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit output almost empty
    .ALMOSTFULL(ALMOSTFULL), // 1-bit output almost full
    .DO(DO), // Output data, width defined by DATA_WIDTH parameter
    .EMPTY(EMPTY), // 1-bit output empty
    .FULL(FULL), // 1-bit output full
    .RDCOUNT(RDCOUNT), // Output read count, width determined by Fifor depth
    .RDERR(RDERR), // 1-bit output read error
    .WRCOUNT(WRCOUNT), // Output write count, width determined by Fifor depth
    .WRERR(WRERR), // 1-bit output write error
    .CLK(CLK), // 1-bit input clock
    .DI(DI), // Input data, width defined by DATA_WIDTH parameter
    .RDEN(RDEN), // 1-bit input read enable
    .RST(RST), // 1-bit input reset
    .WREN(WREN) // 1-bit input write enable
);

// End of FIFO_SYNC_MACRO_inst instantiation

```

詳細情報

[7 シリーズ 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
GTPE2_CHANNEL	プリミティブ : Gigabit Transceiver
GTPE2_COMMON	プリミティブ : Gigabit Transceiver
GTHE2_CHANNEL	プリミティブ : Gigabit Transceiver
GTHE2_COMMON	プリミティブ : Gigabit Transceiver
GTXE2_CHANNEL	プリミティブ : Gigabit Transceiver for 7 Series
GTXE2_COMMON	プリミティブ : Gigabit Transceiver for 7 Series
XADC	プリミティブ : Dual 12-Bit 1MSPS Analog-to-Digital Converter

### 演算ファンクション

デザイン エLEMENT	説明
DSP48E1	プリミティブ : 48-bit Multi-Functional 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 Simple 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
BUFGCTRL	プリミティブ : Global Clock Control Buffer
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
BUFGMUX_CTRL	プリミティブ : 2-to-1 Global Clock MUX Buffer
BUFH	プリミティブ : HROW Clock Buffer for a Single Clocking Region
BUFHCE	プリミティブ : HROW Clock Buffer for a Single Clocking Region with Clock Enable
BUFIO	プリミティブ : Local Clock Buffer for I/O
BUFMR	プリミティブ : Multi-Region Clock Buffer
BUFMRCE	プリミティブ : Multi-Region Clock Buffer with Clock Enable
BUFR	プリミティブ : Regional Clock Buffer for I/O and Logic Resources within a Clock Region
MMCME2_ADV	プリミティブ : Advanced Mixed Mode Clock Manager
MMCME2_BASE	プリミティブ : Base Mixed Mode Clock Manager
PLLE2_ADV	プリミティブ : Advanced Phase Locked Loop (PLL)
PLLE2_BASE	プリミティブ : Base Phase Locked Loop (PLL)

## 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
BSCANE2	プリミティブ : Boundary-Scan User Instruction
CAPTUREE2	プリミティブ : Register Capture
DNA_PORT	プリミティブ : Device DNA Access Port
EFUSE_USR	プリミティブ : 32-bit non-volatile design ID
FRAME_ECCE2	プリミティブ : Configuration Frame Error Correction
ICAPE2	プリミティブ : Internal Configuration Access Port
STARTUPE2	プリミティブ : STARTUP Block
USR_ACCESSE2	プリミティブ : Configuration Data Access

## I/O コンポーネント

デザイン エLEMENT	説明
DCIRESET	プリミティブ : Digitally Controlled Impedance Reset Component
IBUF	プリミティブ : Input Buffer

デザイン エLEMENT	説明
IBUF_IBUFDISABLE	プリミティブ : Single-ended Input Buffer with Input Disable
IBUF_INTERMDISABLE	プリミティブ : Single-ended Input Buffer with Input Termination Disable and Input Disable
IBUFDS	プリミティブ : Differential Signaling Input Buffer
IBUFDS_DIFF_OUT	プリミティブ : Differential Signaling Input Buffer With Differential Output
IBUFDS_DIFF_OUT_IBUFDISABLE	プリミティブ : Input Differential Buffer with Input Disable and Differential Output
IBUFDS_DIFF_OUT_INTERMDISABLE	プリミティブ : Input Differential Buffer with Input Termination Disable, Input Disable, and Differential Output
IBUFDS_GTE2	プリミティブ : Gigabit Transceiver Buffer
IBUFDS_IBUFDISABLE	プリミティブ : Input Differential Buffer with Input Path Disable
IBUFDS_INTERMDISABLE	プリミティブ : Input Differential Buffer with Input Termination Disable and Input Disable
IBUFG	プリミティブ : Dedicated Input Clock Buffer
IBUFGDS	プリミティブ : Differential Signaling Dedicated Input Clock Buffer
IBUFGDS_DIFF_OUT	プリミティブ : Differential Signaling Dedicated Input Clock Buffer with Differential Output
IDELAYCTRL	プリミティブ : IDELAYE2/ODELAYE2 Tap Delay Value Control
IDELAYE2	プリミティブ : Input Fixed or Variable Delay Element
IN_FIFO	プリミティブ : Input First-In, First-Out (FIFO)
IOBUF	プリミティブ : Bi-Directional Buffer
IOBUF_DCIN	プリミティブ : Bi-Directional Single-ended Buffer with DCI and Input Disable.
IOBUF_INTERMDISABLE	プリミティブ : Bi-Directional Single-ended Buffer with Input Termination Disable and Input Path Disable
IOBUFDS	プリミティブ : 3-State Differential Signaling I/O Buffer with Active Low Output Enable
IOBUFDS_DCIN	プリミティブ : Bi-Directional Differential Buffer with DCI Enable/Disable and Input Disable
IOBUFDS_DIFF_OUT	プリミティブ : Differential Bi-directional Buffer with Differential Output
IOBUFDS_DIFF_OUT_DCIN	プリミティブ : Bi-Directional Differential Buffer with DCI Disable, Input Disable, and Differential Output
IOBUFDS_DIFF_OUT_INTERMDISABLE	プリミティブ : Bi-Directional Differential Buffer with Input Termination Disable, Input Disable, and Differential Output
IOBUFDS_INTERMDISABLE	プリミティブ : Bi-Directional Differential Buffer with Input Termination Disable and Input Disable
ISERDESE2	プリミティブ : Input SERIAL/DESerializer with bit-slip
KEEPER	プリミティブ : KEEPER Symbol
OBUF	プリミティブ : Output Buffer

デザイン エLEMENT	説明
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
OBUFDS	プリミティブ : 3-State Output Buffer with Differential Signaling, Active-Low Output Enable
ODELAYE2	プリミティブ : Output Fixed or Variable Delay Element
OSERDESE2	プリミティブ : Output SERIAL/DESerializer with bit-slip
OUT_FIFO	プリミティブ : Output First-In, First-Out (FIFO) Buffer
PHASER_IN	プリミティブ : Phaser in
PHASER_IN_PHY	プリミティブ : Phaser in phy
PHASER_OUT	プリミティブ : Phaser out.
PHASER_OUT_PHY	プリミティブ : Phaser out phy.
PHASER_REF	プリミティブ : Phaser out.
PHY_CONTROL	プリミティブ : Phaser out.
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
FIFO18E1	プリミティブ : 18Kb FIFO (First-In-First-Out) Block RAM Memory
FIFO36E1	プリミティブ : 36Kb FIFO (First-In-First-Out) Block RAM Memory
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

デザイン エLEMENT	説明
RAM64X1S_1	プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAMB18E1	プリミティブ : 18K-bit Configurable Synchronous Block RAM
RAMB36E1	プリミティブ : 36K-bit Configurable Synchronous Block RAM
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
IDDR	プリミティブ : Input Dual Data-Rate Register
IDDR_2CLK	プリミティブ : Input Dual Data-Rate Register with Dual Clock Inputs
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
ODDR	プリミティブ : Dedicated Dual Data Rate (DDR) Output Register

### 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	プリミティブ : 1-Bit Look-Up Table with General Output
LUT1_D	プリミティブ : 1-Bit Look-Up Table with Dual Output
LUT1_L	プリミティブ : 1-Bit Look-Up Table with Local Output
LUT2	プリミティブ : 2-Bit Look-Up Table with General Output
LUT2_D	プリミティブ : 2-Bit Look-Up Table with Dual Output
LUT2_L	プリミティブ : 2-Bit Look-Up Table with Local Output
LUT3	プリミティブ : 3-Bit Look-Up Table with General Output

デザイン エLEMENT	説明
LUT3_D	プリミティブ : 3-Bit Look-Up Table with Dual Output
LUT3_L	プリミティブ : 3-Bit Look-Up Table with Local Output
LUT4	プリミティブ : 4-Bit Look-Up-Table with General Output
LUT4_D	プリミティブ : 4-Bit Look-Up Table with Dual Output
LUT4_L	プリミティブ : 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

---

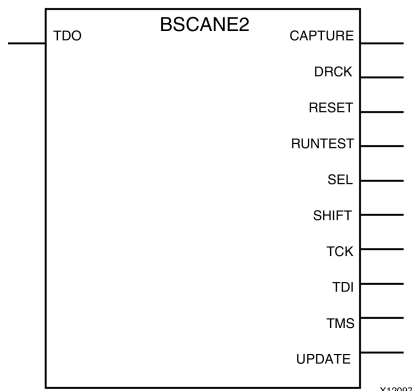
このセクションでは、Zynq™ を含む 7 シリーズ アーキテクチャで利用できるデザイン エLEMENT について説明します。デザイン エLEMENT は、アルファベット順に並べられています。

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

## BSCANE2

### プリミティブ：Boundary-Scan User Instruction



### 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できるので、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。USER 命令の 4 つすべてを処理するには、ELEMENTを 4 つインスタンス化し JTAG\_CHAIN 属性を設定します。バウンダリ スキャンの詳細は、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	説明
CAPTURE	出力	1	TAP コントローラーの CAPTURE 出力
DRCK	出力	1	ゲート付き TCK 出力。SEL がアサートされているときに CAPTURE または SHIFT がアサートされるとグルします。
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ます。
SEL	出力	1	USER 命令アクティブ出力
SHIFT	出力	1	TAP コントローラーの SHIFT 出力
TCK	出力	1	テスト クロック出力。TAP クロック ピンへのファブリック接続です。
TDI	出力	1	TAP コントローラーからのテスト データ入力 (TDI) 出力
TDO	入力	1	USER ファンクションのテスト データ出力 (TDO) 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です。
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	10 進数	1、2、3、4	1	USER コマンドの値を指定します。

## 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;

-- BSCANE2: Boundary-Scan User Instruction
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BSCANE2_inst : BSCANE2
generic map (
    JTAG_CHAIN => 1 -- Value for USER command.
)
port map (
    CAPTURE => CAPTURE, -- 1-bit output: CAPTURE output from TAP controller.
    DRCK => DRCK,        -- 1-bit output: Gated TCK output. When SEL is asserted, DRCK toggles when CAPTURE or
                        -- SHIFT are asserted.

    RESET => RESET,      -- 1-bit output: Reset output for TAP controller.
    RUNTEST => RUNTEST,  -- 1-bit output: Output asserted when TAP controller is in Run Test/Idle state.
    SEL => SEL,          -- 1-bit output: USER instruction active output.
    SHIFT => SHIFT,      -- 1-bit output: SHIFT output from TAP controller.
    TCK => TCK,          -- 1-bit output: Test Clock output. Fabric connection to TAP Clock pin.
    TDI => TDI,          -- 1-bit output: Test Data Input (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: Test Data Output (TDO) input for USER function.
);

-- End of BSCANE2_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BSCANE2: Boundary-Scan User Instruction
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

BSCANE2 #(
    .JTAG_CHAIN(1) // Value for USER command.
)
BSCANE2_inst (
    .CAPTURE(CAPTURE), // 1-bit output: CAPTURE output from TAP controller.
    .DRCK(DRCK),       // 1-bit output: Gated TCK output. When SEL is asserted, DRCK toggles when CAPTURE or
                        // SHIFT are asserted.

    .RESET(RESET),     // 1-bit output: Reset output for TAP controller.
    .RUNTEST(RUNTEST), // 1-bit output: Output asserted when TAP controller is in Run Test/Idle state.
    .SEL(SEL),         // 1-bit output: USER instruction active output.
    .SHIFT(SHIFT),     // 1-bit output: SHIFT output from TAP controller.
    .TCK(TCK),         // 1-bit output: Test Clock output. Fabric connection to TAP Clock pin.
    .TDI(TDI),         // 1-bit output: Test Data Input (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: Test Data Output (TDO) input for USER function.
);

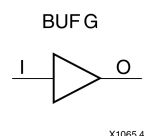
// End of BSCANE2_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFG

### プリミティブ：Global Clock Simple 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配するために、信号をグローバル配線リソースに接続します。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	説明
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	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 Simple Buffer
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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

-- End of BUFG_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFG: Global Clock Simple Buffer
//      7 Series
// Xilinx HDL Libraries Guide, version 14.4

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

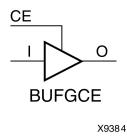
// End of BUFG_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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

### 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

### 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;

-- BUFGCE: Global Clock Buffer with Clock Enable
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFGCE_inst : BUFGCE
port map (
    O => O, -- 1-bit output: Clock output
    CE => CE, -- 1-bit input: Clock enable input for IO
    I => I -- 1-bit input: Primary clock
);

-- End of BUFGCE_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGCE: Global Clock Buffer with Clock Enable
//      7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFGCE BUFGCE_inst (
    .O(O),    // 1-bit output: Clock output
    .CE(CE),  // 1-bit input: Clock enable input for IO
    .I(I)     // 1-bit input: Primary clock
);

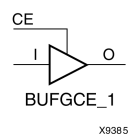
// End of BUFGCE_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 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	説明
CE	入力	1	クロック バッファ イネーブル (アクティブ High)
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	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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFGCE_1_inst : BUFGCE_1
port map (
    O => O,  -- 1-bit output: Clock output
    CE => CE, -- 1-bit input: Clock enable input for IO
    I => I   -- 1-bit input: Primary clock
);

-- 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
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFGCE_1 BUFGCE_1_inst (
    .O(O),    // 1-bit output: Clock output
    .CE(CE),  // 1-bit input: Clock enable input for IO
    .I(I)     // 1-bit input: Primary clock
);

// End of BUFGCE_1_inst instantiation
```

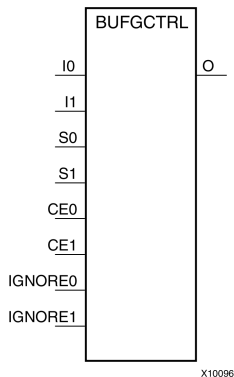
詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## BUFGCTRL

### プリミティブ：Global Clock Control Buffer



### 概要

BUFGCTRL は、2 つのクロック入力を持ち、同期/非同期のグリッチのない 2:1 マルチプレクサーとして機能する 7 シリーズのグローバル クロック バッファです。以前の FPGA に含まれるグローバル クロック バッファに比べ、制御ピンが追加されており、さまざまな機能の使用および効率的な入力の切り替えが可能です。BUFGCTRL は、クロック供給以外の用途にも使用できます。

### ポートの説明

ポート名	方向	幅	説明
CE0	入力	1	I0 クロック入力のクロック イネーブル入力。CE0 ピンを使用する場合、この入力を有効にするにはセットアップ/ホールド タイムを満たす必要があります。要件を満たさない場合、クロックでグリッチが発生する可能性があります。
CE1	入力	1	I1 クロック入力のクロック イネーブル入力。CE1 ピンを使用する場合、この入力を有効にするにはセットアップ/ホールド タイムを満たす必要があります。要件を満たさない場合、クロックでグリッチが発生する可能性があります。
IGNORE0	入力	1	I0 入力のクロック IGNORE 入力。IGNORE ピンがアサートされると、BUFGCTRL がバイパスされ、2 つのクロック入力間の切り替え条件の検出時に無視されます。つまり、IGNORE ピンのアサート時は、セレクトピンが変更されると直ちにマルチプレクサーにより入力が切り替わります。セレクトピンが変更されると、IGNORE0 ピンでは出力が I0 入力から I1 入力に切り替えられ、IGNORE1 ピンでは I1 入力から I0 入力に切り替えられます。
IGNORE1	入力	1	I1 入力のクロック IGNORE 入力。IGNORE ピンがアサートされると、BUFGCTRL がバイパスされ、2 つのクロック入力間の切り替え条件の検出時に無視されます。つまり、IGNORE ピンのアサート時は、セレクトピンが変更されると直ちにマルチプレクサーにより入力が切り替わります。セレクトピンが変更されると、IGNORE0 ピンでは出力が I0 入力から I1 入力に切り替えられ、IGNORE1 ピンでは I1 入力から I0 入力に切り替えられます。

ポート名	方向	幅	説明
I0	入力	1	CE0 入力でイネーブルになり、S0 入力で選択される BUFGCTRL へのプライマリ クロック入力
I1	入力	1	CE1 入力でイネーブルになり、S1 入力で選択される BUFGCTRL へのセカンダリ クロック入力
O	出力	1	クロック出力
S0	入力	1	I0 のクロック セレクト入力。S ピンは、各クロック入力ピンのクロック セレクト入力です。S ピンをクロック セレクト入力として使用する場合は、セットアップおよびホールド タイム要件があります。CE ピンとは異なり、要件を満たさなくてもクロック グリッチが発生することはありませんが、出力クロックがピンに現れるのが 1 クロック サイクル後になる場合があります。
S1	入力	1	I1 のクロック セレクト入力。S ピンは、各クロック入力ピンのクロック セレクト入力です。S ピンをクロック セレクト入力として使用する場合は、セットアップおよびホールド タイム要件があります。CE ピンとは異なり、要件を満たさなくてもクロック グリッチが発生することはありませんが、出力クロックがピンに現れるのが 1 クロック サイクル後になる場合があります。

## 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
INIT_OUT	10 進数	0、1	0	コンフィギュレーション後の BUFGCTRL 出力の初期値を指定します。
PRESELECT_I0	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、コンフィギュレーション後に I0 入力が出力されます。
PRESELECT_I1	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、コンフィギュレーション後に I1 入力が出力されます。

注記：2 つの PRESELECT 属性を同時に TRUE に設定することはできません。

## 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;

-- BUFGCTRL: Global Clock Control Buffer
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFGCTRL_inst : BUFGCTRL
generic map (
    INIT_OUT => 0,           -- Initial value of BUFGCTRL output ($VALUES;)
    PRESELECT_I0 => FALSE,   -- BUFGCTRL output uses I0 input ($VALUES;)
    PRESELECT_I1 => FALSE   -- BUFGCTRL output uses I1 input ($VALUES;)
)
port map (
    O => O,                 -- 1-bit output: Clock output
    CE0 => CE0,             -- 1-bit input: Clock enable input for I0
    CE1 => CE1,             -- 1-bit input: Clock enable input for I1
    I0 => I0,               -- 1-bit input: Primary clock
    I1 => I1,               -- 1-bit input: Secondary clock
    IGNORE0 => IGNORE0,     -- 1-bit input: Clock ignore input for I0
    IGNORE1 => IGNORE1,     -- 1-bit input: Clock ignore input for I1
    S0 => S0,               -- 1-bit input: Clock select for I0
    S1 => S1                -- 1-bit input: Clock select for I1
);

-- End of BUFGCTRL_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGCTRL: Global Clock Control Buffer
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFGCTRL #(
    .INIT_OUT(0),           // Initial value of BUFGCTRL output ($VALUES;)
    .PRESELECT_I0("FALSE"), // BUFGCTRL output uses I0 input ($VALUES;)
    .PRESELECT_I1("FALSE") // BUFGCTRL output uses I1 input ($VALUES;)
)
BUFGCTRL_inst (
    .O(O),                 // 1-bit output: Clock output
    .CE0(CE0),             // 1-bit input: Clock enable input for I0
    .CE1(CE1),             // 1-bit input: Clock enable input for I1
    .I0(I0),               // 1-bit input: Primary clock
    .I1(I1),               // 1-bit input: Secondary clock
    .IGNORE0(IGNORE0),     // 1-bit input: Clock ignore input for I0
    .IGNORE1(IGNORE1),     // 1-bit input: Clock ignore input for I1
    .S0(S0),               // 1-bit input: Clock select for I0
    .S1(S1)                // 1-bit input: Clock select for I1
);

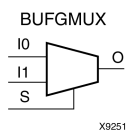
// End of BUFGCTRL_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFGMUX

### プリミティブ：Global Clock Mux Buffer



### 概要

このデザイン エレメントは、BUFGCTRL に基づくグローバル 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 と 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。

### 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	説明
I0	入力	1	クロック バッファ入力。S 入力が 0 のときに O に出力されます。
I1	入力	1	クロック バッファ入力。S 入力が 1 のときに O に出力されます。
O	出力	1	クロック バッファ出力
S	入力	1	クロック バッファ セレクト入力。Low の場合は I0 入力を選択され、High の場合は I1 入力を選択されます。

### 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;

-- BUFGMUX: Global Clock Mux Buffer
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFGMUX_inst : BUFGMUX
port map (
    O => O, -- 1-bit output: Clock output
    IO => IO, -- 1-bit input: Clock input (S=0)
    I1 => I1, -- 1-bit input: Clock input (S=1)
    S => S -- 1-bit input: Clock select
);

-- End of BUFGMUX_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGMUX: Global Clock Mux Buffer
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFGMUX #(
)
BUFGMUX_inst (
    .O(O), // 1-bit output: Clock output
    .IO(IO), // 1-bit input: Clock input (S=0)
    .I1(I1), // 1-bit input: Clock input (S=1)
    .S(S) // 1-bit input: Clock select
);

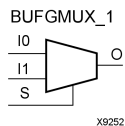
// End of BUFGMUX_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFGMUX\_1

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



### 概要

このデザイン エレメントは、BUFGCTRL に基づくグローバル 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 と 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。

### 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

### 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;

-- BUFGMUX_1: Global Clock Mux Buffer with Output State 1
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFGMUX_1_inst : BUFGMUX_1
port map (
    O => O, -- 1-bit output: Clock output
    I0 => I0, -- 1-bit input: Clock input (S=0)
    I1 => I1, -- 1-bit input: Clock input (S=1)
    S => S -- 1-bit input: Clock select
);

-- End of BUFGMUX_1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGMUX_1: Global Clock Mux Buffer with Output State 1
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFGMUX_1 #(
)
BUFGMUX_1_inst (
    .O(O),    // 1-bit output: Clock output
    .IO(I0),  // 1-bit input: Clock input (S=0)
    .I1(I1),  // 1-bit input: Clock input (S=1)
    .S(S)     // 1-bit input: Clock select
);

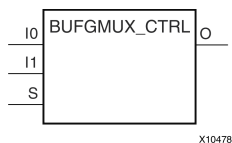
// End of BUFGMUX_1_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFGMUX\_CTRL

### プリミティブ：2-to-1 Global Clock MUX Buffer



### 概要

このデザイン エLEMENTは、2 つのクロック入力、1 つのクロック出力、セレクト入力を持つクロック バッファです。このセレクト入力は、グローバル クロック リソースを駆動する 2 つのクロックのいずれかを選択するときに使用します。このコンポーネントは BUFGCTRL に基づいており、一部のピンが High または Low に接続されています。このELEMENTは、S ピンを 2:1 マルチプレクサーのセレクトピンとして使用します。この S ピンは、バッファの出力にグリッチを発生させることなく、いつでも切り替えることができます。

### ポートの説明

ポート名	方向	幅	説明
I0	入力	1	クロック バッファ入力。S 入力が 0 のときに O に出力されます。
I1	入力	1	クロック バッファ入力。S 入力が 1 のときに O に出力されます。
O	出力	1	クロック バッファ出力
S	入力	1	クロック バッファ セレクト入力。Low の場合は I0 入力を選択され、High の場合は I1 入力を選択されます。

### 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;

-- BUFGMUX_CTRL: 2-to-1 Global Clock MUX Buffer
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFGMUX_CTRL_inst : BUFGMUX_CTRL
port map (
  O => O, -- 1-bit output: Clock output
  IO => IO, -- 1-bit input: Clock input (S=0)
  I1 => I1, -- 1-bit input: Clock input (S=1)
  S => S -- 1-bit input: Clock select
);

-- End of BUFGMUX_CTRL_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGMUX_CTRL: 2-to-1 Global Clock MUX Buffer
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFGMUX_CTRL BUFGMUX_CTRL_inst (
  .O(O), // 1-bit output: Clock output
  .IO(IO), // 1-bit input: Clock input (S=0)
  .I1(I1), // 1-bit input: Clock input (S=1)
  .S(S) // 1-bit input: Clock select
);

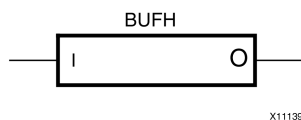
// End of BUFGMUX_CTRL_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFH

### プリミティブ：HROW Clock Buffer for a Single Clocking Region



### 概要

BUFH プリミティブは、グローバル バッファ (BUFG) リソースのクロック領域エントリ ポイントに直接アクセスできます。これにより、グローバル クロック ネットワークの未使用部分を高速、低スキューのローカル (1 つのクロック領域) 配線リソースとして使用できます。このコンポーネントの使用の詳細は、『7 シリーズ FPGA クロッキング リソース 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	説明
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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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
//      7 Series
// Xilinx HDL Libraries Guide, version 14.4

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

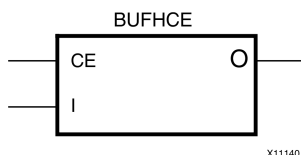
// End of BUFH_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFHCE

**プリミティブ：HROW Clock Buffer for a Single Clocking Region with Clock Enable**



### 概要

BUFHCE を使用すると、グローバル バッファ (BUFG) リソースのクロック領域エントリ ポイントに直接アクセスできます。これにより、グローバル クロック ネットワークの未使用部分を高速、低スキューのローカル (1 つのクロック領域) 配線リソースとして使用できます。また、CE (クロック イネーブル) 入力を使用して、常に使用されるわけではないデザインの回路または部分にクロック イネーブルまたはクロック ゲーティングを適用することにより、消費電力を抑えることができます。このコンポーネントの使用の詳細は、『7 シリーズ FPGA クロッキング リソース 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	説明
CE	入力	1	I から O への信号伝搬をイネーブルにします。Low の場合、出力の INIT_OUT 値への遷移でグリッチが発生しません。
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

### 使用可能な属性

属性	データ型	値	デフォルト	説明
CE_TYPE	文字列	"SYNC"、"ASYN"	"SYNC"	"SYNC" の場合、クロック イネーブルでの INIT_OUT 値への遷移または INIT_OUT 値からの遷移でグリッチが発生しません。"ASYN" の場合、クロックを停止したり、ファンアウト制御またはデータ パス配線のためにクロック バッファの代わりに BUFHCE 使用するときなど、遷移をすぐに発生させるときに使用します。
INIT_OUT	10 進数	0、1	0	初期出力値で、Low で停止するか High で停止するかも指定します。

## 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;

-- BUFHCE: HROW Clock Buffer for a Single Clocking Region with Clock Enable
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFHCE_inst : BUFHCE
generic map (
  CE_TYPE => "SYNC", -- "SYNC" (glitchless switching) or "ASYN" (immediate switch)
  INIT_OUT => 0      -- Initial output value (0-1)
)
port map (
  O => O, -- 1-bit output: Clock output
  CE => CE, -- 1-bit input: Active high enable
  I => I -- 1-bit input: Clock input
);

-- End of BUFHCE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFHCE: HROW Clock Buffer for a Single Clocking Region with Clock Enable
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFHCE #(
  .CE_TYPE("SYNC"), // "SYNC" (glitchless switching) or "ASYN" (immediate switch)
  .INIT_OUT(0)      // Initial output value (0-1)
)
BUFHCE_inst (
  .O(O), // 1-bit output: Clock output
  .CE(CE), // 1-bit input: Active high enable
  .I(I) // 1-bit input: Clock input
);

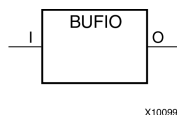
// End of BUFHCE_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFIO

### プリミティブ：Local Clock Buffer for I/O



### 概要

BUFIO は、単純な clock-in/clock-out バッファです。I/O 列の専用クロック ネットを駆動し、グローバル クロック リソースからは独立しているため、ソース同期データ キャプチャ (転送/受信クロック分配) に適しています。これらは、同じクロック領域に配置されている専用の MRCC I/O または複数クロック領域へのクロック供給を可能にする BUFMRCE/BUFMR コンポーネントで駆動できます。BUFIO では、同じバンクに含まれる I/O コンポーネントのみを駆動できます。I/O クロック ネットワークの範囲は I/O 列までなので、CLB やブロック RAM などのロジック リソースは直接駆動できません。

### ポートの説明

ポート名	方向	幅	説明
I	入力	1	クロック バッファへの入力ポート。最上位ポートに接続されている IBUFG または関連する BUFMR バッファに接続します。
O	出力	1	クロック バッファからの出力ポート。ISERDESE2、OSERDESE2、IDDR、または ODDR などの同期 I/O コンポーネントに接続するか、または推論またはインスタンス化されている I/O ポートに直接接続されているレジスタに接続します。

### 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	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;

-- BUFIO: Local Clock Buffer for I/O
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFIO_inst : BUFIO
port map (
  O => O, -- 1-bit output: Clock output (connect to I/O clock loads).
  I => I  -- 1-bit input: Clock input (connect to an IBUFG or BUFMR).
);

-- End of BUFIO_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFIO: Local Clock Buffer for I/O
//      7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFIO BUFIO_inst (
    .O(O), // 1-bit output: Clock output (connect to I/O clock loads).
    .I(I)  // 1-bit input: Clock input (connect to an IBUFG or BUFMR).
);

// End of BUFIO_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFMR

### プリミティブ：Multi-Region Clock Buffer



### 概要

BUFMR は、単純な clock-in/clock-out バッファです。BUFMR は、以前の Virtex アーキテクチャで提供されている BUFR および BUFIO の複数領域/バンク サポートに置き換わるエレメントです。各バンクには BUFMR が 2 個含まれており、バッファは、それぞれ同じバンクの特定の MRCC で駆動できます。BUFMR は、同じ領域/バンクおよび I/O クロッキング バックボーンを介した上下の領域に含まれる BUFIO と BUFR の両方またはいずれかを駆動します。バイパスモードではなくクロック分周器を使用して BUFR を駆動するときは、BUFMR の代わりに BUFMRCE を使用してください。

### ポートの説明

ポート名	方向	幅	説明
I	入力	1	BUFMR クロック入力ピン。MRCC I/O ポートに直接接続されている IBUFG 入力に接続します。
O	出力	1	BUFMR クロック出力ピン。隣接する領域で駆動される BUFIO と BUFR の両方またはいずれかに接続します。

### 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	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;

-- BUFMR: Multi-Region Clock Buffer
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFMR_inst : BUFMR
port map (
  O => O, -- 1-bit output: Clock output (connect to BUFIOs/BUFRs)
  I => I  -- 1-bit input: Clock input (Connect to IBUFG)
);

-- End of BUFMR_inst instantiation
```



## Verilog 記述 (インスタンスレーション)

```
// BUFMR: Multi-Region Clock Buffer
//      7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFMR BUFMR_inst (
    .O(O), // 1-bit output: Clock output (connect to BUFIOs/BUFRs)
    .I(I)  // 1-bit input: Clock input (Connect to IBUFG)
);

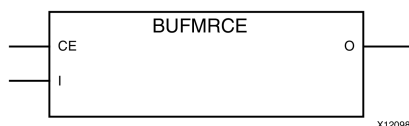
// End of BUFMR_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFMRCE

### プリミティブ：Multi-Region Clock Buffer with Clock Enable



### 概要

BUFMRCE は、クロック イネーブル (CE) が付いた単純な clock-in/clock-out バッファです。CE をアサートすると、出力クロックがユーザー指定の値で停止します。BUFMRCE は、以前の Virtex アーキテクチャで提供されている BUFR および BUFIO の複数領域/バンク サポートに置き換わるエレメントです。各バンクには BUFMRCE が 2 個含まれており、それぞれ同じバンクの特定の MRCC で駆動できます。BUFMRCE は、同じ領域/バンクおよび I/O クロッキング バックボーンを介した上下の領域に含まれる BUFIO と BUFR の両方またはいずれかを駆動します。バイパスされていない BUFR 分周器を使用する場合は、CE ピンをディアサートして BUFMRCE をディスエーブルにし、CLR をアサートして BUFR をリセットした後、CE 信号をアサートします。この順序に従うことで、すべての BUFR 出力クロックの位相が揃います。BUFR 内の分周器を使用しない場合は、この追加回路は不要です。クロック イネーブル回路が不要な場合は、BUFMRCE の代わりに BUFMR コンポーネントを使用してください。

### ポートの説明

ポート名	方向	幅	説明
CE	入力	1	アクティブ High のバッファ イネーブル入力。Low の場合、出力が INIT_OUT 値になります。
I	入力	1	BUFMR クロック入力ピン。MRCC I/O ポートに直接接続されている IBUFG 入力に接続します。
O	出力	1	BUFMR クロック出力ピン。同じまたは隣接する領域で駆動される BUFIO と BUFR の両方またはいずれかに接続します。

### 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	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
CE_TYPE	文字列	"SYNC"、"ASYN"	"SYNC"	"SYNC" に設定すると、CE が入力 I に同期し、出力にグリッチは発生しません。CE 信号のクロックが停止されていたり、またはクロックでの操作がない場合は、"ASYN" に設定します。
INIT_OUT	10 進数	0、1	0	初期出力値で、Low で停止するか High で停止するかも指定します。

## 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;

-- BUFMRCE: Multi-Region Clock Buffer with Clock Enable
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFMRCE_inst : BUFMRCE
generic map (
  CE_TYPE => "SYNC", -- SYNC, ASYNC
  INIT_OUT => 0      -- Initial output and stopped polarity, (0-1)
)
port map (
  O => O, -- 1-bit output: Clock output (connect to BUFIOs/BUFRs)
  CE => CE, -- 1-bit input: Active high buffer enable
  I => I -- 1-bit input: Clock input (Connect to IBUFG)
);

-- End of BUFMRCE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFMRCE: Multi-Region Clock Buffer with Clock Enable
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFMRCE #(
  .CE_TYPE("SYNC"), // SYNC, ASYNC
  .INIT_OUT(0)      // Initial output and stopped polarity, (0-1)
)
BUFMRCE_inst (
  .O(O), // 1-bit output: Clock output (connect to BUFIOs/BUFRs)
  .CE(CE), // 1-bit input: Active high buffer enable
  .I(I) // 1-bit input: Clock input (Connect to IBUFG)
);

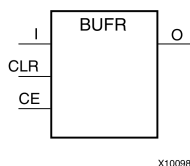
// End of BUFMRCE_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFR

**プリミティブ**：Regional Clock Buffer for I/O and Logic Resources within a Clock Region



## 概要

BUFR は、7 シリーズで利用できるリージョナル クロック バッファです。グローバル クロック ツリーからは独立しており、クロック領域内の専用クロック ネットにクロック信号を供給します。BUFR では、BUFR が配置されている領域内のリージョナル クロック ネットを駆動できます。BUFIO とは異なり、BUFR は I/O ロジックだけでなく、同じクロック領域のロジックリソース (CLB、ブロック RAM など) も駆動できます。BUFR は、IBUFG、BUFMRCE、MMCM、またはローカル インターコネクトの出力のいずれかで駆動できます。クロック入力を分周したクロックを出力することもできます。分周率は、1 ～ 8 の整数です。BUFR は、クロックドメインの切り替えやシリアルからパラレルへの変換が必要なソース同期アプリケーションに適しています。通常、1 つのクロック領域 (リージョナル クロック ネットワーク 2 つ) には BUFR が 2 つ含まれます。複数のクロック領域でローカル クロッキングが必要な場合、BUFMRCE により隣接するクロック領域の BUFR を複数駆動して、クロック機能を拡張できます。詳細は、BUFMRCE を参照してください。

## ポートの説明

ポート名	方向	幅	説明
CE	入力	1	クロック イネーブル ポート。Low になると、出力クロックがディスエーブルになります。High になると、クロックが O に出力されます。“BYPASS”モードでは使用できません。BUFR_DIVIDE を “BYPASS” に設定している場合、または使用しない場合は、VCC に接続します。
CLR	入力	1	分周クロック出力用のカウンタ非同期クリア。High になると、分周クロック出力を生成するために使用されたカウンタがリセットされ、出力が Low になります。“BYPASS”モードでは使用できません。BUFR_DIVIDE を “BYPASS” に設定している場合、または使用しない場合は、グラウンドに接続します。
I	入力	1	クロック入力ポートBUFR のクロック ソース ポートです。IBUFG、BUFMRCE、MMCM、またはローカルのインターコネクトにより駆動できます。
O	出力	1	クロック出力ポートBUFR のクロック領域に含まれるクロックトラックを駆動します。このポートは FPGA のクロックを供給するコンポーネントに接続します。

## デザインの入力方法

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	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
BUFR_DIVIDE	文字列	"BYPASS"、"1"、"2"、 "3"、"4"、"5"、"6"、"7"、 "8"	"BYPASS"	出力クロックに分周した入力クロックを使用する場合の分周率を指定します。
SIM_DEVICE	文字列	"7SERIES"	"7SERIES"	7 シリーズを使用するときは、シミュレーションが正しく動作するよう "7SERIES" に設定する必要があります。

## 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;

-- BUFR: Regional Clock Buffer for I/O and Logic Resources within a Clock Region
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

BUFR_inst : BUFR
generic map (
    BUFR_DIVIDE => "BYPASS", -- Values: "BYPASS", 1, 2, 3, 4, 5, 6, 7, 8"
    SIM_DEVICE => "7SERIES"  -- Must be set to "7SERIES"
)
port map (
    O => O,      -- 1-bit output: Clock output port
    CE => CE,    -- 1-bit input: Active high, clock enable (Divided modes only)
    CLR => CLR,  -- 1-bit input: Active high, asynchronous clear (Divided modes only)
    I => I       -- 1-bit input: Clock buffer input driven by an IBUFG, MMCM or local interconnect
);

-- End of BUFR_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFR: Regional Clock Buffer for I/O and Logic Resources within a Clock Region
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

BUFR #(
    .BUFR_DIVIDE("BYPASS"), // Values: "BYPASS", 1, 2, 3, 4, 5, 6, 7, 8"
    .SIM_DEVICE("7SERIES")  // Must be set to "7SERIES"
)
BUFR_inst (
    .O(O),      // 1-bit output: Clock output port
    .CE(CE),    // 1-bit input: Active high, clock enable (Divided modes only)
    .CLR(CLR),  // 1-bit input: Active high, asynchronous clear (Divided modes only)
    .I(I)       // 1-bit input: Clock buffer input driven by an IBUFG, MMCM or local interconnect
);

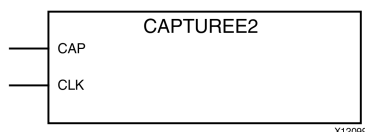
// End of BUFR_inst instantiation
```

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## CAPTUREE2

### プリミティブ：Register Capture



### 概要

このデザイン エレメントは、レジスタ（フリップフロップとラッチ）情報のキャプチャ方法およびそのタイミングを制御します。リードバック機能は、専用のコンフィギュレーション ポート命令により提供されます。このエレメントを使用しない場合は、データはコンフィギュレーション クロックに同期してリードバックされます。レジスタ（フリップフロップとラッチ）の値のみをキャプチャできます。LUT RAM、SRL、ブロック RAM の値もリードバックされますが、キャプチャできません。CAP 信号を High にアサートすると、次にクロックが Low から High に切り替わるときにデバイス内のレジスタがキャプチャされます。デフォルトでは、トリガー（CAP をアサートしているときの CLK の遷移）のたびにデータがキャプチャされます。リードバック処理を 1 回のデータ キャプチャだけに制限するには、ONESHOT 属性を "TRUE" に設定します。

### ポートの説明

ポート名	方向	幅	説明
CAP	入力	1	キャプチャ入力
CLK	入力	1	クロック入力

### 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
ONESHOT	文字列	"TRUE"、"FALSE"	"TRUE"	CAP のトリガーごとにリードバックを 1 回実行します。

## 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;

-- CAPTUREE2: Register Capture
--           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

CAPTUREE2_inst : CAPTUREE2
generic map (
    ONESHOT => "TRUE" -- Specifies the procedure for performing single readback per CAP trigger.
)
port map (
    CAP => CAP, -- 1-bit input: Capture Input
    CLK => CLK  -- 1-bit input: Clock Input
);

-- End of CAPTUREE2_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// CAPTUREE2: Register Capture
//           7 Series
// Xilinx HDL Libraries Guide, version 14.4

CAPTUREE2 #(
    .ONESHOT("TRUE") // Specifies the procedure for performing single readback per CAP trigger.
)
CAPTUREE2_inst (
    .CAP(CAP), // 1-bit input: Capture Input
    .CLK(CLK)  // 1-bit input: Clock Input
);

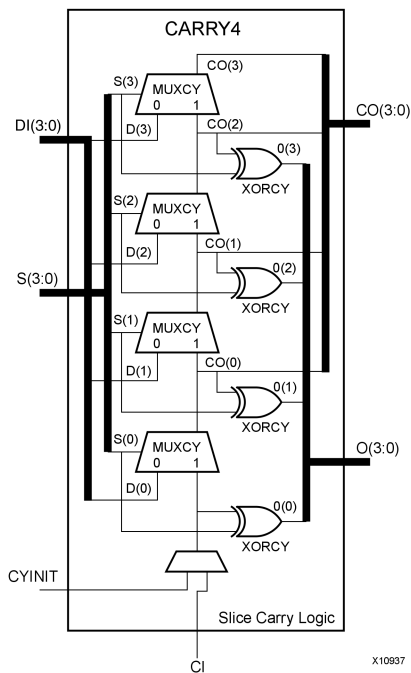
// End of CAPTUREE2_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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、幅広のコンパレータ、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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

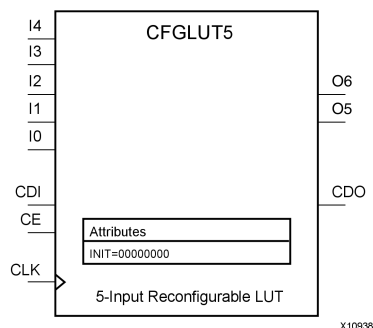
// End of CARRY4_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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、シングル シリアル チェーンのデータ (LUT に付き 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 (オプションで後続の 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 ビットをシフト入力することで、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 にシフト入力されると、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]) までシフト入力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	すべてゼロ	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;

-- CFGLUT5: Reconfigurable 5-input LUT (Mapped to SliceM LUT6)
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスレーション)

```
// CFGLUT5: Reconfigurable 5-input LUT (Mapped to a SliceM LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

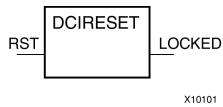
// End of CFGLUT5_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## DCIRESET

### プリミティブ：Digitally Controlled Impedance Reset Component



### 概要

このデザイン エLEMENTは、コンフィギュレーション後にデジタル制御インピーダンス (DCI) ステート マシンをリセットするために使用します。デバイスの動作中に DCIRESET プリミティブへの RST 入力をトグルすることで、DCI ステート マシンがリセットされ、インピーダンス調整の両段階が順に実行されます。DCI を使用するすべての I/O は、DCIRESET ブロックの LOCKED 出力がアサートされるまで使用できません。

### ポートの説明

ポート名	方向	幅	説明
LOCKED	出力	1	DCI ステート マシン LOCK ステータス出力。Low のときは DCI I/O インピーダンスは調整中であり、DCI I/O は使用できません。Low から High に遷移すると、DCI I/O を使用できるようになります。
RST	入力	1	DCI ステート マシンへのアクティブ High の非同期リセット入力。RST をアサートすると、LOCKED がアサートされるまで DCI を使用している I/O を使用できなくなります。

### 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	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;

-- DCIRESET: Digitally Controlled Impedance Reset Component
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

DCIRESET_inst : DCIRESET
port map (
    LOCKED => LOCKED, -- 1-bit output: LOCK status output
    RST => RST        -- 1-bit input: Active-high asynchronous reset input
);

-- End of DCIRESET_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// DCIRESET: Digitally Controlled Impedance Reset Component
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

DCIRESET DCIRESET_inst (
    .LOCKED(LOCKED), // 1-bit output: LOCK status output
    .RST(RST)        // 1-bit input: Active-high asynchronous reset input
);

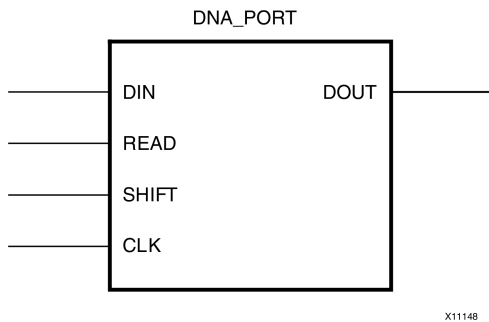
// End of DCIRESET_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## DNA\_PORT

### プリミティブ：Device DNA Access Port



### 概要

DNA\_PORT を使用すると、7 シリーズ デバイスのデバイス DNA データビット（工場でプログラムされた読み取り専用の固有 ID）を読み込むことができる専用のシフトレジスタにアクセスできます。このコンポーネントを使用すると、DNA データビットが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	説明
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 DNA ロード入力、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

## 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'h0000000000000000 ～ 57'h1fffffffff	57'h0000000000000000	シミュレーションで使用する 57 ビットの DNA 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;

-- DNA_PORT: Device DNA Access Port
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

DNA_PORT_inst : DNA_PORT
generic map (
    SIM_DNA_VALUE => X"0000000000000000" -- Specifies a sample 57-bit DNA value for simulation
)
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 (インスタンスレーション)

```
// DNA_PORT: Device DNA Access Port
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

DNA_PORT #(
    .SIM_DNA_VALUE(57'h0000000000000000) // Specifies a sample 57-bit DNA value for simulation
)
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
```

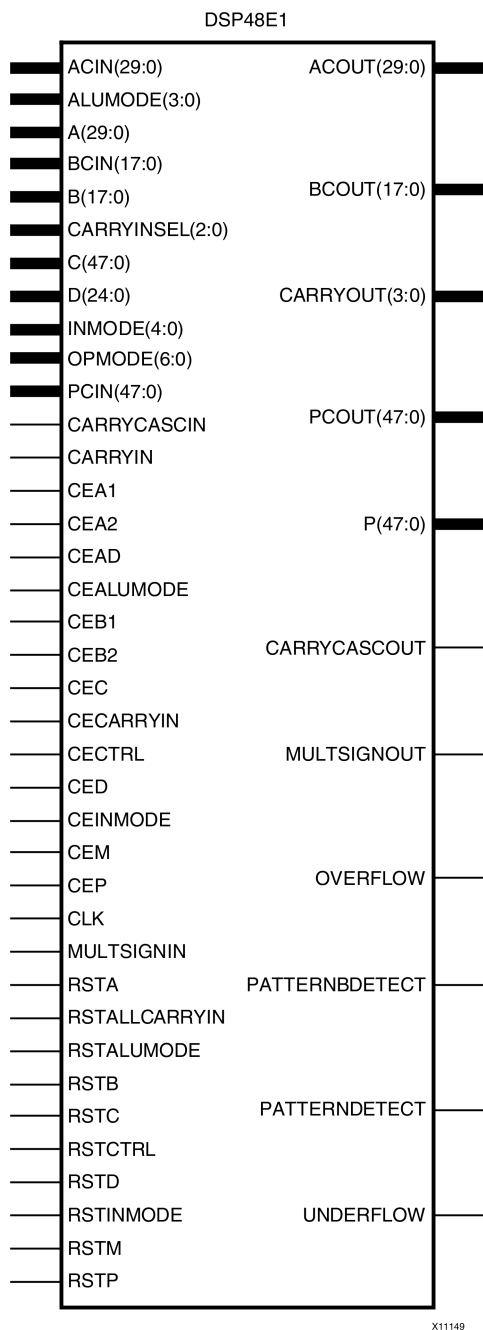
詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## DSP48E1

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



### 概要

このデザイン エLEMENTは、7 シリーズ デバイスに含まれる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、乗算、加算、減算、累積、シフト、論理演算、パターン検出などを実行できます。

データポート A、B、C、および D を使用しない場合でリーク電流を削減するには、データポート入力信号を High にし、ポート入力レジスタを選択し、CE および RST 入力制御信号を Low にする必要があります。次に、未使用の C ポートに推奨する設定の例を示します。C[47:0] = すべて 1、CREG = 1、CEC = 0、および RSTC = 0

すべての信号はアクティブ High です。

## ポートの説明

ポート名	方向	幅	説明
A<29:0>	入力	30	前置加算器、乗算器、加減算器/アキュムレータ、ALU、連結などの演算用のデータ入力。乗算器または前置加算器で使用する場合は、データ A[24:0] の 25 ビットが使用され、上位ビット (A[29:25]) は使用されないで、グラウンドに接続できます。内部加減算器/アキュムレータまたは ALU 回路で使用する場合は、30 ビットすべてが使用されます (A[29:0])。連結モードで使用する場合は、30 ビットすべてが使用され、連結ベクターの MSB (上位) ビットを構成します。
ACIN<29:0>	入力	30	カスケード接続されている前の DSP48E1 スライスの ACOUT からのデータ入力で、A 入力とマルチプレクサー処理されます。使用しない場合は、ポートをすべてゼロにします。
ACOUT<29:0>	出力	30	カスケード接続されている次の DSP48E1 スライスの ACIN へのデータ出力で、使用しない場合は未接続にします。
ALUMODE<3:0>	入力	4	DSP48E1 スライスの論理ファンクションを選択します。
B<17:0>	入力	18	乗算器の B 入力です。A と B の連結入力の下位ビット (LSB) でもあり、2 段目の加減算器または論理ファンクションに入力されます。
BCIN<17:0>	入力	18	カスケード接続されている前の DSP48E1 スライスの BCOUT からのデータ入力で、B 入力とマルチプレクサー処理されます。使用しない場合は、ポートをすべてゼロにします。
BCOUT<17:0>	出力	18	カスケード接続されている次の DSP48E1 スライスの BCIN へのデータ出力で、使用しない場合は未接続にします。この信号は、DSP48E1 列への内部専用配線バスです。ファブリック配線リソースからはアクセスできません。
C<47:0>	入力	48	加減算器、パターン検出器、または論理ファンクションのデータ入力。
CARRYCASCIN	入力	1	カスケード接続されている前の DSP48E1 スライスの CARRYCASCOUT からのキャリー入力
CARRYCASCOUT	出力	1	カスケード接続されている次の DSP48E1 スライスの CARRYCASCIN へのキャリー出力。この信号は、同じ DSP48E1 スライスの CARRYINSEL マルチプレクサー入力に内部フィードバックされます。
CARRYIN	入力	1	FPGA ロジックからのキャリー入力
CARRYINSEL<2:0>	入力	3	キャリーのソースを選択します。 <ul style="list-style-type: none"> <li>0 1 1 - PCIN[47] : PCIN の丸め (0 への丸め)</li> <li>1 0 0 - CARRYCASCOUT : 大型の加減算器/アキュムレータ向け (内部フィードバックを介した連続操作)。PREG=1 も選択する必要があります。</li> <li>1 0 1 - ~P[47] : P の丸め (無限大への丸め)。PREG=1 も選択する必要があります。</li> <li>1 1 0 - A[24] : XNOR B[17] A x B の丸め</li> <li>1 1 1 - P[47] : P の丸め (ゼロへの丸め)。PREG=1 も選択する必要があります。</li> </ul>

ポート名	方向	幅	説明
CARRYOUT<3:0>	出力	4	各累積/加算/論理ユニットの 12 ビットフィールドから出力される 4 ビットキャリー出力。通常の 48 ビット操作では CARRYOUT3 のみが使用されます。SIMD 操作では、4 つのキャリーアウトビット (CARRYOUT[3:0]) を使用できます。
CEAD	入力	1	前置加算器出力 AD パイプラインレジスタのアクティブ High のクロックイネーブル。使用しない場合および ADREG=1 の場合は、論理 1 に接続します。ADREG=0 の場合は論理 0 に接続します。
CEALUMODE	入力	1	ALUMODE (制御入力) レジスタ (ALUMODEREG=1) のアクティブ High のクロックイネーブルで、使用しない場合は論理 1 に接続します。
CEA1	入力	1	最初の A (入力) レジスタのアクティブ High のクロックイネーブルで、AREG=2 または INMODE0 = 1 の場合にのみ使用します。使用しない場合および AREG=2 の場合は、論理 1 に接続します。AREG=0 または 1 の場合は論理 0 に接続します。2 つのレジスタを使用する場合は、このレジスタが最初に使用されます。ダイナミック AB アクセスが使用されている場合、このクロックイネーブルは INMODE[0]=1 に適用されます。
CEA2	入力	1	2 番目の A (入力) レジスタのアクティブ High のクロックイネーブルで、AREG=1 または 2 の場合にのみ使用します。使用しない場合や AREG=1 または 2 の場合は論理 1 に、AREG=0 の場合は論理 0 に接続します。2 つのレジスタを使用する場合は、このレジスタが 2 番目に使用されます。使用するレジスタが 1 つの場合 (AREG=1)、CEA2 はクロックイネーブルになります。
CEB1	入力	1	最初の B (入力) レジスタのアクティブ High のクロックイネーブルで、BREG=2 または INMODE4=1 の場合にのみ使用します。使用しない場合および BREG=2 の場合は論理 1 に、BREG=0 または 1 の場合は論理 0 に接続します。2 つのレジスタを使用する場合は、このレジスタが最初に使用されます。ダイナミック AB アクセスが使用されている場合、このクロックイネーブルは INMODE[4]=1 に適用されます。
CEB2	入力	1	2 番目の B (入力) レジスタのアクティブ High のクロックイネーブルで、BREG=1 または 2 の場合にのみ使用します。使用しない場合および BREG=1 または 2 の場合は論理 1 に、BREG=0 の場合は論理 0 に接続します。2 つのレジスタを使用する場合は、このレジスタが 2 番目に使用されます。使用するレジスタが 1 つの場合 (BREG=1)、CEB2 はクロックイネーブルになります。
CEC	入力	1	C (入力) レジスタ (CREG=1) のアクティブ High のクロックイネーブルで、使用しない場合は論理 1 に接続します。
CECARRYIN	入力	1	CARRYIN (ファブリックからの入力) レジスタ (CARRYINREG=1) のアクティブ High のクロックイネーブルで、使用しない場合は論理 1 に接続します。
CECTRL	入力	1	OPMODE および CARRYINSEL (制御入力) レジスタ (OPMODEREG=1 または CARRYINSELREG=1) のアクティブ High のクロックイネーブルで、使用しない場合は論理 1 に接続します。
CED	入力	1	D (入力) レジスタ (DREG=1) のアクティブ High のクロックイネーブルで、使用しない場合は論理 1 に接続します。
CEINMODE	入力	1	INMODE 制御入力レジスタ (INMODEREG=1) のアクティブ High のクロックイネーブルで、使用しない場合は論理 1 に接続します。
CEM	入力	1	後置乗算 M (パイプライン) レジスタおよび内部乗算丸め CARRYIN レジスタ (MREG=1) のアクティブ High のクロックイネーブルで、使用しない場合は論理 1 に接続します。
CEP	入力	1	P (出力) レジスタ (PREG=1) のアクティブ High のクロックイネーブルで、使用しない場合は論理 1 に接続します。
CLK	入力	1	DSP48E1 の入力クロックで、すべての内部レジスタおよびフリップフロップで使用されます。

