

# Virtex-5 ライブラリ ガイド (HDL 用)

UG621 (v14.5) 2013 年 3 月 20 日

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



## Notice of Disclaimer

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

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

本資料は英語版 (v14.5) を翻訳したもので、内容に相違が生じる場合には原文を優先します。

資料によっては英語版の更新に対応していないものがあります。

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

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

## 概要

---

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

このマニュアルには、次の内容が含まれます。

- ・ 概要
- ・ ターゲット デバイス変更後のエレメントのリスト
- ・ 各マクロの詳細説明
- ・ このアーキテクチャでサポートされるプリミティブとマクロのファンクション別リスト
- ・ 各プリミティブの詳細説明

## デザイン エLEMENTについて

このバージョンのライブラリ ガイドでは、Virtex®-5 デバイスのデザイン エLEMENTの説明とそのインスタンスエーション コード例を示します。インスタンスエーション 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。

- ・ **ターゲット デバイス変更後のエレメント**：このアーキテクチャで使用すると自動的に ISE ツールにより変換される既存のデザイン エLEMENTです。ターゲット デバイスを変更することにより最新の回路設計技術を最大限に利用できるようになります。
- ・ **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エーションしにくいようなプリミティブをインスタンスエーション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エーションして変換 (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エーションし、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。

## ターゲット デバイス変更後のデザイン エレメント

デザイン エレメントの中には、アーキテクチャを変更すると、最新の回路設計技術を最大限に利用できるように、ISE® Design Suite ツールにより自動的に変更されるものがあります。

次の表に、それらのエレメントと変更後のアドバンス エレメントを示します。

元のデザイン エレメント	変更後のエレメント
BUFGCE_1	BUFGCE + INV
BUFGMUX	BUFGMUX_CTRL
BUFGMUX_1	BUFGMUX_CTRL + INV
BUFGMUX_VIRTEX4	BUFGMUX_CTRL
BUFGP	BUFG
DCM_BASE	DCM_ADV
DCM_PS	DCM_ADV
DSP48	DSP48E
FD	FDCPE
FD_1	FDCPE + INV
FDC	FDCPE
FDC_1	FDCPE + INV
FDCE	FDCPE
FDCE_1	FDCPE + INV
FDCP	FDCPE
FDCP_1	FDCPE + INV
FDE	FDCPE
FDE_1	FDCPE + INV
FDPE	FDCPE
FDPE_1	FDCPE + INV
FDR	FDRSE
FDR_1	FDRSE + INV
FDRE	FDRSE
FDRE_1	FDRSE + INV

元のデザイン エLEMENT	変更後のELEMENT
FDRS	FDRSE
FDRS_1	FDRSE + INV
FDS	FDRSE
FDS_1	FDRSE + INV
FDSE	FDRSE
FDSE_1	FDRSE + INV
FIFO16	FIFO18
ISERDES	ISERDES_NODELAY
JTAGPPC	JTAG_PPC440
LD	LDCPE
LD_1	LDCPE + INV
LDC	LDCPE
LDC_1	LDCPE + INV
LDCE	LDCPE
LDCE_1	LDCPE + INV
LDCP	LDCPE
LDCP_1	LDCPE + INV
LDE	LDCPE
LDE_1	LDCPE + INV
LDP	LDCPE
LDP_1	LDCPE + INV
LDPE	LDCPE
LDPE_1	LDCPE + INV
LUT1	LUT5
LUT1_L	LUT5_L
LUT1_D	LUT5_D
LUT2	LUT5
LUT2_L	LUT5_L
LUT2_D	LUT5_D
LUT3	LUT5
LUT3_L	LUT5_L
LUT3_D	LUT56_D
LUT4	LUT5
LUT4_L	LUT5_L
LUT4_D	LUT5_D
MULT_AND	LUT6
MULT18X18	DSP48E
MULT18X18S	DSP48E