ポート名	方向	幅	説明
D<24:0>	入力	25	前置加算器への 25 ビット データ入力、または乗算器への代替入力。前置加算器では、INMODE3 信号の指定に応じた D + A がインプリメントされます。
INMODE<4:0>	入力	5	前置加算器、A、B、D 入力、および入力レジスタの機能を選択します。使用しない場合は、すべて 0 に接続する必要があります。
MULTSIGNIN	入力	1	前の DSP48E1 スライスから出力された乗算結果の符号。MACC 拡張用です。別の DSP ブロックの MULTSIGNOUT に接続するか、使用しない場合はグラウンドに接続します。
MULTSIGNOUT	出力	1	次の DSP48E1 スライスに入力される乗算結果の符号。MACC 拡張用です。別の DSP ブロックの MULTSIGNIN に接続するか、使用しない場合はグラウンドに接続します。この信号は、DSP48E1 列への内部専用配線バスです。ファブリック配線リソースからはアクセスできません。
OPMODE<6:0>	入力	7	DSP48E1 スライスに含まれる X、Y、Z マルチプレクサーへの入力を制御して、DSP スライスのファンクションを指定します。
OVERFLOW	出力	1	適切に設定したパターン検出器および PREG=1 を使用したときのアクティブ High のオーバーフロー出力
P<47:0>	出力	48	2 段目の加減算器または論理ファンクションからのデータ出力
PATTERNBDETECT	出力	1	P[47:0] とパターン バーの一致を示すアクティブ High 出力
PATTERNDETECT	出力	1	P[47:0] と MASK でゲート処理されたパターン的一致を示すアクティブ High 出力で、結果は P と同じクロック サイクルで出力されます。
PCIN<47:0>	入力	48	カスケード接続されている前の DSP48E1 スライスの PCOUT から加算器へのデータ入力。使用する場合は、カスケード接続されている前の DSP スライスの PCOUT に接続し、使用しない場合は、ポートをすべてゼロにします。
PCOUT<47:0>	出力	48	カスケード接続されている次の DSP48E1 スライスの PCIN へのデータ出力。使用する場合は、カスケード接続されている次の DSP スライスの PCIN に接続し、使用しない場合は未接続にします。
RSTA	入力	1	2 つの A (入力) レジスタ (AREG=1 または 2) のアクティブ High の同期リセットで、使用しない場合は論理 0 に接続します。
RSTALLCARRYIN	入力	1	キャリー (内部バス) および CARRYIN レジスタ (CARRYINREG=1) のアクティブ High 同期リセットで、使用しない場合は論理 0 に接続します。
RSTALUMODE	入力	1	ALUMODE (制御入力) レジスタ (ALUMODEREG=1) のアクティブ High の同期リセットで、使用しない場合は論理 0 に接続します。
RSTB	入力	1	2 つの B (入力) レジスタ (BREG=1 または 2) のアクティブ High の同期リセットで、使用しない場合は論理 0 に接続します。
RSTC	入力	1	C (入力) レジスタ (CREG=1) のアクティブ High の同期リセットで、使用しない場合は論理 0 に接続します。
RSTCTRL	入力	1	OPMODE および CARRYINSEL (制御入力) レジスタ (OPMODEREG=1 または CARRYINSELREG=1) のアクティブ High の同期リセットで、使用しない場合は論理 0 に接続します。
RSTD	入力	1	D (入力) レジスタおよび前置加算 (出力) AD バイプライン レジスタ (DREG=1 と ADREG=1 の両方またはいずれか) のアクティブ High の同期リセットで、使用しない場合は論理 0 に接続します。
RSTINMODE	入力	1	INMODE (制御入力) レジスタ (INMODEREG=1) のアクティブ High の同期リセットで、使用しない場合は論理 0 に接続します。
RSTM	入力	1	M (パイプライン) レジスタ (MREG=1) のアクティブ High の同期リセットで、使用しない場合は論理 0 に接続します。

ポート名	方向	幅	説明
RSTP	入力	1	P (出力) レジスタ (PREG=1) のアクティブ High の同期リセットで、使用しない場合は論理 0 に接続します。
UNDERFLOW	出力	1	適切に設定したパターン検出器および PREG=1 を使用したときのアクティブ High のアンダーフロー出力

## 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	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
ACASCREG	10 進数	1、0、2	1	AREG 属性と組み合わせて使用し、A カスケードパス ACOUT の A 入力レジスタの数を指定します。この値は、AREG の値以下にする必要があります。 <ul style="list-style-type: none"> <li>AREG=0：ACASCREG を 0 にする必要があります。</li> <li>AREG=1：ACASCREG を 1 にする必要があります。</li> <li>AREG=2：ACASCREG を 1 または 2 に設定できます。</li> </ul>
ADREG	10 進数	1、0	1	AD バイプライン レジスタの数を選択します。AD バイプライン レジスタを使用する場合は 1 に設定します。
A_INPUT	文字列	"DIRECT"、 "CASCADE"	"DIRECT"	A ポートへの入力をパラレル入力 ("DIRECT") またはカスケード接続されている前のスライスからの入力 ("CASCADE") のいずれかから選択します。
ALUMODEREG	10 進数	1、0	1	ALUMODE 入力レジスタの数を選択します。ALUMODE 入力をレジスタに格納する場合は 1 に設定します。
AREG	10 進数	1、0、2	1	A 入力パイプライン レジスタの数を選択します。
AUTORESET_PATDET	文字列	"NO_RESET"、 "RESET_MATCH"、 "RESET_NOT_MATCH"	"NO_RESET"	あるクロック サイクルでパターン検出イベントが発生した場合、その次のクロック サイクルで P レジスタ (累積値またはカウンタ値) を自動的にリセットします。パターンが一致した場合に自動リセットを実行する場合は "RESET_MATCH"、1 つ前のサイクルでパターンが一致していたが現在のサイクルでパターンが一致しない場合にリセットを実行する場合は "RESET_NOT_MATCH" を選択します。

属性	データ型	値	デフォルト	説明
BCASCREG	10 進数	1、0、2	1	BREG 属性と組み合わせて使用し、B カスケードパス BCOUT の B 入力レジスタの数を指定します。この値は、BREG の値以下にする必要があります。 <ul style="list-style-type: none"> <li>・ BREG=0：BCASCREG を 0 にする必要があります。</li> <li>・ BREG=1：BCASCREG を 1 にする必要があります。</li> <li>・ BREG=2：BCASCREG を 1 または 2 に設定できます。</li> </ul>
B_INPUT	文字列	"DIRECT"、 "CASCADE"	"DIRECT"	B ポートへの入力をパラレル入力 ("DIRECT") またはカスケード接続されている前のスライスからの入力 ("CASCADE") のいずれかから選択します。
BREG	10 進数	1、0、2	1	B 入力レジスタの数を選択します。
CARRYINREG	10 進数	1、0	1	CARRYIN 入力レジスタの数を選択します。CARRYIN 入力をレジスタに格納する場合は 1 に設定します。
CARRYINSELREG	10 進数	1、0	1	CARRYINSEL 入力レジスタの数を選択します。CARRYINSEL 入力をレジスタに格納する場合は 1 に設定します。
CREG	10 進数	1、0	1	C 入力レジスタの数を選択します。C 入力をレジスタに格納する場合は 1 に設定します。
DREG	10 進数	1、0	1	D 入力レジスタの数を選択します。D 入力をレジスタに格納する場合は 1 に設定します。
INMODEREG	10 進数	1、0	1	INMODE 入力レジスタの数を選択します。INMODE 入力をレジスタに格納する場合は 1 に設定します。
MASK	16 進数	48'h00000000000000 ～ 48'hffffffff	48'h3fffffff	48 ビット値を使用してパターン検出中に特定のビットをマスクし、パターン検出対象外にします。MASK ビットを 1 に設定すると対応するパターンビットが無視され、0 に設定するとパターンビットが比較されます。
MREG	10 進数	1、0	1	乗算器出力 (M) パイプライン レジスタの段数を選択します。使用する場合は 1 に設定します。
OPMODEREG	10 進数	1、0	1	OPMODE 入力レジスタの数を選択します。OPMODE 入力を格納する場合は 1 に設定します。
PATTERN	16 進数	48'h00000000000000 ～ 48'hffffffff	48'h0000000000000000	パターン検出器で使用されます。
PREG	10 進数	1、0	1	P 出力レジスタの数を選択します。P 出力をレジスタに格納する場合は 1 に設定します。レジスタが付いている出力には、CARRYOUT、CARRYCASCOUT、MULTSIGNOUT、PATTERNB_DETECT、PATTERN_DETECT、PCOUT が含まれます。



属性	データ型	値	デフォルト	説明
SEL_MASK	文字列	"MASK"、"C"、 "ROUNDING_MODE1"、 "ROUNDING_MODE2"	"MASK"	パターン検出器で使用するマスクを指定します。 "C" および "MASK" は、パターン検出器を標準的に使用する際の設定です (カウンター、オーバーフロー検出など)。 "ROUNDING_MODE1" (C バーを左に 1 シフト) および "ROUNDING_MODE2" (C バーを左に 2 シフト) に設定すると、オプションでレジスタを付けた C ポートに基づき、特別マスクが選択されます。これらの丸めモードは、パターン検出器を使用して DSP48E1 スライスに収束丸めをインプリメントするのに使用できます。
SEL_PATTERN	文字列	"PATTERN"、"C"	"PATTERN"	パターン フィールドの入力ソースを選択します。入力ソースには、48 ビットのダイナミック C 入力または 48 ビットのスタティック PATTERN 属性フィールドを指定できます。
USE_DPORT	ブール代数	FALSE、TRUE	FALSE	前置加算器および D ポートを使用するかどうかを指定します。
USE_MULT	文字列	"MULTIPLY"、 "DYNAMIC"、 "NONE"	"MULTIPLY"	乗算器の使用方法を選択します。 "NONE" に設定すると、加算器/論理ユニットのみを使用する場合に消費電力を節約できます。 "DYNAMIC" は、A*B と A:B をダイナミックに切り替えており、この 2 つのパスのワーストケース タイミングを取得する必要があります。
USE_PATTERN_DETECT	文字列	"NO_PATDET"、 "PATDET"	"NO_PATDET"	パターン検出器および関連機能を使用するか ("PATDET")、使用しないか ("NO_PATDET") を選択します。この属性は、スピード仕様およびシミュレーション モデル用にのみ使用します。
USE_SIMD	文字列	"ONE48"、 "TWO24"、 "FOUR12"	"ONE48"	加減算器の動作モードを選択します。 1 つの 48 ビット加算器モードにするには "ONE48"、2 つの 24 ビット加算器モードにするには "TWO24"、4 つの 12 ビット加算器モードにするには "FOUR12" を指定します。 "ONE48" モードは、Virtex-5 DSP48 の動作と互換性があり、本来の SIMD モードではありません。通常の乗算/加算操作は、"ONE48" が設定されているときにサポートされています。 "TWO24" または "FOUR12" モードを選択する場合は、乗算器は使用せず、USE_MULT を "NONE" に設定する必要があります。

## 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;

-- DSP48E1: 48-bit Multi-Functional Arithmetic Block
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

DSP48E1_inst : DSP48E1
generic map (
  -- Feature Control Attributes: Data Path Selection
  A_INPUT => "DIRECT",      -- Selects A input source, "DIRECT" (A port) or "CASCADE" (ACIN port)
  B_INPUT => "DIRECT",      -- Selects B input source, "DIRECT" (B port) or "CASCADE" (BCIN port)
  USE_DPORT => FALSE,       -- Select D port usage (TRUE or FALSE)
  USE_MULT => "MULTIPLY",   -- Select multiplier usage ("MULTIPLY", "DYNAMIC", or "NONE")
  -- Pattern Detector Attributes: Pattern Detection Configuration
  AUTORESET_PATDET => "NO_RESET", -- "NO_RESET", "RESET_MATCH", "RESET_NOT_MATCH"

```

```

MASK => X"3fffffffffff",      -- 48-bit mask value for pattern detect (1=ignore)
PATTERN => X"000000000000",   -- 48-bit pattern match for pattern detect
SEL_MASK => "MASK",           -- "C", "MASK", "ROUNDING_MODE1", "ROUNDING_MODE2"
SEL_PATTERN => "PATTERN",      -- Select pattern value ("PATTERN" or "C")
USE_PATTERN_DETECT => "NO_PATDET", -- Enable pattern detect ("PATDET" or "NO_PATDET")
-- Register Control Attributes: Pipeline Register Configuration
ACASCREG => 1,                 -- Number of pipeline stages between A/ACIN and ACOUT (0, 1 or 2)
ADREG => 1,                    -- Number of pipeline stages for pre-adder (0 or 1)
ALUMODEREG => 1,               -- Number of pipeline stages for ALUMODE (0 or 1)
AREG => 1,                     -- Number of pipeline stages for A (0, 1 or 2)
BCASCREG => 1,                 -- Number of pipeline stages between B/BCIN and BCOUT (0, 1 or 2)
BREG => 1,                     -- Number of pipeline stages for B (0, 1 or 2)
CARRYINREG => 1,               -- Number of pipeline stages for CARRYIN (0 or 1)
CARRYINSELREG => 1,            -- Number of pipeline stages for CARRYINSEL (0 or 1)
CREG => 1,                     -- Number of pipeline stages for C (0 or 1)
DREG => 1,                     -- Number of pipeline stages for D (0 or 1)
INMODEREG => 1,                -- Number of pipeline stages for INMODE (0 or 1)
MREG => 1,                     -- Number of multiplier pipeline stages (0 or 1)
OPMODEREG => 1,                -- Number of pipeline stages for OPMODE (0 or 1)
PREG => 1,                     -- Number of pipeline stages for P (0 or 1)
USE_SIMD => "ONE48"            -- SIMD selection ("ONE48", "TWO24", "FOUR12")
)
port map (
-- Cascade: 30-bit (each) output: Cascade Ports
ACOUT => ACOUT,                -- 30-bit output: A port cascade output
BCOUT => BCOUT,                -- 18-bit output: B port cascade output
CARRYCASCOUT => CARRYCASCOUT,   -- 1-bit output: Cascade carry output
MULTSIGNOUT => MULTSIGNOUT,     -- 1-bit output: Multiplier sign cascade output
PCOUT => PCOUT,                -- 48-bit output: Cascade output
-- Control: 1-bit (each) output: Control Inputs/Status Bits
OVERFLOW => OVERFLOW,          -- 1-bit output: Overflow in add/acc output
PATTERNBDETECT => PATTERNBDETECT, -- 1-bit output: Pattern bar detect output
PATTERNDETECT => PATTERNDETECT, -- 1-bit output: Pattern detect output
UNDERFLOW => UNDERFLOW,        -- 1-bit output: Underflow in add/acc output
-- Data: 4-bit (each) output: Data Ports
CARRYOUT => CARRYOUT,           -- 4-bit output: Carry output
P => P,                          -- 48-bit output: Primary data output
-- Cascade: 30-bit (each) input: Cascade Ports
ACIN => ACIN,                   -- 30-bit input: A cascade data input
BCIN => BCIN,                   -- 18-bit input: B cascade input
CARRYCASCIN => CARRYCASCIN,     -- 1-bit input: Cascade carry input
MULTSIGNIN => MULTSIGNIN,       -- 1-bit input: Multiplier sign input
PCIN => PCIN,                   -- 48-bit input: P cascade input
-- Control: 4-bit (each) input: Control Inputs/Status Bits
ALUMODE => ALUMODE,              -- 4-bit input: ALU control input
CARRYINSEL => CARRYINSEL,        -- 3-bit input: Carry select input
CEINMODE => CEINMODE,            -- 1-bit input: Clock enable input for INMODEREG
CLK => CLK,                     -- 1-bit input: Clock input
INMODE => INMODE,                -- 5-bit input: INMODE control input
OPMODE => OPMODE,                -- 7-bit input: Operation mode input
RSTINMODE => RSTINMODE,          -- 1-bit input: Reset input for INMODEREG
-- Data: 30-bit (each) input: Data Ports
A => A,                          -- 30-bit input: A data input
B => B,                          -- 18-bit input: B data input
C => C,                          -- 48-bit input: C data input
CARRYIN => CARRYIN,              -- 1-bit input: Carry input signal
D => D,                          -- 25-bit input: D data input
-- Reset/Clock Enable: 1-bit (each) input: Reset/Clock Enable Inputs
CEA1 => CEA1,                   -- 1-bit input: Clock enable input for 1st stage AREG
CEA2 => CEA2,                   -- 1-bit input: Clock enable input for 2nd stage AREG
CEAD => CEAD,                   -- 1-bit input: Clock enable input for ADREG
CEALUMODE => CEALUMODE,          -- 1-bit input: Clock enable input for ALUMODERE
CEB1 => CEB1,                   -- 1-bit input: Clock enable input for 1st stage BREG
CEB2 => CEB2,                   -- 1-bit input: Clock enable input for 2nd stage BREG
CEC => CEC,                     -- 1-bit input: Clock enable input for CREG
CECARRYIN => CECARRYIN,          -- 1-bit input: Clock enable input for CARRYINREG
CECTRL => CECTRL,                -- 1-bit input: Clock enable input for OPMODEREG and CARRYINSELREG
CED => CED,                     -- 1-bit input: Clock enable input for DREG
CEM => CEM,                     -- 1-bit input: Clock enable input for MREG
CEP => CEP,                     -- 1-bit input: Clock enable input for PREG
RSTA => RSTA,                   -- 1-bit input: Reset input for AREG
RSTALLCARRYIN => RSTALLCARRYIN, -- 1-bit input: Reset input for CARRYINREG
RSTALUMODE => RSTALUMODE,        -- 1-bit input: Reset input for ALUMODEREG

```



```

RSTB => RSTB,          -- 1-bit input: Reset input for BREG
RSTC => RSTC,          -- 1-bit input: Reset input for CREG
RSTCTRL => RSTCTRL,    -- 1-bit input: Reset input for OPMODEREG and CARRYINSELREG
RSTD => RSTD,          -- 1-bit input: Reset input for DREG and ADREG
RSTM => RSTM,          -- 1-bit input: Reset input for MREG
RSTP => RSTP,          -- 1-bit input: Reset input for PREG
);

-- End of DSP48E1_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// DSP48E1: 48-bit Multi-Functional Arithmetic Block
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

DSP48E1 #(
    // Feature Control Attributes: Data Path Selection
    .A_INPUT("DIRECT"),          // Selects A input source, "DIRECT" (A port) or "CASCADE" (ACIN port)
    .B_INPUT("DIRECT"),          // Selects B input source, "DIRECT" (B port) or "CASCADE" (BCIN port)
    .USE_DPORT("FALSE"),         // Select D port usage (TRUE or FALSE)
    .USE_MULT("MULTIPLY"),        // Select multiplier usage ("MULTIPLY", "DYNAMIC", or "NONE")
    // Pattern Detector Attributes: Pattern Detection Configuration
    .AUTORESET_PATDET("NO_RESET"), // "NO_RESET", "RESET_MATCH", "RESET_NOT_MATCH"
    .MASK(48'h3fffffff),         // 48-bit mask value for pattern detect (1=ignore)
    .PATTERN(48'h000000000000),   // 48-bit pattern match for pattern detect
    .SEL_MASK("MASK"),            // "C", "MASK", "ROUNDING_MODE1", "ROUNDING_MODE2"
    .SEL_PATTERN("PATTERN"),       // Select pattern value ("PATTERN" or "C")
    .USE_PATTERN_DETECT("NO_PATDET"), // Enable pattern detect ("PATDET" or "NO_PATDET")
    // Register Control Attributes: Pipeline Register Configuration
    .ACASCREG(1),                 // Number of pipeline stages between A/ACIN and ACOUT (0, 1 or 2)
    .ADREG(1),                   // Number of pipeline stages for pre-adder (0 or 1)
    .ALUMODEREG(1),              // Number of pipeline stages for ALUMODE (0 or 1)
    .AREG(1),                    // Number of pipeline stages for A (0, 1 or 2)
    .BCASCREG(1),                // Number of pipeline stages between B/BCIN and BCOUT (0, 1 or 2)
    .BREG(1),                    // Number of pipeline stages for B (0, 1 or 2)
    .CARRYINREG(1),              // Number of pipeline stages for CARRYIN (0 or 1)
    .CARRYINSELREG(1),           // Number of pipeline stages for CARRYINSEL (0 or 1)
    .CREG(1),                    // Number of pipeline stages for C (0 or 1)
    .DREG(1),                    // Number of pipeline stages for D (0 or 1)
    .INMODEREG(1),               // Number of pipeline stages for INMODE (0 or 1)
    .MREG(1),                    // Number of multiplier pipeline stages (0 or 1)
    .OPMODEREG(1),               // Number of pipeline stages for OPMODE (0 or 1)
    .PREG(1),                    // Number of pipeline stages for P (0 or 1)
    .USE_SIMD("ONE48"),           // SIMD selection ("ONE48", "TWO24", "FOUR12")
)
DSP48E1_inst (
    // Cascade: 30-bit (each) output: Cascade Ports
    .ACOUT(ACOUT),               // 30-bit output: A port cascade output
    .BCOUT(BCOUT),               // 18-bit output: B port cascade output
    .CARRYCASCOUT(CARRYCASCOUT), // 1-bit output: Cascade carry output
    .MULTSIGNOUT(MULTSIGNOUT),   // 1-bit output: Multiplier sign cascade output
    .PCOUT(PCOUT),               // 48-bit output: Cascade output
    // Control: 1-bit (each) output: Control Inputs/Status Bits
    .OVERFLOW(OVERFLOW),         // 1-bit output: Overflow in add/acc output
    .PATTERNBDETECT(PATTERNBDETECT), // 1-bit output: Pattern bar detect output
    .PATTERNDETECT(PATTERNDETECT), // 1-bit output: Pattern detect output
    .UNDERFLOW(UNDERFLOW),       // 1-bit output: Underflow in add/acc output
    // Data: 4-bit (each) output: Data Ports
    .CARRYOUT(CARRYOUT),         // 4-bit output: Carry output
    .P(P),                       // 48-bit output: Primary data output
    // Cascade: 30-bit (each) input: Cascade Ports
    .ACIN(ACIN),                 // 30-bit input: A cascade data input
    .BCIN(BCIN),                 // 18-bit input: B cascade input
    .CARRYCASCIN(CARRYCASCIN),   // 1-bit input: Cascade carry input
    .MULTSIGNIN(MULTSIGNIN),     // 1-bit input: Multiplier sign input
    .PCIN(PCIN),                 // 48-bit input: P cascade input
    // Control: 4-bit (each) input: Control Inputs/Status Bits
    .ALUMODE(ALUMODE),           // 4-bit input: ALU control input
    .CARRYINSEL(CARRYINSEL),     // 3-bit input: Carry select input
    .CEINMODE(CEINMODE),         // 1-bit input: Clock enable input for INMODEREG

```

```

.CLK(CLK),                // 1-bit input: Clock input
.INMODE(INMODE),          // 5-bit input: INMODE control input
.OPMODE(OPMODE),          // 7-bit input: Operation mode input
.RSTINMODE(RSTINMODE),    // 1-bit input: Reset input for INMODEREG
// Data: 30-bit (each) input: Data Ports
.A(A),                    // 30-bit input: A data input
.B(B),                    // 18-bit input: B data input
.C(C),                    // 48-bit input: C data input
.CARRYIN(CARRYIN),        // 1-bit input: Carry input signal
.D(D),                    // 25-bit input: D data input
// Reset/Clock Enable: 1-bit (each) input: Reset/Clock Enable Inputs
.CEA1(CEA1),              // 1-bit input: Clock enable input for 1st stage AREG
.CEA2(CEA2),              // 1-bit input: Clock enable input for 2nd stage AREG
.CEAD(CEAD),              // 1-bit input: Clock enable input for ADREG
.CEALUMODE(CEALUMODE),    // 1-bit input: Clock enable input for ALUMODERE
.CEB1(CEB1),              // 1-bit input: Clock enable input for 1st stage BREG
.CEB2(CEB2),              // 1-bit input: Clock enable input for 2nd stage BREG
.CEC(CEC),                // 1-bit input: Clock enable input for CREG
.CECARRYIN(CECARRYIN),    // 1-bit input: Clock enable input for CARRYINREG
.CECTRL(CECTRL),          // 1-bit input: Clock enable input for OPMODEREG and CARRYINSELREG
.CED(CED),                // 1-bit input: Clock enable input for DREG
.CEM(CEM),                // 1-bit input: Clock enable input for MREG
.CEP(CEP),                // 1-bit input: Clock enable input for PREG
.RSTA(RSTA),              // 1-bit input: Reset input for AREG
.RSTALLCARRYIN(RSTALLCARRYIN), // 1-bit input: Reset input for CARRYINREG
.RSTALUMODE(RSTALUMODE),  // 1-bit input: Reset input for ALUMODERE
.RSTB(RSTB),              // 1-bit input: Reset input for BREG
.RSTC(RSTC),              // 1-bit input: Reset input for CREG
.RSTCTRL(RSTCTRL),        // 1-bit input: Reset input for OPMODEREG and CARRYINSELREG
.RSTD(RSTD),              // 1-bit input: Reset input for DREG and ADREG
.RSTM(RSTM),              // 1-bit input: Reset input for MREG
.RSTP(RSTP),              // 1-bit input: Reset input for PREG
);

// End of DSP48E1_inst instantiation

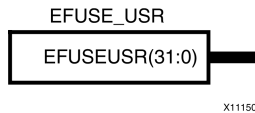
```

## 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## EFUSE\_USR

**プリミティブ：32-bit non-volatile design ID**



### 概要

このデザイン エLEMENTを使用すると、ユーザーがプログラム可能な不揮発性の eFUSE レジスタの 32 ビットにアクセスできます。

### ポートの説明

ポート名	方向	幅	説明
EFUSEUSR<31 :0>	出力	32	ユーザー eFUSE レジスタ値出力

### 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
SIM_EFUSE_VALUE	16 進数	32'h00000000 ~ 32'hffffff	すべてゼロ	シミュレーションで使用される 32 ビットの非揮発性の値

### 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;

-- EFUSE_USR: 32-bit non-volatile design ID
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

EFUSE_USR_inst : EFUSE_USR
generic map (
    SIM_EFUSE_VALUE => X"00000000" -- Value of the 32-bit non-volatile value used in simulation
)
port map (
    EFUSEUSR => EFUSEUSR -- 32-bit output: User eFUSE register value output
);

-- End of EFUSE_USR_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// EFUSE_USR: 32-bit non-volatile design ID
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

EFUSE_USR #(
    .SIM_EFUSE_VALUE(32'h00000000) // Value of the 32-bit non-volatile value used in simulation
)
EFUSE_USR_inst (
    .EFUSEUSR(EFUSEUSR) // 32-bit output: User eFUSE register value output
);

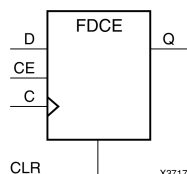
// End of EFUSE_USR_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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\_E2 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	2 進数	1、0	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;

-- FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
--       Clock Enable (posedge clk).
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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).
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

FDCE #(
    .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) 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
);

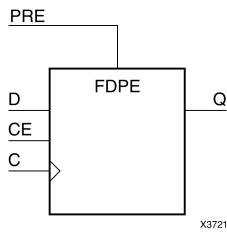
// End of FDCE_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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。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\_E2 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	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;

-- FDPE: Single Data Rate D Flip-Flop with Asynchronous Preset and
--       Clock Enable (posedge clk).
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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).
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

FDPE #(
    .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) 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
```

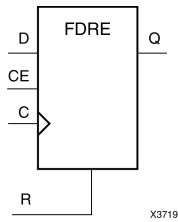
詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## 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\_E2 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	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;

-- FDRE: Single Data Rate D Flip-Flop with Synchronous Reset and
--       Clock Enable (posedge clk).
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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).
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

FDRE #(
    .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) 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
);

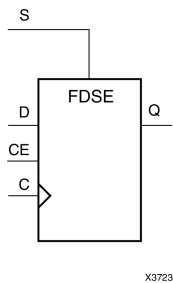
// End of FDRE_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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\_E2 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	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;

-- FDSE: Single Data Rate D Flip-Flop with Synchronous Set and
--       Clock Enable (posedge clk).
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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).
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

FDSE #(
    .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) 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
);

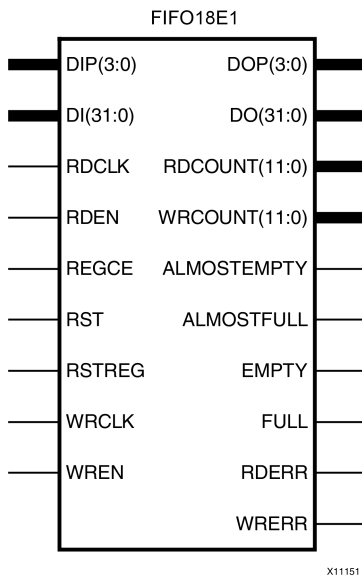
// End of FDSE_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## FIFO18E1

### プリミティブ：18Kb FIFO (First-In-First-Out) Block RAM Memory



## 概要

7 シリーズ デバイスにはブロック RAM が数個含まれ、これらの RAM を個別に FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36Kb または 18Kb) 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。FIFO18E1 では、FIFO 制御ロジックおよび 18Kb ブロック RAM が使用されます。このプリミティブは、4 ビット X 4K、9 ビット X 2K、18 ビット X 1K、または 36 ビット X 512 コンフィギュレーションで使用できます。また、関連するすべての FIFO フラグおよびステータス信号を持つ、同期モードまたはデュアル クロック (非同期) モードにコンフィギュレーションできます。

デュアル クロック モードで独立したクロックを使用する場合、読み出しクロック エッジと書き込みクロック エッジ間のオフセットによっては、Empty、Almost Empty、Full、および Almost Full フラグが 1 クロック サイクル後にディアサートされることがあります。クロックが非同期のため、シミュレーション モデルにはユーザー ガイドに示されているディアサートレイテンシ サイクルのみが反映されます。

**注記：** 36 ビット X 512 ワードの FIFO には、“FIFO18\_36” モードを使用する必要があります。これよりワード数が多く、データ幅の広いコンフィギュレーションには、FIFO36E1 を使用できます。エラー修正回路が必要な場合は、FIFO36E1 を “FIFO36\_72” モードで使用する必要があります。

## ポートの説明

ポート名	方向	幅	説明
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示します。このフラグのしきい値は ALMOST_EMPTY_OFFSET 属性で指定します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示します。このフラグのしきい値は ALMOST_FULL_OFFSET 属性で指定します。
DI<31:0>	入力	32	FIFO データ入力バス
DIP<3:0>	入力	4	FIFO パリティ データ入力バス
DO<31:0>	出力	32	FIFO データ出力バス

ポート名	方向	幅	説明
DOP<3:0>	出力	4	FIFO パリティ データ出力バス
EMPTY	出力	1	FIFO が空であることを示します。
FULL	出力	1	FIFO がフルであることを示します。
RDCLK	入力	1	立ち上がりエッジ リード クロック 入力
RDCOUNT<11:0>	出力	12	読み出しカウント
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
RDERR	出力	1	読み出しエラーが発生したことを示します。
REGCE	入力	1	パイプライン化された同期 FIFO の出力レジスタ クロック イネーブル。このイネーブルを使用するには、DO_REG を 1 にする必要があります。
RST	入力	1	アクティブ High の (FIFO ロジック) 非同期リセット (デュアル クロック FIFO の場合)、または同期リセット (同期 FIFO の場合)。WRCLK/RDCLK の 5 サイクル以上 High にする必要があります。
RSTREG	入力	1	出力レジスタの同期セット/リセット。このリセットを使用するには、DO_REG を 1 にする必要があります。
WRCLK	入力	1	立ち上がりエッジ ライト クロック
WRCOUNT<11:0>	出力	12	書き込みカウント
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
WRERR	出力	1	書き込みエラーが発生したことを示します。FIFO がフルのときに書き込みを行うとアサートされます。WRCLK に同期しています。

## 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	推奨

## 使用可能な属性

属性	データ型	値	デフォルト	説明
ALMOST_EMPTY_OFFSET	16 進数	13'h0000 ~ 13'h1fff	13'h0080	ALMOST_EMPTY フラグをトリガーする RAM のデータ量を指定します。
ALMOST_FULL_OFFSET	16 進数	13'h0000 ~ 13'h1fff	13'h0080	ALMOST_FULL フラグをトリガーする RAM のデータ量を指定します。
DATA_WIDTH	10 進数	4、9、18、36	4	FIFO のデータ幅を指定します。  注記：36 に設定する場合は、FIFO_MODE を "FIFO18_36" に設定する必要があります。
DO_REG	10 進数	1、0	1	EN_SYN のデータ パイプライン レジスタ

属性	データ型	値	デフォルト	説明
EN_SYN	ブール代数	FALSE、TRUE	FALSE	FIFO がデュアル クロック (独立した 2 つのクロック) または同期 (1 つのクロック) のいずれかで動作しているかを示します。デュアル クロックの場合は DO_REG=1 である必要があります。
FIFO_MODE	文字列	"FIFO18"、 "FIFO18_36"	"FIFO18"	"FIFO18" または "FIFO18_36" モードを選択します。"FIFO18_36" に設定する場合は、DATA_WIDTH を 36 に設定する必要があります。
FIRST_WORD_FALL_THROUGH	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、RDEN がアサートされずに FIFO に最初に書き込まれた値が DO に出力されます。
INIT	16 進数	36 ビット値	36'h000000000	コンフィギュレーション後の DO 出力の初期値を指定します。
SIM_DEVICE	文字列	"7SERIES"	"7SERIES"	どの条件でもシミュレーションが正しく動作するよう "7SERIES" に設定する必要があります。
SRVAL	16 進数	36 ビット値	36'h000000000	同期リセット信号 (RSTREG) がアサートされたときの FIFO の出力値を指定します。DO_REG=1 の場合のみ有効です。

## 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;

-- FIFO18E1: 18Kb FIFO (First-In-First-Out) Block RAM Memory
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

FIFO18E1_inst : FIFO18E1
generic map (
    ALMOST_EMPTY_OFFSET => X"0080",    -- Sets the almost empty threshold
    ALMOST_FULL_OFFSET  => X"0080",    -- Sets almost full threshold
    DATA_WIDTH => 4,                  -- Sets data width to 4-36
    DO_REG => 1,                      -- Enable output register (1-0) Must be 1 if EN_SYN = FALSE
    EN_SYN => FALSE,                  -- Specifies FIFO as dual-clock (FALSE) or Synchronous (TRUE)
    FIFO_MODE => "FIFO18",            -- Sets mode to FIFO18 or FIFO18_36
    FIRST_WORD_FALL_THROUGH => FALSE,  -- Sets the FIFO FWFT to FALSE, TRUE
    INIT => X"000000000",              -- Initial values on output port
    SIM_DEVICE => "7SERIES",          -- Must be set to "7SERIES" for simulation behavior
    SRVAL => X"000000000"              -- Set/Reset value for output port
)
port map (
    -- Read Data: 32-bit (each) output: Read output data
    DO => DO,                          -- 32-bit output: Data output
    DOP => DOP,                         -- 4-bit output: Parity data output
    -- Status: 1-bit (each) output: Flags and other FIFO status outputs
    ALMOSTEMPTY => ALMOSTEMPTY,        -- 1-bit output: Almost empty flag
    ALMOSTFULL => ALMOSTFULL,          -- 1-bit output: Almost full flag
    EMPTY => EMPTY,                    -- 1-bit output: Empty flag
    FULL => FULL,                       -- 1-bit output: Full flag
    RDCOUNT => RDCOUNT,                 -- 12-bit output: Read count
    RDERR => RDERR,                     -- 1-bit output: Read error
    WRCOUNT => WRCOUNT,               -- 12-bit output: Write count
    WRERR => WRERR,                     -- 1-bit output: Write error
    -- Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
    RDCLK => RDCLK,                     -- 1-bit input: Read clock

```

```

RDEN => RDEN,           -- 1-bit input: Read enable
REGCE => REGCE,         -- 1-bit input: Clock enable
RST => RST,             -- 1-bit input: Asynchronous Reset
RSTREG => RSTREG,       -- 1-bit input: Output register set/reset
-- Write Control Signals: 1-bit (each) input: Write clock and enable input signals
WRCLK => WRCLK,         -- 1-bit input: Write clock
WREN => WREN,           -- 1-bit input: Write enable
-- Write Data: 32-bit (each) input: Write input data
DI => DI,               -- 32-bit input: Data input
DIP => DIP              -- 4-bit input: Parity input
);

-- End of FIFO18E1_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// FIFO18E1: 18Kb FIFO (First-In-First-Out) Block RAM Memory
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

FIFO18E1 #(
    .ALMOST_EMPTY_OFFSET(13'h0080), // Sets the almost empty threshold
    .ALMOST_FULL_OFFSET(13'h0080),  // Sets almost full threshold
    .DATA_WIDTH(4),                 // Sets data width to 4-36
    .DO_REG(1),                     // Enable output register (1-0) Must be 1 if EN_SYN = FALSE
    .EN_SYN("FALSE"),               // Specifies FIFO as dual-clock (FALSE) or Synchronous (TRUE)
    .FIFO_MODE("FIFO18"),           // Sets mode to FIFO18 or FIFO18_36
    .FIRST_WORD_FALL_THROUGH("FALSE"), // Sets the FIFO FWFT to FALSE, TRUE
    .INIT(36'h000000000),           // Initial values on output port
    .SIM_DEVICE("7SERIES"),         // Must be set to "7SERIES" for simulation behavior
    .SRVAL(36'h000000000)           // Set/Reset value for output port
)
FIFO18E1_inst (
    // Read Data: 32-bit (each) output: Read output data
    .DO(DO),                        // 32-bit output: Data output
    .DOP(DOP),                      // 4-bit output: Parity data output
    // Status: 1-bit (each) output: Flags and other FIFO status outputs
    .ALMOSTEMPTY(ALMOSTEMPTY),     // 1-bit output: Almost empty flag
    .ALMOSTFULL(ALMOSTFULL),       // 1-bit output: Almost full flag
    .EMPTY(EMPTY),                 // 1-bit output: Empty flag
    .FULL(FULL),                   // 1-bit output: Full flag
    .RDCOUNT(RDCOUNT),             // 12-bit output: Read count
    .RDERR(RDERR),                 // 1-bit output: Read error
    .WRCOUNT(WRCOUNT),             // 12-bit output: Write count
    .WRERR(WRERR),                 // 1-bit output: Write error
    // Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
    .RDCLK(RDCLK),                 // 1-bit input: Read clock
    .RDEN(RDEN),                   // 1-bit input: Read enable
    .REGCE(REGCE),                 // 1-bit input: Clock enable
    .RST(RST),                     // 1-bit input: Asynchronous Reset
    .RSTREG(RSTREG),               // 1-bit input: Output register set/reset
    // Write Control Signals: 1-bit (each) input: Write clock and enable input signals
    .WRCLK(WRCLK),                 // 1-bit input: Write clock
    .WREN(WREN),                   // 1-bit input: Write enable
    // Write Data: 32-bit (each) input: Write input data
    .DI(DI),                       // 32-bit input: Data input
    .DIP(DIP)                      // 4-bit input: Parity input
);

// End of FIFO18E1_inst instantiation

```

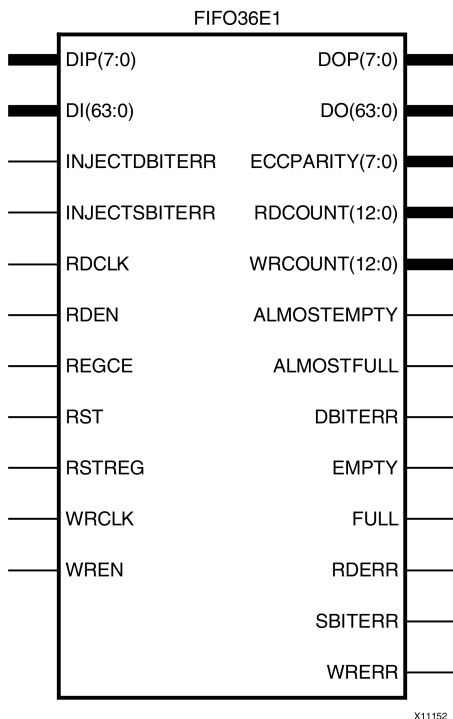
詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## FIFO36E1

プリミティブ：36Kb FIFO (First-In-First-Out) Block RAM Memory



### 概要

7 シリーズ デバイスにはブロック RAM が数個含まれ、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36 Kb または 18 Kb) 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。FIFO36E1 を使用すると、36KB の FIFO のブロック RAM へアクセスできます。このコンポーネントは、関連 FIFO フラグを持つ 4 ビット X 8K ワード、9 ビット X 4K ワード、18 ビット X 2K ワード、36 ビット X 1K ワード、72 ビット X 512 ワードの同期またはデュアル クロック (非同期) FIFO RAM としてコンフィギュレーションできます。

デュアル クロック モードで独立したクロックを使用する場合、読み出しクロック エッジと書き込みクロック エッジ間のオフセットによっては、Empty、Almost Empty、Full、および Almost Full フラグが 1 クロック サイクル後にディアサートされることがあります。クロックが非同期のため、シミュレーション モデルにはユーザー ガイドに示されているディアサートレイテンシ サイクルのみが反映されます。

**注記：** 72 ビット X 512 ワードの FIFO には、“FIFO36\_72” モードを使用する必要があります。これよりワード数が少なく、データ幅の狭いコンフィギュレーションには、FIFO18E1 を使用できます。エラー修正回路が必要な場合は、“FIFO36\_72” モードを使用する必要があります。

## ポートの説明

ポート名	方向	幅	説明
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示します。このフラグをトリガーする位置は ALMOST_EMPTY_OFFSET 属性で指定します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示します。このフラグをトリガーする位置は ALMOST_FULL_OFFSET 属性で指定します。
DBITERR	出力	1	ダブル ビット エラーが検出されたことを示す ECC ファンクションからのステータス出力。ECC 機能を使用する場合は、EN_ECC_READ を TRUE に設定する必要があります。
DI<63:0>	入力	64	FIFO データ入力バス
DIP<7:0>	入力	8	FIFO パリティ データ入力バス
DO<63:0>	出力	64	FIFO データ出力バス
DOP<7:0>	出力	8	FIFO パリティ データ出力バス
ECCPARITY<7:0>	出力	8	ECC デコーダーでメモリ エラーを検出および訂正するために使用される、ECC エンコーダーで生成された 8 ビット データ。
EMPTY	出力	1	FIFO が空であることを示します。
FULL	出力	1	FIFO がフルであることを示します。
INJECTDBITE RR	入力	1	ECC 機能を使用している場合にダブル ビット エラーを挿入します。
INJECTSBITE RR	入力	1	ECC 機能を使用している場合にシングル ビット エラーを挿入します。
RDCLK	入力	1	立ち上がりエッジ リード クロック 入力
RDCOUNT<12:0>	出力	13	読み出しカウント
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
RDERR	出力	1	読み出しエラーが発生したことを示します。
REGCE	入力	1	パイプライン化された同期 FIFO の出力レジスタ クロック イネーブル。このイネーブルを使用するには、DO_REG を 1 にする必要があります。
RST	入力	1	5 クロック サイクル間アクティブ High の (FIFO ロジック) の非同期リセット (デュアル レートの FIFO 向け)、同期リセット (同期 FIFO)
RSTREG	入力	1	出力レジスタの同期セット/リセット。このリセットを使用するには、DO_REG を 1 にする必要があります。
SBITERR	出力	1	シングル ビット エラーが検出されたことを示す ECC 機能のステータス出力。ECC 機能を使用する場合は、EN_ECC_READ を TRUE に設定する必要があります。
WRCLK	入力	1	ライト クロックおよびイネーブル入力信号
WRCOUNT<12:0>	出力	13	書き込みカウント
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
WRERR	出力	1	書き込みエラーが発生したことを示します。FIFO がフルのときに書き込みを行うとアサートされます。WRCLK に同期しています。

## 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	推奨

## 使用可能な属性

属性	データ型	値	デフォルト	説明
ALMOST_EMPTY_OFFSET	16 進数	13'h0000 ~ 13'h1fff	13'h0080	ALMOST_EMPTY フラグをトリガーする RAM のデータ量を指定します。
ALMOST_FULL_OFFSET	16 進数	13'h0000 ~ 13'h1fff	13'h0080	ALMOST_FULL フラグをトリガーする RAM のデータ量を指定します。
DATA_WIDTH	10 進数	4、9、18、36、72	4	FIFO に必要なデータ幅を指定。データ幅が 72 の場合は、FIFO_MODE を "FIFO36_72" に設定する必要があります。
DO_REG	10 進数	1、0	1	読み出しレイテンシ (パイプライン遅延 1 つ) 追加することで clock-to-out のタイミングを向上するように、FIFO の出力レジスタをイネーブルします。EN_SYN が FALSE のときは DO_REG を 1 にする必要があります。
EN_ECC_READ	ブール代数	FALSE、TRUE	FALSE	ECC デコーダー回路をイネーブルにします。
EN_ECC_WRITE	ブール代数	FALSE、TRUE	FALSE	ECC エンコーダー回路をイネーブルにします。
EN_SYN	ブール代数	FALSE、TRUE	FALSE	FALSE のときは非同期モード、TRUE のときは同期 (1 クロック) モードで FIFO が使用されることを示します。
FIFO_MODE	文字列	"FIFO36"、 "FIFO36_72"	"FIFO36"	通常の "FIFO36" または幅広の "FIFO36_72" モードのいずれかを選択します。"FIFO36_72" に設定する場合、DATA_WIDTH 属性は 72 にする必要があります。
FIRST_WORD_FALL_THROUGH	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、RDEN がアサートされずに FIFO に最初に書き込まれた値が DO に出力されます。
INIT	16 進数	72 ビット値	72'h000000 0000000000 00	コンフィギュレーション後の DO 出力の初期値を指定します。
SIM_DEVICE	文字列	"7SERIES"	"7SERIES"	どの条件でもシミュレーションが正しく動作するよう "7SERIES" に設定する必要があります。
SRVAL	16 進数	72 ビット値	72'h000000 0000000000 00	同期リセット信号 (RSTREG) がアサートされたときの FIFO の出力値を指定します。DO_REG=1 の場合のみ有効です。

## 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;

-- FIFO36E1: 36Kb FIFO (First-In-First-Out) Block RAM Memory
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

FIFO36E1_inst : FIFO36E1
generic map (
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    DATA_WIDTH => 4, -- Sets data width to 4-72
    DO_REG => 1, -- Enable output register (1-0) Must be 1 if EN_SYN = FALSE
    EN_ECC_READ => FALSE, -- Enable ECC decoder, FALSE, TRUE
    EN_ECC_WRITE => FALSE, -- Enable ECC encoder, FALSE, TRUE
    EN_SYN => FALSE, -- Specifies FIFO as Asynchronous (FALSE) or Synchronous (TRUE)
    FIFO_MODE => "FIFO36", -- Sets mode to "FIFO36" or "FIFO36_72"
    FIRST_WORD_FALL_THROUGH => FALSE, -- Sets the FIFO FWT to &VALUES
    INIT => X"000000000000000000", -- Initial values on output port
    SIM_DEVICE => "7SERIES", -- Must be set to "7SERIES" for simulation behavior
    SRVAL => X"000000000000000000" -- Set/Reset value for output port
)
port map (
    -- ECC Signals: 1-bit (each) output: Error Correction Circuitry ports
    DBITERR => DBITERR, -- 1-bit output: Double bit error status
    ECCPARITY => ECCPARITY, -- 8-bit output: Generated error correction parity
    SBITERR => SBITERR, -- 1-bit output: Single bit error status
    -- Read Data: 64-bit (each) output: Read output data
    DO => DO, -- 64-bit output: Data output
    DOP => DOP, -- 8-bit output: Parity data output
    -- Status: 1-bit (each) output: Flags and other FIFO status outputs
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit output: Almost empty flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit output: Almost full flag
    EMPTY => EMPTY, -- 1-bit output: Empty flag
    FULL => FULL, -- 1-bit output: Full flag
    RDCOUNT => RDCOUNT, -- 13-bit output: Read count
    RDERR => RDERR, -- 1-bit output: Read error
    WRCOUNT => WRCOUNT, -- 13-bit output: Write count
    WRERR => WRERR, -- 1-bit output: Write error
    -- ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
    INJECTDBITERR => INJECTDBITERR, -- 1-bit input: Inject a double bit error input
    INJECTSBITERR => INJECTSBITERR,
    -- Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
    RDCLK => RDCLK, -- 1-bit input: Read clock
    RDEN => RDEN, -- 1-bit input: Read enable
    REGCE => REGCE, -- 1-bit input: Clock enable
    RST => RST, -- 1-bit input: Reset
    RSTREG => RSTREG, -- 1-bit input: Output register set/reset
    -- Write Control Signals: 1-bit (each) input: Write clock and enable input signals
    WRCLK => WRCLK, -- 1-bit input: Rising edge write clock.
    WREN => WREN, -- 1-bit input: Write enable
    -- Write Data: 64-bit (each) input: Write input data
    DI => DI, -- 64-bit input: Data input
    DIP => DIP, -- 8-bit input: Parity input
);

-- End of FIFO36E1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// FIFO36E1: 36Kb FIFO (First-In-First-Out) Block RAM Memory
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

FIFO36E1 #(
    .ALMOST_EMPTY_OFFSET(13'h0080),    // Sets the almost empty threshold
    .ALMOST_FULL_OFFSET(13'h0080),     // Sets almost full threshold
    .DATA_WIDTH(4),                    // Sets data width to 4-72
    .DO_REG(1),                        // Enable output register (1-0) Must be 1 if EN_SYN = FALSE
    .EN_ECC_READ("FALSE"),             // Enable ECC decoder, FALSE, TRUE
    .EN_ECC_WRITE("FALSE"),            // Enable ECC encoder, FALSE, TRUE
    .EN_SYN("FALSE"),                  // Specifies FIFO as Asynchronous (FALSE) or Synchronous (TRUE)
    .FIFO_MODE("FIFO36"),               // Sets mode to "FIFO36" or "FIFO36_72"
    .FIRST_WORD_FALL_THROUGH("FALSE"), // Sets the FIFO FWFT to &VALUES
    .INIT(72'h00000000000000000000),   // Initial values on output port
    .SIM_DEVICE("7SERIES"),             // Must be set to "7SERIES" for simulation behavior
    .SRVAL(72'h00000000000000000000)   // Set/Reset value for output port
)
FIFO36E1_inst (
    // ECC Signals: 1-bit (each) output: Error Correction Circuitry ports
    .DBITERR(DBITERR),                // 1-bit output: Double bit error status
    .ECCPARITY(ECCPARITY),             // 8-bit output: Generated error correction parity
    .SBITERR(SBITERR),                // 1-bit output: Single bit error status
    // Read Data: 64-bit (each) output: Read output data
    .DO(DO),                          // 64-bit output: Data output
    .DOP(DOP),                        // 8-bit output: Parity data output
    // Status: 1-bit (each) output: Flags and other FIFO status outputs
    .ALMOSTEMPTY(ALMOSTEMPTY),        // 1-bit output: Almost empty flag
    .ALMOSTFULL(ALMOSTFULL),          // 1-bit output: Almost full flag
    .EMPTY(EMPTY),                    // 1-bit output: Empty flag
    .FULL(FULL),                      // 1-bit output: Full flag
    .RDCOUNT(RDCOUNT),                // 13-bit output: Read count
    .RDERR(RDERR),                    // 1-bit output: Read error
    .WRCOUNT(WRCOUNT),                // 13-bit output: Write count
    .WRERR(WRERR),                    // 1-bit output: Write error
    // ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
    .INJECTDBITERR(INJECTDBITERR),    // 1-bit input: Inject a double bit error input
    .INJECTSBITERR(INJECTSBITERR),
    // Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
    .RDCLK(RDCLK),                    // 1-bit input: Read clock
    .RDEN(RDEN),                      // 1-bit input: Read enable
    .REGCE(REGCE),                    // 1-bit input: Clock enable
    .RST(RST),                        // 1-bit input: Reset
    .RSTREG(RSTREG),                  // 1-bit input: Output register set/reset
    // Write Control Signals: 1-bit (each) input: Write clock and enable input signals
    .WRCLK(WRCLK),                    // 1-bit input: Rising edge write clock.
    .WREN(WREN),                      // 1-bit input: Write enable
    // Write Data: 64-bit (each) input: Write input data
    .DI(DI),                          // 64-bit input: Data input
    .DIP(DIP),                        // 8-bit input: Parity input
);

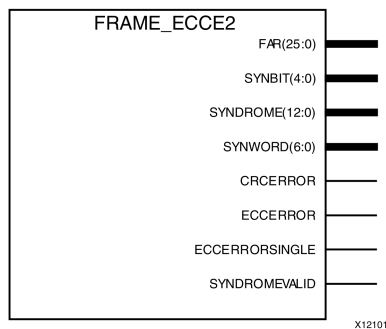
// End of FIFO36E1_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## FRAME\_ECCE2

### プリミティブ：Configuration Frame Error Correction



### 概要

このデザイン エLEMENTでは、FPGA のコンフィギュレーション メモリ向けの専用ビルトインエラー検出および修正 (ECC) が提供されます。このELEMENTには、ECC 回路のステータスおよびリードバック CRC 回路のステータスを監視する出力が含まれています。

### ポートの説明

ポート名	方向	幅	説明
CRCERROR	出力	1	CRC エラーを示す出力
ECCERROR	出力	1	ECC エラーを示す出力
ECCERRORSINGLE	出力	1	シングル ビット フレーム ECC エラーが検出されたことを示す出力
FAR<25:0>	出力	26	フレーム アドレス レジスタ値を示す出力
SYNBIT<4:0>	出力	5	エラーのビット アドレスを示す出力
SYNDROME<12:0>	出力	13	エラー ビットのロケーションを示す出力
SYNDROMEVALID	出力	1	SYNDROME 出力が有効であることを示す フレーム ECC 出力
SYNWORD<6:0>	出力	7	ECC エラーが検出されたフレーム内のワード出力

### 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
FARSRC	文字列	"EFAR"、"FAR"	"EFAR"	FAR[25:0] コンフィギュレーションレジスタの出力先を "FAR" または "EFAR" のいずれかに決定します。コンフィギュレーション オプションレジスタビット CTL0[7] を設定します。
FRAME_RBT_IN_FILENAME	文字列	ファイル名	なし	このファイルは、ICAP_E2 モデルにより出力され、RBT ファイルのフレーム データ情報が含まれています。FRAME_ECC モデルではこのファイルを解析して ECC が算出され、エラーがある場合は出力されます。

## 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;

-- FRAME_ECCE2: Configuration Frame Error Correction
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

FRAME_ECCE2_inst : FRAME_ECCE2
generic map (
    FARSRC => "EFAR",                -- Determines if the output of FAR[25:0] configuration register points
                                     -- to the FAR or EFAR. Sets configuration option register bit CTL0[7].
    FRAME_RBT_IN_FILENAME => "NONE"  -- This file is output by the ICAP_E2 model and it contains Frame Data
                                     -- information for the Raw Bitstream (RBT) file. The FRAME_ECCE2 model
                                     -- will parse this file, calculate ECC and output any error conditions.
)
port map (
    CRCERROR => CRCERROR,            -- 1-bit output: Output indicating a CRC error.
    ECCERROR => ECCERROR,            -- 1-bit output: Output indicating an ECC error.
    ECCERRORSINGLE => ECCERRORSINGLE, -- 1-bit output: Output Indicating single-bit Frame ECC error detected.
    FAR => FAR,                      -- 26-bit output: Frame Address Register Value output.
    SYNBIT => SYNBIT,                -- 5-bit output: Output bit address of error.
    SYNDROME => SYNDROME,            -- 13-bit output: Output location of erroneous bit.
    SYNDROMEVALID => SYNDROMEVALID, -- 1-bit output: Frame ECC output indicating the SYNDROME output is
                                     -- valid.

    SYNWORD => SYNWORD               -- 7-bit output: Word output in the frame where an ECC error has been
                                     -- detected.
);

-- End of FRAME_ECCE2_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// FRAME_ECCE2: Configuration Frame Error Correction
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

FRAME_ECCE2 #(
    .FARSRC("EFAR"),                // Determines if the output of FAR[25:0] configuration register points to
                                    // the FAR or EFAR. Sets configuration option register bit CTL0[7].
    .FRAME_RBT_IN_FILENAME("NONE") // This file is output by the ICAP_E2 model and it contains Frame Data
                                    // information for the Raw Bitstream (RBT) file. The FRAME_ECCE2 model
                                    // will parse this file, calculate ECC and output any error conditions.
)
FRAME_ECCE2_inst (
    .CRCERROR(CRCERROR),            // 1-bit output: Output indicating a CRC error.
    .ECCERROR(ECCERROR),            // 1-bit output: Output indicating an ECC error.
    .ECCERRORSINGLE(ECCERRORSINGLE), // 1-bit output: Output Indicating single-bit Frame ECC error detected.
    .FAR(FAR),                      // 26-bit output: Frame Address Register Value output.
    .SYNBIT(SYNBIT),                // 5-bit output: Output bit address of error.
    .SYNDROME(SYNDROME),            // 13-bit output: Output location of erroneous bit.
    .SYNDROMEVALID(SYNDROMEVALID), // 1-bit output: Frame ECC output indicating the SYNDROME output is
                                    // valid.

    .SYNWORD(SYNWORD)              // 7-bit output: Word output in the frame where an ECC error has been
                                    // detected.
);

// End of FRAME_ECCE2_inst instantiation
```

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## GTHE2\_CHANNEL

プリミティブ：Gigabit Transceiver

### 概要

GTHE2 は、7 シリーズ デバイスのギガビットトランシーバー コンポーネントです。このコンポーネントを正しくコンフィギュレーションするには、直接インスタンス化せずに、CORE Generator を使用してください。詳細は、7 シリーズ FPGA のトランシーバー ユーザー ガイド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	不可
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

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## GTHE2\_COMMON

**プリミティブ**：Gigabit Transceiver

### 概要

GTHE2 は、7 シリーズ デバイスのギガビットトランシーバー コンポーネントです。このコンポーネントを正しくコンフィギュレーションするには、直接インスタンス化せずに、CORE Generator を使用してください。詳細は、7 シリーズ FPGA のトランシーバー ユーザー ガイド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	不可
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

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## GTPE2\_CHANNEL

プリミティブ：Gigabit Transceiver

### 概要

GTPE2 は、7 シリーズ デバイスのギガビットトランシーバー コンポーネントです。このコンポーネントを正しくコンフィギュレーションするには、直接インスタンス化せずに、CORE Generator を使用してください。詳細は、7 シリーズ FPGA のトランシーバー ユーザー ガイド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	不可
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

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## GTPE2\_COMMON

**プリミティブ**：Gigabit Transceiver

### 概要

GTPE2 は、7 シリーズ デバイスのギガビットトランシーバー コンポーネントです。このコンポーネントを正しくコンフィギュレーションするには、直接インスタンス化せずに、CORE Generator を使用してください。詳細は、7 シリーズ FPGA のトランシーバー ユーザー ガイド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	不可
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

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## GTXE2\_CHANNEL

**プリミティブ**：Gigabit Transceiver for 7 Series

### 概要

GTXE2 は、7 シリーズ デバイスのギガビットトランシーバー コンポーネントです。このコンポーネントを正しくコンフィギュレーションするには、直接インスタンス化せずに、CORE Generator を使用してください。詳細は、7 シリーズ FPGA のトランシーバー ユーザー ガイド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	不可
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

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## GTXE2\_COMMON

### プリミティブ：Gigabit Transceiver for 7 Series

#### 概要

GTXE2 は、7 シリーズ デバイスのギガビットトランシーバー コンポーネントです。このコンポーネントを正しくコンフィギュレーションするには、直接インスタンス化せずに、CORE Generator を使用してください。詳細は、7 シリーズ FPGA のトランシーバー ユーザー ガイドを参照してください。

#### デザインの入力方法

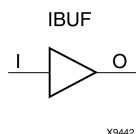
インスタンス化	不可
推論	不可
CORE Generator™ およびウィザード	可
マクロのサポート	不可

#### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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エートすることも可能です。インスタンスエートするには、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

このELEMENTは通常、デザインの最上位入力ポートに対して推論されます。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エートできます。このコンポーネントをインスタンスエートするには、該当するライブラリ ガイドに含まれるインスタンスエーション 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 コンポーネントを必ず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 ポートをこの入力が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
IBUF_LOW_PWR	ブール代数	TRUE、FALSE	TRUE	TRUE に設定すると、LVDS や HSTL などの差動入力規格 (V <sub>REF</sub> が必要) を使用する場合に消費電力を削減できます。FALSE に設定すると、TRUE に設定した場合より消費電力は大きくなりますが、パフォーマンスは向上します。詳細は、『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』を参照してください。
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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUF #(
    .IBUF_LOW_PWR("TRUE"), // Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
    .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
```

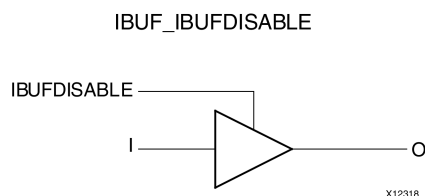
詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## IBUF\_IBUFDISABLE

プリミティブ：Single-ended Input Buffer with Input Disable



### 概要

このデザイン エLEMENTは、内部ロジックを外部ピンに接続する入力バッファです。I/O が長時間使用されないときに電力消費を削減する機能として、入力パス ディスエーブルが含まれています。

### ポートの説明

ポート名	方向	幅	説明
I	入力	1	入力ポート接続。デザインの最上位ポートに直接接続します。
IBUFDISABLE	入力	1	USE_IBUFDISABLE が "TRUE" に設定されている場合にこの信号が High にアサートされると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グランドに接続する必要があります。この機能は、I/O が長時間アイドル状態になったときに電力消費を削減するために使用します。
O	出力	1	デバイスへの入力パスを表すバッファ出力

### デザインの入力方法

インスタンス化	可
推論	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
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	参照 I/O 規格が使用された場合に、消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。

## 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;

-- IBUF_IBUFDISABLE: Single-ended Input Buffer with Disable
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IBUF_IBUFDISABLE_inst : IBUF_IBUFDISABLE
generic map (
    IBUF_LOW_PWR => "TRUE", -- Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
    IOSTANDARD => "DEFAULT", -- Specify the input I/O standard
    USE_IBUFDISABLE => "TRUE") -- Set to "TRUE" to enable IBUFDISABLE feature
port map (
    O => O,      -- Buffer output
    I => I,      -- Buffer input (connect directly to top-level port)
    IBUFDISABLE => IBUFDISABLE -- Buffer disable input, low=disable
);

-- End of IBUF_IBUFDISABLE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUF_IBUFDISABLE: Single-ended Input Buffer with Disable
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUF_IBUFDISABLE #(
    .IBUF_LOW_PWR("TRUE"), // Low power ("TRUE") vs. performance ("FALSE") for referenced I/O standards
    .IOSTANDARD("DEFAULT"), // Specify the input I/O standard
    .USE_IBUFDISABLE("TRUE") // Set to "TRUE" to enable IBUFDISABLE feature
) IBUF_IBUFDISABLE_inst (
    .O(O), // Buffer output
    .I(I), // Buffer input (connect directly to top-level port)
    .IBUFDISABLE(IBUFDISABLE) // Buffer disable input, low=disable
);

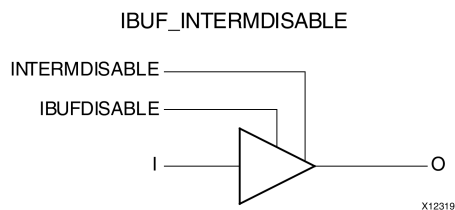
// End of IBUF_IBUFDISABLE_inst instantiation
```

### 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUF\_INTERMDISABLE

**プリミティブ：Single-ended Input Buffer with Input Termination Disable and Input Disable**



### 概要

このデザイン エLEMENTは、内部ロジックを外部ピンに接続する入力バッファです。I/O が長時間使用されないときに電力消費を削減する機能として、入力パス ディスエーブルと入力終端 (INTERM) イネーブル/ディスエーブルが含まれています。

### ポートの説明

ポート名	方向	幅	説明
I	入力	1	入力ポート接続。デザインの最上位ポートに直接接続します。
IBUFDISABLE	入力	1	USE_IBUFDISABLE が "TRUE" に設定されている場合にこの信号が High にアサートされると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グランドに接続する必要があります。この機能は、I/O が長時間アイドル状態になったときに電力消費を削減するために使用します。
INTERMDISABLE	入力	1	入力終端をディスエーブルにします。この機能は、I/O がアイドル状態になったときに電力消費を削減するために使用します。
O	出力	1	デバイスへの入力パスを表すバッファ出力

### デザインの入力方法

インスタンス化	可
推論	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
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	参照 I/O 規格が使用された場合に、消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。

## 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;

-- IBUF_INTERMDISABLE: Single-ended Input Buffer with Termination Input Disable
--                               May only be placed in High Range (HR) Banks
--                               7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IBUF_INTERMDISABLE_inst : IBUF_INTERMDISABLE
generic map (
    IBUF_LOW_PWR => "TRUE", -- Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
    IOSTANDARD => "DEFAULT", -- Specify the input I/O standard
    USE_IBUFDISABLE => "TRUE") -- Set to "TRUE" to enable IBUFDISABLE feature
port map (
    O => O,           -- Buffer output
    I => I,           -- Buffer input (connect directly to top-level port)
    INTERMDISABLE => INTERMDISABLE, -- Input Termination Disable
    IBUFDISABLE => IBUFDISABLE -- Buffer disable input, low=disable
);

-- End of IBUF_INTERMDISABLE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUF_INTERMDISABLE: Single-ended Input Buffer with Termination Input Disable
//                               May only be placed in High Range (HR) Banks
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUF_INTERMDISABLE #(
    .IBUF_LOW_PWR("TRUE"), // Low power ("TRUE") vs. performance ("FALSE") for referenced I/O standards
    .IOSTANDARD("DEFAULT"), // Specify the input I/O standard
    .USE_IBUFDISABLE("TRUE") // Set to "TRUE" to enable IBUFDISABLE feature
) IBUF_INTERMDISABLE_inst (
    .O(O), // Buffer output
    .I(I), // Buffer input (connect directly to top-level port)
    .IBUFDISABLE(IBUFDISABLE), // Buffer disable input, low=disable
    .INTERMDISABLE(INTERMDISABLE) // Input Termination Disable
);

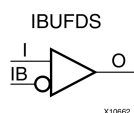
// End of IBUF_INTERMDISABLE_inst instantiation
```

詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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	7 シリーズ FPGA でサポートされる差動入力 I/O 規格用の差動終端属性で、ビルトインの差動終端をオン (TRUE) またはオフ (FALSE) にします。
IBUF_LOW_PWR	ブール代数	TRUE、FALSE	TRUE	TRUE に設定すると、LVDS や HSTL などの差動入力規格 ( $V_{REF}$ が必要) を使用する場合に消費電力を削減できます。FALSE に設定すると、TRUE に設定した場合より消費電力は大きくなりますが、パフォーマンスは向上します。詳細は、『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』を参照してください。
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。

## 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: Differential Input Buffer
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスレーション)

```
// IBUFDS: Differential Input Buffer
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUFDS #(
    .DIFF_TERM("FALSE"), // Differential Termination
    .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performance="FALSE"
    .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)
);

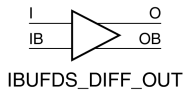
// End of IBUFDS_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFDS\_DIFF\_OUT

プリミティブ：Differential 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。IBUFGDS\_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 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	説明
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
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	7 シリーズ FPGA でサポートされる差動入力 I/O 規格用の差動終端属性で、ビルトインの差動終端をオン (TRUE) またはオフ (FALSE) にします。
IBUF_LOW_PWR	ブール代数	TRUE、FALSE	TRUE	TRUE に設定すると、LVDS や HSTL などの差動入力規格 ( $V_{REF}$ が必要) を使用する場合に消費電力を削減できます。FALSE に設定すると、TRUE に設定した場合より消費電力は大きくなりますが、パフォーマンスは向上します。詳細は、『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』を参照してください。
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。

## 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;

-- IBUFDS_DIFF_OUT: Differential Input Buffer with Differential Output
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IBUFDS_DIFF_OUT_inst : IBUFDS_DIFF_OUT
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", -- Specify the input I/O standard
  USE_IBUF_DISABLE => "TRUE"
)
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
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUFDS_DIFF_OUT #(
    .DIFF_TERM("FALSE"), // Differential Termination, "TRUE"/"FALSE"
    .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performance="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
```

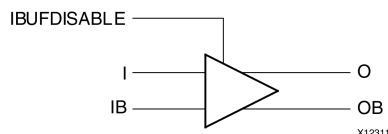
## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFDS\_DIFF\_OUT\_IBUFDISABLE

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

IBUFDS\_DIFF\_OUT\_IBUFDISABLE



### 概要

このデザイン エLEMENTは、内部ロジックを外部差動ピンに接続する差動入力バッファです。入力が長時間アイドル状態になったときに電力消費を削減する機能として、入力パス ディスエーブルが含まれています。IOBUFDS\_DIFF\_OUT\_IBUFDISABLE は、差動信号の両方の位相に内部アクセスできる点が IOBUFDS\_IBUFDISABLE と異なります。

### ポートの説明

ポート名	方向	幅	説明
I	入力	1	入力 p 側ポート接続。デザインの最上位ポートに直接接続します。
IB	入力	1	入力 n 側ポート接続。デザインの最上位ポートに直接接続します。
IBUFDISABLE	入力	1	USE_IBUFDISABLE が "TRUE" に設定されている場合にこの信号が High にアサートされると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グランドに接続する必要があります。この機能は、I/O が長時間アイドル状態になったときに電力消費を削減するために使用します。
O	出力	1	デバイスへの入力パスを表すバッファ p 側出力
OB	出力	1	デバイスへの入力パスを表すバッファ n 側出力

### 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	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
DIFF_TERM	文字列	"TRUE"、"FALSE"	"FALSE"	ビルトインの差動終端をイネーブルにします。
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	差動 I/O 規格が使用された場合に、消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。

## 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;

-- IBUFDS_DIFF_OUT_IBUFDISABLE: Differential Input Buffer with Differential Output w/ Disable
--                               7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IBUFDS_DIFF_OUT_IBUFDISABLE_inst : IBUFDS_DIFF_OUT_IBUFDISABLE
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", -- Specify the input I/O standard
    USE_IBUFDISABLE => "TRUE") -- Set to "TRUE" to enable IBUFDISABLE feature
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)
    IBUFDISABLE => IBUFDISABLE -- Buffer disable input, low=disable
);

-- End of IBUFDS_DIFF_OUT_IBUFDISABLE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUFDS_DIFF_OUT_IBUFDISABLE: Differential Input Buffer with Differential Output with Input Disable
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUFDS_DIFF_OUT_IBUFDISABLE #(
    .DIFF_TERM("FALSE"), // Differential Termination, "TRUE"/"FALSE"
    .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performance="FALSE"
    .IOSTANDARD("DEFAULT"), // Specify the input I/O standard
    .USE_IBUFDISABLE("TRUE") // Set to "TRUE" to enable IBUFDISABLE feature
) IBUFDS_DIFF_OUT_IBUFDISABLE_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)
    .IBUFDISABLE(IBUFDISABLE) // Buffer disable input, low=disable
);

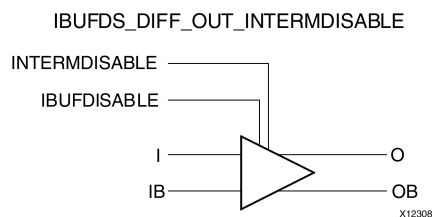
// End of IBUFDS_DIFF_OUT_IBUFDISABLE_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFDS\_DIFF\_OUT\_INTERMDISABLE

**プリミティブ**：Input Differential Buffer with Input Termination Disable, Input Disable, and Differential Output



### 概要

このデザイン エレメントは、内部ロジックを外部差動ピンに接続する差動入力バッファです。I/O が長時間アイドル状態になったときに電力消費を削減する機能として、入力バス ディスエーブルとキャリブレーションされない入力終端 (INTERM) ディスエーブルが含まれています。IOBUFDS\_DIFF\_OUT\_INTERMDISABLE は、差動信号の両方の位相に内部アクセスできる点が IOBUFDS\_INTERMDISABLE と異なります。このエレメントは、7 シリーズ デバイスの HR (High Range) バンクにのみ配置可能です。

### ポートの説明

ポート名	方向	幅	説明
I	入力	1	入力 p 側ポート接続。デザインの最上位ポートに直接接続します。
IB	入力	1	入力 n 側ポート接続。デザインの最上位ポートに直接接続します。
IBUFDSABLE	入力	1	USE_IBUFDSABLE が "TRUE" に設定されている場合にこの信号が High にアサートされると、バッファを介する入力バスがディスエーブルになり、ロジック High になります。USE_IBUFDSABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グランドに接続する必要があります。この機能は、I/O がアイドル状態になったときに電力消費を削減するために使用します。
INTERMDISABLE	入力	1	入力終端をディスエーブルにします。この機能は、I/O がアイドル状態になったときに電力消費を削減するために使用します。
O	出力	1	デバイスへの入力バスを表すバッファ p 側出力
OB	出力	1	デバイスへの入力バスを表すバッファ n 側出力

### 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	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
DIFF_TERM	文字列	"TRUE"、"FALSE"	"FALSE"	ビルトインの差動終端をイネーブルにします。
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	差動 I/O 規格が使用された場合に、消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。

## 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;

-- IBUFDS_DIFF_OUT_INTERMDISABLE: Differential Input Buffer with Differential Output w/ Disable
--                                     7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IBUFDS_DIFF_OUT_INTERMDISABLE_inst : IBUFDS_DIFF_OUT_INTERMDISABLE
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", -- Specify the input I/O standard
    USE_IBUFDISABLE => "TRUE") -- Set to "TRUE" to enable IBUFDISABLE feature
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)
    IBUFDISABLE => IBUFDISABLE, -- Buffer disable input, low=disable
    INTERMDISABLE => INTERMDISABLE -- Input termination disable
);

-- End of IBUFDS_DIFF_OUT_INTERMDISABLE_inst instantiation
```

## Verilog 記述 (インスタンスエーション)

```
// IBUFDS_DIFF_OUT_INTERMDISABLE: Differential Input Buffer with Differential Output with Input Termination Disable
//                                     May only be placed in High Range (HR) Banks
//                                     7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUFDS_DIFF_OUT_INTERMDISABLE #(
    .DIFF_TERM("FALSE"), // Differential Termination, "TRUE"/"FALSE"
    .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performance="FALSE"
    .IOSTANDARD("DEFAULT"), // Specify the input I/O standard
    .USE_IBUFDISABLE("TRUE") // Set to "TRUE" to enable IBUFDISABLE feature
) IBUFDS_DIFF_OUT_INTERMDISABLE_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)
    .IBUFDISABLE(IBUFDISABLE), // Buffer disable input, low=disable
    .INTERMDISABLE(INTERMDISABLE) // Input Termination Disable
);

// End of IBUFDS_DIFF_OUT_INTERMDISABLE_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFDS\_GTE2

**プリミティブ** :Gigabit Transiever Buffer

### 概要

IBUFDS\_GTE2 は、7 シリーズ デバイスのギガビットトランシーバーの入力パッド バッファ コンポーネントです。このコンポーネントを正しくコンフィギュレーションするには、直接インスタンス化せずに、CORE Generator を使用してください。詳細は、7 シリーズ FPGA のトランシーバー ユーザー ガイドを参照してください。

### デザインの入力方法

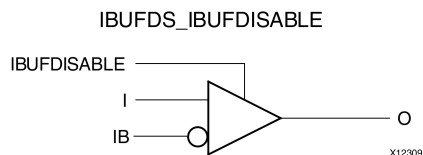
インスタンス化	不可
推論	不可
CORE Generator™ およびウィザード	可
マクロのサポート	不可

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFDS\_IBUFDISABLE

**プリミティブ：Input Differential Buffer with Input Path Disable**



### 概要

このデザイン エレメントは、内部ロジックを外部差動ピンに接続する入力差動バッファです。I/O が長時間使用されないときに電力消費を削減する機能として、入力パス ディスエーブルが含まれています。

### ポートの説明

ポート名	方向	幅	説明
I	入力	1	入力 p 側ポート接続。デザインの最上位ポートに直接接続します。
IB	入力	1	入力 n 側ポート接続。デザインの最上位ポートに直接接続します。
IBUFDISABLE	入力	1	USE_IBUFDISABLE が "TRUE" に設定されている場合にこの信号が High にアサートされると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グランドに接続する必要があります。この機能は、I/O がアイドル状態になったときに電力消費を削減するために使用します。
O	出力	1	デバイスへの入力パスを表すバッファ出力

### デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	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	文字列	"TRUE"、"FALSE"	"FALSE"	ビルトインの差動終端をイネーブルにします。
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	差動 I/O 規格が使用された場合に、消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。



## 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;

-- IBUFDS_IBUFDISABLE: Differential Input Buffer w/ Disable
--                               7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IBUFDS_IBUFDISABLE_inst : IBUFDS_IBUFDISABLE
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" -- Specify the input I/O standard
    USE_IBUFDISABLE => "TRUE") -- Set to "TRUE" to enable IBUFDISABLE feature
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)
    IBUFDISABLE => IBUFDISABLE -- Buffer disable input, low=disable
);

-- End of IBUFDS_IBUFDISABLE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUFDS_IBUFDISABLE: Differential Input Buffer with Input Disable
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUFDS_IBUFDISABLE #(
    .DIFF_TERM("FALSE"), // Differential Termination
    .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performance="FALSE"
    .IOSTANDARD("DEFAULT"), // Specify the input I/O standard
    .USE_IBUFDISABLE("TRUE") // Set to "TRUE" to enable IBUFDISABLE feature
) IBUFDS_IBUFDISABLE_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)
    .IBUFDISABLE(IBUFDISABLE) // Buffer disable input, low=disable
);

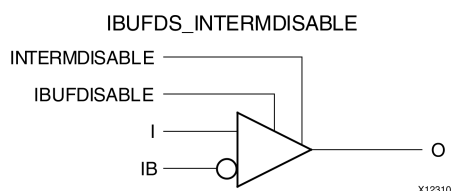
// End of IBUFDS_IBUFDISABLE_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFDS\_INTERMDISABLE

**プリミティブ：Input Differential Buffer with Input Termination Disable and Input Disable**



### 概要

このデザイン エレメントは、内部ロジックを外部差動ピンに接続する入力差動バッファです。入力が長時間アイドル状態になったときに電力消費を削減する機能として、入力パス ディスエーブルとキャリブレーションされない入力終端 (INTERM) ディスエーブルが含まれています。このエレメントは、7 シリーズ デバイスの HR (High Range) バンクにのみ配置可能です。

### ポートの説明

ポート名	方向	幅	説明
I	入力	1	入力 p 側ポート接続。デザインの最上位ポートに直接接続します。
IB	入力	1	入力 n 側ポート接続。デザインの最上位ポートに直接接続します。
IBUFDISABLE	入力	1	USE_IBUFDISABLE が "TRUE" に設定されている場合にこの信号が High にアサートされると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グランドに接続する必要があります。この機能は、I/O がアイドル状態になったときに電力消費を削減するために使用します。
INTERMDISABLE	入力	1	入力終端をディスエーブルにします。この機能は、I/O がアイドル状態になったときに電力消費を削減するために使用します。
O	出力	1	デバイスへの入力パスを表すバッファ出力

### デザインの入力方法

インスタンス化	可
推論	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
DIFF_TERM	文字列	"TRUE"、"FALSE"	"FALSE"	ビルトインの差動終端をイネーブルにします。
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	差動 I/O 規格が使用された場合に、消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE 機能をイネーブルまたはディスエーブルにします。通常、書き込み中の読み出しを許容するため入力パスをディスエーブルにするのが適切でない場合に使用します。

## 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;

-- IBUFDS_INTERMDISABLE: Differential Input Buffer with Input Termination Disable
--                               May only be placed in High Range (HR) Banks
--                               7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IBUFDS_INTERMDISABLE_inst : IBUFDS_INTERMDISABLE
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" -- Specify the input I/O standard
    USE_IBUFDISABLE => "TRUE") -- Set to "TRUE" to enable IBUFDISABLE feature
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)
    IBUFDISABLE => IBUFDISABLE, -- Buffer disable input, low=disable
    INTERMDISABLE => INTERMDISABLE -- Input termination disable
);

-- End of IBUFDS_IBUFDISABLE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUFDS_INTERMDISABLE: Differential Input Buffer with Input Termination Disable
//                               May only be placed in High Range (HR) Banks
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUFDS_INTERMDISABLE #(
    .DIFF_TERM("FALSE"), // Differential Termination
    .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performance="FALSE"
    .IOSTANDARD("DEFAULT"), // Specify the input I/O standard
    .USE_IBUFDISABLE("TRUE") // Set to "TRUE" to enable IBUFDISABLE feature
) IBUFDS_INTERMDISABLE_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)
    .IBUFDISABLE(IBUFDISABLE), // Buffer disable input, low=disable
    .INTERMDISABLE(INTERMDISABLE) // Input Termination Disable
);

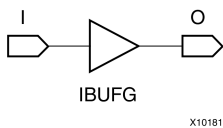
// End of IBUFDS_INTERMDISABLE_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFG

プリミティブ：Dedicated Input Clock Buffer



### 概要

IBUFG は、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 FPGA への専用入力です。最上位ポートからクロック マネージメント タイル (MMCM および PLL を含む) または BUFG への専用接続として使用でき、デバイスのクロック遅延とジッターを最小限に抑えます。IBUFG 入力は、クロック兼用ピン (MRCC または SRCC 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
IBUF_LOW_PWR	ブール代数	TRUE、FALSE	TRUE	消費電力とパフォーマンスのどちらを優先するかを選択します。 <ul style="list-style-type: none"> <li>TRUE に設定すると、LVDS や HSTL などの差動入力規格 (<math>V_{REF}</math> が必要) を使用する場合に消費電力を削減できます。</li> <li>FALSE に設定すると、TRUE に設定した場合より消費電力は大きくなりますが、パフォーマンスは向上します。</li> </ul> 詳細は、『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』を参照してください。
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。

## 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;

-- IBUFG: Single-ended global clock input buffer
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスレーション)

```
// IBUFG: Single-ended global clock input buffer
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUFG #(
    .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performance="FALSE"
    .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)
);

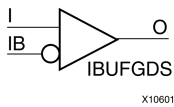
// End of IBUFG_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFGDS

プリミティブ：Differential Signaling Dedicated Input Clock Buffer



### 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。

### 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	内部差動終端抵抗を使用するかどうかを指定します。
IBUF_LOW_PWR	ブール代数	TRUE、FALSE	TRUE	TRUE に設定すると、LVDS や HSTL などの差動入力規格 ( $V_{REF}$ が必要) を使用する場合に消費電力を削減できます。FALSE に設定すると、TRUE に設定した場合より消費電力は大きくなりますが、パフォーマンスは向上します。詳細は、『7 シリーズ FPGA SelectIO リソース ユーザーガイド』を参照してください。
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。

## 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;

-- IBUFGDS: Differential Global Clock Input Buffer
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスエーション)

```
// IBUFGDS: Differential Global Clock Input Buffer
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUFGDS #(
    .DIFF_TERM("FALSE"), // Differential Termination
    .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performance="FALSE"
    .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
```

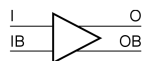
### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## IBUFGDS\_DIFF\_OUT

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



IBUFGDS\_DIFF\_OUT

X12011

## 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 では、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。
IBUF_LOW_PWR	ブール代数	TRUE、FALSE	TRUE	TRUE に設定すると、LVDS や HSTL などの差動入力規格 ( $V_{REF}$ が必要) を使用する場合に消費電力を削減できます。FALSE に設定すると、TRUE に設定した場合より消費電力は大きくなりますが、パフォーマンスは向上します。詳細は、『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』を参照してください。

## 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_DIFF_OUT: Differential Global Clock Buffer with Differential Output
--                               7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IBUFGDS_DIFF_OUT_inst : IBUFGDS_DIFF_OUT
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") -- 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 (インスタンスレーション)

```
// IBUFGDS_DIFF_OUT: Differential Global Clock Buffer with Differential Output
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

IBUFGDS_DIFF_OUT #(
    .DIFF_TERM("FALSE"), // Differential Termination, "TRUE"/"FALSE"
    .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performance="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)
);

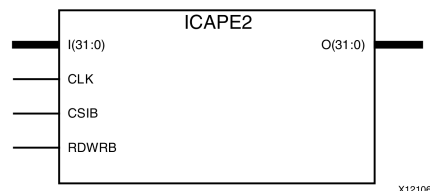
// End of IBUFGDS_DIFF_OUT_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ICAPE2

### 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	説明
CLK	入力	1	クロック入力
CSIB	入力	1	アクティブ Low の ICAP イネーブル
I<31:0>	入力	32	コンフィギュレーション データ入力バス
O<31:0>	出力	32	コンフィギュレーション データ出力バス
RDWRB	入力	1	読み出し/書き込みのセレクト入力

### 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
DEVICE_ID	16 進数	32'h03651093、 32'h036A2093、 32'h036A4093、 32'h036A6093、 32'h036BF093、 32'h036B1093、 32'h036B3093、 32'h036C2093、 32'h036C4093、 32'h036C6093、 32'h036DF093、 32'h036D1093、 32'h036D3093、 32'h036D5093、	0'h3651093	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。

属性	データ型	値	デフォルト	説明
		32'h036D9093、 32'h0362C093、 32'h0362D093、 32'h0363B093、 32'h0364C093、 32'h0371F093、 32'h0372C093、 32'h0377F093、 32'h03627093、 32'h03628093、 32'h03631093、 32'h03636093、 32'h03642093、 32'h03647093、 32'h03656093、 32'h03667093、 32'h03671093、 32'h03676093、 32'h03680093、 32'h03681093、 32'h03682093、 32'h03687093、 32'h03691093、 32'h03692093、 32'h03696093、 32'h03702093、 32'h03704093、 32'h03711093、 32'h03722093、 32'h03727093、 32'h03731093、 32'h03747093、 32'h03751093、 32'h03752093、 32'h03762093、 32'h03771093、 32'h03782093		
ICAP_WIDTH	文字列	"X32"、"X8"、"X16"	"X32"	入力および出力データ幅を指定します。
SIM_CFG_FILE_NAME	文字列	ファイル名	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;

-- ICAPE2: Internal Configuration Access Port
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

ICAPE2_inst : ICAPE2
generic map (
    DEVICE_ID => X"3651093",    -- Specifies the pre-programmed Device ID value to be used for simulation
                                -- purposes.
    ICAP_WIDTH => "X32",        -- Specifies the input and output data width.
    SIM_CFG_FILE_NAME => "NONE" -- Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
                                -- model.
)
port map (

```

```
O => O,          -- 32-bit output: Configuration data output bus
CLK => CLK,       -- 1-bit input: Clock Input
CSIB => CSIB,     -- 1-bit input: Active-Low ICAP Enable
I => I,           -- 32-bit input: Configuration data input bus
RDWRB => RDWRB    -- 1-bit input: Read/Write Select input
);

-- End of ICAPE2_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// ICAPE2: Internal Configuration Access Port
//      7 Series
// Xilinx HDL Libraries Guide, version 14.4

ICAPE2 #(
    .DEVICE_ID(0'h3651093),    // Specifies the pre-programmed Device ID value to be used for simulation
                                // purposes.
    .ICAP_WIDTH("X32"),       // Specifies the input and output data width.
    .SIM_CFG_FILE_NAME("NONE") // Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
                                // model.
)
ICAPE2_inst (
    .O(O),          // 32-bit output: Configuration data output bus
    .CLK(CLK),      // 1-bit input: Clock Input
    .CSIB(CSIB),    // 1-bit input: Active-Low ICAP Enable
    .I(I),          // 32-bit input: Configuration data input bus
    .RDWRB(RDWRB)   // 1-bit input: Read/Write Select input
);

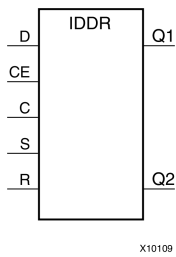
// End of ICAPE2_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IDDR

### プリミティブ：Input Dual Data-Rate Register



### 概要

このデザイン エLEMENTは、ザイリンクス FPGA に入力される外部デュアル 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。データが取り込まれるクロック エッジごとにデータを FPGA ファブリックに入力するモードと、同じクロック エッジで同時にデータを入力するモードがあります。これにより、タイミングが複雑にならず、追加のリソースも必要ありません。

- OPPOSITE\_EDGE モード：**通常の DDR 方式でデータが受信されます。Q1 はクロック C の各立ち上がりエッジの後に変化し、Q2 は各立ち下がりエッジの後に変化します。
- SAME\_EDGE モード：**データはクロック C の反対のエッジで受信されますが、立ち下がりエッジ データレジスタの前にレジスタが追加されます。このレジスタはクロック信号 C の立ち上がりエッジで動作するので、DDR データは同じクロック エッジで FPGA に送信されます。ただし、データ ペアは分離されているように見えます。Q1 と Q2 にはペア 1 および 2 が同時に送信されず、最初のペアがペア 1 とドントケアとなり、次のクロック サイクルでペア 2 と 3 が送信されます。
- SAME\_EDGE\_PIPELINED モード：**SAME\_EDGE モードと同様にデータが処理されますが、SAME\_EDGE モードでのデータ ペアの分離を回避するため、立ち上がりエッジ データレジスタの前にもレジスタが追加されるので、データ ペアが Q1 と Q2 ピンに同時に送信されます。ただし、このモードを使用すると、Q1 と Q2 信号が変化するレイテンシが 1 サイクル分増加します。

IDDR は IDELAYE2 などの SelectIO™ 機能とも使用できます。

**注記：**高速インターフェイスには、IDDR\_2CLK コンポーネントを使用して データの取り込みに 2 つの独立したクロックを指定できます。このコンポーネントは、IDDR のパフォーマンス要件が不十分のときに使用します。IDDR\_2CLK では、必要なクロック リソース数が増え、IDDR コンポーネントを使用するときには不要な配置制限が発生する可能性があります。

### ポートの説明

ポート名	方向	幅	説明
Q1 ~ Q2	出力	1	FPGA に接続する IDDR 出力。Q1 は最初のデータ ペア、Q2 は 2 番目のデータ ペアです。
C	入力	1	クロック入力ピン
CE	入力	1	Low になると、ポート O の出力クロックがディスエーブルになります。
D	入力	1	DDR データを IDDR モジュールに入力するピン  このピンは、最上位の入力または双方向ポート、入力遅延が設定された IDELAYE2、あるいは適切な入力または双方向バッファに接続します。

ポート名	方向	幅	説明
R	入力	1	アクティブ High のリセットで Q1 および Q2 を論理 0 にします。SRTYPE 属性に基づき、同期または非同期に設定できます。
S	入力	1	アクティブ High のリセットで Q1 および Q2 を論理 1 にします。SRTYPE 属性に基づき、同期または非同期に設定できます。

**注記：** このコンポーネントでセットとリセットを両方アクティブにすることはできません。R および S のどちらかまたは両方をグラウンドに接続する必要があります。

## 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	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_CLK_EDGE	文字列	"OPPOSITE_EDGE"、 "SAME_EDGE"、 "SAME_EDGE_PIPELINED"	"OPPOSITE_EDGE"	クロック エッジに対する IDDR の操作モードを指定します。
INIT_Q1	2 進数	0、1	0	コンフィギュレーションのスタートアップ後または GSR がアサートされたときの Q1 ピンの初期値を指定します。
INIT_Q2	2 進数	0、1	0	コンフィギュレーションのスタートアップ後または GSR がアサートされたときの Q1 ピンの初期値を指定します。
SRTYPE	文字列	"SYNC"、"ASYNC"	"SYNC"	セット/リセットのタイプを選択します。"SYNC" に設定すると、リセット (R) およびセット (S) ピンの動作が C クロック ピンの立ち上がりエッジに同期し、"ASYNC" に設定すると非同期動作になります。

## 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;

-- IDDR: Double Data Rate Input Register with Set, Reset
-- and Clock Enable.
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IDDR_inst : IDDR
generic map (
  DDR_CLK_EDGE => "OPPOSITE_EDGE", -- "OPPOSITE_EDGE", "SAME_EDGE"
                                         -- or "SAME_EDGE_PIPELINED"
  INIT_Q1 => '0', -- Initial value of Q1: '0' or '1'
  INIT_Q2 => '0', -- Initial value of Q2: '0' or '1'
```



```

    SRTYPE => "SYNC") -- Set/Reset type: "SYNC" or "ASYNC"
port map (
    Q1 => Q1, -- 1-bit output for positive edge of clock
    Q2 => Q2, -- 1-bit output for negative edge of clock
    C => C,   -- 1-bit clock input
    CE => CE, -- 1-bit clock enable input
    D => D,   -- 1-bit DDR data input
    R => R,   -- 1-bit reset
    S => S    -- 1-bit set
);

-- End of IDDR_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// IDDR: Input Double Data Rate Input Register with Set, Reset
//         and Clock Enable.
//         7 Series
// Xilinx HDL Libraries Guide, version 14.4

IDDR #(
    .DDR_CLK_EDGE("OPPOSITE_EDGE"), // "OPPOSITE_EDGE", "SAME_EDGE"
                                     // or "SAME_EDGE_PIPELINED"
    .INIT_Q1(1'b0), // Initial value of Q1: 1'b0 or 1'b1
    .INIT_Q2(1'b0), // Initial value of Q2: 1'b0 or 1'b1
    .SRTYPE("SYNC") // Set/Reset type: "SYNC" or "ASYNC"
) IDDR_inst (
    .Q1(Q1), // 1-bit output for positive edge of clock
    .Q2(Q2), // 1-bit output for negative edge of clock
    .C(C),   // 1-bit clock input
    .CE(CE), // 1-bit clock enable input
    .D(D),   // 1-bit DDR data input
    .R(R),   // 1-bit reset
    .S(S)    // 1-bit set
);

// End of IDDR_inst instantiation

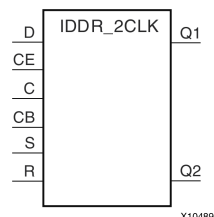
```

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IDDR\_2CLK

### プリミティブ：Input Dual Data-Rate Register with Dual Clock Inputs



### 概要

このデザイン エLEMENTは、ザイリンクス FPGA に入力される外部デュアル 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。IDDR\_2CLK を使用すると、必要なクロック リソース数および消費電力が増加し、IDDR コンポーネントを使用するときには不要な配置制限があるため、超高速インターフェイスにのみ使用してください。IDDR コンポーネントを使用すると、同じ I/O 速度は得られませんが、簡単に使用でき、リソース数も抑えられ、制限も少なくなります。このプリミティブには、データが取り込まれるクロック エッジごとにデータを FPGA ファブリックに入力するモードと、同じクロック エッジで同時にデータを入力するモードがあります。これにより、タイミングが複雑にならず、追加のリソースも必要ありません。

- OPPOSITE\_EDGE モード：**通常の DDR 方式でデータが受信されます。Q1 はクロック C の各立ち上がりエッジの後に変化し、Q2 はクロック CB の各立ち下がりエッジの後に変化します。
- SAME\_EDGE モード：**データは各クロックの立ち上がりエッジで受信されますが、CB クロック データレジスタの前にレジスタが 1 つ追加されます。このレジスタはクロック信号 C の立ち上がりエッジで動作するので、DDR データは同じクロック エッジで FPGA に送信されます。ただし、データ ペアは分離されているように見えます。Q1 と Q2 にはペア 1 および 2 が同時に送信されず、最初のペアがペア 1 とドントケアとなり、次のクロック サイクルでペア 2 と 3 が送信されます。
- SAME\_EDGE\_PIPELINED モード：**SAME\_EDGE モードと同様にデータが処理されますが、SAME\_EDGE モードでのデータ ペアの分離を回避するため、C のクロック データレジスタの前にもレジスタが追加されるので、C の立ち上がりエッジでデータ ペアが Q1 と Q2 ピンに同時に送信されます。ただし、このモードを使用すると、Q1 と Q2 信号が変化するレイテンシが 1 サイクル分増加します。

IDDR は IODELAYE2 などの SelectIO™ 機能とも使用できます。

### ポートの説明

ポート名	方向	幅	機能
Q1 : Q2	出力	1	FPGA に接続する IDDR 出力。Q1 は最初のデータ ペア、Q2 は 2 番目のデータ ペアです。
C	入力	1	立ち上がりエッジのデータをキャプチャするプライマリ クロック 入力ピン
CB	入力	1	立ち下がりエッジのデータをキャプチャするセカンダリ クロック 入力ピン。通常プライマリ クロックと 180 度位相がずれています。
CE	入力	1	Low になると、ポート O の出力クロックがディスエーブルになります。
D	入力	1	DDR データを IDDR モジュールに入力するピン  このピンは、最上位の入力または双方向ポート、入力遅延が設定された IODELAY、あるいは適切な入力または双方向バッファに接続します。

ポート名	方向	幅	機能
R	入力	1	アクティブ High のリセットで Q1 および Q2 を論理 0 にします。SRTYPE 属性に基づき、同期または非同期に設定できます。
S	入力	1	アクティブ High のリセットで Q1 および Q2 を論理 1 にします。SRTYPE 属性に基づき、同期または非同期に設定できます。

## 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

- ・ C ピンを立ち上がりクロック エッジを示すクロック ソースに、CB ピンを立ち下がりクロック エッジを示すクロック ソースに接続します。
- ・ D ピンを最上位の入力または双方向ポート、IODELAY、あるいはインスタンス化された入力または双方向バッファに接続します。
- ・ Q1 および Q2 ピンは、適切なデータ ソースに接続する必要があります。
- ・ CE ピンは、未使用の場合は High に接続し、使用する場合は適切なクロック イネーブル ロジックに接続します。
- ・ R および S ピンは、未使用の場合は Low に接続し、使用する場合は適切なセット/リセット生成ロジックに接続します。
- ・ 目的の動作になるように、コンポーネントに属性を設定します。
- ・ このペアのコンポーネントは同じクロックを使用してインスタンス化し、使用可能な I/O リソースを無駄にしないように、I/O ペアの P および N に LOC 制約を使用して固定します。
- ・ このコンポーネントは、常にほかの I/O コンポーネントと共にコードの最上位階層にインスタンス化します。これにより、階層デザイン フローを適切に実行できるようになります。
- ・ CLK スキューを最小限に抑えるには、CLK および CLKB の両方が、ローカル反転からではなく、グローバル配線 (MMCM) から供給されるようにする必要があります。ローカル反転を使用するとスキューが追加されますが、MMCM を使用するとスキューが抑えられます。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DDR_CLK_EDGE	文字列	"OPPOSITE_EDGE"、 "SAME_EDGE"、 "SAME_EDGE_PIPELINED"	"OPPOSITE_EDGE"	クロック エッジに対する DDR の操作モードを指定します。詳細は、「概要」を参照してください。
INIT_Q1	2 進数	0、1	0	コンフィギュレーションのスタートアップ後または GSR がアサートされたときの Q1 ピンの初期値を指定します。
INIT_Q2	2 進数	0、1	0	コンフィギュレーションのスタートアップ後または GSR がアサートされたときの Q2 ピンの初期値を指定します。
SRTYPE	文字列	"SYNC"、"ASYN"	"SYNC"	セット/リセットのタイプを選択します。 "SYNC" に設定すると、リセット (R) およびセット (S) ピンの動作が C クロックピンの立ち上がりエッジに同期し、"ASYN" に設定すると非同期動作になります。

## 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;

-- IDDR_2CLK: Dual-Clock, Input Double Data Rate Input Register with
--           Set, Reset and Clock Enable.
--           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IDDR_2CLK_inst : IDDR_2CLK
generic map (
  DDR_CLK_EDGE => "OPPOSITE_EDGE", -- "OPPOSITE_EDGE", "SAME_EDGE"
                                   -- or "SAME_EDGE_PIPELINED"
  INIT_Q1 => '0', -- Initial value of Q1: '0' or '1'
  INIT_Q2 => '0', -- Initial value of Q2: '0' or '1'
  SRTYPE => "SYNC") -- Set/Reset type: "SYNC" or "ASYN"
port map (
  Q1 => Q1, -- 1-bit output for positive edge of clock
  Q2 => Q2, -- 1-bit output for negative edge of clock
  C => C,   -- 1-bit primary clock input
  CB => CB, -- 1-bit secondary clock input
  CE => CE, -- 1-bit clock enable input
  D => D,   -- 1-bit DDR data input
  R => R,   -- 1-bit reset
  S => S    -- 1-bit set
);

-- End of IDDR_2CLK_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IDDR_2CLK: Dual-Clock, Input Double Data Rate Input Register with
//           Set, Reset and Clock Enable.
//           7 Series
// Xilinx HDL Libraries Guide, version 14.4

IDDR_2CLK #(
  .DDR_CLK_EDGE("OPPOSITE_EDGE"), // "OPPOSITE_EDGE", "SAME_EDGE"
                                   // or "SAME_EDGE_PIPELINED"
  .INIT_Q1(1'b0), // Initial value of Q1: 1'b0 or 1'b1
  .INIT_Q2(1'b0), // Initial value of Q2: 1'b0 or 1'b1
  .SRTYPE("SYNC")) // Set/Reset type: "SYNC" or "ASYN"
) IDDR_2CLK_inst (
  .Q1(Q1), // 1-bit output for positive edge of clock
  .Q2(Q2), // 1-bit output for negative edge of clock
  .C(C),   // 1-bit primary clock input
  .CB(CB), // 1-bit secondary clock input
  .CE(CE), // 1-bit clock enable input
  .D(D),   // 1-bit DDR data input
  .R(R),   // 1-bit reset
  .S(S)    // 1-bit set
);

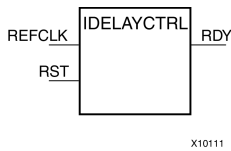
// End of IDDR_2CLK_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IDELAYCTRL

### プリミティブ：IDELAYE2/ODELAYE2 Tap Delay Value Control



### 概要

IDELAYE2 または ODELAYE2 を使用する場合は、このデザイン エLEMENTを使用する必要があります。IDELAYCTRL モジュールは、PVT（プロセス、電圧、および温度）の変動にかかわらず、関連付けられている IDELAYE2 および ODELAYE2 コンポーネント用に正確な遅延タップ値を求めるため、内部回路で電圧バイアスを算出するのに使用する基準クロック入力を提供します。このコンポーネントをインスタンス化するには、どの IDELAYCTRL がどの IDELAYE2 または ODELAYE2 に関連付けられているかを判別するため、IODELAY\_GROUP 属性を使用することをお勧めします。IODELAY\_GROUP 属性の詳細は、『制約ガイド』を参照してください。

### ポートの説明

ポート名	方向	幅	説明
RDY	出力	1	特定の領域の IDELAYE2 および ODELAYE2 モジュールがキャリブレーションされたことを示します。REFCLK が High または Low に 1 クロック周期以上保持されると、RDY 信号はディアサートされます。RDY が Low にディアサートされた場合は、IDELAYCTRL モジュールをリセットする必要があります。RDY を使用しない場合は、未接続にするか無視します。
REFCLK	入力	1	同じ領域にあるすべての IDELAYE2 および ODELAYE2 モジュールをキャリブレーションするための IDELAYCTRL に対する時間の基準です。ユーザー ソースまたは MMCME2/PLLE2 から直接供給でき、グローバル クロック バッファに配線する必要があります。
RST	入力	1	アクティブ High の非同期リセット。IDELAYE2 および ODELAYE2 を正しく動作させるため、コンフィギュレーション後 REFCLK 信号が安定した後 IDELAYCTRL をリセットする必要があります。リセット パルス幅 Tidelayctrl_rpw が必要です。

**RST (モジュールリセット)：**IDELAYCTRL 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。IDELAYCTRL をリセットするには、このポートを 50ns 以上 High にアサートする必要があります。

**REFCLK (基準クロック)：**プロセス、電圧、温度の変化に影響されずに、タップ遅延ラインに電圧バイアスを供給します。タップ遅延をデータシートに記載された値にするには、REFCLK の周波数を 200MHz にする必要があります。

**RDY (Ready 出力)：**基準クロック入力 REFCLK が有効になったことを示します。REFCLK が停止する (REFCLK が High または Low に 1 クロック周期以上保持される) と、ディアサートされます。

### 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

## 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;

-- IDELAYCTRL: IDELAYE2/ODELAYE2 Tap Delay Value Control
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IDELAYCTRL_inst : IDELAYCTRL
port map (
    RDY => RDY,          -- 1-bit output: Ready output
    REFCLK => REFCLK,    -- 1-bit input: Reference clock input
    RST => RST           -- 1-bit input: Active high reset input
);

-- End of IDELAYCTRL_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IDELAYCTRL: IDELAYE2/ODELAYE2 Tap Delay Value Control
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

(* IDELAY_GROUP = <iodelay_group_name> *) // Specifies group name for associated IDELAYs/ODELAYs and IDELAYCTRL

IDELAYCTRL IDELAYCTRL_inst (
    .RDY(RDY),           // 1-bit output: Ready output
    .REFCLK(REFCLK),    // 1-bit input: Reference clock input
    .RST(RST)           // 1-bit input: Active high reset input
);

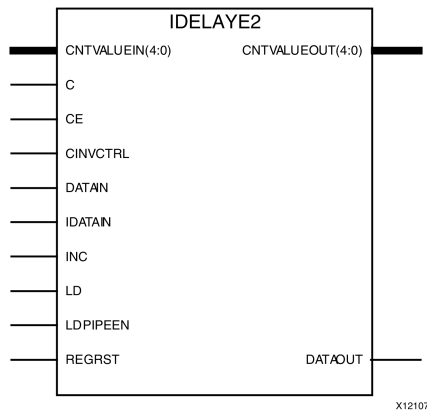
// End of IDELAYCTRL_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IDELAYE2

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



### 概要

各 I/O ブロックには、IDELAYE2 と呼ばれるプログラム可能な絶対遅延エレメントが含まれています。この遅延エレメントは、入力レジスタ/ISERDESE1 に接続するか、または FPGA ロジックを直接駆動できます。IDELAYE2 は、キャリブレーションされたタップ精度を使用する 31 タップのラップアラウンド遅延エレメントです。遅延値は、7 シリーズ FPGA データシートを参照してください。IDELAYE2 を使用すると、入力信号を個別に遅延できます。タップ遅延精度は、IDELAYCTRL 基準クロックを 7 シリーズ FPGA データシートで指定された範囲から選択することによって変更できます。