元のデザイン エLEMENT	変更後のELEMENT
MUXCY	CARRY4
MUXCY_D	CARRY4
MUXCY_L	CARRY4
MUXF5	LUT5
MUXF5_D	LUT5_D
MUXF5_L	LUT5_L
MUXF6	LUT6
MUXF6_D	LUT6_D
MUXF6_L	LUT6_L
PMCD	PLL_ADV
RAM16X1D	RAM64X1D
RAM16X1S	RAM64X1S
RAM32X1S	RAM64X1S
RAMB16	RAMB18
RAMB16BWE	RAMB18
ROM128X1	6 入力 LUT X 2 + MUXF7
ROM16X1	LUT5
ROM256X1	6 入力 LUT X 4 + MUXF6/7
ROM32X1	LUT5
ROM64X1	LUT6
SRLC16	SRLC32E
SRLC16_1	SRLC32E + INV
SRLC16E	SRLC32E
SRLC16E_1	SRLC32E + INV
XORCY	CARRY4
XORCY_D	CARRY4
XORCY_L	CARRY4





## UniMacro

---

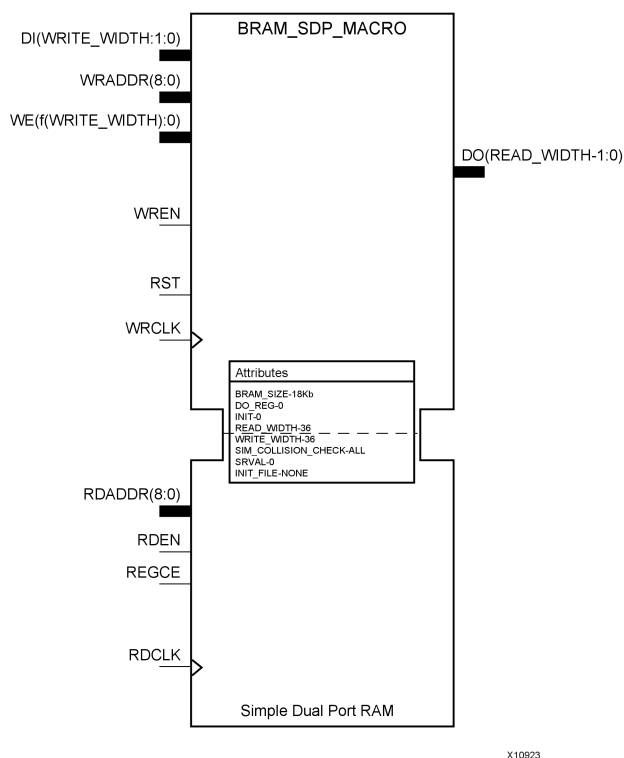
このセクションでは、Virtex®-5 デバイスで利用できる UniMacro について説明します。UniMacro は、アルファベット順に並べられています。

各 UniMacro について、次の情報を示します。

- ・ 名称
- ・ 説明
- ・ 回路図シンボル
- ・ 論理表 (該当するエレメントでのみ)
- ・ ポートの説明
- ・ デザインの入力方法
- ・ 使用可能な属性
- ・ インスタンスエーション 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

FPGA デバイスにはブロック RAM が数個含まれ、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。読み出しと書き込みは、コンポーネントに供給されるクロックに完全に同期して実行されます。ただし、読み出しポートと書き込みポートは完全に独立しており、お互いに非同期で、同じメモリアレイにアクセスします。バイト イネーブルの書き込みが可能で、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