### ポートの説明

ポート名	方向	幅	説明
C	入力	1	IDELAYE2 プリミティブの制御入力 (RST、CE、および INC) は、すべてクロック入力 (C) に同期しています。IDELAYE2 を "VARIABLE"、"VAR_LOAD"、または "VAR_LOAD_PIPE" モードでコンフィギュレーションする場合は、クロックをこのポートに接続する必要があります。C はローカルで反転でき、グローバルまたはリージョナル クロック バッファから供給する必要があります。このクロックは SelectIO ロジックリソースの同じクロックに接続する必要があります (ISERDESE2 および OSERDESE2 を使用する場合は C は CLKDIV に接続)。
CE	入力	1	インクリメント/デクリメントをイネーブル/ディスエーブルにするアクティブ High の信号
CINVCTRL	入力	1	C ピンの極性を動的に切り替えます。この機能は、グリッチが問題にならないアプリケーションで使用します。極性を切り替える際は、2 クロック サイクル間 IDELAYE2 制御ピンを使用しないでください。
CNTVALUEIN<4:0>	入力	5	動的に読み込まれるタップ値用の FPGA ロジックからのカウンター値
CNTVALUEOUT<4:0>	出力	5	遅延エレメントの値の動的な変更をレポートします。IDELAYE2 が "VAR_LOAD" または "VAR_LOAD_PIPE" モードの場合にのみ有効です。
DATAIN	入力	1	FPGA ロジックで直接駆動され、ロジックでアクセス可能な遅延ラインとなります。データは、DATAOUT ポートを介して IDELAY_VALUE で設定された遅延で FPGA ロジックにフィードバックされます。DATAIN はローカルで反転可能です。データを I/O に駆動することはできません。
DATAOUT	出力	1	IDATAIN または DATAIN 入力パスからの遅延データです。ISERDESE2、入力レジスタ、または FPGA ロジックに接続されます。
IDATAIN	入力	1	関連付けられている I/O で駆動されます。データは ISERDESE1 または入力レジスタブロックに入力するか、FPGA ロジックに直接入力するか、ま

ポート名	方向	幅	説明
			または DATAOUT ポートを介して IDELAY_VALUE で設定された遅延で両方に入力できます。
INC	入力	1	タップ遅延のインクリメント/デクリメント数
LD	入力	1	カウンタに IDELAY_VALUE を読み込みます。
LDPIPEEN	入力	1	LD ピンからデータを読み込む際にパイプライン レジスタをイネーブルにします。
REGRST	入力	1	“VARIABLE” モードでは、遅延エレメントを IDELAY_VALUE 属性で設定された値にリセットします。この属性が設定されていない場合は、0 にリセットします。RST はアクティブ High のリセットで、入力クロック信号 (C) に同期しています。“VAR_LOAD” または “VAR_LOAD_PIPE” モードの場合は、遅延エレメントを CNTVALUEIN で設定された値にリセットします。CNTVALUEIN の値が新しいタップ値になります。この場合、IDELAY_VALUE 属性は無視されます。

## 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
CINVCTRL_SEL	文字列	“FALSE”、“TRUE”	“FALSE”	C ピンの極性を動的に切り替える CINVCTRL_SEL ピンをイネーブルにします。
DELAY_SRC	文字列	“IDATAIN”、“DATAIN”	“IDATAIN”	IDELAYE2 に入力する遅延ソースを指定します。 <ul style="list-style-type: none"> <li>“IDATAIN”：IDELAYE2 チェーンの入力は IDATAIN</li> <li>“DATAIN”：IDELAYE2 チェーンの入力は DATAIN</li> </ul>
HIGH_PERFORMANCE_MODE	文字列	“FALSE”、“TRUE”	“FALSE”	“TRUE” に設定すると出力ジッターが減少し、“FALSE” に設定すると消費電力が削減されます。消費電力量の差異は、Xilinx Power Estimator ツールで確認できます。
IDELAY_TYPE	文字列	“FIXED”、“VARIABLE”、“VAR_LOAD”、“VAR_LOAD_PIPE”	“FIXED”	タップ遅延ラインのタイプを設定します。 <ul style="list-style-type: none"> <li>“FIXED”：スタティック遅延値に設定します。</li> <li>“VARIABLE”：遅延値を動的に調整（インクリメントまたはデクリメント）します。</li> <li>“VAR_LOADABLE”：タップ値を動的に読み込みます。</li> </ul>



属性	データ型	値	デフォルト	説明
				・ “VAR_LOAD_PIPE”：タップ値をパイプラインを介して動的に読み込みます。
IDELAY_VALUE	10 進数	0、1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31	0	“FIXED” モードでは遅延タップ数、“VARIABLE” モードでは遅延タップ数の初期値を指定します (入力パス)。“VAR_LOAD” または “VAR_LOAD_PIPE” モードの場合、この属性値は無視されます。
PIPE_SEL	文字列	“FALSE”、“TRUE”	“FALSE”	パイプライン モードを選択します。
REFCLK_FREQUENCY	1 上位ビット FLOAT	190.0 ～ 210.0 および 290.0 ～ 310.0	200.0	Timing Analyzer でスタティック タイミング解析、論理シミュレーション、タイミングシミュレーションに使用するタップ値 (MHz) を設定します。適切なタップ遅延値およびパフォーマンスを得るには、REFCLK の周波数をデータシートに記載された範囲内にする必要があります。
SIGNAL_PATTERN	文字列	“DATA”、“CLOCK”	“DATA”	Timing Analyzer でデータパスまたはクロックパスに対して適切な遅延チェーンジッター量が使用されるようにします。

## 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;

-- IDELAYE2: Input Fixed or Variable Delay Element
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IDELAYE2_inst : IDELAYE2
generic map (
    CINVCTRL_SEL => "FALSE",           -- Enable dynamic clock inversion (FALSE, TRUE)
    DELAY_SRC => "IDATAIN",             -- Delay input (IDATAIN, DATAIN)
    HIGH_PERFORMANCE_MODE => "FALSE",   -- Reduced jitter ("TRUE"), Reduced power ("FALSE")
    IDELAY_TYPE => "FIXED",             -- FIXED, VARIABLE, VAR_LOAD, VAR_LOAD_PIPE
    IDELAY_VALUE => 0,                  -- Input delay tap setting (0-31)
    PIPE_SEL => "FALSE",                -- Select pipelined mode, FALSE, TRUE
    REFCLK_FREQUENCY => 200.0,          -- IDELAYCTRL clock input frequency in MHz (190.0-210.0).
    SIGNAL_PATTERN => "DATA"            -- DATA, CLOCK input signal
)
port map (
    CNTVALUEOUT => CNTVALUEOUT, -- 5-bit output: Counter value output
    DATAOUT => DATAOUT,       -- 1-bit output: Delayed data output
    C => C,                     -- 1-bit input: Clock input
    CE => CE,                   -- 1-bit input: Active high enable increment/decrement input
    CINVCTRL => CINVCTRL,       -- 1-bit input: Dynamic clock inversion input
    CNTVALUEIN => CNTVALUEIN,   -- 5-bit input: Counter value input
    DATAIN => DATAIN,         -- 1-bit input: Internal delay data input
    IDATAIN => IDATAIN,         -- 1-bit input: Data input from the I/O
    INC => INC,                 -- 1-bit input: Increment / Decrement tap delay input
    LD => LD,                   -- 1-bit input: Load IDELAY_VALUE input
    LDPIPEEN => LDPIPEEN,       -- 1-bit input: Enable PIPELINE register to load data input
    REGRST => REGRST            -- 1-bit input: Active-high reset tap-delay input
);

-- End of IDELAYE2_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```
// IDELAYE2: Input Fixed or Variable Delay Element
//      7 Series
// Xilinx HDL Libraries Guide, version 14.4

(* IDELAY_GROUP = <iodelay_group_name> *) // Specifies group name for associated IDELAYs/ODELAYs and IDELAYCTRL

IDELAYE2 #(
    .CINVCTRL_SEL("FALSE"),           // Enable dynamic clock inversion (FALSE, TRUE)
    .DELAY_SRC("IDATAIN"),            // Delay input (IDATAIN, DATAIN)
    .HIGH_PERFORMANCE_MODE("FALSE"), // Reduced jitter ("TRUE"), Reduced power ("FALSE")
    .IDELAY_TYPE("FIXED"),            // FIXED, VARIABLE, VAR_LOAD, VAR_LOAD_PIPE
    .IDELAY_VALUE(0),                 // Input delay tap setting (0-31)
    .PIPE_SEL("FALSE"),               // Select pipelined mode, FALSE, TRUE
    .REFCLK_FREQUENCY(200.0),         // IDELAYCTRL clock input frequency in MHz (190.0-210.0).
    .SIGNAL_PATTERN("DATA")           // DATA, CLOCK input signal
)
IDELAYE2_inst (
    .CNTVALUEOUT(CNTVALUEOUT), // 5-bit output: Counter value output
    .DATAOUT(DATAOUT),         // 1-bit output: Delayed data output
    .C(C),                     // 1-bit input: Clock input
    .CE(CE),                   // 1-bit input: Active high enable increment/decrement input
    .CINVCTRL(CINVCTRL),      // 1-bit input: Dynamic clock inversion input
    .CNTVALUEIN(CNTVALUEIN),   // 5-bit input: Counter value input
    .DATAIN(DATAIN),          // 1-bit input: Internal delay data input
    .IDATAIN(IDATAIN),         // 1-bit input: Data input from the I/O
    .INC(INC),                 // 1-bit input: Increment / Decrement tap delay input
    .LD(LD),                   // 1-bit input: Load IDELAY_VALUE input
    .LDPIPEEN(LDPIPEEN),       // 1-bit input: Enable PIPELINE register to load data input
    .REGRST(REGRST)            // 1-bit input: Active-high reset tap-delay input
);

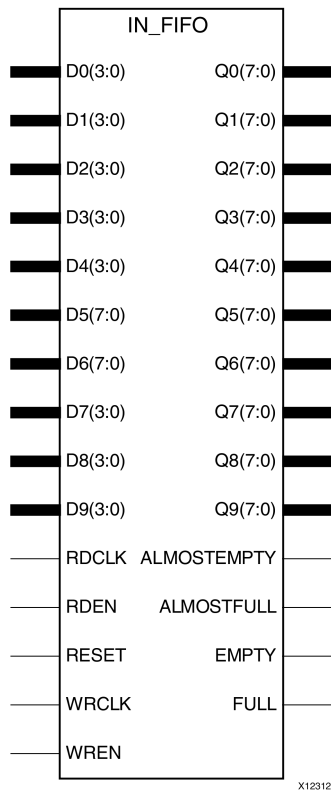
// End of IDELAYE2_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IN\_FIFO

プリミティブ : Input First-In, First-Out (FIFO)



## 概要

入力 FIFO は、I/O の横に配置されている新しいリソースです。この専用ハードウェアは、データを入力ポート、入力レジスタ、IDDR、または ISERDES からファブリックに転送するために設計されたものです。FIFO に入力されたデータが同じレートで出力される 4x4 モードと、データが係数 2 でデシリアライズされてから出力される 4x8 モードがあります。つまり、4x8 モードでは 4 ビットのデータが IN\_FIFO に入力され、8 ビットのデータが出力されます。このコンポーネントには、次の機能があります。

- ・ アレイ サイズ : 幅 80、深さ 8 (4x8 モード)、幅 40、深さ 8 (4x4 モード)
- ・ EMPTY および FULL フラグ
- ・ プログラム可能な ALMOSTEMPTY および ALMOSTFULL フラグ

## ポートの説明

ポート名	方向	幅	説明
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示すアクティブ High の出力フラグ。しきい値は ALMOST_EMPTY_VALUE 属性で設定します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示すアクティブ High の出力フラグ。しきい値は ALMOST_FULL_VALUE 属性で設定します。
D0<3:0>	入力	4	チャネル 0 入力バス
D1<3:0>	入力	4	チャネル 1 入力バス
D2<3:0>	入力	4	チャネル 2 入力バス
D3<3:0>	入力	4	チャネル 3 入力バス
D4<3:0>	入力	4	チャネル 4 入力バス
D5<7:0>	入力	8	チャネル 5 入力バス
D6<7:0>	入力	8	チャネル 6 入力バス
D7<3:0>	入力	4	チャネル 7 入力バス
D8<3:0>	入力	4	チャネル 8 入力バス
D9<3:0>	入力	4	チャネル 9 入力バス
EMPTY	出力	1	FIFO が空であることを示すアクティブ High の出力フラグ
FULL	出力	1	FIFO がフルであることを示すアクティブ High の出力フラグ
Q0<7:0>	出力	8	チャネル 0 出力バス
Q1<7:0>	出力	8	チャネル 1 出力バス
Q2<7:0>	出力	8	チャネル 2 出力バス
Q3<7:0>	出力	8	チャネル 3 出力バス
Q4<7:0>	出力	8	チャネル 4 出力バス
Q5<7:0>	出力	8	チャネル 5 出力バス
Q6<7:0>	出力	8	チャネル 6 出力バス
Q7<7:0>	出力	8	チャネル 7 出力バス
Q8<7:0>	出力	8	チャネル 8 出力バス
Q9<7:0>	出力	8	チャネル 9 出力バス
RDCLK	入力	1	読み出しクロック
RDEN	入力	1	アクティブ High のリード イネーブル
RESET	入力	1	アクティブ Low の非同期リセット
WRCLK	入力	1	書き込みクロック
WREN	入力	1	アクティブ High のライト イネーブル

## デザインの入力方法

インスタンス化	可
推論	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
ALMOST_EMPTY_VALUE	10 進数	1, 2	1	ALMOSTEMPTY 出力信号をアサートする際に FIFO にあるエントリ数を指定します。
ALMOST_FULL_VALUE	10 進数	1, 2	1	ALMOSTFULL 出力信号をアサートする際に FIFO にあるエントリ数を指定します。
ARRAY_MODE	文字列	"ARRAY_MODE_4_X_8", "ARRAY_MODE_4_X_4"	"ARRAY_MODE_4_X_8"	デシリアライザー モードを指定します。  ・ "ARRAY_MODE_8_X_8": 8 ビット データを入力し、8 ビット データを出力します。  ・ "ARRAY_MODE_4_X_8": 4 ビット データを入力し、8 ビット データを出力します。
SYNCHRONOUS_MODE	文字列	"FALSE"	"FALSE"	RDCLK と WRCLK を同期させ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;

-- IN_FIFO: Input First-In, First-Out (FIFO)
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IN_FIFO_inst : IN_FIFO
generic map (
    ALMOST_EMPTY_VALUE => 1,          -- Almost empty offset (1-2)
    ALMOST_FULL_VALUE  => 1,          -- Almost full offset (1-2)
    ARRAY_MODE          => "ARRAY_MODE_4_X_8", -- ARRAY_MODE_4_X_8, ARRAY_MODE_4_X_4
    SYNCHRONOUS_MODE    => "FALSE"    -- Clock synchronous (FALSE)
)
port map (
    -- FIFO Status Flags: 1-bit (each) output: Flags and other FIFO status outputs
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit output: Almost empty
    ALMOSTFULL  => ALMOSTFULL,  -- 1-bit output: Almost full
    EMPTY       => EMPTY,       -- 1-bit output: Empty
    FULL        => FULL,        -- 1-bit output: Full
    -- Q0-Q9: 8-bit (each) output: FIFO Outputs
    Q0 => Q0, -- 8-bit output: Channel 0
    Q1 => Q1, -- 8-bit output: Channel 1
    Q2 => Q2, -- 8-bit output: Channel 2

```

```
Q3 => Q3,          -- 8-bit output: Channel 3
Q4 => Q4,          -- 8-bit output: Channel 4
Q5 => Q5,          -- 8-bit output: Channel 5
Q6 => Q6,          -- 8-bit output: Channel 6
Q7 => Q7,          -- 8-bit output: Channel 7
Q8 => Q8,          -- 8-bit output: Channel 8
Q9 => Q9,          -- 8-bit output: Channel 9
-- D0-D9: 4-bit (each) input: FIFO inputs
D0 => D0,          -- 4-bit input: Channel 0
D1 => D1,          -- 4-bit input: Channel 1
D2 => D2,          -- 4-bit input: Channel 2
D3 => D3,          -- 4-bit input: Channel 3
D4 => D4,          -- 4-bit input: Channel 4
D5 => D5,          -- 8-bit input: Channel 5
D6 => D6,          -- 8-bit input: Channel 6
D7 => D7,          -- 4-bit input: Channel 7
D8 => D8,          -- 4-bit input: Channel 8
D9 => D9,          -- 4-bit input: Channel 9
-- FIFO Control Signals: 1-bit (each) input: Clocks, Resets and Enables
RDCLK => RDCLK,    -- 1-bit input: Read clock
RDEN => RDEN,      -- 1-bit input: Read enable
RESET => RESET,    -- 1-bit input: Reset
WRCLK => WRCLK,    -- 1-bit input: Write clock
WREN => WREN       -- 1-bit input: Write enable
);

-- End of IN_FIFO_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IN_FIFO: Input First-In, First-Out (FIFO)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

IN_FIFO #(
    .ALMOST_EMPTY_VALUE(1),           // Almost empty offset (1-2)
    .ALMOST_FULL_VALUE(1),            // Almost full offset (1-2)
    .ARRAY_MODE("ARRAY_MODE_4_X_8"), // ARRAY_MODE_4_X_8, ARRAY_MODE_4_X_4
    .SYNCHRONOUS_MODE("FALSE")        // Clock synchronous (FALSE)
)
IN_FIFO_inst (
    // FIFO Status Flags: 1-bit (each) output: Flags and other FIFO status outputs
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit output: Almost empty
    .ALMOSTFULL(ALMOSTFULL),   // 1-bit output: Almost full
    .EMPTY(EMPTY),             // 1-bit output: Empty
    .FULL(FULL),               // 1-bit output: Full
    // Q0-Q9: 8-bit (each) output: FIFO Outputs
    .Q0(Q0),                   // 8-bit output: Channel 0
    .Q1(Q1),                   // 8-bit output: Channel 1
    .Q2(Q2),                   // 8-bit output: Channel 2
    .Q3(Q3),                   // 8-bit output: Channel 3
    .Q4(Q4),                   // 8-bit output: Channel 4
    .Q5(Q5),                   // 8-bit output: Channel 5
    .Q6(Q6),                   // 8-bit output: Channel 6
    .Q7(Q7),                   // 8-bit output: Channel 7
    .Q8(Q8),                   // 8-bit output: Channel 8
    .Q9(Q9),                   // 8-bit output: Channel 9
    // D0-D9: 4-bit (each) input: FIFO inputs
    .D0(D0),                   // 4-bit input: Channel 0
    .D1(D1),                   // 4-bit input: Channel 1
    .D2(D2),                   // 4-bit input: Channel 2
    .D3(D3),                   // 4-bit input: Channel 3
    .D4(D4),                   // 4-bit input: Channel 4
    .D5(D5),                   // 8-bit input: Channel 5
    .D6(D6),                   // 8-bit input: Channel 6
    .D7(D7),                   // 4-bit input: Channel 7
    .D8(D8),                   // 4-bit input: Channel 8
    .D9(D9),                   // 4-bit input: Channel 9
    // FIFO Control Signals: 1-bit (each) input: Clocks, Resets and Enables
    .RDCLK(RDCLK),             // 1-bit input: Read clock
    .RDEN(RDEN),               // 1-bit input: Read enable
    .RESET(RESET),             // 1-bit input: Reset
    .WRCLK(WRCLK),             // 1-bit input: Write clock
    .WREN(WREN),               // 1-bit input: Write enable
);

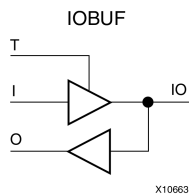
// End of IN_FIFO_inst instantiation
```

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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ために使用します。

## 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	説明
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	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"	"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。

## 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;

-- IOBUF: Single-ended Bi-directional Buffer
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスレーション)

```
// IOBUF: Single-ended Bi-directional Buffer
//       All devices
// Xilinx HDL Libraries Guide, version 14.4

IOBUF #(
    .DRIVE(12), // Specify the output drive strength
    .IBUF_LOW_PWR("TRUE"), // Low Power - "TRUE", High Performance = "FALSE"
    .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
);

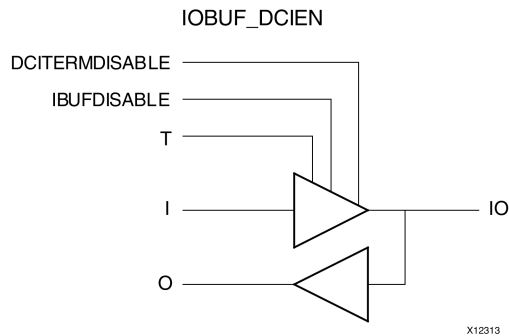
// End of IOBUF_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IOBUF\_DCIEEN

**プリミティブ：Bi-Directional Single-ended Buffer with DCI and Input Disable.**



## 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ために使用します。I/O が長時間使用されないとき、または長時間出力として使用される場合に電力消費を削減する機能として、入力パス ディスエーブルとデジタル制御インピーダンス (DCI) 終端イネーブル/ディスエーブルが含まれています。このELEMENTは、7 シリーズ デバイスの HP (High Performance) バンクにのみ配置可能です。

## ポートの説明

ポート名	方向	幅	説明
IO	入出力	1	双方向ポート接続。デザインの最上位ポートに直接接続します。
I	入力	1	デバイスへの出力パスを表すバッファ入力
IBUFDISABLE	入力	1	入力パスをディスエーブルにします。USE_IBUFDISABLE が "TRUE" に設定されている場合にこの信号が High にアサートされると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グランドに接続する必要があります。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
DCITERMDISABLE	入力	1	DCI 終端をディスエーブルにします。High の場合、DCI 終端はディスエーブルになります。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
T	入力	1	I/O が読み出し (入力) で使用されている場合に I/O をハイインピーダンス (トライステート) モードにします。USE_IBUFDISABLE = "TRUE" の場合、T ピンは IBUFDISABLE 機能にも影響します。
O	出力	1	デバイスへの入力パスを表すバッファ出力

## デザインの入力方法

インスタンス化	可
推論	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
DRIVE	整数	2、4、6、8、12、16、24	12	SelectIO™ バッファの出力駆動電流を指定します。
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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"	"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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。T ビン ディスエーブル入力パスで書き込み中の読み出しを許可しないようにする場合は、"FALSE" に設定します。"TRUE" に設定した場合、T をディアサートするか (I/O を出力として使用)、IBUFDISABLE をアサートすると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。

## 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;

-- IOBUF_DCIE: Single-ended Bi-directional Buffer with Digital Controlled Impedance (DCI)
--           and Input path enable/disable
--           May only be placed in High Performance (HP) Banks
--           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IOBUF_DCIE_inst : IOBUF_DCIE
generic map (
  DRIVE => 12,
  IOSTANDARD => "DEFAULT",
  IBUF_LOW_PWR => "TRUE",
  SLEW => "SLOW")
port map (
  O => O,      -- Buffer output
  IO => IO,    -- Buffer inout port (connect directly to top-level port)
  DCITERMDISABLE => DCITERMDISABLE, -- DCI Termination enable input
  I => I,      -- Buffer input
  IBUFDISABLE => IBUFDISABLE, -- Input disable input, low=disable
  T => T      -- 3-state enable input, high=input, low=output
);

-- End of IOBUF_DCIE_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// IOBUF_DCIEN: Single-ended Bi-directional Buffer with Digital Controlled Impedance (DCI)
//               and Input path enable/disable
//               May only be placed in High Performance (HP) Banks
//               7 Series
// Xilinx HDL Libraries Guide, version 14.4

IOBUF_DCIEN #(
    .DRIVE(12), // Specify the output drive strength
    .IBUF_LOW_PWR("TRUE"), // Low Power - "TRUE", High Performance = "FALSE"
    .IOSTANDARD("DEFAULT"), // Specify the I/O standard
    .SLEW("SLOW"), // Specify the output slew rate
    .USE_IBUFDISABLE("TRUE") // Use IBUFDISABLE function, "TRUE" or "FALSE"
) IOBUF_DCIEN_inst (
    .O(O), // Buffer output
    .IO(IO), // Buffer inout port (connect directly to top-level port)
    .DCITERMDISABLE(DCITERMDISABLE), // DCI Termination enable input
    .I(I), // Buffer input
    .IBUFDISABLE(IBUFDISABLE), // Input disable input, low=disable
    .T(T) // 3-state enable input, high=input, low=output
);

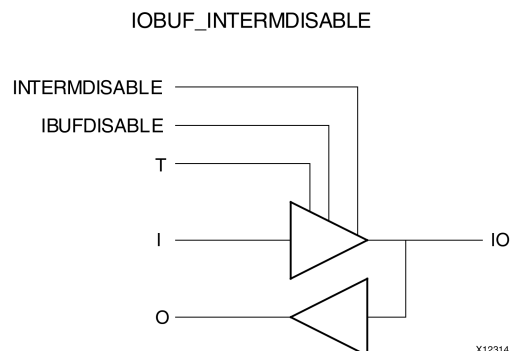
// End of IOBUF_DCIEN_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IOBUF\_INTERMDISABLE

**プリミティブ**：Bi-Directional Single-ended Buffer with Input Termination Disable and Input Path Disable



## 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ために使用します。I/O が長時間使用されないとき、または数クロック サイクル間出力として使用される場合に電力消費を削減する機能として、入力パス ディスエーブルとキャリブレーションされない入力終端 (INTERM) ディスエーブルが含まれています。このELEMENTは、7 シリーズ デバイスの HR (High Range) バンクにのみ配置可能です。

## ポートの説明

ポート名	方向	幅	説明
O	出力	1	デバイスへの入力パスを表すバッファ出力
IO	入出力	1	双方向ポート接続。デザインの最上位ポートに直接接続します。
I	入力	1	デバイスへの出力パスを表すバッファ入力
IBUFDISABLE	入力	1	USE_IBUFDISABLE が "TRUE" に設定されている場合に、バッファを介する入力パスをディスエーブルにし、ロジック High にします。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グラウンドに接続する必要があります。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
INTERMDISABLE	入力	1	入力終端をディスエーブルにします。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
T	入力	1	I/O が読み出し (入力) で使用されている場合に I/O をハイ インピーダンス (トライステート) モードにします。USE_IBUFDISABLE = "TRUE" の場合、T ピンは IBUFDISABLE 機能にも影響します。書き込み (出力) モードの場合は、INTERM もディスエーブルにします。

## 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
DRIVE	整数	2、4、6、8、12、16、24	12	SelectIO™ バッファの出力駆動電流を指定します。
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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"	"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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。T ビン ディスエーブル入力パスで書き込み中の読み出しを許可しないようにする場合に使用します。

## 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;

-- IOBUF_INTERMDISABLE: Single-ended Bi-directional Buffer with Input Termination
--                        and Input path enable/disable
--                        May only be placed in High Range (HR) Banks
--                        7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IOBUF_INTERMDISABLE_inst : IOBUF_INTERMDISABLE
generic map (
    DRIVE => 12,
    IOSTANDARD => "DEFAULT", -- Specify the I/O standard
    IBUF_LOW_PWR => "TRUE", -- Low Power - "TRUE", High Performance = "FALSE"
    USE_IBUFDISABLE => "TRUE", -- Use IBUFDISABLE function "TRUE" or "FALSE"
    SLEW => "SLOW")
port map (
    O => O,      -- Buffer output
    IO => IO,    -- Buffer inout port (connect directly to top-level port)
    DCITERMDISABLE => DCITERMDISABLE, -- DCI Termination enable input
    I => I,      -- Buffer input
    IBUFDISABLE => IBUFDISABLE, -- Input disable input, low=disable
    INTERMDISABLE => INTERMDISABLE, -- Input termination disable input
    T => T      -- 3-state enable input, high=input, low=output
);

-- End of IOBUF_INTERMDISABLE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IOBUF_INTERMDISABLE: Single-ended Bi-directional Buffer with Input Termination
//                        and Input path enable/disable
//                        May only be placed in High Range (HR) Banks
//                        7 Series
// Xilinx HDL Libraries Guide, version 14.4

IOBUF_INTERMDISABLE #(
    .DRIVE(12), // Specify the output drive strength
    .IBUF_LOW_PWR("TRUE"), // Low Power - "TRUE", High Performance = "FALSE"
    .IOSTANDARD("DEFAULT"), // Specify the I/O standard
    .SLEW("SLOW"), // Specify the output slew rate
    .USE_IBUFDISABLE("TRUE") // Use IBUFDISABLE function, "TRUE" or "FALSE"
) IOBUF_INTERMDISABLE_inst (
    .O(O), // Buffer output
    .IO(IO), // Buffer inout port (connect directly to top-level port)
    .I(I), // Buffer input
    .IBUFDISABLE(IBUFDISABLE), // Input disable input, low=disable
    .INTERMDISABLE(INTERMDISABLE), // Input termination disable input
    .T(T) // 3-state enable input, high=input, low=output
);

// End of IOBUF_INTERMDISABLE_inst instantiation
```

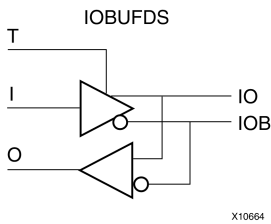
## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## 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、低電圧差動信号をサポート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	説明
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
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	7 シリーズ FPGA でサポートされる差動入力 I/O 規格用の差動終端属性で、ビルトインの差動終端をオン (TRUE) またはオフ (FALSE) にします。
IBUF_LOW_PWR	ブール代数	TRUE、FALSE	TRUE	TRUE に設定すると、LVDS や HSTL などの差動入力規格 (V <sub>REF</sub> が必要) を使用する場合に消費電力を削減できます。FALSE に設定すると、TRUE に設定した場合より消費電力は大きくなりますが、パフォーマンスは向上します。詳細は、『7 シリーズ FPGA SelectIO リソース ユーザーガイド』を参照してください。
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;

-- IOBUFDS: Differential Bi-directional Buffer
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IOBUFDS_inst : IOBUFDS
generic map (
    DIFF_TERM => FALSE, -- Differential Termination (TRUE/FALSE)
    IBUF_LOW_PWR => TRUE, -- Low Power = TRUE, High Performance = FALSE
    IOSTANDARD => "BLVDS_25", -- Specify the I/O standard
    SLEW => "SLOW") -- Specify the output slew rate
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
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

IOBUFDS #(
    .DIFF_TERM("FALSE"),      // Differential Termination ("TRUE"/"FALSE")
    .IBUF_LOW_PWR("TRUE"),    // Low Power - "TRUE", High Performance = "FALSE"
    .IOSTANDARD("BLVDS_25"), // Specify the I/O standard
    .SLEW("SLOW")             // Specify the output slew rate
) 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
);

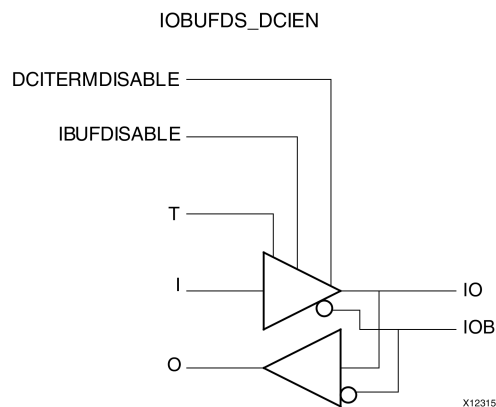
// End of IOBUFDS_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IOBUFDS\_DCIE

**プリミティブ：Bi-Directional Differential Buffer with DCI Enable/Disable and Input Disable**



## 概要

このデザイン エレメントは双方向差動 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ために使用します。I/O が長時間使用されないとき、または長時間出力として使用される場合に電力消費を削減する機能として、入力パス ディスエーブルとデジタル制御インピーダンス (DCI) 終端イネーブル/ディスエーブルが含まれています。このエレメントは、7 シリーズ デバイスの HP (High Performance) バンクにのみ配置可能です。

## ポートの説明

ポート名	方向	幅	説明
IO	入出力	1	双方向 p 側ポート接続。デザインの最上位ポートに直接接続します。
IOB	入出力	1	双方向 p 側ポート接続。デザインの最上位ポートに直接接続します。
I	入力	1	デバイスへの出力パスを表すバッファ入力
IBUFDISABLE	入力	1	入力パスをディスエーブルにします。USE_IBUFDISABLE が "TRUE" に設定されている場合にこの信号が High にアサートされると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グランドに接続する必要があります。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
DCITERMDISABLE	入力	1	DCI 終端をディスエーブルにします。High の場合、DCI 終端はディスエーブルになります。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
T	入力	1	I/O が読み出し (入力) で使用されている場合に I/O をハイ インピーダンス (トライステート) モードにします。USE_IBUFDISABLE = "TRUE" の場合、T ピンは IBUFDISABLE 機能にも影響します。
O	出力	1	デバイスへの入力パスを表すバッファ出力

## デザインの入力方法

インスタンス化	可
推論	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
DIFF_TERM	文字列	"TRUE"、"FALSE"	"FALSE"	ビルトインの差動終端をイネーブルにします。
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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"	"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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。T ビン ディスエーブル入力パスで書き込み中の読み出しを許可しないようにする場合は、"FALSE" に設定します。 "TRUE" に設定した場合、T をディアサートするか (I/O を出力として使用)、IBUFDISABLE をアサートすると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。

## 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;

-- IOBUFDS_DCIEN: Differential Bi-directional Buffer with Digital Controlled Impedance (DCI)
--               and Input path enable/disable
--               May only be placed in High Performance (HP) Banks
--               7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IOBUFDS_DCIEN_inst : IOBUFDS_DCIEN
generic map (
  DIFF_TERM => "FALSE", -- Differential termination (TRUE/FALSE)
  IBUF_LOW_PWR => "TRUE", -- Low Power - TRUE, HIGH Performance = FALSE
  IOSTANDARD => "BLVDS_25", -- Specify the I/O standard
  SLEW => "SLOW", -- Specify the output slew rate
  USE_IBUFDISABLE => "TRUE") -- Use IBUFDISABLE function "TRUE" or "FALSE"
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)
  DCITERMDISABLE => DCITERMDISABLE, -- DCI Termination enable input
  I => I, -- Buffer input
  IBUFDISABLE => IBUFDISABLE, -- Input disable input, low=disable
  T => T -- 3-state enable input, high=input, low=output
);
```

```
-- End of IOBUFDS_DCIEN_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IOBUFDS_DCIEN: Differential Bi-directional Buffer with Digital Controlled Impedance (DCI)
//               and Input path enable/disable
//               May only be placed in High Performance (HP) Banks
//               7 Series
// Xilinx HDL Libraries Guide, version 14.4

IOBUFDS_DCIEN #(
    .DIFF_TERM("FALSE"), // Differential Termination ("TRUE"/"FALSE")
    .IBUF_LOW_PWR("TRUE"), // Low Power - "TRUE", High Performance = "FALSE"
    .IOSTANDARD("BLVDS_25"), // Specify the I/O standard
    .SLEW("SLOW"), // Specify the output slew rate
    .USE_IBUFDISABLE("TRUE") // Use IBUFDISABLE function, "TRUE" or "FALSE"
) IOBUFDS_DCIEN_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)
    .DCITERMDISABLE(DCITERMDISABLE), // DCI Termination enable input
    .I(I), // Buffer input
    .IBUFDISABLE(IBUFDISABLE), // Input disable input, low=disable
    .T(T) // 3-state enable input, high=input, low=output
);

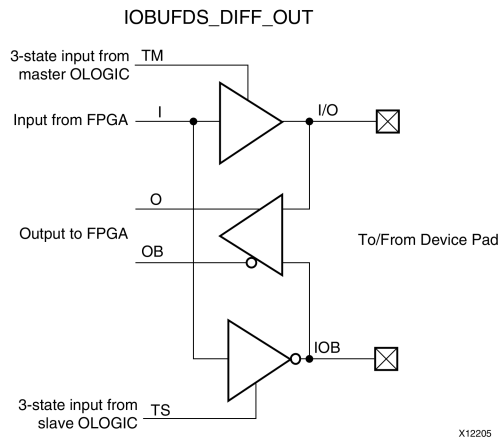
// End of IOBUFDS_DCIEN_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IOBUFDS\_DIFF\_OUT

プリミティブ：Differential Bi-directional Buffer with Differential Output



## 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双方向バッファです。IOBUFDS\_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 (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。IBUFDS\_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 IOBUFDS 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。

## ポートの説明

ポート名	方向	幅	説明
O	出力	1	バッファの p 側出力
OB	出力	1	バッファの n 側出力
IO	入出力	1	Diff_p 入出力 (最上位ポートに直接接続)
IOB	入出力	1	Diff_n 入出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力
TM	入力	1	マスター OLOGIC からのトライステート イネーブル入力。High の場合は入力、Low の場合は出力です。
TS	入力	1	スレーブ OLOGIC からのトライステート イネーブル入力。High の場合は入力、Low の場合は出力です。

## 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
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	7 シリーズ FPGA でサポートされる差動入力 I/O 規格用の差動終端属性で、ビルトインの差動終端をオン (TRUE) またはオフ (FALSE) にします。
IBUF_LOW_PWR	ブール代数	TRUE、FALSE	TRUE	TRUE に設定すると、LVDS や HSTL などの差動入力規格 ( $V_{REF}$ が必要) を使用する場合に消費電力を削減できます。FALSE に設定すると、TRUE に設定した場合より消費電力は大きくなりますが、パフォーマンスは向上します。詳細は、『7 シリーズ FPGA SelectIO リソース ユーザーガイド』を参照してください。
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。

## 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_DIFF_OUT: Differential Bi-directional Buffer with Differential Output
--                               7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IOBUFDS_DIFF_OUT_inst : IOBUFDS_DIFF_OUT
generic map (
    DIFF_TERM => FALSE, -- Differential Termination (TRUE/FALSE)
    IBUF_LOW_PWR => TRUE, -- Low Power - TRUE, High Performance = FALSE
    IOSTANDARD => "BLVDS_25") -- Specify the I/O standard
port map (
    O => O,      -- Buffer p-side output
    OB => OB,    -- Buffer n-side 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
    TM => TM,    -- 3-state enable input, high=input, low=output
    TS => TS     -- 3-state enable input, high=output, low=input
);

-- End of IOBUFDS_DIFF_OUT_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IOBUFDS_DIFF_OUT: Differential Bi-directional Buffer with Differential Output
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

IOBUFDS_DIFF_OUT #(
    .DIFF_TERM("FALSE"), // Differential Termination ("TRUE"/"FALSE")
    .IBUF_LOW_PWR("TRUE"), // Low Power - "TRUE", High Performance = "FALSE"
    .IOSTANDARD("BLVDS_25") // Specify the I/O standard
) IOBUFDS_DIFF_OUT_inst (
    .O(O), // Buffer p-side output
    .OB(OB), // Buffer n-side 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
    .TM(TM), // 3-state enable input, high=input, low=output
    .TS(TS)
);
```



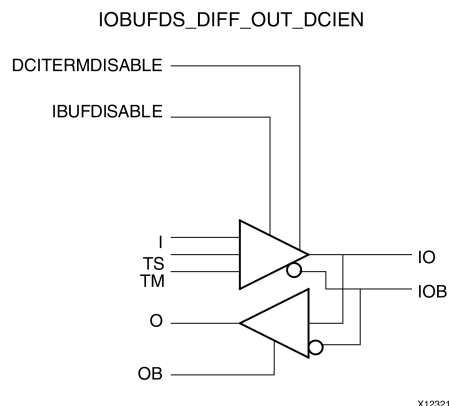
```
.TS(TS)    // 3-state enable input, high=output, low=input  
);  
  
// End of IOBUFDS_DIFF_OUT_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IOBUFDS\_DIFF\_OUT\_DCIEEN

**プリミティブ**：Bi-Directional Differential Buffer with DCI Disable, Input Disable, and Differential Output



## 概要

このデザイン エLEMENTは双方向差動 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ために使用します。I/O が長時間使用されないとき、または長時間出力として使用される場合に電力消費を削減する機能として、入力パス ディスエーブルとデジタル制御インピーダンス (DCI) 終端イネーブル/ディスエーブルが含まれています。IOBUFDS\_DIFF\_OUT\_DCIEEN は、差動信号の両方の位相に内部アクセスできる点が IOBUFDS\_DCIEEN と異なります。このELEMENTは、7 シリーズ デバイスの HP (High Performance) バンクにのみ配置可能です。

## ポートの説明

ポート名	方向	幅	説明
IO	入出力	1	双方向 p 側ポート接続。デザインの最上位ポートに直接接続します。
IOB	入出力	1	双方向 n 側ポート接続。デザインの最上位ポートに直接接続します。
I	入力	1	デバイスへの出力パスを表すバッファ入力
IBUFDISABLE	入力	1	入力パスをディスエーブルにします。USE_IBUFDISABLE が "TRUE" に設定されている場合にこの信号が High にアサートされると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グラウンドに接続する必要があります。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
DCITERMDISABLE	入力	1	DCI 終端をディスエーブルにします。High の場合、DCI 終端はディスエーブルになります。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
TM	入力	1	I/O が読み出し (入力) で使用されている場合に、P 側 (マスター) をハイインピーダンス (トライステート) モードにします。USE_IBUFDISABLE = "TRUE" の場合、TM ピンは IBUFDISABLE 機能にも影響します。
TS	入力	1	I/O が読み出し (入力) で使用されている場合に、N 側 (スレーブ) をハイインピーダンス (トライステート) モードにします。USE_IBUFDISABLE = "TRUE" の場合、TS ピンは IBUFDISABLE 機能にも影響します。

ポート名	方向	幅	説明
O	出力	1	デバイスへの入力パスを表すバッファ p 側出力
OB	出力	1	デバイスへの入力パスを表すバッファ n 側出力

## 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	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
DIFF_TERM	文字列	"TRUE"、"FALSE"	"FALSE"	ビルトインの差動終端をイネーブルにします。
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。T ビン ディスエーブル入力パスで書き込み中の読み出しを許可しないようにする場合は、"FALSE" に設定します。 "TRUE" に設定した場合、T をディアサートするか (I/O を出力として使用)、IBUFDISABLE をアサートすると、バッファを介する入力パスがディスエーブルになり、ロジック High になります。

## 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;

-- IOBUFDSDIFF_OUT_DCIE: Differential Bi-directional Buffer with Differential Output,
--                           Digital Controlled Impedance (DCI) and Input path enable/disable
--                           May only be placed in High Performance (HP) Banks
--                           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IOBUFDSDIFF_OUT_DCIE_inst : IOBUFDSDIFF_OUT_DCIE
generic map (
  DIFF_TERM => "FALSE", -- Differential Termination (TRUE/FALSE)
  IBUF_LOW_PWR => "TRUE", -- Low Power - TRUE, High Performance = FALSE
  IOSTANDARD => "LVCMOS18", -- Specify the I/O standard
  USE_IBUFDISABLE => "TRUE") -- Use IBUFDISABLE function, "TRUE" or "FALSE"
port map (
  O => O, -- Buffer p-side output
  OB => OB, -- Buffer n-side output
  IO => IO, -- Diff_p inout (connect directly to top-level port)
  IOB => IOB, -- Diff_n inout (connect directly to top-level port)
  DCITERMDISABLE => DCITERMDISABLE, -- DCI Termination enable input
  I => I, -- Buffer input
```

```

IBUFTERMDISABLE => IBUFTERMDISABLE, -- input disable input, low=disable
TM => TM,    -- 3-state enable input, high=input, low=output
TS => TS    -- 3-state enable input, high=output, low=input
);

-- End of IOBUFDS_DIFF_OUT_DCIEN_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// IOBUFDS_DIFF_OUT_DCIEN: Differential Bi-directional Buffer with Differential Output,
//                           Digital Controlled Impedance (DCI) and Input path enable/disable
//                           May only be placed in High Performance (HP) Banks
//                           7 Series
// Xilinx HDL Libraries Guide, version 14.4

IOBUFDS_DIFF_OUT_DCIEN #(
    .DIFF_TERM("FALSE"),    // Differential Termination ("TRUE"/"FALSE")
    .IBUF_LOW_PWR("TRUE"),  // Low Power - "TRUE", High Performance = "FALSE"
    .IOSTANDARD("BLVDS_25"), // Specify the I/O standard
    .USE_IBUFDISABLE("TRUE") // Use IBUFDISABLE function, "TRUE" or "FALSE"
) IOBUFDS_DIFF_OUT_DCIEN_inst (
    .O(O),    // Buffer p-side output
    .OB(OB),  // Buffer n-side output
    .IO(IO),  // Diff_p inout (connect directly to top-level port)
    .IOB(IOB), // Diff_n inout (connect directly to top-level port)
    .DCITERMDISABLE(DCITERMDISABLE), // DCI Termination enable input
    .I(I),    // Buffer input
    .IBUFDISABLE(IBUFDISABLE),    // Input disable input, low=disable
    .TM(TM),    // 3-state enable input, high=input, low=output
    .TS(TS)    // 3-state enable input, high=output, low=input
);

// End of IOBUFDS_DIFF_OUT_DCIEN_inst instantiation

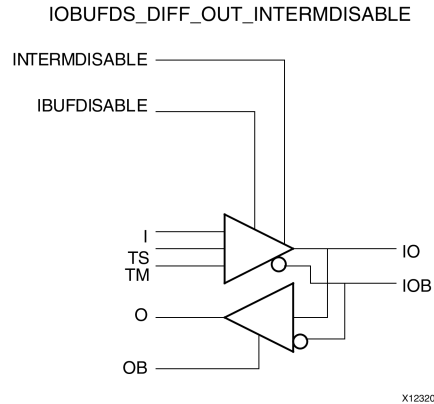
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IOBUFDS\_DIFF\_OUT\_INTERMDISABLE

**プリミティブ：Bi-Directional Differential Buffer with Input Termination Disable, Input Disable, and Differential Output**



## 概要

このデザイン エLEMENTは双方向差動 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ために使用します。I/O が長時間使用されないとき、または数クロック サイクル間出力として使用される場合に電力消費を削減する機能として、入力パス ディスエーブルとキャリブレーションされない入力終端 (INTERM) ディスエーブルが含まれています。IOBUFDS\_DIFF\_OUT\_INTERMDISABLE は、差動信号の両方の位相に内部アクセスできる点が IOBUFDS\_INTERMDISABLE と異なります。このELEMENTは、7 シリーズ デバイスの HR (High Range) バンクにのみ配置可能です。

## ポートの説明

ポート名	方向	幅	説明
IO	入出力	1	双方向 p 側ポート接続。デザインの最上位ポートに直接接続します。
IOB	入出力	1	双方向 n 側ポート接続。デザインの最上位ポートに直接接続します。
I	入力	1	デバイスへの出力パスを表すバッファ入力
IBUFDISABLE	入力	1	USE_IBUFDISABLE が "TRUE" に設定されている場合に、バッファを介する入力パスをディスエーブルにし、ロジック High にします。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グラウンドに接続する必要があります。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
INTERMDISABLE	入力	1	入力終端をディスエーブルにします。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
TM	入力	1	I/O が読み出し (入力) で使用されている場合に、P 側 (マスター) をハイインピーダンス (トライステート) モードにします。USE_IBUFDISABLE = "TRUE" の場合、TM ピンは IBUFDISABLE 機能にも影響します。書き込み (出力) モードの場合は、INTERM もディスエーブルになります。
TS	入力	1	I/O が読み出し (入力) で使用されている場合に、N 側 (スレーブ) をハイインピーダンス (トライステート) モードにします。USE_IBUFDISABLE = "TRUE" の場合、TS ピンは IBUFDISABLE 機能にも影響します。書き込み (出力) モードの場合は、INTERM もディスエーブルになります。

ポート名	方向	幅	説明
O	出力	1	デバイスへの入力パスを表すバッファ p 側出力
OB	出力	1	デバイスへの入力パスを表すバッファ n 側出力

## 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	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
DIFF_TERM	文字列	"TRUE"、"FALSE"	"FALSE"	ビルトインの差動終端をイネーブルにします。
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	参照 I/O 規格が使用された場合に、消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。T ビン ディスエーブル入力パスで書き込み中の読み出しを許可しないようにする場合に使用します。

## 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;

-- IOBUFDSDIFF_OUT_INTERMDISABLE: Differential Global Clock Buffer with Differential Output
--                               Input Termination and Input Path Disable
--                               May only be placed in High Range (HR) Banks
--                               7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IOBUFDSDIFF_OUT_INTERMDISABLE_inst : IOBUFDSDIFF_OUT_INTERMDISABLE
generic map (
    DIFF_TERM => "FALSE", -- Differential Termination (TRUE/FALSE)
    IBUF_LOW_PWR => "TRUE", -- Low Power - TRUE, High Performance = FALSE
    IOSTANDARD => "BLVDS_25", -- Specify the I/O standard
    USE_IBUFDISABLE => "TRUE") -- Use IBUFDISABLE function, "TRUE" or "FALSE"
port map (
    O => O, -- Buffer p-side output
    OB => OB, -- Buffer n-side output
    IO => IO, -- Diff_p inout (connect directly to top-level port)
    IOB => IOB, -- Diff_n inout (connect directly to top-level port)
    DCITERMDISABLE => DCITERMDISABLE, -- DCI Termination enable input
    I => I, -- Buffer input
    IBUFDISABLE => IBUFDISABLE, -- input disable input, low=disable
    INTERMDISABLE => INTERMDISABLE, -- Input termination disable input
    TM => TM, -- 3-state enable input, high=input, low=output
    TS => TS -- 3-state enable input, high=output, low=input
);