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

### ポートの説明

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

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

## コンフィギュレーション表

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

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

```
Library UNISIM;

use UNISIM.vcomponents.all;


-- BRAM_SDP_MACRO: Simple Dual Port RAM
--                   Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5


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

[illegible]

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

## Verilog 記述 (インスタンス化)

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日



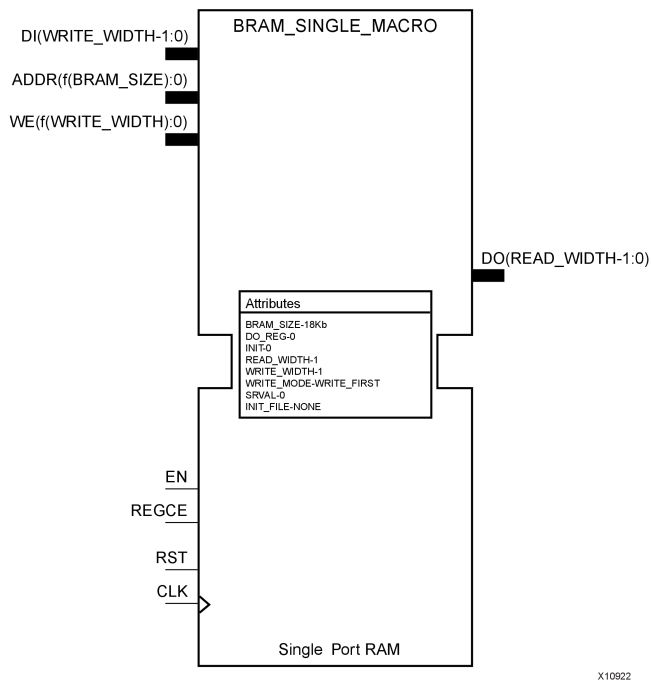
Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## BRAM\_SINGLE\_MACRO

### マクロ : Single Port RAM



### 概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。これらのシングル ポートのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。バイトイネーブルの書き込みが可能で、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

### ポートの説明

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

## コンフィギュレーション表

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

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

インスタンスレーション	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
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。
DO_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロック サイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能ですが、clock-to-out タイムが長くなります。
READ_WIDTH、WRITE_WIDTH	整数	1 ～ 36	1	DI および DO バスの幅を指定します。  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。
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 0	コンフィギュレーション後の出力の初期値を指定します。
SRVAL	16 進数	72 ビット値	すべて 0	同期リセット信号 (RST) がアサートされたときの DO ポートの出力値を指定します。
SIM_MODE	文字列	"SAFE"、"FAST"	"SAFE"	シミュレーションのみの属性です。"FAST" に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。
INIT_00 ～ INIT_FF	16 進数	256 ビット値	すべて 0	16Kb または 32Kb のデータ メモリ アレイの初期値を指定します。
INITP_00 ～ INITP_0F	16 進数	256 ビット値	すべて 0	2Kb または 4Kb のパリティ データ メモリ アレイの初期値を指定します。

## 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;

-- BRAM_SINGLE_MACRO: Single Port RAM
--                               Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

BRAM_SINGLE_MACRO_inst : BRAM_SINGLE_MACRO
generic map (
  BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
  DEVICE => "VIRTEX5", -- Target Device: "VIRTEX5", "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")
  SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST",
                        -- see "Synthesis and Simulation Design Guide" for details
  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",
  INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
```

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

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日



```

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 INITP_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
ADDR => ADDR,   -- Input address
CLK => CLK,     -- Input clock
DI => DI,       -- Input data port
EN => EN,       -- Input RAM enable
REGCE => REGCE, -- Input output register enable
RST => RST,     -- Input reset
WE => WE        -- Input write enable
);

-- End of BRAM_SINGLE_MACRO_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// BRAM_SINGLE_MACRO: Single Port RAM
//                               Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

////////////////////////////////////
// READ_WIDTH | BRAM_SIZE | READ Depth | ADDR Width | //
// WRITE_WIDTH |          | WRITE Depth |          | WE 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 | 13-bit | 1-bit //
// 2 | "18Kb" | 8192 | 12-bit | 1-bit //
// 1 | "36Kb" | 32768 | 13-bit | 1-bit //
// 1 | "18Kb" | 16384 | 12-bit | 1-bit //
////////////////////////////////////

```

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

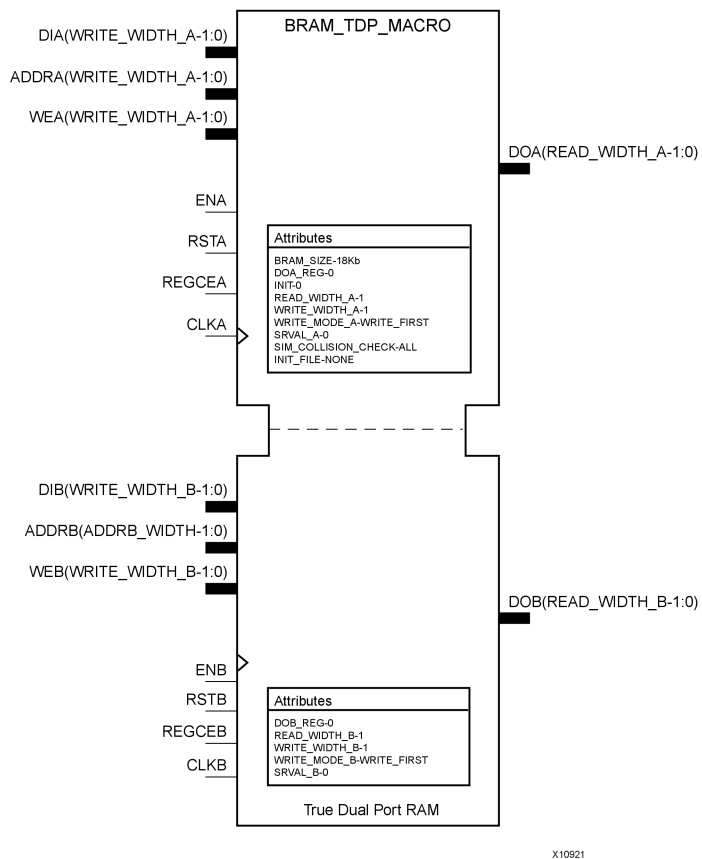
```
.INITP_02(256'h0000000000000000000000000000000000000000000000000),  
.INITP_03(256'h0000000000000000000000000000000000000000000000000),  
.INITP_04(256'h0000000000000000000000000000000000000000000000000),  
.INITP_05(256'h0000000000000000000000000000000000000000000000000),  
.INITP_06(256'h0000000000000000000000000000000000000000000000000),  
.INITP_07(256'h0000000000000000000000000000000000000000000000000),  
  
// The next set of INIT xx are valid when configured as 36Kb  
.INITP_08(256'h0000000000000000000000000000000000000000000000000),  
.INITP_09(256'h0000000000000000000000000000000000000000000000000),  
.INITP_0A(256'h0000000000000000000000000000000000000000000000000),  
.INITP_0B(256'h0000000000000000000000000000000000000000000000000),  
.INITP_0C(256'h0000000000000000000000000000000000000000000000000),  
.INITP_0D(256'h0000000000000000000000000000000000000000000000000),  
.INITP_0E(256'h0000000000000000000000000000000000000000000000000),  
.INITP_0F(256'h0000000000000000000000000000000000000000000000000)  
) BRAM_SINGLE_MACRO inst (  
    .DO(DO),      // Output data  
    .ADDR(ADDR),  // Input address  
    .CLK(CLK),    // Input clock  
    .DI(DI),       // Input data port  
    .EN(EN),       // Input RAM enable  
    .RGCE(REGCE), // Input output register enable  
    .RST(RST),     // Input reset  
    .WE(WE)        // Input write enable  
);  
  
// End of BRAM SINGLE_MACRO inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## BRAM\_TDP\_MACRO

マクロ : True Dual Port RAM



## 概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (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。読み出しと書き込みは、コンポーネントに供給されるクロックに完全に同期して実行されます。ただし、読み出しポートと書き込みポートは完全に独立しており、お互いに非同期で、同じメモリ アレイにアクセスします。バイト イネーブルの書き込みが可能になり、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

## ポートの説明

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

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

## コンフィギュレーション表

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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
BRAM_SIZE	文字列	"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。
DO_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロックサイクルは増加します。値を 0 にすると、1 クロックサイクルで読み出しが可能です。clock-to-out タイムが長くなります。
INIT	16 進数	72 ビット値	すべて 0	コンフィギュレーション後の出力の初期値を指定します。
INIT_FILE	文字列	ファイルの名前と場所	なし	初期値を含むファイルの名前を指定します。
READ_WIDTH、WRITE_WIDTH	整数	1 ~ 72	36	DI および DO バスの幅を指定します。  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) にする必要があります。</li> </ul>
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>



属性	データ型	値	デフォルト	説明
				<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。
SIM_MODE	文字列	“SAFE”、“FAST”	“SAFE”	シミュレーションのみの属性です。“FAST” に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。
SRVAL_A、SRVAL_B	16 進数	72 ビット値	すべて 0	同期リセット信号 (RST) がアサートされたときの DO ポートの出力値を指定します。
INIT_00 ～ INIT_FF	16 進数	256 ビット値	すべて 0	16Kb または 32Kb のデータ メモリ アレイの初期値を指定します。
INITP_00 ～ INITP_0F	16 進数	256 ビット値	すべて 0	2Kb または 4Kb のパリティ データ メモリ アレイの初期値を指定します。

## 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。

[illegible]

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