```

```
-- End of IOBUFDS_DIFF_OUT_INTERMDISABLE_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IOBUFDS_DIFF_OUT_INTERMDISABLE: Differential Global Clock Buffer with Differential Output
//                                     Input Termination and Input Path Disable
//                                     May only be placed in High Range (HR) Banks
//                                     7 Series
// Xilinx HDL Libraries Guide, version 14.4

IOBUFDS_DIFF_OUT_INTERMDISABLE #(
    .DIFF_TERM("FALSE"), // Differential Termination, "TRUE"/"FALSE"
    .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performance="FALSE"
    .IOSTANDARD("DEFAULT"), // Specify the input I/O standard
    .USE_IBUFDISABLE("TRUE") // Set to "TRUE" to enable IBUFDISABLE feature
) IOBUFDS_DIFF_OUT_INTERMDISABLE_inst (
    .O(O), // Buffer p-side output
    .OB(OB), // Buffer n-side 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
    .INTERMDISABLE(INTERMDISABLE), // Input termination disable input
    .IBUFDISABLE(IBUFDISABLE), // Input disable input, low=disable
    .TM(TM), // 3-state enable input, high=input, low=output
    .TS(TS) // 3-state enable input, high=output, low=input
);

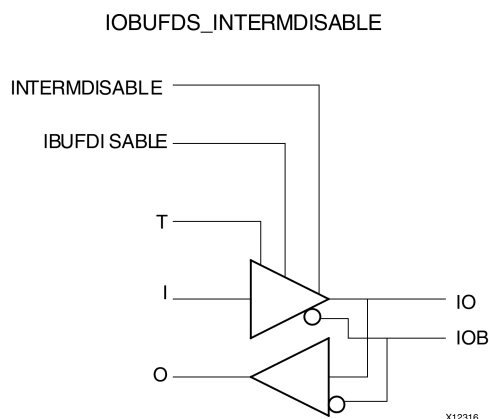
// End of IOBUFDS_DIFF_OUT_INTERMDISABLE_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IOBUFDS\_INTERMDISABLE

プリミティブ：Bi-Directional Differential Buffer with Input Termination Disable and Input Disable



## 概要

このデザイン エレメントは双方向差動 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ために使用します。I/O が長時間使用されないとき、または長時間出力として使用される場合に電力消費を削減する機能として、入力パス ディスエーブルとキャリブレーションされない入力終端 (INTERM) ディスエーブルが含まれています。このエレメントは、7 シリーズ デバイスの HR (High Range) バンクにのみ配置可能です。

## ポートの説明

ポート名	方向	幅	説明
IO	入出力	1	双方向 p 側ポート接続。デザインの最上位ポートに直接接続します。
IOB	入出力	1	双方向 n 側ポート接続。デザインの最上位ポートに直接接続します。
I	入力	1	デバイスへの出力パスを表すバッファ入力
IBUFDISABLE	入力	1	USE_IBUFDISABLE が "TRUE" に設定されている場合に、バッファを介する入力パスをディスエーブルにし、ロジック High にします。USE_IBUFDISABLE が "FALSE" に設定されている場合はこの入力は無視されるので、グランドに接続する必要があります。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
INTERMDISABLE	入力	1	入力終端をディスエーブルにします。この機能は、I/O が長時間アイドル状態になったとき、または長時間書き込み状態 (出力) となったときに、電力消費を削減するために使用します。
T	入力	1	I/O が読み出し (入力) で使用されている場合に I/O をハイ インピーダンス (トライステート) モードにします。USE_IBUFDISABLE = "TRUE" の場合、T ピンは IBUFDISABLE 機能にも影響します。書き込み (出力) モードの場合は、INTERM もディスエーブルにします。
O	出力	1	デバイスへの入力パスを表すバッファ出力



## デザインの入力方法

インスタンス化	可
推論	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
DIFF_TERM	文字列	"TRUE"、"FALSE"	"FALSE"	ビルトインの差動終端をイネーブルにします。
IBUF_LOW_PWR	文字列	"TRUE"、"FALSE"	"TRUE"	参照 I/O 規格が使用された場合に、消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定します。
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"	"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。
USE_IBUFDISABLE	文字列	"TRUE"、"FALSE"	"TRUE"	IBUFDISABLE の機能をイネーブルまたはディスエーブルにします。T ビン ディスエーブル入力パスで書き込み中の読み出しを許可しないようにする場合に使用します。

## 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;

-- IOBUFDIS_INTERMDISABLE: Differential Bi-directional Buffer with Input Termination
-- and Input path enable/disable
-- May only be placed in High Range (HR) Banks
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

IOBUFDIS_INTERMDISABLE_inst : IOBUFDIS_INTERMDISABLE
generic map (
    DIFF_TERM => "FALSE", -- Differential termination (TRUE/FALSE)
    IBUF_LOW_PWR => "TRUE", -- Low Power - TRUE, HIGH Performance = FALSE
    IOSTANDARD => "BLVDS_25", -- Specify the I/O standard
    SLEW => "SLOW", -- Specify the output slew rate
    USE_IBUFDISABLE => "TRUE") -- Use IBUFDISABLE function "TRUE" or "FALSE"
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)
    DCITERMDISABLE => DCITERMDISABLE, -- DCI Termination enable input
    I => I, -- Buffer input
    IBUFDISABLE => IBUFDISABLE, -- Input disable input, low=disable
    INTERMDISABLE => INTERMDISABLE, -- Input termination disable input
    T => T -- 3-state enable input, high=input, low=output
);

-- End of IOBUFDIS_INTERMDISABLE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IOBUFDS_INTERMDISABLE: Differential Bi-directional Buffer with Input Termination
//                               and Input path enable/disable
//                               May only be placed in High Range (HR) Banks
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

IOBUFDS_INTERMDISABLE #(
    .DIFF_TERM("FALSE"), // Differential Termination ("TRUE"/"FALSE")
    .IBUF_LOW_PWR("TRUE"), // Low Power - "TRUE", High Performance = "FALSE"
    .IOSTANDARD("BLVDS_25"), // Specify the I/O standard
    .SLEW("SLOW"), // Specify the output slew rate
    .USE_IBUFDISABLE("TRUE") // Use IBUFDISABLE function, "TRUE" or "FALSE"
) IOBUFDS_INTERMDISABLE_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
    .IBUFDISABLE(IBUFDISABLE), // Input disable input, low=disable
    .INTERMDISABLE(INTERMDISABLE), // Input termination disable input
    .T(T) // 3-state enable input, high=input, low=output
);

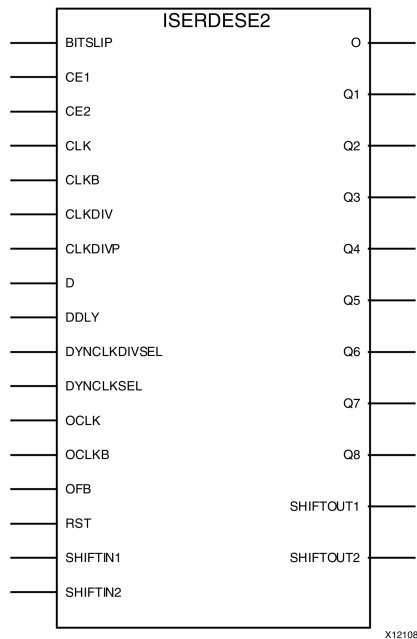
// End of IOBUFDS_INTERMDISABLE_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ISERDESE2

プリミティブ：Input SERial/DESerializer with bitslip



### 概要

7 シリーズ FPGA に含まれる ISERDESE2 は、高速ソース同期アプリケーションのインプリメンテーションに特化したクロックおよびロジック機能を持つ、専用シリアル/パラレル コンバーターです。FPGA でデシリアライザーを設計する際の複雑なタイミング問題を回避するために使用します。ISERDESE2 には、次の機能があります。

- ・ 専用デシリアライザー、シリアル/パラレル コンバーター：FPGA ファブリックが入力データ周波数に一致していなくても、高速データ転送を可能にします。このコンバーターでは、シングル データ レート (SDR) モードとダブル データ レート (DDR) モードがサポートされます。SDR モードでは 2、3、4、5、6、7、または 8 ビット幅のパラレル ワードが作成され、DDR モードでは 4、6、8、10、または 14 ビット幅のパラレル ワードが作成されます。
- ・ Bitslip サブモジュール：FPGA ファブリックに入力するパラレル データ ストリームのシーケンスを並べ替えます。トレーニング パターンを含むトレーニング ソース同期インターフェイスに使用できます。
- ・ ストローブ ベース メモリ インターフェイスの専用サポート (OCLK 入力ピンを含む)：ISERDESE2 ブロック内でストローブから FPGA クロックドメインへの切り替えを処理します。これにより、パフォーマンスが向上し、インプリメンテーションが簡略化されます。
- ・ ネットワーク インターフェイスの専用サポート
- ・ メモリ インターフェイスの専用サポート

## ポートの説明

ポート名	方向	幅	説明
BITSLIP	入力	1	BITSLIP ピンがアサートされると (アクティブ High)、CLKDIV に同期してビットスリップ処理が実行されます。ビットスリップ処理が実行されるごとに、パレル シフターと同様、Q1 ~ Q8 出力ポートのデータが 1 ずつシフトします (DDR と SDR では動作が異なる)。
CE1、CE2	入力	1	各 ISERDESE2 ブロックには、入力クロック イネーブル モジュールが含まれています。NUM_CE = 1 の場合、CE2 入力には使用されず、CE1 入力が高レベルのクロック イネーブルとして ISERDESE2 の入力レジスタに直接接続されます。NUM_CE = 2 の場合、CE1 および CE2 入力の両方が使用され、CLKDIV の半サイクル CE1 がイネーブルとして使用され、残りの半サイクルは CE2 がイネーブルとして使用されます。この場合、クロック イネーブル モジュールは 2:1 シリアル/パラレル コンバーターとして使用され、CLKDIV によりクロックが供給されます。クロック イネーブル モジュールは、ISERDESE2 を DDR モードで 1:4 デシリアル化をコンフィギュレーションする際、双方向メモリ インターフェイスに必要です。NUM_CE = 2 の場合、クロック イネーブル モジュールがイネーブルになり、CE1 および CE2 ポートの両方が使用できるようになります。NUM_CE = 1 の場合は、CE1 のみが使用可能で、通常のクロック イネーブルとして機能します。
CLK	入力	1	入力シリアル データ ストリームの入力に使用される高速クロック入力
CLKB	入力	1	入力シリアル データ ストリームの入力に使用される高速セカンダリ クロック入力。“MEMORY_QDR” 以外のモードでは、CLKB を CLK を反転したクロックに接続します。“MEMORY_QDR” モードでは、CLKB を固有の位相シフトされたクロックに接続する必要があります。
CLKDIV	入力	1	分周クロック入力。通常は CLK を分周したクロックです (インプリメントされたデシリアルライザーの幅による)。シリアル/パラレル コンバーターの出力、Bitslip サブモジュール、および CE モジュールを駆動します。
CLKDIVP	入力	1	MIG でのみサポートされます。MEMORY_DDR3 モードの場合に PHASER_IN で分周された CLK が供給され、その他のモードではグラウンドに接続されます。
D	入力	1	ISERDESE2 のシリアル (高速) データ入力ポート。7 シリーズ FPGA の I/O リソースと使用した場合にのみ機能します。
DDLY	入力	1	ISERDESE2 のシリアル (高速) データ入力ポート。7 シリーズ FPGA の IDELAYE2 リソースと使用した場合にのみ機能します。
DYNCLKDIVSEL	入力	1	CLKDIV の反転を動的に選択します。
DYNCLKSEL	入力	1	CLK および CLKB の反転を動的に選択します。
O	出力	1	組み合わせ出力。ISERDESE2 モジュールのレジスタを介さない出力で、データ入力 (D) または IDELAYE2 を介したデータ入力 (DDLY) が直接出力されます。
OCLK	入力	1	ストロブ ベース メモリ インターフェイスのデータ転送を同期化します。INTERFACE_TYPE が “MEMORY” に設定されている場合にのみ使用されます。ストロブ ベース メモリ データをフリー ランニング クロックドメインに転送するために使用するクロック入力です。OCLK は、CLK 入力上のストロブと同じ周波数のフリーランニング FPGA クロックです。ドメイン転送のタイミングは、ストロブ信号から CLK 入力への遅延を IDELAY を使用するなどして調整することにより、ユーザーが設定します。このドメイン転送のタイミングの設定例は、Memory Interface Generator (MIG) に示されます。INTERFACE_TYPE が “NETWORKING” に設定されている場合は、このポートは使用されず、GND に接続する必要があります。
OCLKB	入力	1	ストロブ ベース メモリ インターフェイスのデータ転送を同期化します。INTERFACE_TYPE が “MEMORY” に設定されている場合にのみ使用されます。

ポート名	方向	幅	説明
OFB	入力	1	ISERDESE2 のシリアル (高速) データ入力ポート。7 シリーズ FPGA の OSERDESE2 の OFB ポートと使用した場合にのみ機能します。
Q1 ~ Q8	出力	1	ISERDESE2 モジュールのレジスタ付き出力。1 つの ISERDESE2 ブロックで最大 8 ビット (1:8 デシリアル化) までサポートできます。データ幅拡張を使用して 8 ビットより広い幅 (14 ビットまで) もサポート可能です。受信される最初のデータビットは、Q 出力の最上位ビットに出力されます。OSERDESE2 の入力でのビット順は、ISERDESE2 ブロックの出力とは逆になります。たとえば、ワード FEDCBA の最下位ビット A は OSERDESE2 の D1 入力に配置されますが、同じビット A は ISERDESE2 ブロックでは Q8 出力に現れます。つまり、OSERDESE2 の最下位入力は D1 であり、ISERDESE2 ブロックの最下位出力は Q8 です。データ幅拡張を使用すると、マスター OSERDESE1 の D1 が最下位入力となり、スレーブ ISERDESE2 ブロックの Q7 が最下位出力になります。
RST	入力	1	CLK および CLKDIV ドメインのすべてのデータフリップフロップの出力を非同期で Low に駆動します。ISERDESE2 回路が CLK ドメインで動作しており、タイミングがクリティカルな場合は、内部専用回路を使用して RST 入力のタイミングを調整し、CLK ドメインに同期したリセット信号を生成してください。同様に、CLKDIV ドメインに同期したリセット信号を生成する RST 入力のタイミングを調整する専用回路があります。ISERDESE2 は非同期でリセットされますが、クロックに同期してリセット状態から戻るので、CLKDIV ドメインへの同期リセットとして扱い、CLKDIV の 1 サイクル以上パルスする必要があります。複数の ISERDESE2 ポートを含むインターフェイスを構築する場合は、すべての ISERDESE2 ポートを同期化する必要があります。RST 入力の内部タイミングは、同じリセットパルスを受信するすべての ISERDESE2 ブロックがお互いに同期してリセット状態から戻るように調整されます。
SHIFTIN1、SHIFTIN2	入力	1	SERDES_MODE が "SLAVE" の場合は、マスターの SHIFTOUT1 と SHIFTOUT2 出力に接続します。それ以外の場合は、SHIFTOUT1 および SHIFTOUT2 を未接続のままにするか、SHIFTIN1 および SHIFTIN2 をグランドに接続します。
SHIFTOUT1、SHIFTOUT2	出力	1	SERDES_MODE が "MASTER" に設定されており、2 つの ISERDESE2 をカスケード接続する場合は、スレーブ ISERDESE2 の SHIFTIN1 と SHIFTIN2 入力に接続します。

## 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
DATA_RATE	文字列	"DDR"、"SDR"	"DDR"	入力データストリームをシングル データレート (SDR) またはダブル データレート (DDR) のいずれかで処理するかを指定します。
DATA_WIDTH	10 進数	4、2、3、5、6、7、8、10、14	4	シリアル/パラレル コンバーターの幅を指定します。有効な値は DATA_RATE 属性によって異なります。  ・ DATA_RATE が "DDR" の場合は 4、6、8、10、14、

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> <li>DATA_RATE が "SDR" の場合は 2、3、4、5、6、7、8</li> </ul>
DYN_CLKDIV_INV_EN	文字列	"FALSE"、"TRUE"	"FALSE"	"TRUE" の場合、DYNCLKDIVINVSEL の反転がイネーブルになり、CLKDIV ピンの HDL 反転がディスエーブルになります。
DYN_CLK_INV_EN	文字列	"FALSE"、"TRUE"	"FALSE"	"TRUE" の場合、DYNCLKINVSEL の反転がイネーブルになり、CLK および CLKB ピンの HDL 反転がディスエーブルになります。
INIT_Q1、 INIT_Q2、 INIT_Q3、INIT_Q4	2 進数	1'b0 ~ 1'b1	1'b0	コンフィギュレーション後のポート Q1 ~ Q4 の出力の初期値を指定します。
INTERFACE_TYPE	文字列	"MEMORY"、 "MEMORY_DDR3"、 "MEMORY_QDR"、 "NETWORKING"、 "OVERSAMPLE"	"MEMORY"	ISERDESE2 の操作モードを指定します。詳細は、『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』を参照してください。
IOBDELAY	文字列	"NONE"、"BOTH"、 "IBUF"、"IFD"	"NONE"	<p>ISERDESE2 モジュールの入力ソースを指定します。D および DDLY ピンは、ISERDESE2 への専用入力です。D 入力は I/O への直接接続で、DDLY ピンは IODELAYE2 への直接接続です。これにより、レジスタを介した (Q1 ~ Q6) 出力または組み合わせパス (O) 出力への入力に、遅延のあるものまたは遅延のないものを使用できます。出力に適用する入力は、IOBDELAY 属性で指定します。</p> <ul style="list-style-type: none"> <li>"NONE" : O =&gt; D   Q1 ~ Q6 =&gt; D</li> <li>"IBUF" : O =&gt; DDLY   Q1 ~ Q6 =&gt; D</li> <li>"IFD" : O =&gt; D   Q1 ~ Q6 =&gt; DDLY</li> <li>"BOTH" : O =&gt; DDLY   Q1 ~ Q6 =&gt; DDLY</li> </ul>
NUM_CE	10 進数	2、1	2	使用するクロック イネーブル (CE1 および CE2) の数を指定します。
OFB_USED	文字列	"FALSE"、"TRUE"	"FALSE"	OLOGIC、OSERDES の OFB ピンから ISERDES の OFB ピンへのパスをイネーブルにし、D 入力ピンの使用をディスエーブルにします。
SERDES_MODE	文字列	"MASTER"、"SLAVE"	"MASTER"	データ幅を拡張する場合に ISERDESE2 モジュールがマスターかスレーブかを指定します。データ幅拡張を使用しない場合は、"MASTER" に設定します。
SRVAL_Q1、 SRVAL_Q2、 SRVAL_Q3、 SRVAL_Q4	2 進数	1'b0 ~ 1'b1	1'b0	SR ピンをアサートした場合の Q1 ~ Q4 出力の値 (セットまたはリセット) 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;

-- ISERDESE2: Input SERIAL/DESerializer with bitslip
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

ISERDESE2_inst : ISERDESE2
generic map (
    DATA_RATE => "DDR",           -- DDR, SDR
    DATA_WIDTH => 4,               -- Parallel data width (2-8,10,14)
    DYN_CLKDIV_INV_EN => "FALSE",   -- Enable DYNCLKDIVINVSEL inversion (FALSE, TRUE)
    DYN_CLK_INV_EN => "FALSE",     -- Enable DYNCLKINVSEL inversion (FALSE, TRUE)
    -- INIT_Q1 - INIT_Q4: Initial value on the Q outputs (0/1)
    INIT_Q1 => '0',
    INIT_Q2 => '0',
    INIT_Q3 => '0',
    INIT_Q4 => '0',
    INTERFACE_TYPE => "MEMORY",    -- MEMORY, MEMORY_DDR3, MEMORY_QDR, NETWORKING, OVERSAMPLE
    IOBDELAY => "NONE",            -- NONE, BOTH, IBUF, IFD
    NUM_CE => 2,                   -- Number of clock enables (1,2)
    OFB_USED => "FALSE",           -- Select OFB path (FALSE, TRUE)
    SERDES_MODE => "MASTER",       -- MASTER, SLAVE
    -- SRVAL_Q1 - SRVAL_Q4: Q output values when SR is used (0/1)
    SRVAL_Q1 => '0',
    SRVAL_Q2 => '0',
    SRVAL_Q3 => '0',
    SRVAL_Q4 => '0'
)
port map (
    O => O,                         -- 1-bit output: Combinatorial output
    -- Q1 - Q8: 1-bit (each) output: Registered data outputs
    Q1 => Q1,
    Q2 => Q2,
    Q3 => Q3,
    Q4 => Q4,
    Q5 => Q5,
    Q6 => Q6,
    Q7 => Q7,
    Q8 => Q8,
    -- SHIFTOUT1-SHIFTOUT2: 1-bit (each) output: Data width expansion output ports
    SHIFTOUT1 => SHIFTOUT1,
    SHIFTOUT2 => SHIFTOUT2,
    BITSLLIP => BITSLLIP,           -- 1-bit input: The BITSLLIP pin performs a Bitslip operation synchronous to
    -- CLKDIV when asserted (active High). Subsequently, the data seen on the
    -- Q1 to Q8 output ports will shift, as in a barrel-shifter operation, one
    -- position every time Bitslip is invoked (DDR operation is different from
    -- SDR).

    -- CE1, CE2: 1-bit (each) input: Data register clock enable inputs
    CE1 => CE1,
    CE2 => CE2,
    CLKDIVP => CLKDIVP,            -- 1-bit input: TBD
    -- Clocks: 1-bit (each) input: ISERDESE2 clock input ports
    CLK => CLK,                    -- 1-bit input: High-speed clock
    CLKB => CLKB,                  -- 1-bit input: High-speed secondary clock
    CLKDIV => CLKDIV,              -- 1-bit input: Divided clock
    OCLK => OCLK,                  -- 1-bit input: High speed output clock used when INTERFACE_TYPE="MEMORY"
    -- Dynamic Clock Inversions: 1-bit (each) input: Dynamic clock inversion pins to switch clock polarity
    DYNCLKDIVSEL => DYNCLKDIVSEL, -- 1-bit input: Dynamic CLKDIV inversion
    DYNCLKSEL => DYNCLKSEL,        -- 1-bit input: Dynamic CLK/CLKB inversion
    -- Input Data: 1-bit (each) input: ISERDESE2 data input ports
    D => D,                        -- 1-bit input: Data input
    DDLY => DDLY,                  -- 1-bit input: Serial data from IDELAYE2
    OFB => OFB,                    -- 1-bit input: Data feedback from OSERDESE2
    OCLKB => OCLKB,                -- 1-bit input: High speed negative edge output clock
    RST => RST,                    -- 1-bit input: Active high asynchronous reset
```

```

-- SHIFTIN1-SHIFTIN2: 1-bit (each) input: Data width expansion input ports
SHIFTIN1 => SHIFTIN1,
SHIFTIN2 => SHIFTIN2
);

-- End of ISERDESE2_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// ISERDESE2: Input SERIAL/DESerializer with bitslip
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

ISERDESE2 #(
    .DATA_RATE("DDR"),           // DDR, SDR
    .DATA_WIDTH(4),              // Parallel data width (2-8,10,14)
    .DYN_CLKDIV_INV_EN("FALSE"), // Enable DYNCLKDIVINVSEL inversion (FALSE, TRUE)
    .DYN_CLK_INV_EN("FALSE"),    // Enable DYNCLKINVSEL inversion (FALSE, TRUE)
    // INIT_Q1 - INIT_Q4: Initial value on the Q outputs (0/1)
    .INIT_Q1(1'b0),
    .INIT_Q2(1'b0),
    .INIT_Q3(1'b0),
    .INIT_Q4(1'b0),
    .INTERFACE_TYPE("MEMORY"),   // MEMORY, MEMORY_DDR3, MEMORY_QDR, NETWORKING, OVERSAMPLE
    .IOBDelay("NONE"),           // NONE, BOTH, IBUF, IFD
    .NUM_CE(2),                  // Number of clock enables (1,2)
    .OFB_USED("FALSE"),          // Select OFB path (FALSE, TRUE)
    .SERDES_MODE("MASTER"),      // MASTER, SLAVE
    // SRVAL_Q1 - SRVAL_Q4: Q output values when SR is used (0/1)
    .SRVAL_Q1(1'b0),
    .SRVAL_Q2(1'b0),
    .SRVAL_Q3(1'b0),
    .SRVAL_Q4(1'b0)
)
ISERDESE2_inst (
    .O(0),                       // 1-bit output: Combinatorial output
    // Q1 - Q8: 1-bit (each) output: Registered data outputs
    .Q1(Q1),
    .Q2(Q2),
    .Q3(Q3),
    .Q4(Q4),
    .Q5(Q5),
    .Q6(Q6),
    .Q7(Q7),
    .Q8(Q8),
    // SHIFTOUT1-SHIFTOUT2: 1-bit (each) output: Data width expansion output ports
    .SHIFTOUT1(SHIFTOUT1),
    .SHIFTOUT2(SHIFTOUT2),
    .BITSLIP(BITSLIP),           // 1-bit input: The BITSLIP pin performs a Bitslip operation synchronous to
                                // CLKDIV when asserted (active High). Subsequently, the data seen on the Q1
                                // to Q8 output ports will shift, as in a barrel-shifter operation, one
                                // position every time Bitslip is invoked (DDR operation is different from
                                // SDR).

    // CE1, CE2: 1-bit (each) input: Data register clock enable inputs
    .CE1(CE1),
    .CE2(CE2),
    .CLKDIVP(CLKDIVP),           // 1-bit input: TBD
    // Clocks: 1-bit (each) input: ISERDESE2 clock input ports
    .CLK(CLK),                   // 1-bit input: High-speed clock
    .CLKB(CLKB),                 // 1-bit input: High-speed secondary clock
    .CLKDIV(CLKDIV),             // 1-bit input: Divided clock
    .OCLK(OCLK),                 // 1-bit input: High speed output clock used when INTERFACE_TYPE="MEMORY"
    // Dynamic Clock Inversions: 1-bit (each) input: Dynamic clock inversion pins to switch clock polarity
    .DYNCLKDIVSEL(DYNCLKDIVSEL), // 1-bit input: Dynamic CLKDIV inversion
    .DYNCLKSEL(DYNCLKSEL),       // 1-bit input: Dynamic CLK/CLKB inversion
    // Input Data: 1-bit (each) input: ISERDESE2 data input ports
    .D(D),                       // 1-bit input: Data input
    .DDL(DDL),                   // 1-bit input: Serial data from IDELAYE2
    .OFB(OFB),                   // 1-bit input: Data feedback from OSERDESE2
    .OCLKB(OCLKB),               // 1-bit input: High speed negative edge output clock

```



```
.RST(RST),                // 1-bit input: Active high asynchronous reset
// SHIFTIN1-SHIFTIN2: 1-bit (each) input: Data width expansion input ports
.SHIFTIN1(SHIFTIN1),
.SHIFTIN2(SHIFTIN2)
);

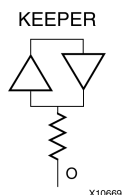
// End of ISERDESE2_inst instantiation
```

#### 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## KEEPER

### プリミティブ：KEEPER Symbol



### 概要

このデザイン エレメントは、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 エレメントです。たとえば、ネットに対して論理 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	説明
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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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

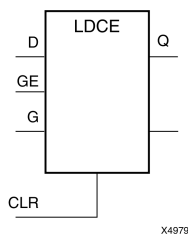
// End of KEEPER_inst instantiation
```

## 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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。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。

## 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.
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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.
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

LDCE #(
    .INIT(1'b0) // Initial value of latch (1'b0 or 1'b1)
) 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
);

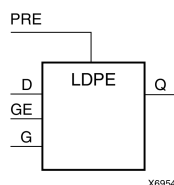
// End of LDCE_inst instantiation
```

詳細情報

[7 シリーズ 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。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。

## 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.
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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.
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

LDPE #(
    .INIT(1'b1) // Initial value of latch (1'b0 or 1'b1)
) 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
);

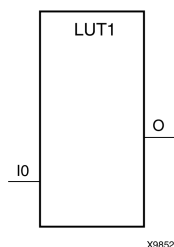
// End of LDPE_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT1

### プリミティブ：1-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 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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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
--      7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//      7 Series
// Xilinx HDL Libraries Guide, version 14.4

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

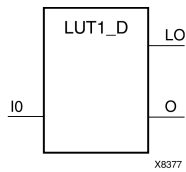
// End of LUT1_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT1\_D

### プリミティブ：1-Bit Look-Up Table with Dual Output



### 概要

このデザイン エレメントは 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。

出力 O は汎用インターコネクトです。LO 出力は、同じ CLB スライス内の別の出力に接続されます。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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	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	すべてゼロ	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_D: 1-input Look-Up Table with general and local outputs
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスレーション)

```
// LUT1_D: 1-input Look-Up Table with general and local outputs (Mapped to a LUT6)
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

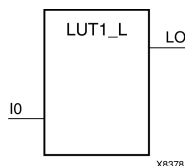
// End of LUT1_D_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT1\_L

### プリミティブ：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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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

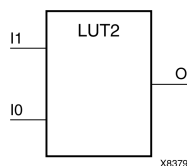
// End of LUT1_L_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT2

### プリミティブ：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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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

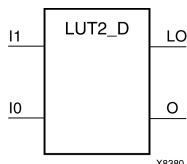
// End of LUT2_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT2\_D

### プリミティブ：2-Bit Look-Up Table with Dual 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 スライス内の別の出力に接続されます。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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

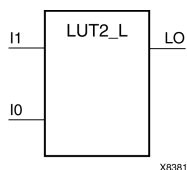
// End of LUT2_D_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT2\_L

### プリミティブ：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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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

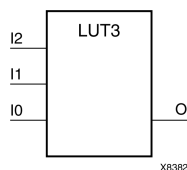
// End of LUT2_L_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT3

### プリミティブ：3-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 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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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 (Mapped to a LUT6)
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

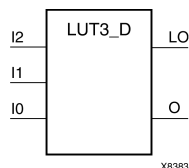
// End of LUT3_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT3\_D

### プリミティブ：3-Bit Look-Up Table with Dual Output



### 概要

このデザイン エレメントは 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 スライス内の別の出力に接続されます。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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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;

-- LUT3_D: 3-input Look-Up Table with general and local outputs
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスレーション)

```
// LUT3_D: 3-input Look-Up Table with general and local outputs (Mapped to a LUT6)
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

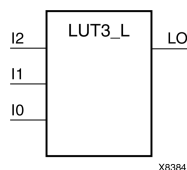
// End of LUT3_D_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT3\_L

### プリミティブ：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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

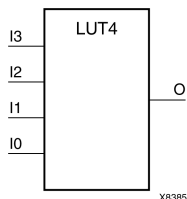
// End of LUT3_L_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT4

### プリミティブ：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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//      7 Series
// Xilinx HDL Libraries Guide, version 14.4

LUT4 #(
    .INIT(16'h0000) // Specify LUT Contents
) LUT4_inst (
    .O(0), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3) // LUT input
);

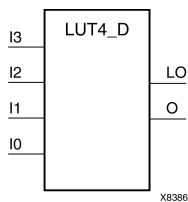
// End of LUT4_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT4\_D

### プリミティブ：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 スライス内の別の出力に接続されます。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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンス化)

```
// LUT4_D: 4-input Look-Up Table with general and local outputs (Mapped to a LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

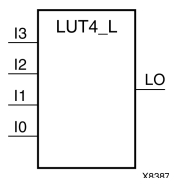
// End of LUT4_D_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT4\_L

### プリミティブ：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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

### 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	すべてゼロ	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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

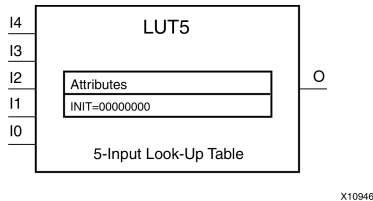
// End of LUT4_L_inst instantiation
```

詳細情報

[7 シリーズ 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、入力がすべてゼロ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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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 (Mapped to SliceM LUT6)
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

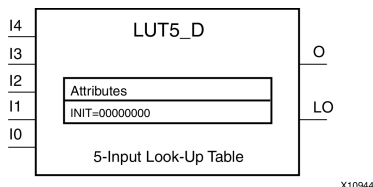
// End of LUT5_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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、入力がすべてゼロ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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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 (Mapped to SliceM LUT6)
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
```

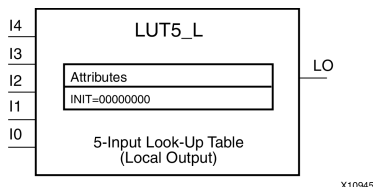


詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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、入力がすべてゼロ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。バイナリの真理値表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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 (Mapped to SliceM LUT6)
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

LUT5_L_inst : LUT5_L
generic map (
  INIT => X"00000000") -- Specify LUT Contents
port map (
  LO => LO, -- LUT local output
  IO => IO, -- 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 (Mapped to a LUT6)
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

LUT5_L #(
  .INIT(32'h00000000) // Specify LUT Contents
) LUT5_L_inst (
  .LO(IO), // LUT local output
  .IO(IO), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4)  // LUT input
);

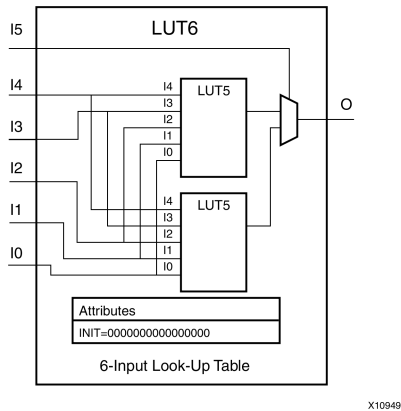
// End of LUT5_L_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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、入力すべてがゼロ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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
```

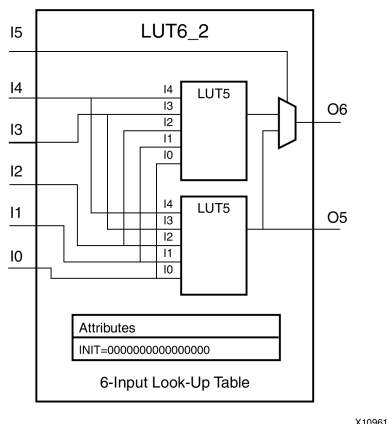


詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT6\_2

**プリミティブ：Six-input, 2-output, Look-Up Table**



### 概要

このデザイン エLEMENT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、または入力とロジック値を共有する 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'hfffffffffffffe` (VHDL では `X"FFFFFFFFFFFFFFFFFE"`) に設定すると、入力がすべてゼロの場合以外は O6 出力は 1 になり、`I[4:0]` がすべてゼロの場合以外は 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 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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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	すべてゼロ	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;

-- LUT6_2: 6-input 2 output Look-Up Table
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスレーション)

```
// LUT6_2: 6-input, 2 output Look-Up Table
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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)
);

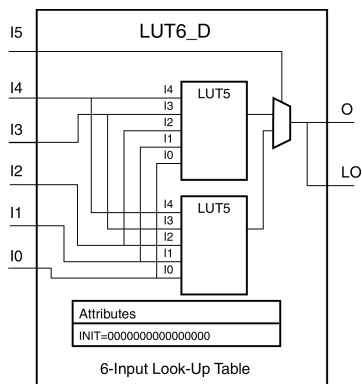
// End of LUT6_2_inst instantiation
```

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT6\_D

### プリミティブ：6-Input Lookup Table with General and Local Outputs



X10947

## 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、入力すべてがゼロ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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

## 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	すべてゼロ	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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

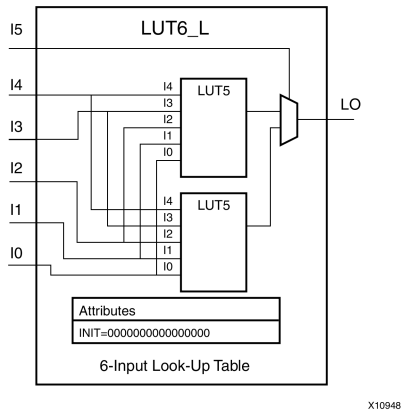
// End of LUT6_D_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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'hffffffffffffff` (VHDL では `X"FFFFFFFFFFFFFFFF"`) に設定すると、入力すべてがゼロ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。バイナリの真理値表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から初期値を作成します。

**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、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

### 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	すべてゼロ	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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

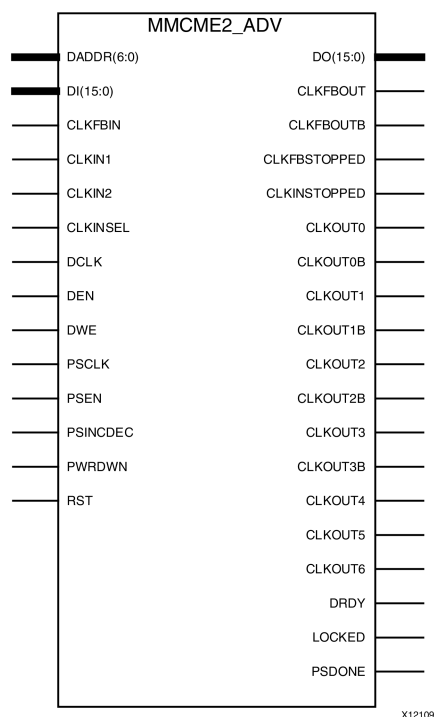
// End of LUT6_L_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MMCM2\_ADV

プリミティブ：Advanced Mixed Mode Clock Manager



### 概要

MMCM2 は、周波数合成、クロック ネットワークのスキュー調整、ジッター低減をサポートするための混合信号ブロックです。各クロック出力に対して、同じ VCO 周波数を基準に分周、位相シフト、デューティ サイクルを個別に設定できます。ダイナミック位相シフトおよび分数分周もサポートされます。

### ポートの説明

ポート名	方向	幅	説明
CLKFBIN	入力	1	MMCM へのフィードバック クロック ピン
CLKFBOUT	出力	1	専用 MMCM フィードバック クロック出力
CLKFBOUTB	出力	1	CLKFBOUT を反転したクロック出力
CLKFBSTOPPED	出力	1	フィードバック クロックが停止したことを示すステータス ピン
CLKINSEL	入力	1	入力マルチプレクサーのステートを制御する信号で、High の場合は CLKIN1、Low の場合は CLKIN2 です。
CLKINSTOPPED	出力	1	入力クロックが停止したことを示すステータス ピン
CLKIN1	入力	1	プライマリ クロック入力
CLKIN2	入力	1	MMCM 基準クロックを動的に切り替えるためのセカンダリ クロック入力
CLKOUT0	出力	1	CLKOUT0 出力
CLKOUT0B	出力	1	CLKOUT0 の反転出力
CLKOUT1	出力	1	CLKOUT1 出力



ポート名	方向	幅	説明
CLKOUT1B	出力	1	CLKOUT1 の反転出力
CLKOUT2	出力	1	CLKOUT2 出力
CLKOUT2B	出力	1	CLKOUT2 の反転出力
CLKOUT3	出力	1	CLKOUT3 出力
CLKOUT3B	出力	1	CLKOUT3 の反転出力
CLKOUT4	出力	1	CLKOUT4 出力
CLKOUT5	出力	1	CLKOUT5 出力
CLKOUT6	出力	1	CLKOUT6 出力
DADDR<6:0>	入力	7	ダイナミック リコンフィギュレーション用のリコンフィギュレーション アドレスを供給する入力バス。このバスを使用しない場合は、すべてのビットを 0 にする必要があります。
DCLK	入力	1	ダイナミック リコンフィギュレーション ポートの基準クロック
DEN	入力	1	ダイナミック リコンフィギュレーション機能にアクセスするためのイネーブル制御信号。ダイナミック リコンフィギュレーションを使用しない場合は、Low に接続する必要があります。
DI<15:0>	入力	16	リコンフィギュレーション データを供給するデータ入力バス。このバスを使用しない場合は、すべてのビットを 0 にする必要があります。
DO<15:0>	出力	16	ダイナミック リコンフィギュレーションを使用する場合の MMCM データ出力バス
DRDY	出力	1	MMCM のダイナミック リコンフィギュレーション機能の DEN 信号への応答を供給する READY 出力です。
DWE	入力	1	DADDR アドレスへの DI データの書き込みを制御するライト イネーブル信号。使用しない場合は、Low に接続する必要があります。
LOCKED	出力	1	位相アライメントが定義されている時間内で完了し、周波数が定義されている PPM 範囲内で一致したことを示します。MMCM は電源投入時に自動的にロックされるので、リセットは必要ありません。入力クロックが停止した場合、または位相アライメントに違反が発生した場合（入力クロックの位相シフトなど）は、LOCKED がデアサートされます。LOCKED がデアサートされると、自動的にロックが達成されます。
PSCLK	入力	1	位相シフト クロック
PSDONE	出力	1	位相シフト終了
PSEN	入力	1	位相シフト イネーブル
PSINCDEC	入力	1	位相シフト インクリメント/デクリメント制御
PWRDWN	入力	1	インスタンス化されているが未使用の MMCM をパワー ダウンします。
RST	入力	1	非同期リセット信号。この信号が解放されると、MMCM はクロックに同期して再びイネーブルになります。入力クロックの条件（周波数など）を変更する場合、リセットが必要です。

## 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
BANDWIDTH	文字列	"OPTIMIZED"、 "HIGH"、"LOW"	"OPTIMIZED"	ジッター、位相マージンなどの MMCM 特性に影響する MMCM プログラム アルゴリズムを指定します。
CLKFBOUT_MULT_F	3 上位ビット 浮動小数点	2.000 ~ 64.000	5.000	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 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKFBOUT_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	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。フィードバック クロックをシフトすると、MMCM の出力クロックがすべて負の値に位相シフトします。
CLKIN1_PERIOD、 CLKIN2_PERIOD	浮動小数点 (nS)	0.000 ~ 100.000	0.000	CLKIN 入力の周期を指定します。精度は ps です。たとえば、値 33.333 は 30 MHz の入力クロックを示します。この値は必ず設定する必要があります。CLKIN1_PERIOD は CLKIN1 入力の入力クロック周期、CLKIN2_PERIOD は CLKIN2 入力の入力クロック周期を指定します。
CLKOUT1_DIVIDE、 CLKOUT2_DIVIDE、 CLKOUT3_DIVIDE、 CLKOUT4_DIVIDE、 CLKOUT5_DIVIDE、 CLKOUT6_DIVIDE	10 進数	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、CLKFBOUT_MULT_F 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKOUT0_DIVIDE_F	3 上位ビット 浮動小数点	1.000 ~ 128.000	1.000	CLKOUT0 クロック出力を分周する値を指定します。この値と、CLKFBOUT_MULT_F 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKOUT0_DUTY_CYCLE ~ CLKOUT6_DUTY_CYCLE	3 上位ビット 浮動小数点	0.001 ~ 0.999	0.500	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。
CLKOUT0_PHASE ~ CLKOUT6_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	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。フィードバック クロックをシフトすると、MMCM の出力クロックがすべて負の値に位相シフトします。
CLKOUT4_CASCADE	ブール代数	FALSE、TRUE	FALSE	出力クロック分周が 128 よりも大きい場合、出力分周 (カウンタ) を CLKOUT4 分周の入力にカスケード接続します。

属性	データ型	値	デフォルト	説明
COMPENSATION	文字列	"ZHOLD"、"BUF_IN"、 "EXTERNAL"、 "INTERNAL"	"ZHOLD"	<p>クロック入力の補正を指定します。 "ZHOLD" に設定する必要があります。 MMCM フィードバックのコンフィギュレーション方法を定義します。</p> <ul style="list-style-type: none"> <li>・ "ZHOLD"：I/O レジスタでのホールドタイムが負になるように MMCM がコンフィギュレーションされます。</li> <li>・ "INTERNAL"：MMCM の内部フィードバックパスが使用され、遅延は補正されません。</li> <li>・ "EXTERNAL"：FPGA の外部ネットワークが補正されます。</li> <li>・ "BUF_IN"：その他の補正モードには一致せず、遅延は補正されません。クロック入力が BUFG/BUFH/BUFR/GT で駆動される場合です。</li> </ul>
DIVCLK_DIVIDE	10 進数	1 ～ 106	1	すべての出力クロックの入力クロックに対する分周比を指定し、PFD に入力される CLKIN を分周します。
REF_JITTER1、 REF_JITTER2	3 上位ビット 浮動小数点	0.000 ～ 0.999	0.010	MMCM パフォーマンスを最適化するため、CLKIN 入力に予測されるジッター値を指定します。BANDWIDTH が "OPTIMIZED" に設定されている場合、値が既知でない場合は入力クロックに最適なパラメーターが選択されます。値が既知である場合は、値を入力クロックに予測されるジッターの UI パーセント (最大ピークトゥピーク値) で指定する必要があります。REF_JITTER1 は CLKIN1 の入力ジッター、REF_JITTER2 は CLKIN2 の入力ジッターを指定します。
SS_EN	文字列	"FALSE"、"TRUE"	"FALSE"	MMCM のスペクトラム拡散機能をイネーブルにします。SS_MODE および SS_MOD_PERIOD 属性と共に使用します。
SS_MOD_PERIOD	10 進数 (ns)	4000 ～ 40000	10000	スペクトラム拡散の変調周期 (ns) を指定します。
SS_MODE	文字列	"CENTER_HIGH"、 "CENTER_LOW"、 "DOWN_HIGH"、 "DOWN_LOW"	"CENTER_HIGH"	スペクトラム拡散の周波数偏差および拡散タイプを指定します。
STARTUP_WAIT	ブール代数	FALSE、TRUE	FALSE	コンフィギュレーション DONE 信号がアサートされるのを、MMCM がロックされるまで遅延します。
CLKFBOUT_USE_ FINE_PS ～ CLKOUT6_USE_ FINE_PS	ブール代数	FALSE、TRUE	FALSE	カウンター可変ファイン位相シフト イネーブル

## 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;

-- MMCME2_ADV: Advanced Mixed Mode Clock Manager
--           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

MMCME2_ADV_inst : MMCME2_ADV
generic map (
    BANDWIDTH => "OPTIMIZED",           -- Jitter programming (OPTIMIZED, HIGH, LOW)
    CLKFBOUT_MULT_F => 5.0,             -- Multiply value for all CLKOUT (2.000-64.000).
    CLKFBOUT_PHASE => 0.0,              -- Phase offset in degrees of CLKFB (-360.000-360.000).
    -- CLKIN_PERIOD: Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    CLKIN1_PERIOD => 0.0,
    CLKIN2_PERIOD => 0.0,
    -- CLKOUT0_DIVIDE - CLKOUT6_DIVIDE: Divide amount for CLKOUT (1-128)
    CLKOUT1_DIVIDE => 1,
    CLKOUT2_DIVIDE => 1,
    CLKOUT3_DIVIDE => 1,
    CLKOUT4_DIVIDE => 1,
    CLKOUT5_DIVIDE => 1,
    CLKOUT6_DIVIDE => 1,
    CLKOUT0_DIVIDE_F => 1.0,             -- Divide amount for CLKOUT0 (1.000-128.000).
    -- CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for CLKOUT outputs (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,
    CLKOUT6_DUTY_CYCLE => 0.5,
    -- CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for CLKOUT outputs (-360.000-360.000).
    CLKOUT0_PHASE => 0.0,
    CLKOUT1_PHASE => 0.0,
    CLKOUT2_PHASE => 0.0,
    CLKOUT3_PHASE => 0.0,
    CLKOUT4_PHASE => 0.0,
    CLKOUT5_PHASE => 0.0,
    CLKOUT6_PHASE => 0.0,
    CLKOUT4_CASCADE => FALSE,           -- Cascade CLKOUT4 counter with CLKOUT6 (FALSE, TRUE)
    COMPENSATION => "ZHOLD",            -- ZHOLD, BUF_IN, EXTERNAL, INTERNAL
    DIVCLK_DIVIDE => 1,                 -- Master division value (1-106)
    -- REF_JITTER: Reference input jitter in UI (0.000-0.999).
    REF_JITTER1 => 0.0,
    REF_JITTER2 => 0.0,
    STARTUP_WAIT => FALSE,             -- Delays DONE until MMCM is locked (FALSE, TRUE)
    -- Spread Spectrum: Spread Spectrum Attributes
    SS_EN => "FALSE",                  -- Enables spread spectrum (FALSE, TRUE)
    SS_MODE => "CENTER_HIGH",          -- CENTER_HIGH, CENTER_LOW, DOWN_HIGH, DOWN_LOW
    SS_MOD_PERIOD => 10000,            -- Spread spectrum modulation period (ns) (VALUES)
    -- USE_FINE_PS: Fine phase shift enable (TRUE/FALSE)
    CLKFBOUT_USE_FINE_PS => FALSE,
    CLKOUT0_USE_FINE_PS => FALSE,
    CLKOUT1_USE_FINE_PS => FALSE,
    CLKOUT2_USE_FINE_PS => FALSE,
    CLKOUT3_USE_FINE_PS => FALSE,
    CLKOUT4_USE_FINE_PS => FALSE,
    CLKOUT5_USE_FINE_PS => FALSE,
    CLKOUT6_USE_FINE_PS => FALSE
)
port map (
    -- Clock Outputs: 1-bit (each) output: User configurable clock outputs
    CLKOUT0 => CLKOUT0,                -- 1-bit output: CLKOUT0
    CLKOUT0B => CLKOUT0B,              -- 1-bit output: Inverted CLKOUT0
    CLKOUT1 => CLKOUT1,                -- 1-bit output: CLKOUT1
    CLKOUT1B => CLKOUT1B,              -- 1-bit output: Inverted CLKOUT1
    CLKOUT2 => CLKOUT2,                -- 1-bit output: CLKOUT2
```

```

CLKOUT2B => CLKOUT2B,          -- 1-bit output: Inverted CLKOUT2
CLKOUT3 => CLKOUT3,            -- 1-bit output: CLKOUT3
CLKOUT3B => CLKOUT3B,          -- 1-bit output: Inverted CLKOUT3
CLKOUT4 => CLKOUT4,            -- 1-bit output: CLKOUT4
CLKOUT5 => CLKOUT5,            -- 1-bit output: CLKOUT5
CLKOUT6 => CLKOUT6,            -- 1-bit output: CLKOUT6
-- DRP Ports: 16-bit (each) output: Dynamic reconfiguration ports
DO => DO,                      -- 16-bit output: DRP data
DRDY => DRDY,                  -- 1-bit output: DRP ready
-- Dynamic Phase Shift Ports: 1-bit (each) output: Ports used for dynamic phase shifting of the outputs
PSDONE => PSDONE,              -- 1-bit output: Phase shift done
-- Feedback Clocks: 1-bit (each) output: Clock feedback ports
CLKFBOUT => CLKFBOUT,          -- 1-bit output: Feedback clock
CLKFBOUTB => CLKFBOUTB,        -- 1-bit output: Inverted CLKFBOUT
-- Status Ports: 1-bit (each) output: MMCM status ports
CLKFBSTOPPED => CLKFBSTOPPED,  -- 1-bit output: Feedback clock stopped
CLKINSTOPPED => CLKINSTOPPED,  -- 1-bit output: Input clock stopped
LOCKED => LOCKED,              -- 1-bit output: LOCK
-- Clock Inputs: 1-bit (each) input: Clock inputs
CLKIN1 => CLKIN1,              -- 1-bit input: Primary clock
CLKIN2 => CLKIN2,              -- 1-bit input: Secondary clock
-- Control Ports: 1-bit (each) input: MMCM control ports
CLKINSEL => CLKINSEL,          -- 1-bit input: Clock select, High=CLKIN1 Low=CLKIN2
PWRDWN => PWRDWN,              -- 1-bit input: Power-down
RST => RST,                    -- 1-bit input: Reset
-- DRP Ports: 7-bit (each) input: Dynamic reconfiguration ports
DADDR => DADDR,                -- 7-bit input: DRP address
DCLK => DCLK,                  -- 1-bit input: DRP clock
DEN => DEN,                    -- 1-bit input: DRP enable
DI => DI,                      -- 16-bit input: DRP data
DWE => DWE,                    -- 1-bit input: DRP write enable
-- Dynamic Phase Shift Ports: 1-bit (each) input: Ports used for dynamic phase shifting of the outputs
PSClk => PSClk,                -- 1-bit input: Phase shift clock
PSEN => PSEN,                  -- 1-bit input: Phase shift enable
PSINCDEC => PSINCDEC,          -- 1-bit input: Phase shift increment/decrement
-- Feedback Clocks: 1-bit (each) input: Clock feedback ports
CLKFBIN => CLKFBIN,            -- 1-bit input: Feedback clock
);

-- End of MMCME2_ADV_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// MMCME2_ADV: Advanced Mixed Mode Clock Manager
// 7 Series
// Xilinx HDL Libraries Guide, version 14.4

MMCME2_ADV #(
    .BANDWIDTH("OPTIMIZED"),      // Jitter programming (OPTIMIZED, HIGH, LOW)
    .CLKFBOUT_MULT_F(5.0),        // Multiply value for all CLKOUT (2.000-64.000).
    .CLKFBOUT_PHASE(0.0),         // Phase offset in degrees of CLKFB (-360.000-360.000).
    // CLKIN_PERIOD: Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    .CLKIN1_PERIOD(0.0),
    .CLKIN2_PERIOD(0.0),
    // CLKOUT0_DIVIDE - CLKOUT6_DIVIDE: Divide amount for CLKOUT (1-128)
    .CLKOUT1_DIVIDE(1),
    .CLKOUT2_DIVIDE(1),
    .CLKOUT3_DIVIDE(1),
    .CLKOUT4_DIVIDE(1),
    .CLKOUT5_DIVIDE(1),
    .CLKOUT6_DIVIDE(1),
    .CLKOUT0_DIVIDE_F(1.0),        // Divide amount for CLKOUT0 (1.000-128.000).
    // CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for CLKOUT outputs (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),
    .CLKOUT6_DUTY_CYCLE(0.5),
    // CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for CLKOUT outputs (-360.000-360.000).

```

```

.CLKOUT0_PHASE(0.0),
.CLKOUT1_PHASE(0.0),
.CLKOUT2_PHASE(0.0),
.CLKOUT3_PHASE(0.0),
.CLKOUT4_PHASE(0.0),
.CLKOUT5_PHASE(0.0),
.CLKOUT6_PHASE(0.0),
.CLKOUT4_CASCADE("FALSE"),      // Cascade CLKOUT4 counter with CLKOUT6 (FALSE, TRUE)
.COMPENSATION("ZHOLD"),          // ZHOLD, BUF_IN, EXTERNAL, INTERNAL
.DIVCLK_DIVIDE(1),              // Master division value (1-106)
// REF_JITTER: Reference input jitter in UI (0.000-0.999).
.REF_JITTER1(0.0),
.REF_JITTER2(0.0),
.STARTUP_WAIT("FALSE"),         // Delays DONE until MMCM is locked (FALSE, TRUE)
// Spread Spectrum: Spread Spectrum Attributes
.SS_EN("FALSE"),               // Enables spread spectrum (FALSE, TRUE)
.SS_MODE("CENTER_HIGH"),       // CENTER_HIGH, CENTER_LOW, DOWN_HIGH, DOWN_LOW
.SS_MOD_PERIOD(10000),         // Spread spectrum modulation period (ns) (VALUES)
// USE_FINE_PS: Fine phase shift enable (TRUE/FALSE)
.CLKFBOUT_USE_FINE_PS("FALSE"),
.CLKOUT0_USE_FINE_PS("FALSE"),
.CLKOUT1_USE_FINE_PS("FALSE"),
.CLKOUT2_USE_FINE_PS("FALSE"),
.CLKOUT3_USE_FINE_PS("FALSE"),
.CLKOUT4_USE_FINE_PS("FALSE"),
.CLKOUT5_USE_FINE_PS("FALSE"),
.CLKOUT6_USE_FINE_PS("FALSE")
)
MMCME2_ADV_inst (
// Clock Outputs: 1-bit (each) output: User configurable clock outputs
.CLKOUT0(CLKOUT0),             // 1-bit output: CLKOUT0
.CLKOUT0B(CLKOUT0B),           // 1-bit output: Inverted CLKOUT0
.CLKOUT1(CLKOUT1),             // 1-bit output: CLKOUT1
.CLKOUT1B(CLKOUT1B),           // 1-bit output: Inverted CLKOUT1
.CLKOUT2(CLKOUT2),             // 1-bit output: CLKOUT2
.CLKOUT2B(CLKOUT2B),           // 1-bit output: Inverted CLKOUT2
.CLKOUT3(CLKOUT3),             // 1-bit output: CLKOUT3
.CLKOUT3B(CLKOUT3B),           // 1-bit output: Inverted CLKOUT3
.CLKOUT4(CLKOUT4),             // 1-bit output: CLKOUT4
.CLKOUT5(CLKOUT5),             // 1-bit output: CLKOUT5
.CLKOUT6(CLKOUT6),             // 1-bit output: CLKOUT6
// DRP Ports: 16-bit (each) output: Dynamic reconfiguration ports
.DO(DO),                       // 16-bit output: DRP data
.DRDY(DRDY),                   // 1-bit output: DRP ready
// Dynamic Phase Shift Ports: 1-bit (each) output: Ports used for dynamic phase shifting of the outputs
.PSDONE(PSDONE),               // 1-bit output: Phase shift done
// Feedback Clocks: 1-bit (each) output: Clock feedback ports
.CLKFBOUT(CLKFBOUT),           // 1-bit output: Feedback clock
.CLKFBOUTB(CLKFBOUTB),         // 1-bit output: Inverted CLKFBOUT
// Status Ports: 1-bit (each) output: MMCM status ports
.CLKFBSTOPPED(CLKFBSTOPPED),   // 1-bit output: Feedback clock stopped
.CLKINSTOPPED(CLKINSTOPPED),   // 1-bit output: Input clock stopped
.LOCKED(LOCKED),               // 1-bit output: LOCK
// Clock Inputs: 1-bit (each) input: Clock inputs
.CLKIN1(CLKIN1),               // 1-bit input: Primary clock
.CLKIN2(CLKIN2),               // 1-bit input: Secondary clock
// Control Ports: 1-bit (each) input: MMCM control ports
.CLKINSEL(CLKINSEL),           // 1-bit input: Clock select, High=CLKIN1 Low=CLKIN2
.PWRDWN(PWRDWN),               // 1-bit input: Power-down
.RST(RST),                     // 1-bit input: Reset
// DRP Ports: 7-bit (each) input: Dynamic reconfiguration ports
.DADDR(DADDR),                 // 7-bit input: DRP address
.DCLK(DCLK),                   // 1-bit input: DRP clock
.DEN(DEN),                     // 1-bit input: DRP enable
.DI(DI),                       // 16-bit input: DRP data
.DWE(DWE),                     // 1-bit input: DRP write enable
// Dynamic Phase Shift Ports: 1-bit (each) input: Ports used for dynamic phase shifting of the outputs
.PSCLK(PSCLK),                 // 1-bit input: Phase shift clock
.PSEN(PSEN),                   // 1-bit input: Phase shift enable
.PSINCDEC(PSINCDEC),           // 1-bit input: Phase shift increment/decrement
// Feedback Clocks: 1-bit (each) input: Clock feedback ports
.CLKFBIN(CLKFBIN)              // 1-bit input: Feedback clock
);

```

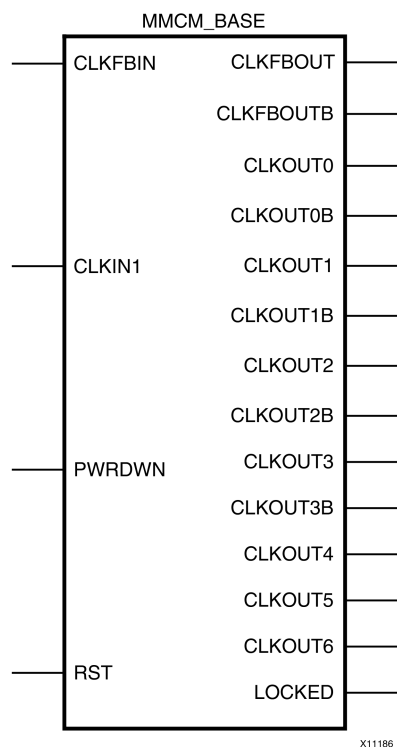
```
// End of MMCME2_ADV_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MMCME2\_BASE

プリミティブ：Base Mixed Mode Clock Manager



### 概要

MMCME2 は、周波数合成、クロック ネットワークのスキュー調整、ジッター低減をサポートするための混合信号ブロックです。各クロック出力に対して、同じ VCO 周波数を基準に分周、位相シフト、デューティ サイクルを個別に設定できます。ダイナミック位相シフトおよび分数分周もサポートされます。

### ポートの説明

ポート名	方向	幅	説明
CLKFBIN	入力	1	MMCM へのフィードバック クロック ピン
CLKFBOUT	出力	1	専用 MMCM フィードバック クロック出力
CLKFBOUTB	出力	1	CLKFBOUT の反転出力
CLKOUT0	出力	1	CLKOUT0 出力
CLKOUT0B	出力	1	CLKOUT0 の反転出力
CLKOUT1	出力	1	CLKOUT1 出力
CLKOUT1B	出力	1	CLKOUT1 の反転出力
CLKOUT2	出力	1	CLKOUT2 出力
CLKOUT2B	出力	1	CLKOUT2 の反転出力
CLKOUT3	出力	1	CLKOUT3 出力



ポート名	方向	幅	説明
CLKOUT3B	出力	1	CLKOUT3 の反転出力
CLKOUT4	出力	1	CLKOUT4 出力
CLKOUT5	出力	1	CLKOUT5 出力
CLKOUT6	出力	1	CLKOUT6 出力
クロック入力	入力	1	汎用クロック入力
PWRDWN	入力	1	インスタンシエートされているが未使用の MMCM をパワー ダウン します。
RST	入力	1	非同期リセット信号。この信号が解放されると、MMCM はクロックに同期して再びイネーブルになります。入力クロックの条件 (周波数など) を変更する場合、リセットが必要です。
LOCKED	出力	1	位相アライメントが定義されている時間内で完了し、周波数が定義されている PPM 範囲内で一致したことを示します。MMCM は電源投入時に自動的にロックされるので、リセットは必要ありません。入力クロックが停止した場合、または位相アライメントに違反が発生した場合 (入力クロックの位相シフトなど) は、LOCKED がディアサートされます。LOCKED がディアサートされると、自動的にロックが達成されます。

## デザインの入力方法

インスタンシエーション	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
BANDWIDTH	文字列	"OPTIMIZED"、 "HIGH"、"LOW"	"OPTIMIZED"	ジッター、位相マージンなどの MMCM 特性に影響する MMCM プログラム アルゴリズムを指定します。
CLKFBOUT_MULT_F	3 上位ビット 浮動小数点	2.000 ~ 64.000	5.000	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 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKFBOUT_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	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。フィードバック クロックをシフトすると、MMCM の出力クロックがすべて負の値に位相シフトします。
CLKIN1_PERIOD	浮動小数点 (nS)	0.000 ~ 100.000	0.000	CLKIN1 入力の周期を指定します。精度は ps です (小数点以下 3 桁)。たとえば、値 33.333 は 30MHz の入力クロックを示します。この値は必ず設定する必要があります。
CLKOUT1_DIVIDE、 CLKOUT2_DIVIDE、 CLKOUT3_DIVIDE、 CLKOUT4_DIVIDE、 CLKOUT5_DIVIDE、 CLKOUT6_DIVIDE	10 進数	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、CLKFBOUT_MULT_F 値および DIVCLK_DIVIDE 値により出力周波数が決まります。

属性	データ型	値	デフォルト	説明
CLKOUT0_DIVIDE_F	3 上位ビット 浮動小数点	1.000 ～ 128.000	1.000	CLKOUT0 クロック出力を分周する値を指定します。この値と、CLKFBOUT_MULT_F 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKOUT0_DUTY_CYCLE ～ CLKOUT6_DUTY_CYCLE	3 上位ビット 浮動小数点	0.001 ～ 0.999	0.500	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。
CLKOUT0_PHASE ～ CLKOUT6_PHASE	3 上位ビット 浮動小数点	-360.000 ～ 360.000	0.000	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。フィードバッククロックをシフトすると、MMCM の出力クロックがすべて負の値に位相シフトします。
CLKOUT4_CASCADE	ブール代数	FALSE、TRUE	FALSE	出力クロック分周が 128 よりも大きい場合、出力分周 (カウンタ) CLKOUT6 を CLKOUT4 分周の入力にカスケード接続します。
DIVCLK_DIVIDE	10 進数	1 ～ 106	1	すべての出力クロックの入力クロックに対する分周比を指定し、PFD に入力される CLKIN を分周します。
REF_JITTER1	3 上位ビット 浮動小数点	0.000 ～ 0.999	0.010	MMCM のパフォーマンスを最適化するため、CLKIN1 に予測されるジッター値を指定します。BANDWIDTH が "OPTIMIZED" に設定されている場合、値が既知でない場合は入力クロックに最適なパラメーターが選択されます。値が既知である場合は、値を入力クロックに予測されるジッターの UI パーセント (最大ピークトゥピーク値) で指定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 信号がアサートされるのを、MMCM がロックされるまで遅延します。

## 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;

-- MMCME2_BASE: Base Mixed Mode Clock Manager
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

MMCME2_BASE_inst : MMCME2_BASE
generic map (
    BANDWIDTH => "OPTIMIZED",    -- Jitter programming (OPTIMIZED, HIGH, LOW)
    CLKFBOUT_MULT_F => 5.0,      -- Multiply value for all CLKOUT (2.000-64.000).
    CLKFBOUT_PHASE => 0.0,       -- Phase offset in degrees of CLKFB (-360.000-360.000).
    CLKIN1_PERIOD => 0.0,        -- Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    -- CLKOUT0_DIVIDE - CLKOUT6_DIVIDE: Divide amount for each CLKOUT (1-128)
    CLKOUT1_DIVIDE => 1,
    CLKOUT2_DIVIDE => 1,
    CLKOUT3_DIVIDE => 1,
    CLKOUT4_DIVIDE => 1,
    CLKOUT5_DIVIDE => 1,
    CLKOUT6_DIVIDE => 1,
    CLKOUT0_DIVIDE_F => 1.0,     -- Divide amount for CLKOUT0 (1.000-128.000).
    -- CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for each CLKOUT (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,
CLKOUT6_DUTY_CYCLE => 0.5,
-- CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for each CLKOUT (-360.000-360.000).
CLKOUT0_PHASE => 0.0,
CLKOUT1_PHASE => 0.0,
CLKOUT2_PHASE => 0.0,
CLKOUT3_PHASE => 0.0,
CLKOUT4_PHASE => 0.0,
CLKOUT5_PHASE => 0.0,
CLKOUT6_PHASE => 0.0,
CLKOUT4_CASCADE => FALSE, -- Cascade CLKOUT4 counter with CLKOUT6 (FALSE, TRUE)
DIVCLK_DIVIDE => 1, -- Master division value (1-106)
REF_JITTER1 => 0.0, -- Reference input jitter in UI (0.000-0.999).
STARTUP_WAIT => FALSE -- Delays DONE until MMCM is locked (FALSE, TRUE)
)
port map (
-- Clock Outputs: 1-bit (each) output: User configurable clock outputs
CLKOUT0 => CLKOUT0, -- 1-bit output: CLKOUT0
CLKOUT0B => CLKOUT0B, -- 1-bit output: Inverted CLKOUT0
CLKOUT1 => CLKOUT1, -- 1-bit output: CLKOUT1
CLKOUT1B => CLKOUT1B, -- 1-bit output: Inverted CLKOUT1
CLKOUT2 => CLKOUT2, -- 1-bit output: CLKOUT2
CLKOUT2B => CLKOUT2B, -- 1-bit output: Inverted CLKOUT2
CLKOUT3 => CLKOUT3, -- 1-bit output: CLKOUT3
CLKOUT3B => CLKOUT3B, -- 1-bit output: Inverted CLKOUT3
CLKOUT4 => CLKOUT4, -- 1-bit output: CLKOUT4
CLKOUT5 => CLKOUT5, -- 1-bit output: CLKOUT5
CLKOUT6 => CLKOUT6, -- 1-bit output: CLKOUT6
-- Feedback Clocks: 1-bit (each) output: Clock feedback ports
CLKFBOUT => CLKFBOUT, -- 1-bit output: Feedback clock
CLKFBOUTB => CLKFBOUTB, -- 1-bit output: Inverted CLKFBOUT
-- Status Ports: 1-bit (each) output: MMCM status ports
LOCKED => LOCKED, -- 1-bit output: LOCK
-- Clock Inputs: 1-bit (each) input: Clock input
CLKIN1 => CLKIN1, -- 1-bit input: Clock
-- Control Ports: 1-bit (each) input: MMCM control ports
PWRDWN => PWRDWN, -- 1-bit input: Power-down
RST => RST, -- 1-bit input: Reset
-- Feedback Clocks: 1-bit (each) input: Clock feedback ports
CLKFBIN => CLKFBIN -- 1-bit input: Feedback clock
);

-- End of MMCME2_BASE_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// MMCME2_BASE: Base Mixed Mode Clock Manager
//                               7 Series
// Xilinx HDL Libraries Guide, version 14.4

MMCME2_BASE #(
    .BANDWIDTH("OPTIMIZED"), // Jitter programming (OPTIMIZED, HIGH, LOW)
    .CLKFBOUT_MULT_F(5.0),   // Multiply value for all CLKOUT (2.000-64.000).
    .CLKFBOUT_PHASE(0.0),    // Phase offset in degrees of CLKFB (-360.000-360.000).
    .CLKIN1_PERIOD(0.0),     // Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    // CLKOUT0_DIVIDE - CLKOUT6_DIVIDE: Divide amount for each CLKOUT (1-128)
    .CLKOUT1_DIVIDE(1),
    .CLKOUT2_DIVIDE(1),
    .CLKOUT3_DIVIDE(1),
    .CLKOUT4_DIVIDE(1),
    .CLKOUT5_DIVIDE(1),
    .CLKOUT6_DIVIDE(1),
    .CLKOUT0_DIVIDE_F(1.0),   // Divide amount for CLKOUT0 (1.000-128.000).
    // CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for each CLKOUT (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),
    .CLKOUT6_DUTY_CYCLE(0.5),
    // CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for each CLKOUT (-360.000-360.000).
    .CLKOUT0_PHASE(0.0),
    .CLKOUT1_PHASE(0.0),
    .CLKOUT2_PHASE(0.0),
    .CLKOUT3_PHASE(0.0),
    .CLKOUT4_PHASE(0.0),
    .CLKOUT5_PHASE(0.0),
    .CLKOUT6_PHASE(0.0),
    .CLKOUT4_CASCADE("FALSE"), // Cascade CLKOUT4 counter with CLKOUT6 (FALSE, TRUE)
    .DIVCLK_DIVIDE(1),         // Master division value (1-106)
    .REF_JITTER1(0.0),        // Reference input jitter in UI (0.000-0.999).
    .STARTUP_WAIT("FALSE")    // Delays DONE until MMCM is locked (FALSE, TRUE)
)
MMCME2_BASE_inst (
    // Clock Outputs: 1-bit (each) output: User configurable clock outputs
    .CLKOUT0(CLKOUT0),        // 1-bit output: CLKOUT0
    .CLKOUT0B(CLKOUT0B),      // 1-bit output: Inverted CLKOUT0
    .CLKOUT1(CLKOUT1),        // 1-bit output: CLKOUT1
    .CLKOUT1B(CLKOUT1B),      // 1-bit output: Inverted CLKOUT1
    .CLKOUT2(CLKOUT2),        // 1-bit output: CLKOUT2
    .CLKOUT2B(CLKOUT2B),      // 1-bit output: Inverted CLKOUT2
    .CLKOUT3(CLKOUT3),        // 1-bit output: CLKOUT3
    .CLKOUT3B(CLKOUT3B),      // 1-bit output: Inverted CLKOUT3
    .CLKOUT4(CLKOUT4),        // 1-bit output: CLKOUT4
    .CLKOUT5(CLKOUT5),        // 1-bit output: CLKOUT5
    .CLKOUT6(CLKOUT6),        // 1-bit output: CLKOUT6
    // Feedback Clocks: 1-bit (each) output: Clock feedback ports
    .CLKFBOUT(CLKFBOUT),      // 1-bit output: Feedback clock
    .CLKFBOUTB(CLKFBOUTB),    // 1-bit output: Inverted CLKFBOUT
    // Status Ports: 1-bit (each) output: MMCM status ports
    .LOCKED(LOCKED),          // 1-bit output: LOCK
    // Clock Inputs: 1-bit (each) input: Clock input
    .CLKIN1(CLKIN1),          // 1-bit input: Clock
    // Control Ports: 1-bit (each) input: MMCM control ports
    .PWRDWN(PWRDWN),          // 1-bit input: Power-down
    .RST(RST),                // 1-bit input: Reset
    // Feedback Clocks: 1-bit (each) input: Clock feedback ports
    .CLKFBIN(CLKFBIN)         // 1-bit input: Feedback clock
);

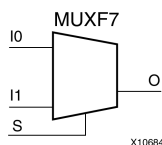
// End of MMCME2_BASE_inst instantiation
```

詳細情報

[7 シリーズ 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 エLEMENTを組み合わせ、7 入力ファンクション、8:1 マルチプレクサー、または 12 ビット幅までのロジック ファンクションを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、LUT6 のローカル出力を接続します。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 (LUT6 の LO 出力に接続)
I1	入力	1	入力 (LUT6 の 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;

-- MUXF7: CLB MUX to tie two LUT6's together with general output
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

MUXF7_inst : MUXF7
port map (
    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
```

## Verilog 記述 (インスタンス化)

```
// MUXF7: CLB MUX to tie two LUT6's together with general output
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

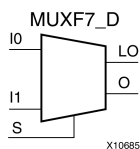
// End of MUXF7_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 ELEMENTを組み合わせ、7 入力ファンクション、8:1 マルチプレクサー、または 12 ビット幅までのロジック ファンクションを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、LUT6 のローカル出力を接続します。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 (LUT6 の LO 出力に接続)
I1	入力	1	入力 (LUT6 の 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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 LUT6 O6 pin)
    I1 => I1, -- Input (tie to 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
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

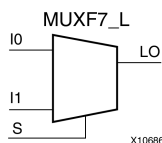
// End of MUXF7_D_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 エLEMENTを組み合わせ、7 入力ファンクション、8:1 マルチプレクサー、または 12 ビット幅までのロジック ファンクションを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、LUT6 のローカル出力を接続します。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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

MUXF7_L_inst : MUXF7_L
port map (
    LO => LO, -- 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
```

## 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
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

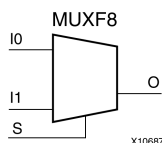
// End of MUXF7_L_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 エレメントは、2 つの MUXF7 マルチプレクサーおよび 4 つの LUT6 エレメントと組み合わせて、8 入力ファンクション、16:1 マルチプレクサー、または 24 ビット幅までのロジック ファンクションを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力を接続します。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 MUXF8\_D および MUXF8\_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予測をより正確に行う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 (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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

MUXF8_inst : MUXF8
port map (
    O => O,    -- Output of MUX to general routing
    I0 => I0,   -- Input (tie to MUXF7 L/LO out)
    I1 => I1,   -- Input (tie to MUXF7 L/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
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

MUXF8 MUXF8_inst (
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie to MUXF7 L/LO out)
    .I1(I1),    // Input (tie to MUXF7 L/LO out)
    .S(S)       // Input select to MUX
);

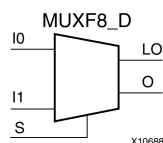
// End of MUXF8_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 エレメントは、2 つの MUXF7 マルチプレクサーおよび 4 つの LUT6 エレメントと組み合わせて、8 入力ファンクション、16:1 マルチプレクサー、または 24 ビット幅までのロジック ファンクションを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力を接続します。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。

「MUXF8」および「MUXF8\_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 (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
--           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 L/LO out)
    I1 => I1, -- Input (tie to MUXF7 L/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
//           7 Series
// Xilinx HDL Libraries Guide, version 14.4

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 L/LO out)
    .I1(I1), // Input (tie to MUXF7 L/LO out)
    .S(S)    // Input select to MUX
);

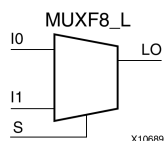
// End of MUXF8_D_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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、2 つの MUXF7 マルチプレクサーおよび 4 つの LUT6 エLEMENTと組み合わせて、8 入力ファンクション、16:1 マルチプレクサー、または 24 ビット幅までのロジック ファンクションを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力を接続します。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。

「MUXF8」および「MUXF8\_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 (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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

MUXF8_L_inst : MUXF8_L
port map (
    LO => LO,  -- Output of MUX to local routing
    IO => IO,  -- Input (tie to MUXF7 L/LO out)
    I1 => I1,  -- Input (tie to MUXF7 L/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
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

MUXF8_L MUXF8_L_inst (
    .LO(LO), // Output of MUX to local routing
    .IO(IO), // Input (tie to MUXF7 L/LO out)
    .I1(I1), // Input (tie to MUXF7 L/LO out)
    .S(S)    // Input select to MUX
);

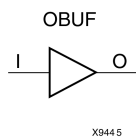
// End of MUXF8_L_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## OBUF

### プリミティブ：Output Buffer



### 概要

このデザイン エレメントは単純な出力バッファで、トライステートする必要のない（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。

このエレメント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。このエレメントでは LVCMOS18 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。

### 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します。

### 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
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することが推奨されます。
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"	"SLOW"	出力ドライバースルー 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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

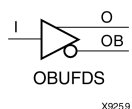
// End of OBUF_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## OBUFDS

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



### 概要

このデザイン エレメントは、低電圧差動信号をサポートする単一出力バッファです。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
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;

-- OBUFDS: Differential Output Buffer
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

OBUFDS_inst : OBUFDS
generic map (
    IOSTANDARD => "DEFAULT", -- Specify the output I/O standard
    SLEW => "SLOW")         -- Specify the output slew rate
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 (インスタンスレーション)

```
// OBUFDS: Differential Output Buffer
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

OBUFDS #(
    .IOSTANDARD("DEFAULT"), // Specify the output I/O standard
    .SLEW("SLOW")           // Specify the output slew rate
) 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
);

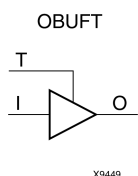
// End of OBUFDS_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 LVCMOS18 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。

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	説明
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
DRIVE	整数	2、4、6、8、12、16、24	12	I/O の出力駆動電流を指定します。許容範囲で最も低い値を使用することが推奨されます。
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"	"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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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
//       All devices
// Xilinx HDL Libraries Guide, version 14.4

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
);

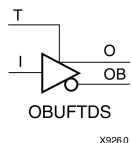
// End of OBUFT_inst instantiation
```

## 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 エレメントは、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	説明
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。
SLEW	文字列	"SLOW"、"FAST"	"SLOW"	出力ドライバのスルー 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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

OBUFTDS #(
    .IOSTANDARD("DEFAULT"), // Specify the output I/O standard
    .SLEW("SLOW")           // Specify the output slew rate
) 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
);

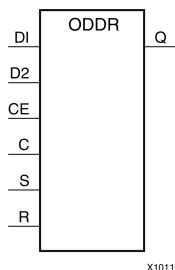
// End of OBUFTDS_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ODDR

### プリミティブ：Dedicated Dual Data Rate (DDR) Output Register



### 概要

このデザイン エLEMENTは、FPGA デバイスからデュアル データレート (DDR) 信号を送信するための専用出力レジスタです。ODDR プリミティブでは、FPGA からのデータを送信するのに反対のクロック エッジだけではなく、同じクロック エッジを使用することも可能です。これにより、タイミングが複雑にならず、追加の CLB リソースも必要ありません。ODDR は SelectIO™ 機能とも使用できます。

#### ODDR のモード

このELEMENTは 2 つのモードで動作します。これらのモードは、DDR\_CLK\_EDGE 属性で設定します。

- OPPOSITE\_EDGE モード：**通常の DDR 方式でデータを送信します。D1 はクロック C の立ち上がりエッジごとにサンプリングされ、D2 は立ち下がりエッジごとにサンプリングされます。Q は各クロック エッジで変化します。
- SAME\_EDGE モード：**データはクロック C の反対のエッジで ODDR 出力から送信されますが、ODDR への 2 つの入力はクロック信号 C の立ち上がりエッジで動作し、追加されたレジスタがクロック信号 C の立ち下がりエッジで動作します。この機能を使用すると、DDR データは同じクロック エッジで ODDR に取り込まれます。

### ポートの説明

ポート名	方向	幅	機能
Q	出力	1	データ出力 (DDR)。IOB パッドに接続されます。
C	入力	1	クロック入力
CE	入力	1	クロック イネーブル入力。High になると、ポート C のクロック入力がいネーブルになります。
D1 : D2	入力	1 (それぞれ)	データ入力。DDR データを ODDR モジュールに入力するピンです。
R	入力	1	リセット。SRTYPE の設定によって異なります。
S	入力	1	アクティブ High の非同期セットピン。SRTYPE 属性の設定により、同期にもなります。

### 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_CLK_EDGE	文字列	"OPPOSITE_EDGE"、 "SAME_EDGE"	"OPPOSITE_EDGE"	クロック エッジに対する DDR の操作モードを指定します。
INIT	整数	0、1	1	Q の初期値を設定します。
SRTYPE	文字列	"SYNC"、"ASYN"	"SYNC"	セット/リセットのタイプを選択します。

## 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;

-- ODDR: Output Double Data Rate Output Register with Set, Reset
-- and Clock Enable.
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

ODDR_inst : ODDR
generic map(
  DDR_CLK_EDGE => "OPPOSITE_EDGE", -- "OPPOSITE_EDGE" or "SAME_EDGE"
  INIT => '0', -- Initial value for Q port ('1' or '0')
  SRTYPE => "SYNC") -- Reset Type ("ASYN" or "SYNC")
port map (
  Q => Q, -- 1-bit DDR output
  C => C, -- 1-bit clock input
  CE => CE, -- 1-bit clock enable input
  D1 => D1, -- 1-bit data input (positive edge)
  D2 => D2, -- 1-bit data input (negative edge)
  R => R, -- 1-bit reset input
  S => S -- 1-bit set input
);

-- End of ODDR_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// ODDR: Output Double Data Rate Output Register with Set, Reset
// and Clock Enable.
// 7 Series
// Xilinx HDL Libraries Guide, version 14.4

ODDR #(
  .DDR_CLK_EDGE("OPPOSITE_EDGE"), // "OPPOSITE_EDGE" or "SAME_EDGE"
  .INIT(1'b0), // Initial value of Q: 1'b0 or 1'b1
  .SRTYPE("SYNC") // Set/Reset type: "SYNC" or "ASYN"
) ODDR_inst (
  .Q(Q), // 1-bit DDR output
  .C(C), // 1-bit clock input
  .CE(CE), // 1-bit clock enable input
  .D1(D1), // 1-bit data input (positive edge)
  .D2(D2), // 1-bit data input (negative edge)
  .R(R), // 1-bit reset
  .S(S) // 1-bit set
);

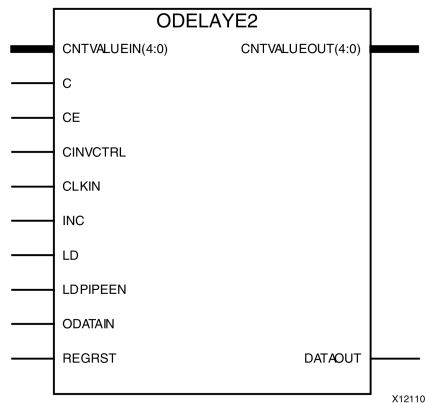
// End of ODDR_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ODELAYE2

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



### 概要

このデザイン エLEMENTは、7 シリーズ FPGA の出力パスに固定遅延または可変遅延を追加します。この遅延は、外部データのアライメント、外部位相オフセットおよび同時スイッチ ノイズ (SSN) の低減、プロセス、温度、電圧 (PVT) の変動による外部データ アライメントの監視に有益です。IDELAYCTRL コンポーネントと共に使用すると、正確なタイミングで遅延を増減できます。VARIABLE モードでは、出力パスを調整して遅延の量を増分または減分できます。7 シリーズ デバイスでは、HR (High Range) バンクに ODELAYE2 はありません。

### ポートの説明

ポート名	方向	幅	説明
C	入力	1	ODELAYE2 プリミティブの制御入力 (CNTVALUEIN、RST、CE、LD、LDPIPEEN、および INC) は、すべてクロック入力 (C) に同期しています。ODELAYE2 を "VARIABLE"、"VAR_LOAD"、または "VAR_LOAD_PIPE" モードでコンフィギュレーションする場合は、クロックをこのポートに接続する必要があります。C はローカルで反転でき、グローバルまたはリージョナル クロック バッファから供給する必要があります。このクロックは SelectIO ロジックリソースの同じクロックに接続する必要があります (OSERDESE2 を使用する場合は C は CLKDIV に接続)。ODELAYE2 を "FIXED" モードでコンフィギュレーションする場合は、GND に接続してください。
CE	入力	1	インクリメント/デクリメントをイネーブル/ディスエーブルにするアクティブ High の信号ODELAYE2 を "FIXED" モードでコンフィギュレーションする場合は、GND に接続してください。
CINVCTRL	入力	1	C ピンの極性を動的に切り替えます。この機能は、グリッチが問題にならないアプリケーションで使用します。極性を切り替える際は、2 クロック サイクル間 ODELAYE2 制御ピンを使用しないでください。ODELAYE2 を "FIXED" モードでコンフィギュレーションする場合は、GND に接続してください。
CLKIN	入力	1	遅延されたクロック入力
CNTVALUEIN<4:0>	入力	5	"VAR_LOAD" または "VAR_LOAD_PIPE" モードでコンフィギュレーションする場合に、動的に読み込まれるタプル値用の FPGA ロジックからのカウンタ値。ODELAYE2 を "FIXED" または "VARIABLE" モードでコンフィギュレーションする場合は、GND に接続してください。
CNTVALUEOUT<4:0>	出力	5	遅延エレメントの値の動的な変更をレポートします。ODELAYE2 が "VAR_LOAD" または "VAR_LOAD_PIPE" モードの場合にのみ有効です。

ポート名	方向	幅	説明
DATAOUT	出力	1	CLKIN または ODATAIN ポートからの遅延されたデータ/クロック。データの場合は I/O ポートに接続され、クロックの場合はクロック構造に戻されます。
INC	入力	1	インクリメント/デクリメントは、イネーブル信号 (CE) により制御されます。 "VARIABLE"、"VAR_LOAD"、または "VAR_LOAD_PIPE" モードの場合にのみ有効です。
LD	入力	1	初期値または読み込まれた値をカウンタに読み込みます。
LDPIPEEN	入力	1	LD ピンからデータを読み込む際にパイプライン レジスタをイネーブルにします。
ODATAIN	入力	1	OSERDESE2 または出力レジスタで駆動される、遅延される出力データ
REGRST	入力	1	アクティブ High のリセットで、入力クロック入力 (C) に同期しています。アサートすると、LDPIPEEN がアサートされていない場合はタップ値が 0 に戻り、LDPIPEEN がアサートされている場合はタップ値は CNTVALUEIN ポートの値になります。

## 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
CINVCTRL_SEL	文字列	"FALSE"、"TRUE"	"FALSE"	C ピンの極性を動的に切り替える CINVCTRL_SEL ピンをイネーブルにします。
DELAY_SRC	文字列	"ODATAIN"、"CLKIN"	"ODATAIN"	データ入力のソースを選択します。 <ul style="list-style-type: none"> <li>"ODATAIN" : ODELAYE2 チェーンの           入力 は ODATAIN</li> <li>"CLKIN" : ODELAYE2 チェーンの           入力 は CLKIN</li> </ul>
HIGH_PERFORMANCE_MODE	文字列	"FALSE"、"TRUE"	"FALSE"	"TRUE" に設定すると出力ジッターが減少し、 "FALSE" に設定すると消費電力が削減されます。 消費電力量の差異は、Xilinx Power Estimator ツールで確認できます。

属性	データ型	値	デフォルト	説明
ODELAY_TYPE	文字列	"FIXED"、 "VARIABLE"、 "VAR_LOAD"、 "VAR_LOAD_PIPE"	"FIXED"	タップ遅延ラインのタイプを設定します。 <ul style="list-style-type: none"> <li>"FIXED" : スタティック遅延値を設定します。</li> <li>"VARIABLE" : 遅延値を動的に調整 (インクリメントまたはデクリメント) します。</li> <li>"VAR_LOAD" : タップ値を動的に読み込みます。</li> <li>"VAR_LOAD_PIPE" : タップ値をパイプラインを介して動的に読み込みます。</li> </ul>
ODELAY_VALUE	10 進数	0、1、2、3、4、5、6、7、 8、9、10、11、12、13、 14、15、16、17、18、19、 20、21、22、23、24、25、 26、27、28、29、30、31	0	"FIXED" モードでは遅延タップ数、 "VARIABLE" モードでは遅延タップ数の初期値を指定します (出力パス)。 "VAR_LOAD" または "VAR_LOAD_PIPE" モードの場合、この属性値は無視されます。
PIPE_SEL	文字列	"FALSE"、"TRUE"	"FALSE"	パイプライン モードを選択します。
REFCLK_FREQUENCY	1 上位ビット 浮動小数点	190.0 ~ 210.0 および 290.0 ~ 310.0	200.0	Timing Analyzer でスタティック タイミング解析、論理シミュレーション、タイミングシミュレーションに使用するタップ値 (MHz) を設定します。適切なタップ遅延値およびパフォーマンスを得るには、REFCLK の周波数をデータシートに記載された範囲内にする必要があります。
SIGNAL_PATTERN	文字列	"DATA"、"CLOCK"	"DATA"	タイミング解析で、不規則な遷移を含む "DATA" パターンまたは規則的な立ち上がり/立下りパターンを含む "CLOCK" パターンが現れたときに、適切な遅延チェーン ジッター量が考慮されます。

## 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;

-- ODELAYE2: Output Fixed or Variable Delay Element
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

ODELAYE2_inst : ODELAYE2
generic map (
    CINVCTRL_SEL => "FALSE",           -- Enable dynamic clock inversion (FALSE, TRUE)
    DELAY_SRC => "ODATAIN",             -- Delay input (ODATAIN, CLKIN)
    HIGH_PERFORMANCE_MODE => "FALSE",  -- Reduced jitter ("TRUE"), Reduced power ("FALSE")
    ODELAY_TYPE => "FIXED",             -- FIXED, VARIABLE, VAR_LOAD, VAR_LOAD_PIPE
    ODELAY_VALUE => 0,                  -- Output delay tap setting (0-31)
    PIPE_SEL => "FALSE",                -- Select pipelined mode, FALSE, TRUE
    REFCLK_FREQUENCY => 200.0,          -- IDELAYCTRL clock input frequency in MHz (190.0-210.0).
    SIGNAL_PATTERN => "DATA"            -- DATA, CLOCK input signal
)
port map (

```

```

CNTVALUEOUT => CNTVALUEOUT, -- 5-bit output: Counter value output
DATAOUT => DATAOUT,        -- 1-bit output: Delayed data/clock output
C => C,                      -- 1-bit input: Clock input
CE => CE,                    -- 1-bit input: Active high enable increment/decrement input
CINVCTRL => CINVCTRL,        -- 1-bit input: Dynamic clock inversion input
CLKIN => CLKIN,              -- 1-bit input: Clock delay input
CNTVALUEIN => CNTVALUEIN,    -- 5-bit input: Counter value input
INC => INC,                  -- 1-bit input: Increment / Decrement tap delay input
LD => LD,                    -- 1-bit input: Loads ODELAY_VALUE tap delay in VARIABLE mode, in VAR_LOAD or
                                -- VAR_LOAD_PIPE mode, loads the value of CNTVALUEIN

LDPIPEEN => LDPIPEEN,        -- 1-bit input: Enables the pipeline register to load data
ODATAIN => ODATAIN,          -- 1-bit input: Output delay data input
REGRST => REGRST             -- 1-bit input: Active-high reset tap-delay input
);

-- End of ODELAYE2_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// ODELAYE2: Output Fixed or Variable Delay Element
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

(* IODELAY_GROUP = <iodelay_group_name> *) // Specifies group name for associated IDELAYs/ODELAYs and IDELAYCTRL

ODELAYE2 #(
    .CINVCTRL_SEL("FALSE"), // Enable dynamic clock inversion (FALSE, TRUE)
    .DELAY_SRC("ODATAIN"),  // Delay input (ODATAIN, CLKIN)
    .HIGH_PERFORMANCE_MODE("FALSE"), // Reduced jitter ("TRUE"), Reduced power ("FALSE")
    .ODELAY_TYPE("FIXED"),  // FIXED, VARIABLE, VAR_LOAD, VAR_LOAD_PIPE
    .ODELAY_VALUE(0),       // Output delay tap setting (0-31)
    .PIPE_SEL("FALSE"),    // Select pipelined mode, FALSE, TRUE
    .REFCLK_FREQUENCY(200.0), // IDELAYCTRL clock input frequency in MHz (190.0-210.0)
    .SIGNAL_PATTERN("DATA") // DATA, CLOCK input signal
)
ODELAYE2_inst (
    .CNTVALUEOUT(CNTVALUEOUT), // 5-bit output: Counter value output
    .DATAOUT(DATAOUT),         // 1-bit output: Delayed data/clock output
    .C(C),                     // 1-bit input: Clock input
    .CE(CE),                   // 1-bit input: Active high enable increment/decrement input
    .CINVCTRL(CINVCTRL),      // 1-bit input: Dynamic clock inversion input
    .CLKIN(CLKIN),             // 1-bit input: Clock delay input
    .CNTVALUEIN(CNTVALUEIN),   // 5-bit input: Counter value input
    .INC(INC),                 // 1-bit input: Increment / Decrement tap delay input
    .LD(LD),                   // 1-bit input: Loads ODELAY_VALUE tap delay in VARIABLE mode, in VAR_LOAD or
                                // VAR_LOAD_PIPE mode, loads the value of CNTVALUEIN

    .LDPIPEEN(LDPIPEEN),      // 1-bit input: Enables the pipeline register to load data
    .ODATAIN(ODATAIN),        // 1-bit input: Output delay data input
    .REGRST(REGRST)           // 1-bit input: Active-high reset tap-delay input
);

// End of ODELAYE2_inst instantiation

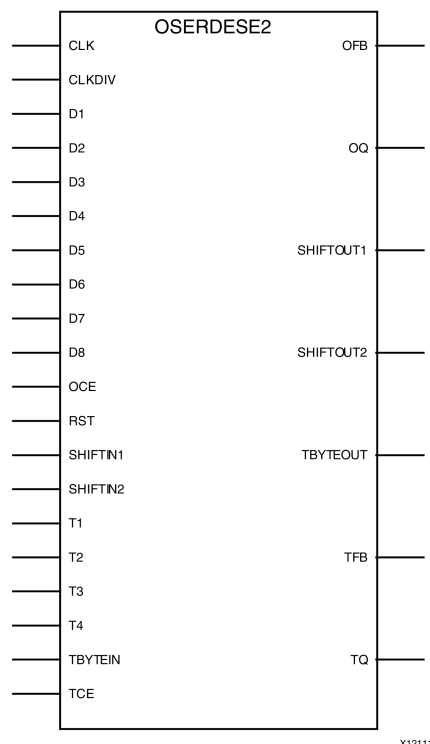
```

### 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## OSERDESE2

プリミティブ：Output SERial/DESerializer with bitslip



### 概要

7 シリーズ デバイスの OSERDES は、高速ソース同期インターフェイスのインプリメンテーションに特化したクロックおよびロジックリソースを持つ、専用パラレル/シリアル コンバーターです。各 OSERDES モジュールに、データおよびトライステート制御用の専用シリアライザーが含まれています。データ シリアライザーおよびトライステート シリアライザーは、両方ともシングル データ レート (SDR) およびダブル データ レート (DDR) モードにコンフィギュレーションできます。データでは 8:1 まで (OSERDES 幅を拡張している場合は 10:1 または 14:1 まで) のシリアル化、トライステートでは 4:1 までのシリアル化が可能です。

### ポートの説明

ポート名	方向	幅	説明
CLK	入力	1	高速クロック入力。パラレル/シリアル コンバーターのシリアル側を駆動します。
CLKDIV	入力	1	パラレル/シリアル コンバーターのパラレル側を駆動します。CLK ポートに接続されたクロックを分周したクロックです。
D1 ~ D8	入力	1	パラレル データ入力。パラレル データは、D1 ~ D8 から OSERDES モジュールに入力されます。これらのポートは FPGA に接続され、2 ~ 8 ビット (8:1 シリアル化) にコンフィギュレーションできます。2 つ目の OSERDES を SLAVE モードで使用することにより、14 ビットまでの幅をサポートできます。
OCE	入力	1	データ パス用のアクティブ High のクロック イネーブル
OFB	出力	1	出力フィードバック ポート。OSERDESE2 のシリアル (高速) データ出力ポートです。



ポート名	方向	幅	説明
OQ	出力	1	OSERDES モジュールのデータ出力ポート。入力ポート D1 のデータが OQ に一番最初に出力されます。このポートは、データ パラレル/シリアル コンバーター の出力と IOB のデータ入力を接続します。このポートで ODELAYE2 を駆動することはできません。OFB ピンを使用する必要があります。
RST	入力	1	CLK および CLKDIV ドメインのすべてのデータ フリップフロップの出力を非同期で Low に駆動します。OSERDES 回路が CLK ドメインで動作しており、タイミングがクリティカルな場合は、内部専用回路を使用して RST 入力のタイミングを調整し、CLK ドメインに同期したリセット信号を生成してください。同様に、CLKDIV ドメインに同期したリセット信号を生成する RST 入力のタイミングを調整する専用回路があります。RST 入力のタイミングを調整する OSERDES 回路があるので、CLKDIV 周波数ドメイン (CLKDIV に同期) でタイミングを満たす RST 入力にリセットパルスを供給するだけですみます。RST は、CLKDIV の 1 サイクル以上 High に駆動する必要があります。複数の OSERDES ポートを含むインターフェイスを構築する場合は、すべての OSERDES ポートを同期化する必要があります。RST 入力の内部タイミングは、同じリセットパルスを受信するすべての OSERDES ブロックがお互いに同期してリセット状態から戻るように調整されます。
SHIFTIN1/ SHIFTIN2	入力	1	データ入力を拡張するためのカスケード入力です。スレーブの SHIFTOUT1、SHIFTOUT2 に接続します。
SHIFTOUT1/ SHIFTOUT2	出力	1	データ入力を拡張するためのカスケード出力です。マスターの SHIFTIN1、SHIFTIN2 に接続します。
TBYTEIN	入力	1	ソースからのバイト グループ トライステート入力
TBYTEOUT	出力	1	IOB へのバイト グループ トライステート出力
TCE	入力	1	トライステート制御パス用のアクティブ High のクロック イネーブル
TFB	出力	1	ODELAYE2 に送信される OSERDES モジュールのトライステート制御出力。トライステート パラレル/シリアル コンバーターの出力を ODELAYE2 の制御/トライステート入力に接続します。
TQ	出力	1	OSERDES モジュールのトライステート制御出力。トライステート パラレル/シリアル コンバーターの出力を IOB の制御/トライステート入力に接続します。
T1 ~ T4	入力	1	パラレル トライステート入力。パラレル トライステート信号は、T1 ~ T4 から OSERDES モジュールに入力されます。このポートは FPGA に接続され、1、2、または 4 ビット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	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
DATA_RATE_OQ	文字列	"DDR"、"SDR"	"DDR"	データをシングル データ レート (SDR) またはダブル データ レート (DDR) のいずれで処理するかを指定します。
DATA_RATE_TQ	文字列	"DDR"、"BUF"、"SDR"	"DDR"	トライステート制御をシングル データ レート (SDR) またはダブル データ レート (DDR) のいずれで処理するかを指定します。
DATA_WIDTH	10 進数	4、2、3、5、6、7、8、10、14	4	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。有効な値は、DATA_RATE_OQ が "SDR" の場合は 2、3、4、5、6、7、および 8、"DDR" の場合は 4、6、8、10、および 14 です。DATA_WIDTH を 10 以上の値に設定した場合、2 つの OSERDES をマスターおよびスレーブとしてコンフィギュレーションする必要があります。
INIT_OQ	2 進数	1'b0 ~ 1'b1	1'b0	OQ 出力の初期値を指定します。
INIT_TQ	2 進数	1'b0 ~ 1'b1	1'b0	TQ 出力の初期値を指定します。
SERDES_MODE	文字列	"MASTER"、"SLAVE"	"MASTER"	データ幅を拡張する場合に OSERDES モジュールがマスターであるかスレーブであるかを指定します。
SRVAL_OQ	2 進数	1'b0 ~ 1'b1	1'b0	SR をアサートした場合の OQ 出力の値を指定します。
SRVAL_TQ	2 進数	1'b0 ~ 1'b1	1'b0	SR をアサートした場合の YQ 出力の値を指定します。
TBYTE_CTL	文字列	"FALSE"、"TRUE"	"FALSE"	DDR3 モードのトライステート バイト操作をイネーブルにします。この操作をイネーブルにすると、トライステート信号の値をソースとなるトライステート出力のいずれかからの値にできます。
TBYTE_SRC	文字列	"FALSE"、"TRUE"	"FALSE"	DDR3 モードで OSERDES をトライステート バイト操作のソースとして使用できるようにします。
TRISTATE_WIDTH	10 進数	4、1	4	トライステート制御パラレル/シリアル コンバーターのパラレル トライステート入力幅を定義します。有効な値は DATA_RATE_TQ 属性の設定によって異なり、DATA_RATE_TQ が "SDR" または "BUF" の場合は 1 に設定し、"DDR" の場合は 4 に設定します。5 以上の幅に設定することはできません。DATA_WIDTH が 5 以上の場合は、TRISTATE_WIDTH を 1 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;

-- OSERDESE2: Output SERIAL/DESerializer with bitslip
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

OSERDESE2_inst : OSERDESE2
generic map (
    DATA_RATE_OQ => "DDR",      -- DDR, SDR
    DATA_RATE_TQ => "DDR",      -- DDR, BUF, SDR
    DATA_WIDTH => 4,            -- Parallel data width (2-8,10,14)
    INIT_OQ => '0',              -- Initial value of OQ output (1'b0,1'b1)
    INIT_TQ => '0',              -- Initial value of TQ output (1'b0,1'b1)
    SERDES_MODE => "MASTER",    -- MASTER, SLAVE
    SRVAL_OQ => '0',             -- OQ output value when SR is used (1'b0,1'b1)
    SRVAL_TQ => '0',             -- TQ output value when SR is used (1'b0,1'b1)
    TBYTE_CTL => "FALSE",       -- Enable tristate byte operation (FALSE, TRUE)
    TBYTE_SRC => "FALSE",       -- Tristate byte source (FALSE, TRUE)
    TRISTATE_WIDTH => 4         -- 3-state converter width (1,4)
)
port map (
    OFB => OFB,                  -- 1-bit output: Feedback path for data
    OQ => OQ,                     -- 1-bit output: Data path output
    -- SHIFTOUT1 / SHIFTOUT2: 1-bit (each) output: Data output expansion (1-bit each)
    SHIFTOUT1 => SHIFTOUT1,
    SHIFTOUT2 => SHIFTOUT2,
    TBYTEOUT => TBYTEOUT,        -- 1-bit output: Byte group tristate
    TFB => TFB,                  -- 1-bit output: 3-state control
    TQ => TQ,                    -- 1-bit output: 3-state control
    CLK => CLK,                  -- 1-bit input: High speed clock
    CLKDIV => CLKDIV,            -- 1-bit input: Divided clock
    -- D1 - D8: 1-bit (each) input: Parallel data inputs (1-bit each)
    D1 => D1,
    D2 => D2,
    D3 => D3,
    D4 => D4,
    D5 => D5,
    D6 => D6,
    D7 => D7,
    D8 => D8,
    OCE => OCE,                  -- 1-bit input: Output data clock enable
    RST => RST,                  -- 1-bit input: Reset
    -- SHIFTIN1 / SHIFTIN2: 1-bit (each) input: Data input expansion (1-bit each)
    SHIFTIN1 => SHIFTIN1,
    SHIFTIN2 => SHIFTIN2,
    -- T1 - T4: 1-bit (each) input: Parallel 3-state inputs
    T1 => T1,
    T2 => T2,
    T3 => T3,
    T4 => T4,
    TBYTEIN => TBYTEIN,         -- 1-bit input: Byte group tristate
    TCE => TCE                   -- 1-bit input: 3-state clock enable
);

-- End of OSERDESE2_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// OSERDESE2: Output SERIAL/DESerializer with bitslip
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

OSERDESE2 #(
    .DATA_RATE_OQ("DDR"),    // DDR, SDR
    .DATA_RATE_TQ("DDR"),    // DDR, BUF, SDR
    .DATA_WIDTH(4),          // Parallel data width (2-8,10,14)
    .INIT_OQ(1'b0),          // Initial value of OQ output (1'b0,1'b1)
    .INIT_TQ(1'b0),          // Initial value of TQ output (1'b0,1'b1)
    .SERDES_MODE("MASTER"), // MASTER, SLAVE
    .SRVAL_OQ(1'b0),          // OQ output value when SR is used (1'b0,1'b1)
    .SRVAL_TQ(1'b0),          // TQ output value when SR is used (1'b0,1'b1)
    .TBYTE_CTL("FALSE"),     // Enable tristate byte operation (FALSE, TRUE)
    .TBYTE_SRC("FALSE"),     // Tristate byte source (FALSE, TRUE)
    .TRISTATE_WIDTH(4)        // 3-state converter width (1,4)
)
OSERDESE2_inst (
    .OFB(OFB),                // 1-bit output: Feedback path for data
    .OQ(OQ),                  // 1-bit output: Data path output
    // SHIFTOUT1 / SHIFTOUT2: 1-bit (each) output: Data output expansion (1-bit each)
    .SHIFTOUT1(SHIFTOUT1),
    .SHIFTOUT2(SHIFTOUT2),
    .TBYTEOUT(TBYTEOUT),      // 1-bit output: Byte group tristate
    .TFB(TFB),                // 1-bit output: 3-state control
    .TQ(TQ),                  // 1-bit output: 3-state control
    .CLK(CLK),                // 1-bit input: High speed clock
    .CLKDIV(CLKDIV),          // 1-bit input: Divided clock
    // D1 - D8: 1-bit (each) input: Parallel data inputs (1-bit each)
    .D1(D1),
    .D2(D2),
    .D3(D3),
    .D4(D4),
    .D5(D5),
    .D6(D6),
    .D7(D7),
    .D8(D8),
    .OCE(OCE),                // 1-bit input: Output data clock enable
    .RST(RST),                // 1-bit input: Reset
    // SHIFTTIN1 / SHIFTTIN2: 1-bit (each) input: Data input expansion (1-bit each)
    .SHIFTTIN1(SHIFTTIN1),
    .SHIFTTIN2(SHIFTTIN2),
    // T1 - T4: 1-bit (each) input: Parallel 3-state inputs
    .T1(T1),
    .T2(T2),
    .T3(T3),
    .T4(T4),
    .TBYTEIN(TBYTEIN),        // 1-bit input: Byte group tristate
    .TCE(TCE)                 // 1-bit input: 3-state clock enable
);

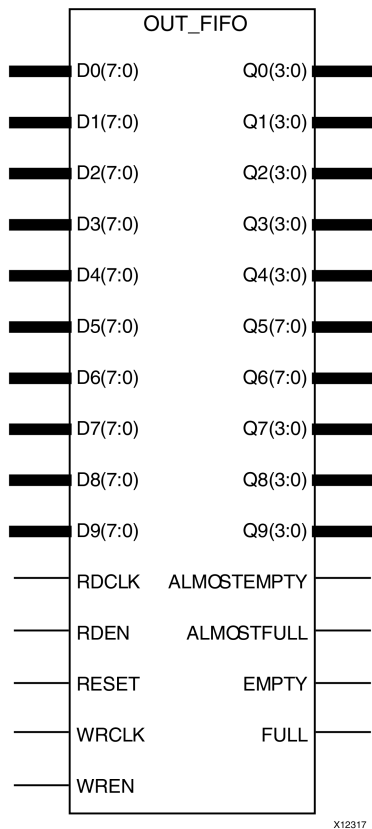
// End of OSERDESE2_inst instantiation
```

## 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』\(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## OUT\_FIFO

プリミティブ：Output First-In, First-Out (FIFO) Buffer



出力 FIFO は、I/O の横に配置されている新しいリソースです。この専用ハードウェアは、データをファブリックから I/O、ODDR、または OSERDES をに転送するために設計されたものです。FIFO に入力されたデータが同じレートで出力される 4x4 モードと、データが係数 2 でシリアルライズされてから出力される 8x4 モードがあります。つまり、8x4 モードでは 8 ビットのデータが OUT\_FIFO に入力され、4 ビットのデータが出力されます。

出力 FIFO は、I/O の横に配置されている新しいリソースです。この専用ハードウェアは、データをファブリックから I/O、ODDR、または OSERDES をに転送するために設計されたものです。FIFO に入力されたデータが同じレートで出力される 4x4 モードと、データが係数 2 でシリアルライズされてから出力される 8x4 モードがあります。つまり、8x4 モードでは 8 ビットのデータが OUT\_FIFO に入力され、4 ビットのデータが出力されます。このコンポーネントには、次の機能があります。

- ・ アレイ サイズ：幅 80、深さ 8 (8x4 モード)、幅 40、深さ 8 (4x4 モード)
- ・ EMPTY および FULL フラグ
- ・ プログラム可能な ALMOSTEMPTY および ALMOSTFULL フラグ

## ポートの説明

ポート名	方向	幅	説明
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示すアクティブ High の出力フラグ。しきい値は ALMOST_EMPTY_VALUE 属性で設定します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示すアクティブ High の出力フラグ。しきい値は ALMOST_FULL_VALUE 属性で設定します。
D0<7:0>	入力	8	チャネル 0 入力バス
D1<7:0>	入力	8	チャネル 1 入力バス
D2<7:0>	入力	8	チャネル 2 入力バス
D3<7:0>	入力	8	チャネル 3 入力バス
D4<7:0>	入力	8	チャネル 4 入力バス
D5<7:0>	入力	8	チャネル 5 入力バス
D6<7:0>	入力	8	チャネル 6 入力バス
D7<7:0>	入力	8	チャネル 7 入力バス
D8<7:0>	入力	8	チャネル 8 入力バス
D9<7:0>	入力	8	チャネル 9 入力バス
EMPTY	出力	1	FIFO が空であることを示すアクティブ High の出力フラグ
FULL	出力	1	FIFO がフルであることを示すアクティブ High の出力フラグ
Q0<3:0>	出力	4	チャネル 0 出力バス
Q1<3:0>	出力	4	チャネル 1 出力バス
Q2<3:0>	出力	4	チャネル 2 出力バス
Q3<3:0>	出力	4	チャネル 3 出力バス
Q4<3:0>	出力	4	チャネル 4 出力バス
Q5<7:0>	出力	8	チャネル 5 出力バス
Q6<7:0>	出力	8	チャネル 6 出力バス
Q7<3:0>	出力	4	チャネル 7 出力バス
Q8<3:0>	出力	4	チャネル 8 出力バス
Q9<3:0>	出力	4	チャネル 9 出力バス
RDCLK	入力	1	読み出しクロック
RDEN	入力	1	アクティブ High のリード イネーブル
RESET	入力	1	アクティブ High の非同期リセット
WRCLK	入力	1	書き込みクロック
WREN	入力	1	アクティブ High のライト イネーブル

## デザインの入力方法

インスタンス化	可
推論	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
ALMOST_EMPTY_VALUE	10 進数	1, 2	1	ALMOSTEMPTY 出力信号をアサートする際に FIFO にあるエントリ数を指定します。
ALMOST_FULL_VALUE	10 進数	1, 2	1	ALMOSTFULL 出力信号をアサートする際に FIFO にあるエントリ数を指定します。
ARRAY_MODE	文字列	"ARRAY_MODE_8_X_4"、 "ARRAY_MODE_4_X_4"	"ARRAY_MODE_8_X_4"	シリアルライザー モードを指定します。 <ul style="list-style-type: none"> <li>"ARRAY_MODE_4_X_4": 4 ビット データを入力し、4 ビット データを出力します。</li> <li>"ARRAY_MODE_8_X_8": 4 ビット データを入力し、8 ビット データを出力します。</li> </ul>
OUTPUT_DISABLE	文字列	"FALSE"、"TRUE"	"FALSE"	出力をディスエーブルにします。
SYNCHRONOUS_MODE	文字列	"FALSE"	"FALSE"	"FALSE" に設定する必要があります。

## 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;

-- OUT_FIFO: Output First-In, First-Out (FIFO) Buffer
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

OUT_FIFO_inst : OUT_FIFO
generic map (
    ALMOST_EMPTY_VALUE => 1,          -- Almost empty offset (1-2)
    ALMOST_FULL_VALUE  => 1,          -- Almost full offset (1-2)
    ARRAY_MODE         => "ARRAY_MODE_8_X_4", -- ARRAY_MODE_8_X_4, ARRAY_MODE_4_X_4
    OUTPUT_DISABLE     => "FALSE",    -- Disable output (FALSE, TRUE)
    SYNCHRONOUS_MODE   => "FALSE"    -- Must always be set to false.
)
port map (
    -- FIFO Status Flags: 1-bit (each) output: Flags and other FIFO status outputs
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit output: Almost empty flag
    ALMOSTFULL  => ALMOSTFULL,  -- 1-bit output: Almost full flag
    EMPTY       => EMPTY,      -- 1-bit output: Empty flag
    FULL        => FULL,        -- 1-bit output: Full flag
    -- Q0-Q9: 4-bit (each) output: FIFO Outputs
    Q0 => Q0,                  -- 4-bit output: Channel 0 output bus
```

```

Q1 => Q1,          -- 4-bit output: Channel 1 output bus
Q2 => Q2,          -- 4-bit output: Channel 2 output bus
Q3 => Q3,          -- 4-bit output: Channel 3 output bus
Q4 => Q4,          -- 4-bit output: Channel 4 output bus
Q5 => Q5,          -- 8-bit output: Channel 5 output bus
Q6 => Q6,          -- 8-bit output: Channel 6 output bus
Q7 => Q7,          -- 4-bit output: Channel 7 output bus
Q8 => Q8,          -- 4-bit output: Channel 8 output bus
Q9 => Q9,          -- 4-bit output: Channel 9 output bus
-- D0-D9: 8-bit (each) input: FIFO inputs
D0 => D0,          -- 8-bit input: Channel 0 input bus
D1 => D1,          -- 8-bit input: Channel 1 input bus
D2 => D2,          -- 8-bit input: Channel 2 input bus
D3 => D3,          -- 8-bit input: Channel 3 input bus
D4 => D4,          -- 8-bit input: Channel 4 input bus
D5 => D5,          -- 8-bit input: Channel 5 input bus
D6 => D6,          -- 8-bit input: Channel 6 input bus
D7 => D7,          -- 8-bit input: Channel 7 input bus
D8 => D8,          -- 8-bit input: Channel 8 input bus
D9 => D9,          -- 8-bit input: Channel 9 input bus
-- FIFO Control Signals: 1-bit (each) input: Clocks, Resets and Enables
RDCLK => RDCLK,    -- 1-bit input: Read clock
RDEN => RDEN,      -- 1-bit input: Read enable
RESET => RESET,    -- 1-bit input: Active high reset
WRCLK => WRCLK,    -- 1-bit input: Write clock
WREN => WREN       -- 1-bit input: Write enable
);

-- End of OUT_FIFO_inst instantiation

```



## Verilog 記述 (インスタンスレーション)

```
// OUT_FIFO: Output First-In, First-Out (FIFO) Buffer
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

OUT_FIFO #(
    .ALMOST_EMPTY_VALUE(1),           // Almost empty offset (1-2)
    .ALMOST_FULL_VALUE(1),            // Almost full offset (1-2)
    .ARRAY_MODE("ARRAY_MODE_8_X_4"), // ARRAY_MODE_8_X_4, ARRAY_MODE_4_X_4
    .OUTPUT_DISABLE("FALSE"),         // Disable output (FALSE, TRUE)
    .SYNCHRONOUS_MODE("FALSE")        // Must always be set to false.
)
OUT_FIFO_inst (
    // FIFO Status Flags: 1-bit (each) output: Flags and other FIFO status outputs
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit output: Almost empty flag
    .ALMOSTFULL(ALMOSTFULL),   // 1-bit output: Almost full flag
    .EMPTY(EMPTY),             // 1-bit output: Empty flag
    .FULL(FULL),               // 1-bit output: Full flag
    // Q0-Q9: 4-bit (each) output: FIFO Outputs
    .Q0(Q0),                   // 4-bit output: Channel 0 output bus
    .Q1(Q1),                   // 4-bit output: Channel 1 output bus
    .Q2(Q2),                   // 4-bit output: Channel 2 output bus
    .Q3(Q3),                   // 4-bit output: Channel 3 output bus
    .Q4(Q4),                   // 4-bit output: Channel 4 output bus
    .Q5(Q5),                   // 8-bit output: Channel 5 output bus
    .Q6(Q6),                   // 8-bit output: Channel 6 output bus
    .Q7(Q7),                   // 4-bit output: Channel 7 output bus
    .Q8(Q8),                   // 4-bit output: Channel 8 output bus
    .Q9(Q9),                   // 4-bit output: Channel 9 output bus
    // D0-D9: 8-bit (each) input: FIFO inputs
    .D0(D0),                   // 8-bit input: Channel 0 input bus
    .D1(D1),                   // 8-bit input: Channel 1 input bus
    .D2(D2),                   // 8-bit input: Channel 2 input bus
    .D3(D3),                   // 8-bit input: Channel 3 input bus
    .D4(D4),                   // 8-bit input: Channel 4 input bus
    .D5(D5),                   // 8-bit input: Channel 5 input bus
    .D6(D6),                   // 8-bit input: Channel 6 input bus
    .D7(D7),                   // 8-bit input: Channel 7 input bus
    .D8(D8),                   // 8-bit input: Channel 8 input bus
    .D9(D9),                   // 8-bit input: Channel 9 input bus
    // FIFO Control Signals: 1-bit (each) input: Clocks, Resets and Enables
    .RDCLK(RDCLK),             // 1-bit input: Read clock
    .RDEN(RDEN),               // 1-bit input: Read enable
    .RESET(RESET),             // 1-bit input: Active high reset
    .WRCLK(WRCLK),             // 1-bit input: Write clock
    .WREN(WREN),               // 1-bit input: Write enable
);

// End of OUT_FIFO_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## PHASER\_IN

プリミティブ：Phaser in

### 概要

PHASER\_IN はほかの PHASER エLEMENTと共に使用され、高速メモリ インターフェイスでのデータのアライメントおよびキャプチャを処理します。Memory Interface Generator (MIG) で使用することを目的としており、インスタンス化したり、サイリンクスで生成された IP 外で使用または変更することはできません。

### デザインの入力方法

インスタンス化	不可
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

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## PHASER\_IN\_PHY

プリミティブ：Phaser in phy

### 概要

PHASER\_IN\_PHY はほかの PHASER エLEMENTと共に使用され、高速メモリ インターフェイスでのデータのアライメントおよびキャプチャを処理します。Memory Interface Generator (MIG) で使用することを目的としており、インスタンス化したり、サイリンクスで生成された IP 外で使用または変更することはできません。

### デザインの入力方法

インスタンス化	不可
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

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

# PHASER\_OUT

プリミティブ : Phaser out.