```
port map  $\bar{m}$  (
    DOA => DOA,      -- Output port-A data
    DOB => DOB,      -- Output port-B data
    ADDR_A => ADDR_A, -- Input port-A address
    ADDR_B => ADDR_B, -- Input port-B address
```

```

    CLKA => CLKA,      -- Input port-A clock
    CLKB => CLKB,      -- Input port-B clock
    DIA => DIA,        -- Input port-A data
    DIB => DIB,        -- Input port-B data
    ENA => ENA,        -- Input port-A enable
    ENB => ENB,        -- Input port-B enable
    REGCEA => REGCEA,  -- Input port-A output register enable
    REGCEB => REGCEB,  -- Input port-B output register enable
    RSTA => RSTA,      -- Input port-A reset
    RSTB => RSTB,      -- Input port-B reset
    WEA => WEA,        -- Input port-A write enable
    WEB => WEB         -- Input port-B write enable
);

-- End of BRAM TDP MACRO inst instantiation

```

## Verilog 記述 (インスタンス化)

[illegible]

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

```
.ENA(ENA),          // Input port-A enable
.ENB(ENB),          // Input port-B enable
.REGCEA(REGCEA),    // Input port-A output register enable
.REGCEB(REGCEB),    // Input port-B output register enable
.RSTA(RSTA),        // Input port-A reset
.RSTB(RSTB),        // Input port-B reset
.WEA(WEA),          // Input port-A write enable
.WEB(WEB)           // Input port-B write enable
);

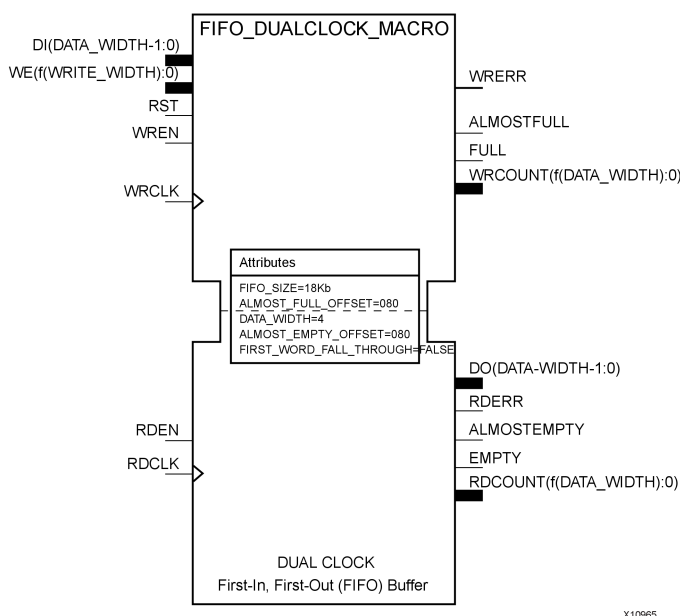
// End of BRAM_TDP_MACRO_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## FIFO\_DUALCLOCK\_MACRO

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



X10965

### 概要

FPGA デバイスにはブロック RAM が数個含まれ、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。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
出力ポート			
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	「コンフィギュレーション表」を参照	ADDR で指定されたデータ出力バス
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	「コンフィギュレーション表」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	「コンフィギュレーション表」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。
入力ポート			
DI	入力	「コンフィギュレーション表」を参照	ADDR で指定されたデータ入力バス
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

## コンフィギュレーション表

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

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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
ALMOST_EMPTY_OFFSET	16 進数	13 ビット値	すべて 0	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 0	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。
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。
SIM_MODE	文字列	"SAFE"、 "FAST"	"SAFE"	シミュレーションのみの属性です。"FAST" に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。

## 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
--                               Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

FIFO_DUALCLOCK_MACRO_inst : FIFO_DUALCLOCK_MACRO
generic map (
    DEVICE => "VIRTEX5",           -- Target Device: "VIRTEX5", "VIRTEX6"
    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
    SIM_MODE => "SAFE") -- Simulation "SAFE" vs "FAST",
                        -- see "Synthesis and Simulation Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY,    -- Output almost empty
    ALMOSTFULL => ALMOSTFULL,      -- Output almost full
    DO => DO,                       -- Output data
    EMPTY => EMPTY,                -- Output empty
    FULL => FULL,                  -- Output full
    RDCOUNT => RDCOUNT,            -- Output read count
    RDERR => RDERR,               -- Output read error
    WRCOUNT => WRCOUNT,           -- Output write count
    WRERR => WRERR,               -- Output write error
    DI => DI,                     -- Input data
    RDCLK => RDCLK,               -- Input read clock
    RDEN => RDEN,                 -- Input read enable
    RST => RST,                   -- Input reset
    WRCLK => WRCLK,               -- Input write clock
    WREN => WREN                  -- 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 (FIFO) RAM Buffer
//                               Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

////////////////////////////////////
// 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 #(
    .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("VIRTEX5"), // Target device: "VIRTEX5", "VIRTEX6"
    .FIFO_SIZE ("18Kb"), // Target BRAM: "18Kb" or "36Kb"
    .FIRST_WORD_FALL_THROUGH ("FALSE"), // Sets the FIFO FWFT to "TRUE" or "FALSE"
    .SIM_MODE("SAFE") // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
) FIFO_DUALCLOCK_MACRO_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // Output almost empty
    .ALMOSTFULL(ALMOSTFULL), // Output almost full
    .DO(DO), // Output data
    .EMPTY(EMPTY), // Output empty
    .FULL(FULL), // Output full
    .RDCOUNT(RDCOUNT), // Output read count
    .RDERR(RDERR), // Output read error
    .WRCOUNT(WRCOUNT), // Output write count
    .WRERR(WRERR), // Output write error
    .DI(DI), // Input data
    .RDCLK(RDCLK), // Input read clock
    .RDEN(RDEN), // Input read enable
    .RST(RST), // Input reset
    .WRCLK(WRCLK), // Input write clock
    .WREN(WREN) // Input write enable
);