## 概要

PHASER\_OUT はほかの PHASER エLEMENTと共に使用され、高速メモリ インターフェイスでのデータのアライメントおよびキャプチャを処理します。Memory Interface Generator (MIG) で使用することを目的としており、インスタンスエートしたり、サイリンクスで生成された IP 外で使用または変更することはできません。

## デザインの入力方法

インスタンスエーション	不可
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

## 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## PHASER\_OUT\_PHY

プリミティブ：Phaser out phy.

### 概要

PHASER\_OUT\_PHY はほかの PHASER エLEMENTと共に使用され、高速メモリ インターフェイスでのデータのアライメントおよびキャプチャを処理します。Memory Interface Generator (MIG) で使用することを目的としており、インスタンス化したり、ザイリンクスで生成された IP 外で使用または変更することはできません。

### デザインの入力方法

インスタンス化	不可
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

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## PHASER\_REF

プリミティブ：Phaser out.

### 概要

PHASER\_REF はほかの PHASER エLEMENTと共に使用され、高速メモリ インターフェイスでのデータのアライメントおよびキャプチャを処理します。Memory Interface Generator (MIG) で使用することを目的としており、インスタンスエートしたり、サイリンクスで生成された IP 外で使用または変更することはできません。

### デザインの入力方法

インスタンスエーション	不可
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

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## PHY\_CONTROL

プリミティブ：Phaser out.

### 概要

PHASER\_CONTROL はほかの PHASER エLEMENTと共に使用され、高速メモリ インターフェイスでのデータのアライメントおよびキャプチャを処理します。Memory Interface Generator (MIG) で使用することを目的としており、インスタンス化したり、サイリンクスで生成された IP 外で使用または変更することはできません。

### デザインの入力方法

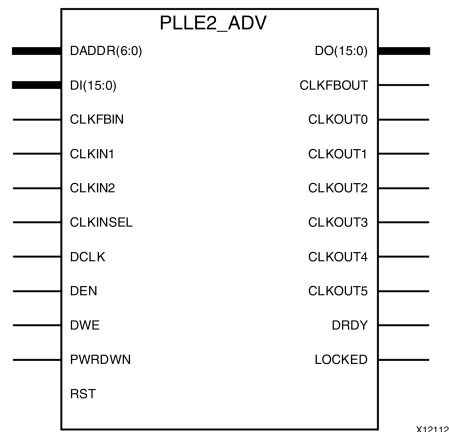
インスタンス化	不可
推論	不可
CORE Generator™ およびウィザード	可
マクロのサポート	不可

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## PLLE2\_ADV

プリミティブ：Advanced Phase Locked Loop (PLL)



### 概要

PLLE2 は、周波数合成、クロック ネットワークのスキュー調整、ジッター低減をサポートするための混合信号ブロックです。各クロック出力に対して、同じ VCO 周波数を基準に分周、位相シフト、デューティ サイクルを個別に設定できます。MMCM が一般的なクロックの機能を提供するのに対し、PLLE2 は高速クロックをサポートします。ほとんどの場合は PLLE2\_BASE を使用し、PLLE2\_ADV はクロックのスイッチオーバーやダイナミック リコンフィギュレーションが必要な場合に使用します。

### ポートの説明

ポート名	方向	幅	説明
CLKFBIN	入力	1	PLL へのフィードバック クロック ピン
CLKFBOUT	出力	1	専用 PLL フィードバック クロック出力
CLKINSEL	入力	1	入力マルチプレクサーのステートを制御する信号で、High の場合は CLKIN1、Low の場合は CLKIN2 です。
CLKIN1	入力	1	プライマリ クロック入力
CLKIN2	入力	1	セカンダリ クロック入力
CLKOUT0	出力	1	CLKOUT0 出力
CLKOUT1	出力	1	CLKOUT1 出力
CLKOUT2	出力	1	CLKOUT2 出力
CLKOUT3	出力	1	CLKOUT3 出力
CLKOUT4	出力	1	CLKOUT4 出力
CLKOUT5	出力	1	CLKOUT5 出力
DADDR<6:0>	入力	7	ダイナミック リコンフィギュレーション用のリコンフィギュレーション アドレスを供給する入力バス。このバスを使用しない場合は、すべてのビットを 0 にする必要があります。
DCLK	入力	1	ダイナミック リコンフィギュレーション ポートの基準クロック
DEN	入力	1	ダイナミック リコンフィギュレーション機能にアクセスするためのイネーブル制御信号。ダイナミック リコンフィギュレーションを使用しない場合は、Low に接続する必要があります。



ポート名	方向	幅	説明
DI<15:0>	入力	16	リコンフィギュレーション データを供給するデータ入力バス。このバスを使用しない場合は、すべてのビットを 0 にする必要があります。
DO<15:0>	出力	16	ダイナミックリコンフィギュレーションを使用する場合の PLL データ出力バス
DRDY	出力	1	PLL のダイナミックリコンフィギュレーション機能の DEN 信号への応答を供給する READY 出力
DWE	入力	1	DADDR アドレスへの DI データの書き込みを制御するライト イネーブル信号。使用しない場合は、Low に接続する必要があります。
PWRDWN	入力	1	インスタンシエートされているが未使用の PLL をパワー ダウンします。
RST	入力	1	PLL の非同期リセットです。この信号が解放されると PLL はクロックに同期して再びイネーブルになり、新たに位相アライメントおよびロック サイクルが実行されます。入力クロックの条件 (周波数など) を変更する場合、リセットが必要です。
LOCKED	出力	1	位相アライメントが定義されている時間内で完了し、周波数が定義されている PPM 範囲内で一致したことを示します。PLL は電源投入時に自動的にロックされるので、リセットは必要ありません。入力クロックが停止した場合、または位相アライメントに違反が発生した場合 (入力クロックの位相シフトなど) は、LOCKED がディASSERTされます。LOCKED がディASSERTされると、自動的にロックが達成されます。

## デザインの入力方法

インスタンシエーション	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
BANDWIDTH	文字列	"OPTIMIZED"、 "HIGH"、"LOW"	"OPTIMIZED"	ジッター、位相マージンなどの PLLE2 特性に影響する PLLE2 プログラム アルゴリズムを指定します。
CLKFBOUT_MULT	10 進数	2 ~ 64	5	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 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKFBOUT_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	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。フィードバッククロックをシフトすると、PLL の出力クロックがすべて負の方向に位相シフトされます。
CLKIN1_PERIOD、 CLKIN2_PERIOD	浮動小数点 (nS)	0.000 ~ 52.631	0.000	CLKIN 入力の周期を指定します。精度は ps です。たとえば、値 33.333 は 30MHz の入力クロックを示します。この値は必ず設定する必要があります。CLKIN1_PERIOD は CLKIN1 入力の入力クロック周期、CLKIN2_PERIOD は CLKIN2 入力の入力クロック周期を指定します。

属性	データ型	値	デフォルト	説明
CLKOUT0_DIVIDE、 CLKOUT1_DIVIDE、 CLKOUT2_DIVIDE、 CLKOUT3_DIVIDE、 CLKOUT4_DIVIDE、 CLKOUT5_DIVIDE	10 進数	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、CLKFBOUT_MULT_F 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKOUT0_DUTY_CYCLE、 CLKOUT1_DUTY_CYCLE、 CLKOUT2_DUTY_CYCLE、 CLKOUT3_DUTY_CYCLE、 CLKOUT4_DUTY_CYCLE、 CLKOUT5_DUTY_CYCLE	3 上位ビット 浮動小数点	0.001 ~ 0.999	0.500	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。0.500 に設定すると、デューティ サイクルは 50% になります。
CLKOUT0_PHASE、 CLKOUT1_PHASE、 CLKOUT2_PHASE、 CLKOUT3_PHASE、 CLKOUT4_PHASE、 CLKOUT5_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	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。フィードバック クロックをシフトすると、PLL の出力クロックがすべて負の方向に位相シフトされます。
COMPENSATION	文字列	"ZHOLD"、"BUF_IN"、 "EXTERNAL"、 "INTERNAL"	"ZHOLD"	<p>クロック入力の補正を指定します。"ZHOLD" に設定することをお勧めします。PLL フィードバックのコンフィギュレーション方法を定義します。</p> <ul style="list-style-type: none"> <li>・ "ZHOLD" : I/O レジスタでのホールドタイムが負になるようコンフィギュレーションします。</li> <li>・ "INTERNAL" : PLL の内部フィードバックパスが使用され、遅延は補正されません。</li> <li>・ "EXTERNAL" : FPGA の外部ネットワークが補正されます。</li> <li>・ "BUF_IN" : いずれの補正モードにも一致していないことを示し、遅延は補正されません。</li> </ul>
DIVCLK_DIVIDE	10 進数	1 ~ 56	1	すべての出力クロックの入力クロックに対する分周比を指定し、PFD に入力される CLKIN を分周します。

属性	データ型	値	デフォルト	説明
REF_JITTER1、 REF_JITTER2	3 上位ビット 浮動小数点	0.000 ～ 0.999	0.010	PLL パフォーマンスを最適化するため、CLKIN 入力に予測されるジッター値を指定します。BANDWIDTH が "OPTIMIZED" に設定されている場合、値が既知でない場合は、入力クロックに最適なパラメーターが選択されます。値が既知である場合は、値を入力クロックに予測されるジッターの UI パーセント (最大ピークトゥピーク値) で指定する必要があります。REF_JITTER1 は CLKIN1 の入力ジッター、REF_JITTER2 は CLKIN2 の入力ジッターを指定します。
STARTUP_WAIT	文字列	"FALSE"、"TRUE"	"FALSE"	"TRUE" の場合、ロックが達成されるまで DONE が High になるのが遅延されます。

## 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;

-- PLLE2_ADV: Advanced Phase Locked Loop (PLL)
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

PLLE2_ADV_inst : PLLE2_ADV
generic map (
    BANDWIDTH => "OPTIMIZED", -- OPTIMIZED, HIGH, LOW
    CLKFBOUT_MULT => 5, -- Multiply value for all CLKOUT, (2-64)
    CLKFBOUT_PHASE => 0.0, -- Phase offset in degrees of CLKFB, (-360.000-360.000).
    -- CLKIN_PERIOD: Input clock period in nS to ps resolution (i.e. 33.333 is 30 MHz).
    CLKIN1_PERIOD => 0.0,
    CLKIN2_PERIOD => 0.0,
    -- CLKOUT0_DIVIDE - CLKOUT5_DIVIDE: Divide amount for CLKOUT (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 outputs (0.001-0.999).
    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: Phase offset for CLKOUT outputs (-360.000-360.000).
    CLKOUT0_PHASE => 0.0,
    CLKOUT1_PHASE => 0.0,
    CLKOUT2_PHASE => 0.0,
    CLKOUT3_PHASE => 0.0,
    CLKOUT4_PHASE => 0.0,
    CLKOUT5_PHASE => 0.0,
    COMPENSATION => "ZHOLD", -- ZHOLD, BUF_IN, EXTERNAL, INTERNAL
    DIVCLK_DIVIDE => 1, -- Master division value (1-56)
    -- REF_JITTER: Reference input jitter in UI (0.000-0.999).
    REF_JITTER1 => 0.0,
    REF_JITTER2 => 0.0,
    STARTUP_WAIT => "FALSE" -- Delay DONE until PLL Locks, ("TRUE"/"FALSE")
)
port map (

```

```

-- Clock Outputs: 1-bit (each) output: User configurable clock outputs
CLKOUT0 => CLKOUT0,    -- 1-bit output: CLKOUT0
CLKOUT1 => CLKOUT1,    -- 1-bit output: CLKOUT1
CLKOUT2 => CLKOUT2,    -- 1-bit output: CLKOUT2
CLKOUT3 => CLKOUT3,    -- 1-bit output: CLKOUT3
CLKOUT4 => CLKOUT4,    -- 1-bit output: CLKOUT4
CLKOUT5 => CLKOUT5,    -- 1-bit output: CLKOUT5
-- DRP Ports: 16-bit (each) output: Dynamic reconfiguration ports
DO => DO,              -- 16-bit output: DRP data
DRDY => DRDY,          -- 1-bit output: DRP ready
-- Feedback Clocks: 1-bit (each) output: Clock feedback ports
CLKFBOUT => CLKFBOUT,  -- 1-bit output: Feedback clock
-- Status Ports: 1-bit (each) output: PLL status ports
LOCKED => LOCKED,     -- 1-bit output: LOCK
-- Clock Inputs: 1-bit (each) input: Clock inputs
CLKIN1 => CLKIN1,     -- 1-bit input: Primary clock
CLKIN2 => CLKIN2,     -- 1-bit input: Secondary clock
-- Control Ports: 1-bit (each) input: PLL control ports
CLKINSEL => CLKINSEL, -- 1-bit input: Clock select, High=CLKIN1 Low=CLKIN2
PWRDWN => PWRDWN,     -- 1-bit input: Power-down
RST => RST,           -- 1-bit input: Reset
-- DRP Ports: 7-bit (each) input: Dynamic reconfiguration ports
DADDR => DADDR,       -- 7-bit input: DRP address
DCLK => DCLK,         -- 1-bit input: DRP clock
DEN => DEN,          -- 1-bit input: DRP enable
DI => DI,             -- 16-bit input: DRP data
DWE => DWE,          -- 1-bit input: DRP write enable
-- Feedback Clocks: 1-bit (each) input: Clock feedback ports
CLKFBIN => CLKFBIN    -- 1-bit input: Feedback clock
);

-- End of PLLE2_ADV_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// PLLE2_ADV: Advanced Phase Locked Loop (PLL)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

PLLE2_ADV #(
    .BANDWIDTH("OPTIMIZED"), // OPTIMIZED, HIGH, LOW
    .CLKFBOUT_MULT(5),       // Multiply value for all CLKOUT, (2-64)
    .CLKFBOUT_PHASE(0.0),    // Phase offset in degrees of CLKFB, (-360.000-360.000).
    // CLKIN_PERIOD: Input clock period in nS to ps resolution (i.e. 33.333 is 30 MHz).
    .CLKIN1_PERIOD(0.0),
    .CLKIN2_PERIOD(0.0),
    // CLKOUT0_DIVIDE - CLKOUT5_DIVIDE: Divide amount for CLKOUT (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 outputs (0.001-0.999).
    .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: Phase offset for CLKOUT outputs (-360.000-360.000).
    .CLKOUT0_PHASE(0.0),
    .CLKOUT1_PHASE(0.0),
    .CLKOUT2_PHASE(0.0),
    .CLKOUT3_PHASE(0.0),
    .CLKOUT4_PHASE(0.0),
    .CLKOUT5_PHASE(0.0),
    .COMPENSATION("ZHOLD"), // ZHOLD, BUF_IN, EXTERNAL, INTERNAL
    .DIVCLK_DIVIDE(1),      // Master division value (1-56)
    // REF_JITTER: Reference input jitter in UI (0.000-0.999).
    .REF_JITTER1(0.0),

```

```

    .REF_JITTER2(0.0),
    .STARTUP_WAIT("FALSE")    // Delay DONE until PLL Locks, ("TRUE"/"FALSE")
)
PLLE2_ADV_inst (
    // Clock Outputs: 1-bit (each) output: User configurable clock outputs
    .CLKOUT0(CLKOUT0),        // 1-bit output: CLKOUT0
    .CLKOUT1(CLKOUT1),        // 1-bit output: CLKOUT1
    .CLKOUT2(CLKOUT2),        // 1-bit output: CLKOUT2
    .CLKOUT3(CLKOUT3),        // 1-bit output: CLKOUT3
    .CLKOUT4(CLKOUT4),        // 1-bit output: CLKOUT4
    .CLKOUT5(CLKOUT5),        // 1-bit output: CLKOUT5
    // DRP Ports: 16-bit (each) output: Dynamic reconfiguration ports
    .DO(DO),                  // 16-bit output: DRP data
    .DRDY(DRDY),              // 1-bit output: DRP ready
    // Feedback Clocks: 1-bit (each) output: Clock feedback ports
    .CLKFBOUT(CLKFBOUT),      // 1-bit output: Feedback clock
    // Status Ports: 1-bit (each) output: PLL status ports
    .LOCKED(LOCKED),          // 1-bit output: LOCK
    // Clock Inputs: 1-bit (each) input: Clock inputs
    .CLKIN1(CLKIN1),          // 1-bit input: Primary clock
    .CLKIN2(CLKIN2),          // 1-bit input: Secondary clock
    // Control Ports: 1-bit (each) input: PLL control ports
    .CLKINSEL(CLKINSEL),      // 1-bit input: Clock select, High=CLKIN1 Low=CLKIN2
    .PWRDWN(PWRDWN),          // 1-bit input: Power-down
    .RST(RST),                // 1-bit input: Reset
    // DRP Ports: 7-bit (each) input: Dynamic reconfiguration ports
    .DADDR(DADDR),           // 7-bit input: DRP address
    .DCLK(DCLK),              // 1-bit input: DRP clock
    .DEN(DEN),                // 1-bit input: DRP enable
    .DI(DI),                  // 16-bit input: DRP data
    .DWE(DWE),                // 1-bit input: DRP write enable
    // Feedback Clocks: 1-bit (each) input: Clock feedback ports
    .CLKFBIN(CLKFBIN)        // 1-bit input: Feedback clock
);

// End of PLLE2_ADV_inst instantiation

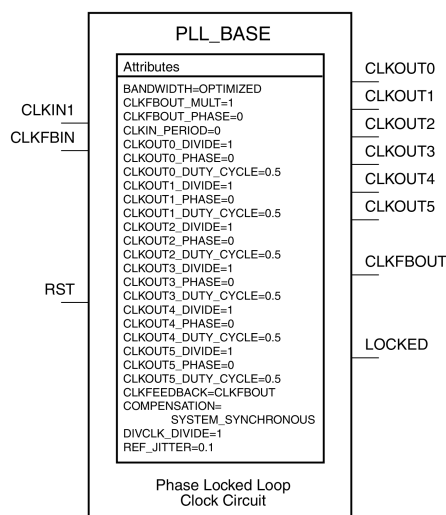
```

## 詳細情報

### [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## PLLE2\_BASE

プリミティブ：Base Phase Locked Loop (PLL)



X10951

### 概要

PLLE2 は、周波数合成、クロック ネットワークのスキュー調整、ジッター低減をサポートするための混合信号ブロックです。各クロック出力に対して、同じ VCO 周波数を基準に分周、位相シフト、デューティ サイクルを個別に設定できます。MMCM が一般的なクロックの機能を提供するのに対し、PLLE2 は高速クロックをサポートします。ほとんどの場合は PLLE2\_BASE を使用し、PLLE2\_ADV はクロックのスイッチオーバーやダイナミック リコンフィギュレーションが必要な場合に使用します。

### ポートの説明

ポート名	方向	幅	説明
CLKFBIN	入力	1	PLL へのフィードバック クロック ピン
CLKFBOUT	出力	1	専用 PLL フィードバック クロック出力
CLKIN1	入力	1	汎用クロック入力
CLKOUT0 ~ CLKOUT5	出力	1	コンフィギュレーション可能なクロック出力で、VCO 位相出力 (ユーザー制御可能) を 1 (バイパス) から 128 までの値で分周したものに設定できます。出力クロックは、位相シフトされていない場合はお互いに位相が揃っており、適切なフィードバック コンフィギュレーションにより入力クロックに揃えられます。
LOCKED	出力	1	位相アライメントが定義されている時間内で完了し、周波数が定義されている PPM 範囲内で一致したことを示します。PLL は電源投入時に自動的にロックされるので、リセットは必要ありません。入力クロックが停止した場合、または位相アライメントに違反が発生した場合 (入力クロックの位相シフトなど) は、LOCKED がディアサートされます。LOCKED がディアサートされると、自動的にロックが達成されます。
PWRDWN	入力	1	インスタンス化されているが未使用の PLL をパワー ダウンします。

ポート名	方向	幅	説明
RST	入力	1	PLL の非同期リセットです。この信号が解放されると PLL はクロックに同期して再びイネーブルになり、新たに位相アライメントおよびロック サイクルが実行されます。入力クロックの条件 (周波数など) を変更する場合、リセットが必要です。

## 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
BANDWIDTH	文字列	"OPTIMIZED"、 "HIGH"、"LOW"	"OPTIMIZED"	ジッター、位相マージンなどの PLLE2 特性に影響する PLLE2 プログラム アルゴリズムを指定します。
CLKFBOUT_MULT	10 進数	2 ~ 64	5	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 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKFBOUT_PHASE	3 上位 ビット浮 動小数点	-360.000 ~ 360.000	0.000	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。フィードバッククロックをシフトすると、PLL の出力クロックがすべて負の方向に位相シフトされます。
CLKIN1_PERIOD	浮動小数 点 (ns)	0.000 ~ 52.631	0.000	CLKIN1 の入力周期を ns で指定します。精度は ps です (小数点以下 3 桁)。たとえば、値 33.333 は 30 MHz の入力クロックを示します。この値は必ず設定する必要があります。
CLKOUT0_DIVIDE、 CLKOUT1_DIVIDE、 CLKOUT2_DIVIDE、 CLKOUT3_DIVIDE、 CLKOUT4_DIVIDE、 CLKOUT5_DIVIDE	10 進数	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、CLKFBOUT_MULT_F 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKOUT0_DUTY_CYCLE、 CLKOUT1_DUTY_CYCLE、 CLKOUT2_DUTY_CYCLE、 CLKOUT3_DUTY_CYCLE、 CLKOUT4_DUTY_CYCLE、 CLKOUT5_DUTY_CYCLE	3 上位 ビット浮 動小数点	0.001 ~ 0.999	0.500	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。0.500 に設定すると、デューティ サイクルは 50% になります。
CLKOUT0_PHASE、 CLKOUT1_PHASE、 CLKOUT2_PHASE、 CLKOUT3_PHASE、 CLKOUT4_PHASE、 CLKOUT5_PHASE	3 上位 ビット浮 動小数点	-360.000 ~ 360.000	0.000	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。フィードバッククロックをシフトすると、PLL の出力クロックがすべて負の方向に位相シフトされます。

属性	データ型	値	デフォルト	説明
DIVCLK_DIVIDE	10 進数	1 ~ 56	1	すべての出力クロックの入力クロックに対する分周比を指定し、PFD に入力される CLKIN を分周します。
REF_JITTER1	3 上位 ビット浮 動小数点	0.000 ~ 0.999	0.010	PLL のパフォーマンスを最適化するため、CLKIN1 に予測されるジッター値を指定します。BANDWIDTH が "OPTIMIZED" に設定されている場合、値が既知でない場合は入力クロックに最適なパラメーターが選択されます。値が既知である場合は、値を入力クロックに予測されるジッターの UI パーセント (最大ピークトゥピーク値) で指定する必要があります。
STARTUP_WAIT	文字列	"FALSE"、"TRUE"	"FALSE"	"TRUE" の場合、ロックが達成されるまで DONE が High になるのが遅延されます。

## 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;

-- PLLE2_BASE: Base Phase Locked Loop (PLL)
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

PLLE2_BASE_inst : PLLE2_BASE
generic map (
    BANDWIDTH => "OPTIMIZED", -- OPTIMIZED, HIGH, LOW
    CLKFBOUT_MULT => 5,        -- Multiply value for all CLKOUT, (2-64)
    CLKFBOUT_PHASE => 0.0,     -- Phase offset in degrees of CLKFB, (-360.000-360.000).
    CLKIN1_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 each CLKOUT (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 each CLKOUT (0.001-0.999).
    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: Phase offset for each CLKOUT (-360.000-360.000).
    CLKOUT0_PHASE => 0.0,
    CLKOUT1_PHASE => 0.0,
    CLKOUT2_PHASE => 0.0,
    CLKOUT3_PHASE => 0.0,
    CLKOUT4_PHASE => 0.0,
    CLKOUT5_PHASE => 0.0,
    DIVCLK_DIVIDE => 1,        -- Master division value, (1-56)
    REF_JITTER1 => 0.0,       -- Reference input jitter in UI, (0.000-0.999).
    STARTUP_WAIT => "FALSE"   -- Delay DONE until PLL Locks, ("TRUE"/"FALSE")
)
port map (
    -- Clock Outputs: 1-bit (each) output: User configurable clock outputs
    CLKOUT0 => CLKOUT0,
    CLKOUT1 => CLKOUT1,

```



```
CLKOUT2 => CLKOUT2,  
CLKOUT3 => CLKOUT3,  
CLKOUT4 => CLKOUT4,  
CLKOUT5 => CLKOUT5,  
-- Feedback Clocks: 1-bit (each) output: Clock feedback ports  
CLKFBOUT => CLKFBOUT, -- 1-bit output: Feedback clock  
-- Status Port: 1-bit (each) output: PLL status ports  
LOCKED => LOCKED, -- 1-bit output: LOCK  
-- Clock Input: 1-bit (each) input: Clock input  
CLKIN1 => CLKIN1, -- 1-bit input: Input clock  
-- Control Ports: 1-bit (each) input: PLL control ports  
PWRDWN => PWRDWN, -- 1-bit input: Power-down  
RST => RST, -- 1-bit input: Reset  
-- Feedback Clocks: 1-bit (each) input: Clock feedback ports  
CLKFBIN => CLKFBIN -- 1-bit input: Feedback clock  
);  
  
-- End of PLLE2_BASE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// PLLE2_BASE: Base Phase Locked Loop (PLL)
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

PLLE2_BASE #(
    .BANDWIDTH("OPTIMIZED"), // OPTIMIZED, HIGH, LOW
    .CLKFBOUT_MULT(5),       // Multiply value for all CLKOUT, (2-64)
    .CLKFBOUT_PHASE(0.0),    // Phase offset in degrees of CLKFB, (-360.000-360.000).
    .CLKIN1_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 each CLKOUT (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 each CLKOUT (0.001-0.999).
    .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: Phase offset for each CLKOUT (-360.000-360.000).
    .CLKOUT0_PHASE(0.0),
    .CLKOUT1_PHASE(0.0),
    .CLKOUT2_PHASE(0.0),
    .CLKOUT3_PHASE(0.0),
    .CLKOUT4_PHASE(0.0),
    .CLKOUT5_PHASE(0.0),
    .DIVCLK_DIVIDE(1),       // Master division value, (1-56)
    .REF_JITTER1(0.0),       // Reference input jitter in UI, (0.000-0.999).
    .STARTUP_WAIT("FALSE")  // Delay DONE until PLL Locks, ("TRUE"/"FALSE")
)
PLLE2_BASE_inst (
    // Clock Outputs: 1-bit (each) output: User configurable clock outputs
    .CLKOUT0(CLKOUT0),
    .CLKOUT1(CLKOUT1),
    .CLKOUT2(CLKOUT2),
    .CLKOUT3(CLKOUT3),
    .CLKOUT4(CLKOUT4),
    .CLKOUT5(CLKOUT5),
    // Feedback Clocks: 1-bit (each) output: Clock feedback ports
    .CLKFBOUT(CLKFBOUT), // 1-bit output: Feedback clock
    // Status Port: 1-bit (each) output: PLL status ports
    .LOCKED(LOCKED),     // 1-bit output: LOCK
    // Clock Input: 1-bit (each) input: Clock input
    .CLKIN1(CLKIN1),     // 1-bit input: Input clock
    // Control Ports: 1-bit (each) input: PLL control ports
    .PWRDWN(PWRDWN),     // 1-bit input: Power-down
    .RST(RST),           // 1-bit input: Reset
    // Feedback Clocks: 1-bit (each) input: Clock feedback ports
    .CLKFBIN(CLKFBIN)    // 1-bit input: Feedback clock
);

// End of PLLE2_BASE_inst instantiation
```

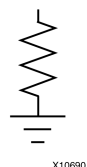
詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
--           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

PULLDOWN_inst : PULLDOWN
port map (
  O => O      -- Pulldown output (connect directly to top-level port)
);

-- End of PULLDOWN_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// PULLDOWN: I/O Buffer Weak Pull-down
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

PULLDOWN PULLDOWN_inst (
    .O(0)      // Pulldown output (connect directly to top-level port)
);

// End of PULLDOWN_inst instantiation
```

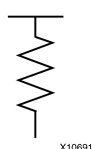
詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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ていないときに、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
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

PULLUP_inst : PULLUP
port map (
  O => O      -- Pullup output (connect directly to top-level port)
);

-- End of PULLUP_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// PULLUP: I/O Buffer Weak Pull-up
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

PULLUP PULLUP_inst (
    .O(0)      // Pullup output (connect directly to top-level port)
);

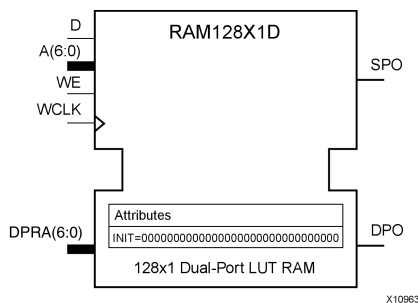
// End of PULLUP_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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。読み出しポートでは、アドレス バス DPRA の値を変更することにより、非同期読み出しを実行でき、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 で指定された読み出し/書き込みポートのデータ出力
DPO	出力	1	アドレス バス DPRA で指定された読み出しポートのデータ出力
D	入力	1	アドレス バス A で指定された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	データ型	値	デフォルト	説明
INIT	16 進数	128 ビット値	すべてゼロ	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 (Mapped to two SliceM LUT6s)
--             7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to two SliceM LUT6s)
//             dual-port distributed LUT RAM
//             7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
```

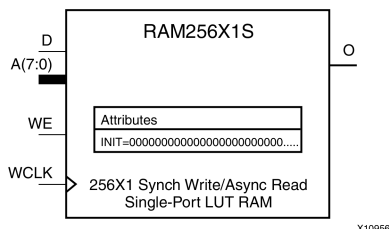


詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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	すべてゼロ	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 (Mapped to four SliceM LUT6s)
--             7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to four SliceM LUT6s)
//             single-port distributed LUT RAM
//             7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

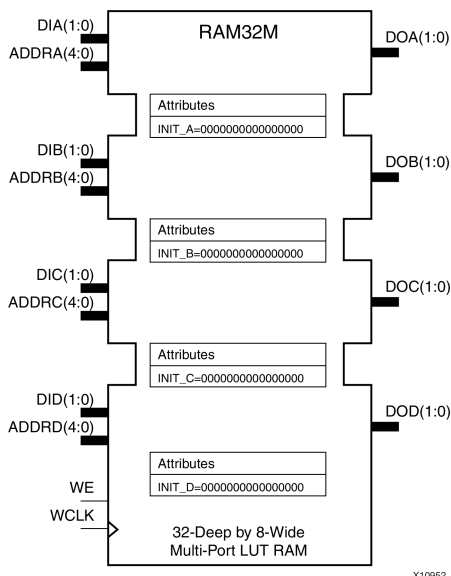
// End of RAM256X1S_inst instantiation
```

### 詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 をグラウンドに接続した場合、DOD は使用されません。
- ・ ADDRA、ADDRB、ADDR C 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。
- ・ ADDR D を ADDRA、ADDRB、ADDR C 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	説明
DOA	出力	2	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	2	アドレス バス ADDR B で指定される読み出しポートのデータ出力
DOC	出力	2	アドレス バス ADDR C で指定される読み出しポートのデータ出力
DOD	出力	2	アドレス バス ADDR D 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	2	ADDR D 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)

ポート名	方向	幅	説明
DIB	入力	2	ADDRD で指定される書き込みデータ入力 (読み出し出力は ADDRDB で指定)
DIC	入力	2	ADDRD で指定される書き込みデータ入力 (読み出し出力は ADDRDC で指定)
DID	入力	2	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	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。RAM ファンクションを暗示的に指定する必要がある場合、または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は、このコンポーネントをインスタンス化してください。同期読み出しが必要な場合は、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を次のように接続します。

- ・ WCLK 入力をクロック ソースに接続
- ・ DIA、DIB、DIC、DID 入力を格納するデータ ソースに接続
- ・ 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
- ・ クロック イネーブル ピン (WE) を適切なライト イネーブル ソースに接続
- ・ ADDRD バスを読み出し/書き込みアドレスのソースに接続
- ・ ADDRA、ADDRB、ADDRC バスを読み出しアドレスに接続

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、 $\text{ADDRy}[z] = \text{INIT}_y[2*z+1:2*z]$  で計算されます。たとえば、RAM の ADDRDC 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	すべてゼロ	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	すべてゼロ	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	すべてゼロ	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	すべてゼロ	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 (Mapped to four SliceM LUT6s)
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to four SliceM LUT6s)
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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), // Read/Write 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
);

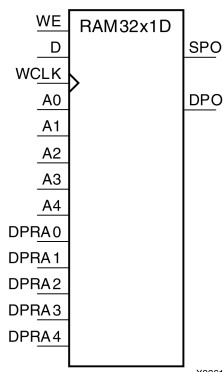
// End of RAM32M_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 エレメントは 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) 用と書き込みアドレス (A4 ~ A0) 用に独立したアドレス ポートがあります。この 2 つのアドレス 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	データ型	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべてゼロ	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 (Mapped to SliceM LUT6)
--           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a SliceM LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4
```

```
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
);

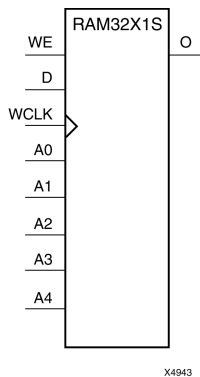
// End of RAM32X1D_inst instantiation
```

詳細情報

[7 シリーズ 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	データ型	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべてゼロ	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;

-- RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM (Mapped to SliceM LUT6)
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスレーション)

```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM (Mapped to a SliceM LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

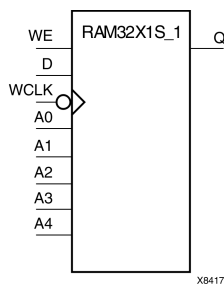
// End of RAM32X1S_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 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	データ型	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;

-- RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM (Mapped to SliceM LUT6)
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスレーション)

```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM (Mapped to a SliceM LUT6)
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

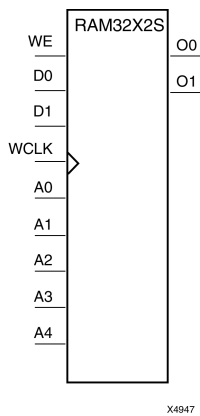
// End of RAM32X1S_1_inst instantiation
```

詳細情報

[7 シリーズ 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 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 (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	データ型	値	デフォルト	説明
INIT_00	16 進数	32 ビット値	すべてゼロ	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	すべてゼロ	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;

-- RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM (Mapped to SliceM LUT6)
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスレーション)

```
// RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM (Mapped to a SliceM LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
```

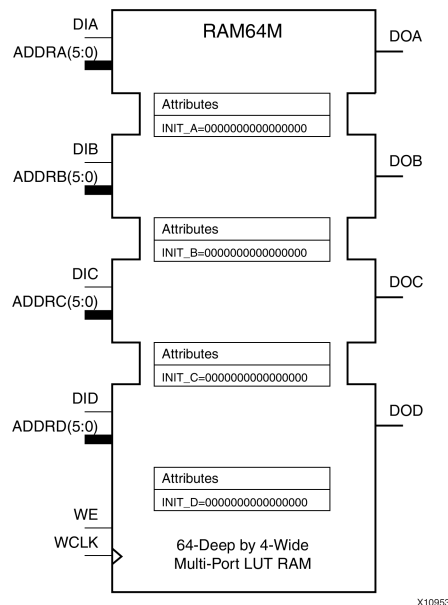


詳細情報

[7 シリーズ 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™ とも呼ばれる) 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、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	説明
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)

ポート名	方向	幅	説明
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 ADDRDB で指定)
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。RAM ファンクションを暗示的に指定する必要がある場合、または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は、このコンポーネントをインスタンス化してください。同期読み出しが必要な場合は、出力を FDRE (非同期リセットが必要な場合は FDCE) 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を次のように接続します。

- ・ WCLK 入力をクロック ソースに接続
- ・ DIA、DIB、DIC、DID 入力を格納するデータ ソースに接続
- ・ 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
- ・ クロック イネーブル ピン (WE) を適切なライト イネーブル ソースに接続
- ・ ADDRD バスを読み出し/書き込みアドレスのソースに接続
- ・ ADDRA、ADDRB、ADDRC バスを読み出しアドレスに接続

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、 $ADDRy[z] = INIT_y[z]$  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	すべてゼロ	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	すべてゼロ	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	すべてゼロ	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	すべてゼロ	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 (Mapped to four SliceM LUT6s)
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to four SliceM LUT6s)
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

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 ADDRDRD,
                //   read addressed by ADDRA
.DIB(DIB),      // RAM 1-bit data write input addressed by ADDRDRD,
                //   read addressed by ADDRDB
.DIC(DIC),      // RAM 1-bit data write input addressed by ADDRDRD,
                //   read addressed by ADDRDC
.DID(DID),      // RAM 1-bit data write input addressed by ADDRDRD,
                //   read addressed by ADDRDRD
.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(ADDRDRD), // Readw/rite port D 6-bit address input
.WE(WE),        // Write enable input
.WCLK(WCLK)     // Write clock input
);

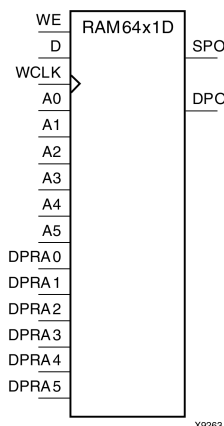
// End of RAM64M_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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) 用と書き込みアドレス (A5 ~ A0) 用に独立したアドレス ポートがあります。この 2 つのアドレス 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	すべてゼロ	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;

-- RAM64X1D: 64 x 1 negative edge write, asynchronous read
--           dual-port distributed RAM (Mapped to SliceM LUT6)
--           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a SliceM LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
```

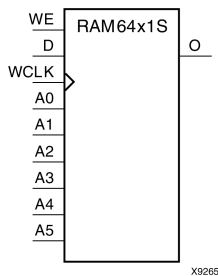
詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## RAM64X1S

プリミティブ：64-Deep by 1-Wide Static Synchronous RAM



### 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 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ELEMENT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	すべてゼロ	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;

-- RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM (Mapped to SliceM LUT6)
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (インスタンスエーション)

```
// RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM (Mapped to a SliceM LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

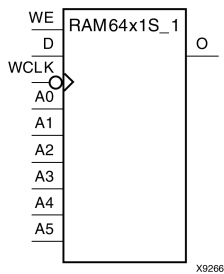
// End of RAM64X1S_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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	すべてゼロ	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 (Mapped to SliceM LUT6)
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a SliceM LUT6)
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

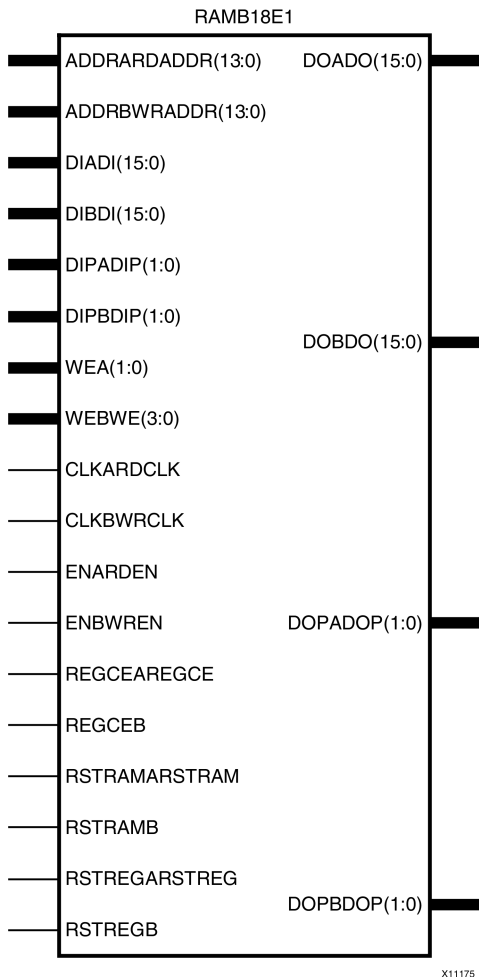
// End of RAM64X1S_1_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## RAMB18E1

### プリミティブ：18K-bit Configurable Synchronous Block RAM



## 概要

7 シリーズ デバイスにはブロック RAM が数個含まれ、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36 Kb または 18 Kb) 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、18Kb の FIFO のブロック RAM へアクセスできます。このエレメントは、1 ビット X 16K ワード～ 18 ビット X 1024 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 512 ワードのシンプル 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	説明
ADDRARDADDR <13:0>	入力	14	ポート A アドレス入力バス/読み出しアドレス入力バス
ADDRBRWADDR <13:0>	入力	14	ポート B アドレス入力バス/書き込みアドレス入力バス

ポート名	方向	幅	説明
CLKARDCLK	入力	1	ポート A クロック入力/読み出しクロック入力 (立ち上がりエッジ)
CLKBWRCLK	入力	1	ポート B クロック入力/書き込みクロック入力 (立ち上がりエッジ)
DIADK<15:0>	入力	16	ポート A データ入力バス/WRADDR でアドレス指定されるデータ入力バス。RAM_MODE が "SDP" の場合は DI<15:0> となります。
DIBDK<15:0>	入力	16	ポート B データ入力バス/WRADDR でアドレス指定されるデータ入力バス。RAM_MODE が "SDP" の場合は DI<31:16> となります。
DIPADIP<1:0>	入力	2	ポート A パリティ データ入力バス/WRADDR でアドレス指定されるデータ パリティ入力バス。RAM_MODE が "SDP" の場合は DIP<1:0> となります。
DIPBDIP<1:0>	入力	2	ポート B パリティ データ入力バス/WRADDR でアドレス指定されるデータ パリティ入力バス。RAM_MODE が "SDP" の場合は DIP<3:2> となります。
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 が "SDP" の場合は DO<15:0> となります。
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 が "SDP" の場合は DO<31:16> となります。
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 が "SDP" の場合は DOP<1:0> となります。
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 が "SDP" の場合は DOP<3:2> となります。
ENARDEN	入力	1	ポート A RAM イネーブル/リード イネーブル
ENBWREN	入力	1	ポート B RAM イネーブル/ライト イネーブル
REGCEAREGCE	入力	1	ポート A 出力レジスタ クロック イネーブル入力/出力レジスタ クロック イネーブル入力 (DOA_REG=1 の場合のみ有効)。
REGCEB	入力	1	ポート B 出力レジスタ クロック イネーブル (DOB_REG=1 および RAM_MODE="TDP" の場合のみ有効)
RSTRAMARSTRAM	入力	1	同期データ ラッチを SRVAL_A で指定された値にセット/リセットします。DO_REG=0 または 1 のときに BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合、RSTRAMARSTRAM でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。RAM_MODE が "TDP" の場合はポート A の RAM 出力がリセットされ、RAM_MODE が "SDP" の場合は RAM 出力全体がリセットされます。
RSTRAMB	入力	1	同期データ ラッチを SRVAL_B で指定された値にセット/リセットします。DO_REG=0 または 1 のときに BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合、RSTRAMB でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。RAM_MODE が "SDP" の場合は使用されません。
RSTREGARSTREG	入力	1	同期出力レジスタを SRVAL_A で指定された値にセット/リセットします。DO_REG=1 のときに出力レジスタをセット/リセットします。RSTREG_PRIORITY_A は、この信号と REGCEAREGCE の優先順位を指定します。RAM_MODE が "TDP" の場合はポート A の出力がリセットされ、RAM_MODE が "SDP" の場合は出力全体がリセットされます。
RSTREGB	入力	1	同期出力レジスタを SRVAL_B で指定された値にセット/リセットします。DO_REG=1 のときに出力レジスタをセット/リセットします。RSTREG_PRIORITY_B は、この信号と REGCEB の優先順位を指定します。RAM_MODE が "SDP" の場合は使用されません。

ポート名	方向	幅	説明
WEA<1:0>	入力	2	ポート A のバイト幅ライト イネーブル。RAM_MODE が "SDP" の場合は使用されません。異なるポート幅の WEA マップについてはユーザー ガイドを参照してください。
WEBWE<3:0>	入力	4	ポート B のバイト幅ライト イネーブル/ライト イネーブル。異なるポート幅の WEBWE マップについてはユーザー ガイド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	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
RDADDR_COLLISION_HWCONFIG	文字列	"DELAYED_WRITE"、 "PERFORMANCE"	"DELAYED_WRITE"	"PERFORMANCE" に設定すると、READ_FIRST モードでのクロック パフォーマンス (周波数) が向上します。RAM の両方のポートで同じクロックを使用している場合に "PERFORMANCE" に設定すると、アドレスが重なった場合の競合規則が適用されます。"DELAYED_WRITE" モードでは、競合を発生させずに RAM を使用できます。
SIM_COLLISION_CHECK	文字列	"ALL"、 "GENERATE_X_ONLY"、 "NONE"、 "WARNING_ONLY"	"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> "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。

属性	データ型	値	デフォルト	説明
DOA_REG、 DOB_REG	10 進数	0、1	0	1 に設定すると、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。0 に設定すると、1 クロック サイクルで読み出すことができますが、clock-to-out タイムが長くなります。DOA_REG は 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、DOB_REG は 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_A、INIT_B	16 進数	18 ビット値	18'h0000	コンフィギュレーション後のポート出力の初期値を指定します。INIT_A は 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、INIT_B は TDP ではポート B に、SDP 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_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16Kb のデータ メモリ アレイの初期値を指定します。
INIT_FILE	文字列	ファイル名	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。
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2Kb 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"	シングル デュアル ポート ("SDP") または完全なデュアル ポート ("TDP") を選択します。
READ_WIDTH_A	10 進数	0、1、2、4、9、18、36、72	0	ポート A の読み出しのデータ幅を指定します (パリティビットを含む)。ポート A を使用しない場合は、ポート幅を 0 にする必要があります。ポートを使用する場合は、適切なポート幅に設定してください。SDP モードの場合は、パリティビットを含む読み出し幅です。
READ_WIDTH_B	10 進数	0、1、2、4、9、18	0	ポート B の読み出しのデータ幅を指定します (パリティビットを含む)。ポート B を使用しない場合は、ポート幅を 0 にする必要があります。ポートを使用する場合は、適切なポート幅に設定してください。SDP モードでは使用されません。
RSTREG_PRIORITY_A、 RSTREG_PRIORITY_B	文字列	"RSTREG"、"REGCE"	"RSTREG"	RSTREG または REGCE のレジスタ優先順位を選択します。RSTREG_PRIORITY_A は 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、RSTREG_PRIORITY_B は 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。
SIM_DEVICE	文字列	"7SERIES"	"7SERIES"	どの条件でもシミュレーションが正しく動作するよう "7SERIES" に設定する必要があります。



属性	データ型	値	デフォルト	説明
SRVAL_A、 SRVAL_B	16 進数	18 ビット値	18'h0000	同期リセット信号 (RSTREG) がアサートされたときの RAM の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	"WRITE_FIRST"、 "NO_CHANGE"、 "READ_FIRST"	"WRITE_FIRST"	書き込みが実行されるとき出力の動作を指定します。  <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> RAM_MODE="SDP" の場合は "NO_CHANGE" には設定できません。シンプル デュアル ポート インプリメンテーションでは、両方のポートで同じクロックを使用する場合はこの属性を "READ_FIRST" に、異なるクロックを使用する場合は "WRITE_FIRST" に設定してください。これにより、このコンフィギュレーションで BRAM を使用した場合の競合またはアドレス オーバーラップ時の動作が向上します。
WRITE_WIDTH_A	10 進数	0、1、2、4、9、18	0	ポート A への書き込みのデータ幅を指定します (パリティ ビットを含む)。ポートを使用しない場合は、0 に設定する必要があります。それ以外の場合は、適切なデータ幅に設定してください。SDP モードでは使用されません。
WRITE_WIDTH_B	10 進数	0、1、2、4、9、18、36、72	0	ポート B への書き込みのデータ幅を指定します (パリティ ビットを含む)。ポートを使用しない場合は、0 に設定する必要があります。それ以外の場合は、適切なデータ幅に設定してください。SDP の場合は、パリティ ビットを含む書き込み幅です。

## 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;

-- RAMB18E1: 18K-bit Configurable Synchronous Block RAM
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

RAMB18E1_inst : RAMB18E1
generic map (
  -- Address Collision Mode: "PERFORMANCE" or "DELAYED_WRITE"
  RDADDR_COLLISION_HWCONFIG => "DELAYED_WRITE",
  -- Collision check: Values ("ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE")
  SIM_COLLISION_CHECK => "ALL",
  -- DOA_REG, DOB_REG: Optional output register (0 or 1)
```