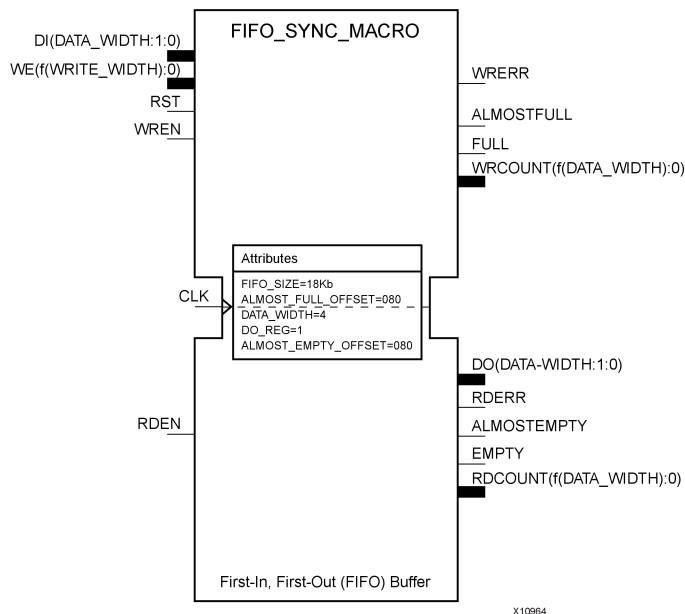
// End of FIFO_DUALCLOCK_MACRO_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート: DC 特性およびスイッチ特性』\(DS202\)](#)

## 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、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。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
出力ポート			
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	「コンフィギュレーション表」を参照	ADDR で指定されたデータ出力バス
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	「コンフィギュレーション表」を参照	FIFO データ読み出しポインター
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	「コンフィギュレーション表」を参照	FIFO データ書き込みポインター

ポート名	方向	幅	機能
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。
入力ポート			
CLK	入力	1	読み出し/書き込みクロック
DI	入力	「コンフィギュレーション表」を参照	ADDR で指定されたデータ入力バス
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

## コンフィギュレーション表

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

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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
ALMOST_EMPTY_OFFSET	16 進数	13 ビット値	すべて 0	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 0	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。
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。
SIM_MODE	文字列	"SAFE"、 "FAST"	"SAFE"	シミュレーションのみの属性です。"FAST" に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。

## 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;

-- FIFO_SYNC_MACRO: Synchronous First-In, First-Out (FIFO) RAM Buffer
--                      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

FIFO_SYNC_MACRO_inst : FIFO_SYNC_MACRO
generic map (
    DEVICE => "VIRTEX5",           -- Target Device: "VIRTEX5", "VIRTEX6"
    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"
    SIM_MODE => "SAFE")           -- Simulation) "SAFE" vs "FAST",
                                -- see "Synthesis and Simulation Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY,    -- Output almost empty
    ALMOSTFULL => ALMOSTFULL,      -- Output almost full
    DO => DO,                      -- Output data
    EMPTY => EMPTY,               -- Output empty
    FULL => FULL,                 -- Output full
    RDCOUNT => RDCOUNT,           -- Output read count
    RDERR => RDERR,              -- Output read error
    WRCOUNT => WRCOUNT,          -- Output write count
    WRERR => WRERR,              -- Output write error
    CLK => CLK,                  -- Input clock
    DI => DI,                    -- Input data
    RDEN => RDEN,                -- Input read enable
    RST => RST,                  -- Input reset
    WREN => WREN                 -- 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 (FIFO) RAM Buffer
//                               Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

////////////////////////////////////
// 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 #(
    .DEVICE("VIRTEX5"), // Target device: "VIRTEX5", "VIRTEX6"
    .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"
    .SIM_MODE("SAFE") // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
) FIFO_SYNC_MACRO_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // Output almost empty
    .ALMOSTFULL(ALMOSTFULL), // Output almost full
    .DO(DO), // Output data
    .EMPTY(EMPTY), // Output empty
    .FULL(FULL), // Output full
    .RDCOUNT(RDCOUNT), // Output read count
    .RDERR(RDERR), // Output read error
    .WRCOUNT(WRCOUNT), // Output write count
    .WRERR(WRERR), // Output write error
    .CLK(CLK), // Input clock
    .DI(DI), // Input data
    .RDEN(RDEN), // Input read enable
    .RST(RST), // Input reset
    .WREN(WREN) // Input write enable
);

// End of FIFO_SYNC_MACRO_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート: DC 特性およびスイッチ特性』\(DS202\)](#)