UG768 (V 14.4) 2012 年 12 月 18 日

```

INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- INIT_A, INIT_B: Initial values on output ports
INIT_A => X"00000",
INIT_B => X"00000",
-- Initialization File: RAM initialization file
INIT_FILE => "NONE",
-- RAM Mode: "SDP" or "TDP"
RAM_MODE => "TDP",
-- READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per port
READ_WIDTH_A => 0, -- 0-72
READ_WIDTH_B => 0, -- 0-18
WRITE_WIDTH_A => 0, -- 0-18
WRITE_WIDTH_B => 0, -- 0-72
-- RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG" or "REGCE")
RSTREG_PRIORITY_A => "RSTREG",
RSTREG_PRIORITY_B => "RSTREG",
-- SRVAL_A, SRVAL_B: Set/reset value for output
SRVAL_A => X"00000",
SRVAL_B => X"00000",
-- Simulation Device: Must be set to "7SERIES" for simulation behavior
SIM_DEVICE => "7SERIES",
-- WriteMode: Value on output upon a write ("WRITE_FIRST", "READ_FIRST", or "NO_CHANGE")
WRITE_MODE_A => "WRITE_FIRST",
WRITE_MODE_B => "WRITE_FIRST"
)
port map (
-- Port A Data: 16-bit (each) output: Port A data
DOADO => DOADO, -- 16-bit output: A port data/LSB data
DOPADOP => DOPADOP, -- 2-bit output: A port parity/LSB parity
-- Port B Data: 16-bit (each) output: Port B data
DOBDO => DOBDO, -- 16-bit output: B port data/MSB data
DOPBDOP => DOPBDOP, -- 2-bit output: B port parity/MSB parity
-- Port A Address/Control Signals: 14-bit (each) input: Port A address and control signals (read port
-- when RAM_MODE="SDP")
ADDRARDADDR => ADDRARDADDR, -- 14-bit input: A port address/Read address
CLKARDCLK => CLKARDCLK, -- 1-bit input: A port clock/Read clock
ENARDEN => ENARDEN, -- 1-bit input: A port enable/Read enable
REGCEAREGCE => REGCEAREGCE, -- 1-bit input: A port register enable/Register enable
RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit input: A port set/reset
RSTREGARSTREG => RSTREGARSTREG, -- 1-bit input: A port register set/reset
WEA => WEA, -- 2-bit input: A port write enable
-- Port A Data: 16-bit (each) input: Port A data
DIADI => DIADI, -- 16-bit input: A port data/LSB data
DIPADIP => DIPADIP, -- 2-bit input: A port parity/LSB parity
-- Port B Address/Control Signals: 14-bit (each) input: Port B address and control signals (write port
-- when RAM_MODE="SDP")
ADDRBWRADDR => ADDRBWRADDR, -- 14-bit input: B port address/Write address
CLKBWRCLK => CLKBWRCLK, -- 1-bit input: B port clock/Write clock
ENBWREN => ENBWREN, -- 1-bit input: B port enable/Write enable
REGCEB => REGCEB, -- 1-bit input: B port register enable
RSTRAMB => RSTRAMB, -- 1-bit input: B port set/reset
RSTREGB => RSTREGB, -- 1-bit input: B port register set/reset
WEBWE => WEBWE, -- 4-bit input: B port write enable/Write enable
-- Port B Data: 16-bit (each) input: Port B data
DIBDI => DIBDI, -- 16-bit input: B port data/MSB data
DIPBDIP => DIPBDIP, -- 2-bit input: B port parity/MSB parity
);
-- End of RAMB18E1_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// RAMB18E1: 18K-bit Configurable Synchronous Block RAM
// 7 Series
// Xilinx HDL Libraries Guide, version 14.4

RAMB18E1 #(
// Address Collision Mode: "PERFORMANCE" or "DELAYED_WRITE"
.RDADDR_COLLISION_HWCONFIG("DELAYED_WRITE"),

```

7 シリーズ FPGA および Zynq ライブラリ ガイド (HDL 用)

```

.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// INIT_A, INIT_B: Initial values on output ports
.INIT_A(18'h00000),
.INIT_B(18'h00000),
// Initialization File: RAM initialization file
.INIT_FILE("NONE"),
// RAM Mode: "SDP" or "TDP"
.RAM_MODE("TDP"),
// READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per port
.READ_WIDTH_A(0), // 0-72
.READ_WIDTH_B(0), // 0-18
.WRITE_WIDTH_A(0), // 0-18
.WRITE_WIDTH_B(0), // 0-72
// RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG" or "REGCE")
.RSTREG_PRIORITY_A("RSTREG"),
.RSTREG_PRIORITY_B("RSTREG"),
// SRVAL_A, SRVAL_B: Set/reset value for output
.SRVAL_A(18'h00000),
.SRVAL_B(18'h00000),
// Simulation Device: Must be set to "7SERIES" for simulation behavior
.SIM_DEVICE("7SERIES"),
// WriteMode: Value on output upon a write ("WRITE_FIRST", "READ_FIRST", or "NO_CHANGE")
.WRITE_MODE_A("WRITE_FIRST"),
.WRITE_MODE_B("WRITE_FIRST")
)
RAMB18E1_inst (
// Port A Data: 16-bit (each) output: Port A data
.DOADO(DOADO), // 16-bit output: A port data/LSB data
.DOPADOP(DOPADOP), // 2-bit output: A port parity/LSB parity
// Port B Data: 16-bit (each) output: Port B data
.DOBDO(DOBDO), // 16-bit output: B port data/MSB data
.DOPBDOP(DOPBDOP), // 2-bit output: B port parity/MSB parity
// Port A Address/Control Signals: 14-bit (each) input: Port A address and control signals (read port
// when RAM_MODE="SDP")
.ADDRARDADDR(ADDRARDADDR), // 14-bit input: A port address/Read address
.CLKARDCLK(CLKARDCLK), // 1-bit input: A port clock/Read clock
.ENARDEN(ENARDEN), // 1-bit input: A port enable/Read enable
.REGCEAREGCE(REGCEAREGCE), // 1-bit input: A port register enable/Register enable
.RSTRAMARSTRAM(RSTRAMARSTRAM), // 1-bit input: A port set/reset
.RSTREGARSTREG(RSTREGARSTREG), // 1-bit input: A port register set/reset
.WEA(WEA), // 2-bit input: A port write enable
// Port A Data: 16-bit (each) input: Port A data
.DIADI(DIADI), // 16-bit input: A port data/LSB data
.DIPADIP(DIPADIP), // 2-bit input: A port parity/LSB parity
// Port B Address/Control Signals: 14-bit (each) input: Port B address and control signals (write port
// when RAM_MODE="SDP")
.ADDRBWRADDR(ADDRBWRADDR), // 14-bit input: B port address/Write address
.CLKBWRCLK(CLKBWRCLK), // 1-bit input: B port clock/Write clock
.ENBWREN(ENBWREN), // 1-bit input: B port enable/Write enable
.REGCEB(REGCEB), // 1-bit input: B port register enable
.RSTRAMB(RSTRAMB), // 1-bit input: B port set/reset
.RSTREGB(RSTREGB), // 1-bit input: B port register set/reset
.WEBWE(WEBWE), // 4-bit input: B port write enable/Write enable
// Port B Data: 16-bit (each) input: Port B data
.DIBDI(DIBDI), // 16-bit input: B port data/MSB data
.DIPBDIP(DIPBDIP) // 2-bit input: B port parity/MSB parity
);
// End of RAMB18E1_inst instantiation

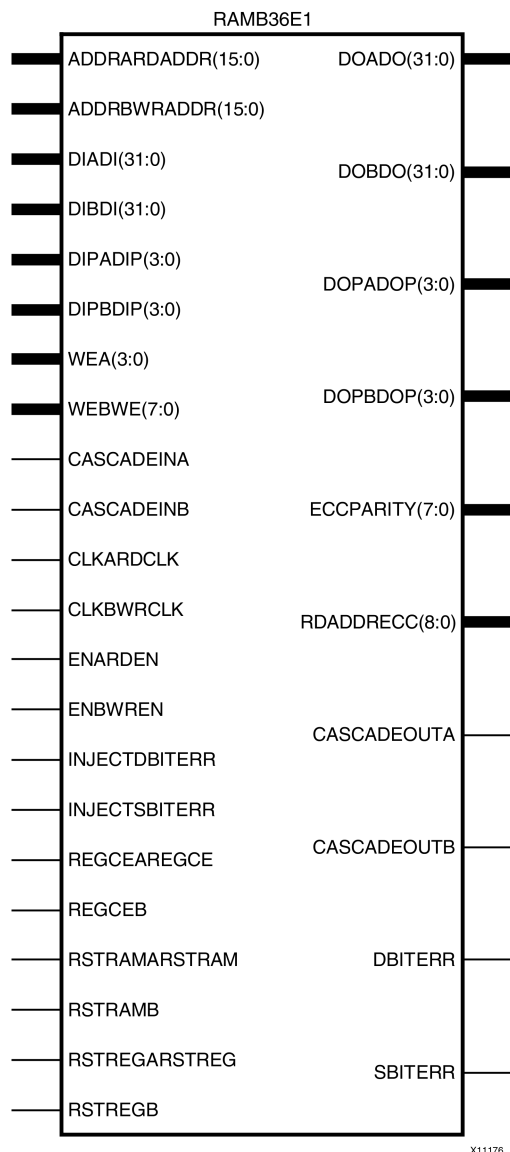
```

## 詳細情報

## 7 シリーズ FPGA の資料 (ユーザー ガイドおよびデータシート)

## RAMB36E1

プリミティブ : 36K-bit Configurable Synchronous Block RAM



## 概要

7 シリーズ デバイスにはブロック RAM が数個含まれ、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36Kb または 18Kb) 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。RAMB36E1 を使用すると、36Kb 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。カスケード 接続すると、大型の RAM を作成できます。このエレメントは、1 ビット幅 X 32K ワード ~ 36 ビット幅 X 1K 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、72 ビット幅 X 512 ワードのシンプル 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。エラー検出と訂正回路をイネーブルにすると、メモリ破損を検出し、修正することもできます。

## ポートの説明

ポート名	方向	幅	機能
ADDRARDADDR<15:0>	入力	16	ポート A アドレス入力バス/読み出しアドレス入力バス
ADDRBWRADDR<15:0>	入力	16	ポート B アドレス入力バス/書き込みアドレス入力バス
CASCADEINA	入力	1	ポート A カスケード入力。RAM_MODE が "SDP" の場合は使用されません。
CASCADEINB	入力	1	ポート B カスケード入力。RAM_MODE が "SDP" の場合は使用されません。
CASCADEOUTA	出力	1	ポート A カスケード出力。RAM_MODE が "SDP" の場合は使用されません。
CASCADEOUTB	出力	1	ポート B カスケード出力。RAM_MODE が "SDP" の場合は使用されません。
CLKARDCLK	入力	1	ポート A クロック入力/読み出しクロック入力 (立ち上がりエッジ)
CLKBWRCLK	入力	1	ポート B クロック入力/書き込みクロック入力 (立ち上がりエッジ)
DBITERR	出力	1	ダブル ビット エラーが検出されたことを示す ECC 機能からのステータス出力。ECC 機能を使用する場合は、EN_ECC_READ を TRUE に設定する必要があります。RAM_MODE が "TDP" の場合は使用されません。
DIADI<31:0>	入力	32	ポート A データ入力バス/WRADDR でアドレス指定されるデータ入力バス。RAM_MODE が "SDP" の場合は DI<31:0> となります。
DIBDI<31:0>	入力	32	ポート B データ入力バス/WRADDR でアドレス指定されるデータ入力バス。RAM_MODE が "SDP" の場合は DI<63:32> となります。
DIPADIP<3:0>	入力	4	ポート A パリティ データ入力バス/WRADDR でアドレス指定されるパリティ データ入力バス。RAM_MODE が "SDP" の場合は DIP<3:0> となります。
DIPBDIP<3:0>	入力	4	ポート B パリティ データ入力バス/WRADDR でアドレス指定されるパリティ データ入力バス。RAM_MODE が "SDP" の場合は DIP<7:4> となります。
DOADO<31:0>	出力	32	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 が "SDP" の場合は DO<31:0> となります。
DOBDO<31:0>	出力	32	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 が "SDP" の場合は DO<63:32> となります。
DOPADOP<3:0>	出力	4	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 が "SDP" の場合は DOP<3:0> となります。
DOPBDOP<3:0>	出力	4	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 が "SDP" の場合は DOP<7:4> となります。
ECCPARITY<7:0>	出力	8	ECC デコーダーでメモリ エラーを検出および訂正するために使用される、ECC エンコーダーで生成された 8 ビット データ。RAM_MODE が "TDP" の場合は使用されません。
ENARDEN	入力	1	ポート A RAM イネーブル/リード イネーブル
ENBWREN	入力	1	ポート B RAM イネーブル/ライト イネーブル

ポート名	方向	幅	機能
INJECTDBITERR	入力	1	ECC 機能を使用している場合にダブル ビット エラーを挿入します。
INJECTSBITERR	入力	1	ECC 機能を使用している場合にシングル ビット エラーを挿入します。
RDADDRECC<8:0>	出力	9	ECC 読み出しアドレス。RAM_MODE が "TDP" の場合は使用されません。
REGCEAREGCE	入力	1	ポート A 出力レジスタ クロック イネーブル入力/出力レジスタ クロック イネーブル入力 (DO_REG=1 の場合のみ有効)
REGCEB	入力	1	ポート B 出力レジスタ クロック イネーブル (DO_REG=1 および RAM_MODE="TDP" の場合のみ有効)
RSTRAMARSTRAM	入力	1	同期データ ラッチを SRVAL_A で指定された値にセット/リセットします。DO_REG=0 または 1 のとき BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合、RSTRAMARSTRAM でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。RAM_MODE が "TDP" の場合はポート A の RAM 出力がリセットされ、RAM_MODE が "SDP" の場合は RAM 出力全体がリセットされます。
RSTRAMB	入力	1	同期データ ラッチを SRVAL_B で指定された値にセット/リセットします。DO_REG=0 または 1 のとき BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合、RSTRAMB でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。RAM_MODE が "SDP" の場合は使用されません。
RSTREGARSTREG	入力	1	同期出力レジスタを SRVAL_A で指定された値にセット/リセットします。DO_REG=1 のとき出力レジスタをセット/リセットします。RSTREG_PRIORITY_A は、この信号と REGCEAREGCE の優先順位を指定します。RAM_MODE が "TDP" の場合はポート A の出力がリセットされ、RAM_MODE が "SDP" の場合は出力全体がリセットされます。
RSTREGB	入力	1	同期出力レジスタを SRVAL_B で指定された値にセット/リセットします。DO_REG=1 のとき出力レジスタをセット/リセットします。RSTREG_PRIORITY_B は、この信号と REGCEB の優先順位を指定します。RAM_MODE が "SDP" の場合は使用されません。
SBITERR	出力	1	シングル ビット エラーが検出されたことを示す ECC 機能のステータス出力。ECC 機能を使用する場合は、EN_ECC_READ を TRUE に設定する必要があります。RAM_MODE が "TDP" の場合は使用されません。
WEA<3:0>	入力	4	ポート A のバイト幅ライト イネーブル。RAM_MODE が "SDP" の場合は使用されません。異なるポート幅の WEA マップについてはユーザー ガイドを参照してください。
WEBWE<7:0>	入力	8	ポート B のバイト幅ライト イネーブル/ライト イネーブル。異なるポート幅の WEBWE マップについてはユーザー ガイドを参照してください。

## 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
RDADDR_COLLISION_HWCONFIG	文字列	"DELAYED_WRITE"、 "PERFORMANCE"	"DELAYED_WRITE"	"PERFORMANCE" に設定すると、READ_FIRST モードでのクロック パフォーマンス (周波数) が向上します。RAM の両方のポートで同じクロックを使用する場合に "PERFORMANCE" に設定すると、アドレスが重なった場合の競合規則が適用されます。"DELAYED_WRITE" モードでは、競合を発生させずに RAM を使用できます。
SIM_COLLISION_CHECK	文字列	"ALL"、 "GENERATE_X_ONLY"、 "NONE"、 "WARNING_ONLY"	"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> <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。
DOA_REG、DOB_REG	10 進数	0、1	0	1 に設定すると、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 サイクル数は増加します。0 に設定すると、1 クロック サイクルで読み出すことができますが、clock-to-out タイムが長くなります。DOA_REG は TDP モードではポート A に、SDP モードでは下位 36 ビット (パリティビットを含む) に適用され、DOB_REG は 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_ECC_READ、 EN_ECC_WRITE	ブール代数	FALSE、TRUE	FALSE	EN_ECC_WRITE は ECC エンコーダーをイネーブルにし、EN_ECC_READ は ECC デコーダーをイネーブルにします。

属性	データ型	値	デフォルト	説明
INIT_A、INIT_B	16 進数	36 ビット値	36'h000000000	コンフィギュレーション後のポート出力の初期値を指定します。INIT_A は TDP モードではポート A に、SDP モードでは下位 36 ビット (パリティビットを含む) に適用され、INIT_B は TDP ではポート B に、SDP 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_00 ~ INIT_7F	16 進数	256'h00000000 000000000000 000000000000 000000000000 000000000000 00000000 ~ 256'h 00000000 00000000 00000000	256'h00000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000	32Kb のデータ メモリ アレイ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	RAM の初期内容を記述するファイルの名前を指定します。
INITP_00 ~ INITP_0F	16 進数	256'h00000000 000000000000 000000000000 000000000000 000000000000 0000 ~ 256'h 00000000 00000000 00000000	256'h00000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000	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。
RAM_EXTENSION_A、 RAM_EXTENSION_B	文字列	"NONE"、"LOWER"、 "UPPER"	"NONE"	カスケード モードを選択します。2 つのブロック RAM をカスケード接続して 64K X 1 RAM を作成しない場合は、"NONE" に設定します。カスケード接続する場合は、RAM を正しくコンフィギュレーションするために、RAM の相対位置を "UPPER" または "LOWER" で指定します。RAM_MODE が "SDP" の場合は使用されません。
RAM_MODE	文字列	"TDP"、"SDP"	"TDP"	シンプル デュアル ポート ("SDP") または完全なデュアル ポート ("TDP") を選択します。
READ_WIDTH_A、 READ_WIDTH_B、 WRITE_WIDTH_A、 WRITE_WIDTH_B	10 進数	0、1、2、4、9、18、 36、72	0	ポート A/B の読み出し/書き込みデータ幅を指定します (パリティビットを含む)。ポートを使用しない場合は、0 に設定する必要があります。ポートを使用する場合は、適切なポート幅に設定してください。
RSTREG_PRIORITY_A、 RSTREG_PRIORITY_B	文字列	"RSTREG"、 "REGCE"	"RSTREG"	RSTREG と REGCE レジスタの優先順位を選択します。RSTREG_PRIORITY_A は 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、RSTREG_PRIORITY_B は 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。
SIM_DEVICE	文字列	"VIRTEX6"、 "7SERIES"	"VIRTEX6"	どの条件でも正しくシミュレーションが動作するよう "7SERIES" に設定する必要があります。

属性	データ型	値	デフォルト	説明
SRVAL_A, SRVAL_B	16 進数	36 ビット値	36'h000000000	同期リセット信号 (RSTREG) がアサートされたときの RAM の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	"WRITE_FIRST"、 "NO_CHANGE"、 "READ_FIRST"	"WRITE_FIRST"	<p>ポート A で書き込みが実行されるとき の出力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="SDP" の場合は "NO_CHANGE" には設定できません。 シンプル デュアル ポート インプリメン テーションでは、両方のポートで同じ クロックを使用する場合はこの属性を "READ_FIRST" に、異なるクロックを使用 する場合は "WRITE_FIRST" に設定する ことをお勧めします。これにより、このコン フィギュレーションで BRAM を使用した 場合の競合またはアドレス オーバーラッ プ時の動作が向上します。</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;

-- RAMB36E1: 36K-bit Configurable Synchronous Block RAM
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

RAMB36E1_inst : RAMB36E1
generic map (
  -- Address Collision Mode: "PERFORMANCE" or "DELAYED_WRITE"
  RDADDR_COLLISION_HWCONFIG => "DELAYED_WRITE",
  -- Collision check: Values ("ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE")
  SIM_COLLISION_CHECK => "ALL",
  -- DOA_REG, DOB_REG: Optional output register (0 or 1)
  DOA_REG => 0,
  DOB_REG => 0,
  EN_ECC_READ => FALSE,
  EN_ECC_WRITE => FALSE,

  -- INITP_00 to INITP_0F: Initial contents of the parity memory array
  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",
  INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",

  -- Enable ECC decoder,
  -- FALSE, TRUE
  -- Enable ECC encoder,
  -- FALSE, TRUE

```

UG768 (V 14.4) 2012 年 12 月 18 日

7 シリーズ FPGA および Zynq ライブラリ ガイド (HDL 用)

```

READ_WIDTH_A => 0, -- 0-72
READ_WIDTH_B => 0, -- 0-36
WRITE_WIDTH_A => 0, -- 0-36
WRITE_WIDTH_B => 0, -- 0-72
-- RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG" or "REGCE")
RSTREG_PRIORITY_A => "RSTREG",
RSTREG_PRIORITY_B => "RSTREG",
-- SRVAL_A, SRVAL_B: Set/reset value for output
SRVAL_A => X"000000000",
SRVAL_B => X"000000000",
-- Simulation Device: Must be set to "7SERIES" for simulation behavior
SIM_DEVICE => "7SERIES",
-- WriteMode: Value on output upon a write ("WRITE_FIRST", "READ_FIRST", or "NO_CHANGE")
WRITE_MODE_A => "WRITE_FIRST",
WRITE_MODE_B => "WRITE_FIRST"
)
port map (
-- Cascade Signals: 1-bit (each) output: BRAM cascade ports (to create 64kx1)
CASCADEOUTA => CASCADEOUTA, -- 1-bit output: A port cascade
CASCADEOUTB => CASCADEOUTB, -- 1-bit output: B port cascade
-- ECC Signals: 1-bit (each) output: Error Correction Circuitry ports
DBITERR => DBITERR, -- 1-bit output: Double bit error status
ECCPARITY => ECCPARITY, -- 8-bit output: Generated error correction parity
RDADDRECC => RDADDRECC, -- 9-bit output: ECC read address
SBITERR => SBITERR, -- 1-bit output: Single bit error status
-- Port A Data: 32-bit (each) output: Port A data
DOADO => DOADO, -- 32-bit output: A port data/LSB data
DOPADOP => DOPADOP, -- 4-bit output: A port parity/LSB parity
-- Port B Data: 32-bit (each) output: Port B data
DOBDO => DOBDO, -- 32-bit output: B port data/MSB data
DOPBDOP => DOPBDOP, -- 4-bit output: B port parity/MSB parity
-- Cascade Signals: 1-bit (each) input: BRAM cascade ports (to create 64kx1)
CASCADEINA => CASCADEINA, -- 1-bit input: A port cascade
CASCADEINB => CASCADEINB, -- 1-bit input: B port cascade
-- ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
INJECTDBITERR => INJECTDBITERR, -- 1-bit input: Inject a double bit error
INJECTSBITERR => INJECTSBITERR, -- 1-bit input: Inject a single bit error
-- Port A Address/Control Signals: 16-bit (each) input: Port A address and control signals (read port
-- when RAM_MODE="SDP")
ADDRARDADDR => ADDRARDADDR, -- 16-bit input: A port address/Read address
CLKARDCLK => CLKARDCLK, -- 1-bit input: A port clock/Read clock
ENARDEN => ENARDEN, -- 1-bit input: A port enable/Read enable
REGCEAREGCE => REGCEAREGCE, -- 1-bit input: A port register enable/Register enable
RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit input: A port set/reset
RSTREGARSTREG => RSTREGARSTREG, -- 1-bit input: A port register set/reset
WEA => WEA, -- 4-bit input: A port write enable
-- Port A Data: 32-bit (each) input: Port A data
DIADI => DIADI, -- 32-bit input: A port data/LSB data
DIPADIP => DIPADIP, -- 4-bit input: A port parity/LSB parity
-- Port B Address/Control Signals: 16-bit (each) input: Port B address and control signals (write port
-- when RAM_MODE="SDP")
ADDRBWRADDR => ADDRBWRADDR, -- 16-bit input: B port address/Write address
CLKBWRCLK => CLKBWRCLK, -- 1-bit input: B port clock/Write clock
ENBWREN => ENBWREN, -- 1-bit input: B port enable/Write enable
REGCEB => REGCEB, -- 1-bit input: B port register enable
RSTRAMB => RSTRAMB, -- 1-bit input: B port set/reset
RSTREGB => RSTREGB, -- 1-bit input: B port register set/reset
WEBWE => WEBWE, -- 8-bit input: B port write enable/Write enable
-- Port B Data: 32-bit (each) input: Port B data
DIBDI => DIBDI, -- 32-bit input: B port data/MSB data
DIPBDIP => DIPBDIP, -- 4-bit input: B port parity/MSB parity
);

-- End of RAMB36E1_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// RAMB36E1: 36K-bit Configurable Synchronous Block RAM
// 7 Series
// Xilinx HDL Libraries Guide, version 14.4

```

```
RAMB36E1 #(
    // Address Collision Mode: "PERFORMANCE" or "DELAYED_WRITE"
    .RDADDR_COLLISION_HWCONFIG("DELAYED_WRITE"),
    // Collision check: Values ("ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE")
    .SIM_COLLISION_CHECK("ALL"),
    // DOA_REG, DOB_REG: Optional output register (0 or 1)
    .DOA_REG(0),
    .DOB_REG(0),
    .EN_ECC_READ("FALSE"), // Enable ECC decoder,
                                // FALSE, TRUE
    .EN_ECC_WRITE("FALSE"), // Enable ECC encoder,
                                // FALSE, TRUE

    // INITP_00 to INITP_0F: Initial contents of the parity memory array
    .INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // INIT_00 to INIT_7F: Initial contents of the data memory array
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

UG768 (V 14.4) 2012 年 12 月 18 日



7 シリーズ FPGA および Zynq ライブラリ ガイド (HDL 用)

```
// when RAM_MODE="SDP")
.ADDRBWRADDR(ADDRBWRADDR),      // 16-bit input: B port address/Write address
.CLKBWRCLK(CLKBWRCLK),          // 1-bit input: B port clock/Write clock
.ENBWREN(ENBWREN),              // 1-bit input: B port enable/Write enable
.REGCEB(REGCEB),                // 1-bit input: B port register enable
.RSTRAMB(RSTRAMB),              // 1-bit input: B port set/reset
.RSTREGB(RSTREGB),              // 1-bit input: B port register set/reset
.WEBWE(WEBWE),                  // 8-bit input: B port write enable/Write enable
// Port B Data: 32-bit (each) input: Port B data
.DIBDI(DIBDI),                  // 32-bit input: B port data/MSB data
.DIPBDIP(DIPBDIP)               // 4-bit input: B port parity/MSB parity
);

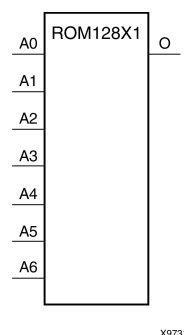
// End of RAMB36E1_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ROM128X1

プリミティブ：128-Deep by 1-Wide ROM



X9731

### 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	すべてゼロ	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
--           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to two SliceM LUT6s)
//           7 Series
// Xilinx HDL Libraries Guide, version 14.4

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]
);

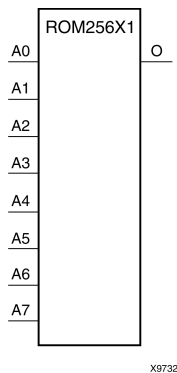
// End of ROM128X1_inst instantiation
```

詳細情報

[7 シリーズ 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	すべてゼロ	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
--           7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to forur SliceM LUT6s)
//           7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
```

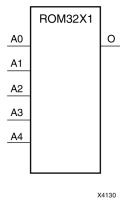
詳細情報

[7 シリーズ 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	すべてゼロ	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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a SliceM LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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]
);

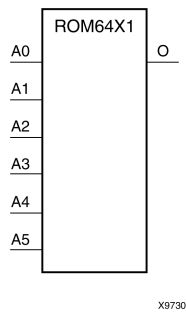
// End of ROM32X1_inst instantiation
```

詳細情報

[7 シリーズ 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 INITINIT 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	すべてゼロ	ROM の値を指定

## 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
--          7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a SliceM LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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]
);

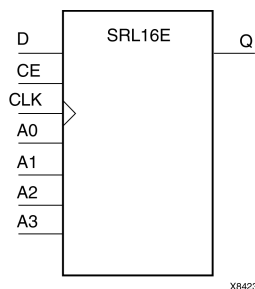
// End of ROM64X1_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 エレメントは、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 x A3) + (4 x A2) + (2 x A1) + A0 + 1 という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (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 の値を変更します。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が入力 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ゼロ (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 ビットに読み込まれます。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、新しい値が読み込まれます。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
A0	入力	1	SRL の長さ選択のビット 0
A1	入力	1	SRL の長さ選択のビット 1
A2	入力	1	SRL の長さ選択のビット 2
A3	入力	1	SRL の長さ選択のビット 3

## 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	すべてゼロ	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;

-- SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock (Mapped to SliceM LUT6)
-- 7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a SliceM LUT6)
//          7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
```

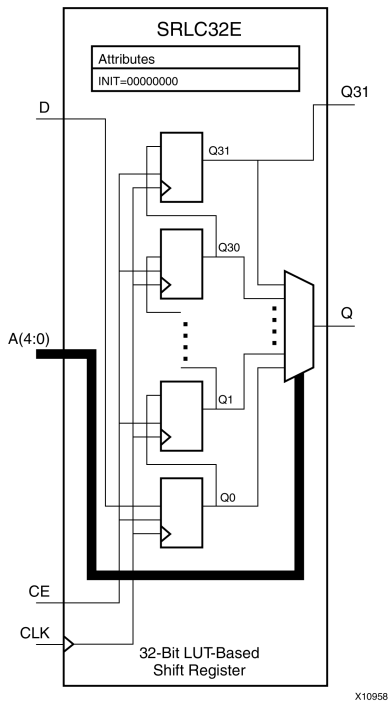
詳細情報

[7 シリーズ 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 FDCE 入力または FDRE 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	すべてゼロ	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 (Mapped to a SliceM LUT6)
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

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 (Mapped to a SliceM LUT6)
//           with clock enable
//           7 Series
// Xilinx HDL Libraries Guide, version 14.4

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
);

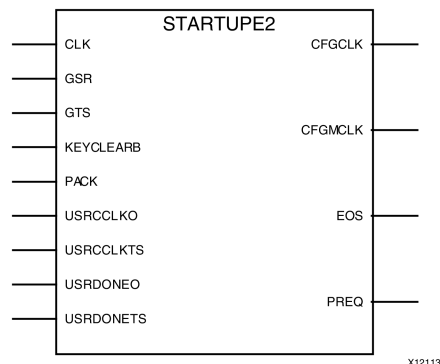
// End of SRLC32E_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## STARTUPE2

### プリミティブ：STARTUP Block



### 概要

このデザイン エLEMENTは、デバイス ピンおよびロジックを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、または専用コンフィギュレーション ピンに接続するために使用します。

### ポートの説明

ポート名	方向	幅	説明
CFGCLK	出力	1	コンフィギュレーションのメイン 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	スタートアップ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 (BBRAM) からのクリア AES デクリプタ キー入力
PACK	入力	1	PROGRAM 確認入力
PREQ	出力	1	デバイス出力への PROGRAM 要求
USRCCLKO	入力	1	ユーザー CCLK 入力
USRCCLKTS	入力	1	ユーザー CCLK トライステート イネーブル入力
USRDONEO	入力	1	ユーザー DONE ピンの出力を制御
USRDONETS	入力	1	ユーザー DONE ピンのトライステート イネーブル出力

### 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
PROG_USR	文字列	"FALSE"、"TRUE"	"FALSE"	プログラム イベント セキュリティ機能を有効にします。暗号化ビットストリームが必要です。
SIM_CCLK_FREQ	浮動小数点 (ns)	0.0 ~ 10.0	0.0	シミュレーション用のコンフィギュレーション クロック周波数 (ns) 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;

-- STARTUPE2: STARTUP Block
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

STARTUPE2_inst : STARTUPE2
generic map (
    PROG_USR => "FALSE", -- Activate program event security feature. Requires encrypted bitstreams.
    SIM_CCLK_FREQ => 0.0 -- Set the Configuration Clock Frequency(ns) for simulation.
)
port map (
    CFGCLK => CFGCLK,      -- 1-bit output: Configuration main clock output
    CFGMCLK => CFGMCLK,    -- 1-bit output: Configuration internal oscillator clock output
    EOS => EOS,            -- 1-bit output: Active high output signal indicating the End Of Startup.
    PREQ => PREQ,          -- 1-bit output: PROGRAM request to fabric output
    CLK => CLK,            -- 1-bit input: User start-up 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)
    PACK => PACK,          -- 1-bit input: PROGRAM acknowledge input
    USRCCLKO => USRCCLKO,  -- 1-bit input: User CCLK input
    USRCCLKTS => USRCCLKTS, -- 1-bit input: User CCLK 3-state enable input
    USRDONEO => USRDONEO,  -- 1-bit input: User DONE pin output control
    USRDONETS => USRDONETS -- 1-bit input: User DONE 3-state enable output
);

-- End of STARTUPE2_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// STARTUPE2: STARTUP Block
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

STARTUPE2 #(
    .PROG_USR("FALSE"), // Activate program event security feature. Requires encrypted bitstreams.
    .SIM_CCLK_FREQ(0.0) // Set the Configuration Clock Frequency(ns) for simulation.
)
STARTUPE2_inst (
    .CFGCLK(CFGCLK),      // 1-bit output: Configuration main clock output
    .CFGMCLK(CFGMCLK),    // 1-bit output: Configuration internal oscillator clock output
    .EOS(EOS),            // 1-bit output: Active high output signal indicating the End Of Startup.
    .PREQ(PREQ),          // 1-bit output: PROGRAM request to fabric output
    .CLK(CLK),            // 1-bit input: User start-up 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)
    .PACK(PACK),          // 1-bit input: PROGRAM acknowledge input
    .USRCCLKO(USRCCLKO),  // 1-bit input: User CCLK input
```

```
.USRCCLKTS(USRCCLKTS), // 1-bit input: User CCLK 3-state enable input
.USRDONEO(USRDONEO),   // 1-bit input: User DONE pin output control
.USRDONETS(USRDONETS)  // 1-bit input: User DONE 3-state enable output
);

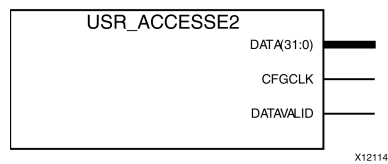
// End of STARTUPE2_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## USR\_ACESSE2

### プリミティブ：Configuration Data Access



### 概要

このデザイン エLEMENTを使用すると、コンフィギュレーション ロジック内の 32 ビットのレジスタにアクセスできます。これにより、ファブリックからビットストリームで設定可能なデータにアクセスできるようになります。

### ポートの説明

ポート名	方向	幅	説明
CFGCLK	出力	1	コンフィギュレーション クロック出力
DATA<31:0>	出力	32	コンフィギュレーション データ出力
DATAVALID	出力	1	データが有効であることを示す出力 (アクティブ High)

### 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;

-- USR_ACESSE2: Configuration Data Access
--              7 Series
-- Xilinx HDL Libraries Guide, version 14.4

USR_ACESSE2_inst : USR_ACESSE2
port map (
    CFGCLK => CFGCLK,      -- 1-bit output: Configuration Clock output
    DATA => DATA,        -- 32-bit output: Configuration Data output
    DATAVALID => DATAVALID -- 1-bit output: Active high data valid output
);

-- End of USR_ACESSE2_inst instantiation

```

## Verilog 記述 (インスタンス化)

```
// USR_ACESSE2: Configuration Data Access
//              7 Series
// Xilinx HDL Libraries Guide, version 14.4

USR_ACESSE2 USR_ACESSE2_inst (
    .CFGCLK(CFGCLK),           // 1-bit output: Configuration Clock output
    .DATA(DATA),               // 32-bit output: Configuration Data output
    .DATAVALID(DATAVALID)     // 1-bit output: Active high data valid output
);

// End of USR_ACESSE2_inst instantiation
```

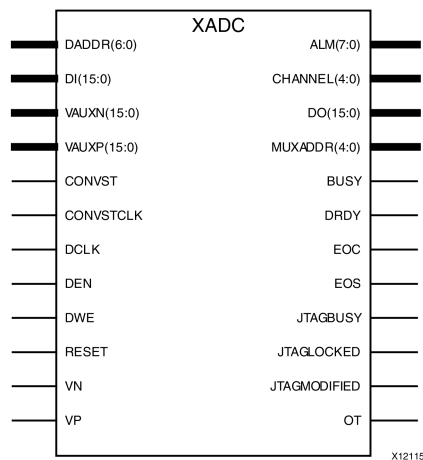
### 詳細情報

- ・ [『7 シリーズ FPGA SelectIO リソース ユーザー ガイド』 \(UG471\)](#)
- ・ [7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## XADC

### プリミティブ：Dual 12-Bit 1MSPS Analog-to-Digital Converter



## 概要

XADC には、デュアル 12 ビット 1MSPS (メガサンプル/秒) ADC とオンチップ センサーが含まれています。この ADC は完全にテストおよび特性化されています (該当する 7 シリーズ FPGA データシートを参照)。ADC は、さまざまなアプリケーションで汎用の高精度アナログ インターフェイスとして使用できます。デュアル ADC では、両方の ADC を外部からトリガーし、同時にサンプリングするなどのさまざまな動作モード、単極性や差動などのさまざまなアナログ入力信号タイプがサポートされます。ADC は 17 個までの外部アナログ入力チャンネルにアクセスできます。

XADC には、オンチップ電源電圧、チップ温度などを計測するオンチップ センサーが多数含まれています。ADC 変換データは、ステータスレジスタという専用レジスタに保存されます。これらのレジスタには、ダイナミックリコンフィギュレーション ポート (DRP) という 16 ビットの同期読み出し/書き込みポートを使用し、FPGA インターコネクトを介してアクセスできます。ADC 変換データは、JTAG TAP を使用してアクセスすることもできます。その場合、既存の FPGA JTAG インフラストラクチャを使用するのは専用インターフェイスであるため、XADC をインスタンスエートする必要はありません。デザインに XADC がインスタンスエートされていない場合、デバイスは定義済みのモード (デフォルト モード) で動作し、オンチップの温度と電源電圧が計測されます。

XADC の動作は、DRP または JTAG インターフェイスを使用して制御レジスタに書き込むことにより、ユーザー定義できます。XADC をインスタンスエートする場合は、ブロック属性を使用してこれらのレジスタの内容を初期化できます。

## ポートの説明

ポート名	方向	幅	説明
ALM<7:0>	出力	8	温度、Vccint、Vccaux、および Vccbram 用の警告出力 <ul style="list-style-type: none"> <li>ALM[0] : XADC 温度センサーの警告出力</li> <li>ALM[1] : XADC Vccint センサーの警告出力</li> <li>ALM[2] : XADC Vccaux センサーの警告出力</li> <li>ALM[3] : XADC Vccbram センサーの警告出力</li> <li>ALM[6:4] : 未定義</li> </ul>
BUSY	出力	1	ADC ビジー信号。ADC 変換中に High になります。ADC またはセンサーのキャリブレーション中にも長期で High になります。

ポート名	方向	幅	説明
CHANNEL<4:0>	出力	5	チャンネル選択出力。これらの出力では、ADC 変換の終わりに変換用の ADC 入力 MUX チャンネルが選択されます。
CONVST	入力	1	変換開始入力。ADC 入力のサンプリングを制御し、イベント モード タイミングでのみ使用されます。FPGA ロジックの汎用インターコネクトから供給されます。
CONVSTCLK	入力	1	変換開始クロック入力。クロック ネットに接続されます。CONVST と同様、ADC 入力のサンプリングを制御し、イベント モード タイミングでのみ使用されます。FPGA ロジックのローカル クロック分配ネットワークから供給されます。そのため、サンプリング インスタンス (遅延およびジッター) を制御するには、グローバル クロック入力を CONVST のソースとして使用するのが最適です。
DADDR<6:0>	入力	7	ダイナミック リコンフィギュレーション ポートのアドレス バス
DCLK	入力	1	ダイナミック リコンフィギュレーション ポートのクロック入力
DEN	入力	1	ダイナミック リコンフィギュレーション ポートのイネーブル信号
DI<15:0>	入力	16	ダイナミック リコンフィギュレーション ポートの入力データ バス
DO<15:0>	出力	16	ダイナミック リコンフィギュレーション ポートの出力データ バス
DRDY	出力	1	ダイナミック リコンフィギュレーション ポートの Data Ready 信号
DWE	入力	1	ダイナミック リコンフィギュレーション ポートのライト イネーブル
EOC	出力	1	変換終了 (End of Conversion) 信号。ADC 変換が終了し、測定データがステータス レジスタに書き込まれると、High に遷移します。
EOS	出力	1	シーケンス終了 (End of Sequence) 信号。自動 チャンネル シーケンスの最後のチャンネルから測定データがステータス レジスタに書き込まれると、High に遷移します。
JTAGBUSY	出力	1	JTAG DRP トランザクションが実行されていることを示します。
JTAGLOCKED	出力	1	JTAG インターフェイスにより DRP ポート ロックが要求されたことを示します。DRP がアクセス可能になったこと (Low の場合) を示すためにも使用されます。
JTAGMODIFIED	出力	1	DRP に対して JTAG Write が発生したことを示します。
MUXADDR<4:0>	出力	5	外部マルチプレクサー モードで使用され、シーケンスで次に変換されるチャンネルのアドレスを示します。外部マルチプレクサーにチャンネル アドレスを供給します。
OT	出力	1	温度超過警告出力
RESET	入力	1	XADC 制御ロジックのリセット信号
VAUXN<15:0>	入力	16	N 側補助アナログ入力
VAUXP<15:0>	入力	16	P 側補助アナログ入力
VN	入力	1	N 側アナログ入力
VP	入力	1	P 側アナログ入力

## 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
INIT_4A	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	シーケンスレジスタ 2
INIT_4B	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	シーケンスレジスタ 3
INIT_4C	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	シーケンスレジスタ 4
INIT_4D	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	シーケンスレジスタ 5
INIT_4E	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	シーケンスレジスタ 6
INIT_4F	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	シーケンスレジスタ 7
INIT_5C	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	VbRam の警告下限しきい値
INIT_40	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	コンフィギュレーションレジスタ 0
INIT_41	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	コンフィギュレーションレジスタ 1
INIT_42	16 進数	16'h0000 ～ 16'hffff	16'h0800	コンフィギュレーションレジスタ 2
INIT_43	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	テストレジスタ 0
INIT_44	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	テストレジスタ 1
INIT_45	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	テストレジスタ 2
INIT_46	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	テストレジスタ 3
INIT_47	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	テストレジスタ 4
INIT_48	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	シーケンスレジスタ 0
INIT_49	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	シーケンスレジスタ 1
INIT_50	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	警告制限レジスタ 0
INIT_51	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	警告制限レジスタ 1
INIT_52	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	警告制限レジスタ 2
INIT_53	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	警告制限レジスタ 3
INIT_54	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	警告制限レジスタ 4
INIT_55	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	警告制限レジスタ 5
INIT_56	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	警告制限レジスタ 6
INIT_57	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	警告制限レジスタ 7
INIT_58	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	VbRam の警告上限しきい値
INIT_59、INIT_5A、 INIT_5B、INIT_5D、 INIT_5E、INIT_5F	16 進数	16'h0000 ～ 16'hffff	すべてゼロ	今後の使用のために予約

属性	データ型	値	デフォルト	説明
SIM_DEVICE	文字列	"7SERIES"、"ZYNQ"	"7SERIES"	シミュレーションを正しく実行するためターゲット デバイスを指定します。
SIM_MONITOR_FILE	文字列	ファイル名	なし	XADC シミュレーション ビヘイビア一用のアナログ電圧および温度データを含むファイルの名前（およびシミュレーション ディレクトリと異なる場合はディレクトリ）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;

-- XADC: Dual 12-Bit 1MSPS Analog-to-Digital Converter
--       7 Series
-- Xilinx HDL Libraries Guide, version 14.4

XADC_inst : XADC
generic map (
  -- INIT_40 - INIT_42: XADC configuration registers
  INIT_40 => X"0000",
  INIT_41 => X"0000",
  INIT_42 => X"0800",
  -- INIT_48 - INIT_4F: Sequence Registers
  INIT_48 => X"0000",
  INIT_49 => X"0000",
  INIT_4A => X"0000",
  INIT_4B => X"0000",
  INIT_4C => X"0000",
  INIT_4D => X"0000",
  INIT_4E => X"0000",
  INIT_4F => X"0000",
  INIT_4E => X"0000", -- Sequence register 6
  -- INIT_50 - INIT_58, INIT5C: Alarm Limit Registers
  INIT_50 => X"0000",
  INIT_51 => X"0000",
  INIT_52 => X"0000",
  INIT_53 => X"0000",
  INIT_54 => X"0000",
  INIT_55 => X"0000",
  INIT_56 => X"0000",
  INIT_57 => X"0000",
  INIT_58 => X"0000",
  INIT_5C => X"0000",
  -- Simulation attributes: Set for proper simulation behavior
  SIM_DEVICE => "7SERIES", -- Select target device (values)
  SIM_MONITOR_FILE => "design.txt" -- Analog simulation data file name
)
port map (
  -- ALARMS: 8-bit (each) output: ALM, OT
  ALM => ALM, -- 8-bit output: Output alarm for temp, Vccint, Vccaux and Vccbram
  OT => OT, -- 1-bit output: Over-Temperature alarm
  -- Dynamic Reconfiguration Port (DRP): 16-bit (each) output: Dynamic Reconfiguration Ports
  DO => DO, -- 16-bit output: DRP output data bus
  DRDY => DRDY, -- 1-bit output: DRP data ready
  -- STATUS: 1-bit (each) output: XADC status ports
  BUSY => BUSY, -- 1-bit output: ADC busy output
  CHANNEL => CHANNEL, -- 5-bit output: Channel selection outputs
  EOC => EOC, -- 1-bit output: End of Conversion
  EOS => EOS, -- 1-bit output: End of Sequence
  JTAGBUSY => JTAGBUSY, -- 1-bit output: JTAG DRP transaction in progress output
  JTAGLOCKED => JTAGLOCKED, -- 1-bit output: JTAG requested DRP port lock
  JTAGMODIFIED => JTAGMODIFIED, -- 1-bit output: JTAG Write to the DRP has occurred
  MUXADDR => MUXADDR, -- 5-bit output: External MUX channel decode

```

```
-- Auxiliary Analog-Input Pairs: 16-bit (each) input: VAUXP[15:0], VAUXN[15:0]
VAUXN => VAUXN,          -- 16-bit input: N-side auxiliary analog input
VAUXP => VAUXP,          -- 16-bit input: P-side auxiliary analog input
-- CONTROL and CLOCK: 1-bit (each) input: Reset, conversion start and clock inputs
CONVST => CONVST,        -- 1-bit input: Convert start input
CONVSTCLK => CONVSTCLK,   -- 1-bit input: Convert start input
RESET => RESET,          -- 1-bit input: Active-high reset
-- Dedicated Analog Input Pair: 1-bit (each) input: VP/VN
VN => VN,                -- 1-bit input: N-side analog input
VP => VP,                -- 1-bit input: P-side analog input
-- Dynamic Reconfiguration Port (DRP): 7-bit (each) input: Dynamic Reconfiguration Ports
DADDR => DADDR,          -- 7-bit input: DRP address bus
DCLK => DCLK,            -- 1-bit input: DRP clock
DEN => DEN,              -- 1-bit input: DRP enable signal
DI => DI,                -- 16-bit input: DRP input data bus
DWE => DWE               -- 1-bit input: DRP write enable
);

-- End of XADC_inst instantiation
```

## Verilog 記述 (インスタンスーション)

```
// XADC: Dual 12-Bit 1MSPS Analog-to-Digital Converter
//       7 Series
// Xilinx HDL Libraries Guide, version 14.4

XADC #(
    // INIT_40 - INIT_42: XADC configuration registers
    .INIT_40(16'h0000),
    .INIT_41(16'h0000),
    .INIT_42(16'h0800),
    // INIT_48 - INIT_4F: Sequence Registers
    .INIT_48(16'h0000),
    .INIT_49(16'h0000),
    .INIT_4A(16'h0000),
    .INIT_4B(16'h0000),
    .INIT_4C(16'h0000),
    .INIT_4D(16'h0000),
    .INIT_4E(16'h0000),
    .INIT_4F(16'h0000),
    .INIT_4E(16'h0000), // Sequence register 6
    // INIT_50 - INIT_58, INIT5C: Alarm Limit Registers
    .INIT_50(16'h0000),
    .INIT_51(16'h0000),
    .INIT_52(16'h0000),
    .INIT_53(16'h0000),
    .INIT_54(16'h0000),
    .INIT_55(16'h0000),
    .INIT_56(16'h0000),
    .INIT_57(16'h0000),
    .INIT_58(16'h0000),
    .INIT_5C(16'h0000),
    // Simulation attributes: Set for proper simulation behavior
    .SIM_DEVICE("7SERIES"), // Select target device (values)
    .SIM_MONITOR_FILE("design.txt") // Analog simulation data file name
)
XADC_inst (
    // ALARMS: 8-bit (each) output: ALM, OT
    .ALM(ALM), // 8-bit output: Output alarm for temp, Vccint, Vccaux and Vccbram
    .OT(OT), // 1-bit output: Over-Temperature alarm
    // Dynamic Reconfiguration Port (DRP): 16-bit (each) output: Dynamic Reconfiguration Ports
    .DO(DO), // 16-bit output: DRP output data bus
    .DRDY(DRDY), // 1-bit output: DRP data ready
    // STATUS: 1-bit (each) output: XADC status ports
    .BUSY(BUSY), // 1-bit output: ADC busy output
    .CHANNEL(CHANNEL), // 5-bit output: Channel selection outputs
    .EOC(EOC), // 1-bit output: End of Conversion
    .EOS(EOS), // 1-bit output: End of Sequence
    .JTAGBUSY(JTAGBUSY), // 1-bit output: JTAG DRP transaction in progress output
    .JTAGLOCKED(JTAGLOCKED), // 1-bit output: JTAG requested DRP port lock
    .JTAGMODIFIED(JTAGMODIFIED), // 1-bit output: JTAG Write to the DRP has occurred
    .MUXADDR(MUXADDR), // 5-bit output: External MUX channel decode
    // Auxiliary Analog-Input Pairs: 16-bit (each) input: VAUXP[15:0], VAUXN[15:0]
    .VAUXN(VAUXN), // 16-bit input: N-side auxiliary analog input
    .VAUXP(VAUXP), // 16-bit input: P-side auxiliary analog input
    // CONTROL and CLOCK: 1-bit (each) input: Reset, conversion start and clock inputs
    .CONVST(CONVST), // 1-bit input: Convert start input
    .CONVSTCLK(CONVSTCLK), // 1-bit input: Convert start input
    .RESET(RESET), // 1-bit input: Active-high reset
    // Dedicated Analog Input Pair: 1-bit (each) input: VP/VN
    .VN(VN), // 1-bit input: N-side analog input
    .VP(VP), // 1-bit input: P-side analog input
    // Dynamic Reconfiguration Port (DRP): 7-bit (each) input: Dynamic Reconfiguration Ports
    .DADDR(DADDR), // 7-bit input: DRP address bus
    .DCLK(DCLK), // 1-bit input: DRP clock
    .DEN(DEN), // 1-bit input: DRP enable signal
    .DI(DI), // 16-bit input: DRP input data bus
    .DWE(DWE) // 1-bit input: DRP write enable
);

// End of XADC_inst instantiation
```

詳細情報

[7 シリーズ FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)