## 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	ギガビット I/O	レジスタおよびラッチ
演算ファンクション	I/O コンポーネント	シフト レジスタ LUT
クロック コンポーネント	プロセッサ	スライス/CLB プリミティブ
コンフィギュレーション/BSCAN コンポーネント	RAM/ROM	

### アドバンス

デザイン エLEMENT	説明
<a href="#">CRC32</a>	プリミティブ : Cyclic Redundancy Check Calculator for 32 bits
<a href="#">PCIE_EP</a>	プリミティブ : PCI Express
<a href="#">SYSMON</a>	プリミティブ : System Monitor
<a href="#">TEMAC</a>	プリミティブ : Tri-mode Ethernet Media Access Controller (MAC)

### 演算ファンクション

デザイン エLEMENT	説明
<a href="#">DSP48E</a>	プリミティブ : 25x18 Two's Complement Multiplier with Integrated 48-Bit, 3-Input Adder/Subtractor/Accumulator or 2-Input Logic Unit

## クロック コンポーネント

デザイン エLEMENT	説明
BUFG	プリミティブ : Global Clock Buffer
BUFGCE	プリミティブ : Global Clock Buffer with Clock Enable
BUFGCE_1	プリミティブ : Global Clock Buffer with Clock Enable and Output State 1
BUFGCTRL	プリミティブ : Global Clock MUX Buffer
BUFGMUX_CTRL	プリミティブ : 2-to-1 Global Clock MUX Buffer
BUFIO	プリミティブ : Local Clock Buffer for I/O
BUFR	プリミティブ : Regional Clock Buffer for I/O and Logic Resources
DCM_ADV	プリミティブ : Advanced Digital Clock Manager Circuit
DCM_BASE	プリミティブ : Base Digital Clock Manager Circuit
DCM_PS	プリミティブ : Digital Clock Manager with Basic and Phase Shift Features
PLL_ADV	プリミティブ : Advanced Phase Locked Loop Clock Circuit
PLL_BASE	プリミティブ : Basic Phase Locked Loop Clock Circuit

## コンフィギュレーション/BSCAN コンポーネント

デザイン エLEMENT	説明
BSCAN_VIRTEX5	プリミティブ : Virtex®-5 JTAG Boundary-Scan Logic Access Circuit
CAPTURE_VIRTEX5	プリミティブ : Virtex®-5 Readback Register Capture Control
FRAME_ECC_VIRTEX5	プリミティブ : Virtex®-5 Configuration Frame Error Detection and Correction Circuitry
ICAP_VIRTEX5	プリミティブ : Internal Configuration Access Port
KEY_CLEAR	プリミティブ : Virtex-5 Configuration Encryption Key Erase
STARTUP_VIRTEX5	プリミティブ : Virtex®-5 Configuration Start-Up Sequence Interface
USR_ACCESS_VIRTEX5	プリミティブ : Virtex-5 User Access Register

## ギガビット I/O

デザイン エLEMENT	説明
GTP_DUAL	プリミティブ : Dual Gigabit Transceiver
GTX_DUAL	プリミティブ : Dual Gigabit Transceiver

## I/O コンポーネント

デザイン エLEMENT	説明
DCIRESET	プリミティブ : DCI State Machine Reset (After Configuration Has Been Completed)
IBUF	プリミティブ : Input Buffer
IBUFDS	プリミティブ : Differential Signaling Input Buffer
IBUFG	プリミティブ : Dedicated Input Clock Buffer
IBUFGDS	プリミティブ : Differential Signaling Dedicated Input Clock Buffer and Optional Delay
IDELAYCTRL	プリミティブ : IDELAY Tap Delay Value Control
IOBUF	プリミティブ : Bi-Directional Buffer
IOBUFDS	プリミティブ : 3-State Differential Signaling I/O Buffer with Active Low Output Enable
IODELAY	プリミティブ : Input and Output Fixed or Variable Delay Element
ISERDES	プリミティブ : Dedicated I/O Buffer Input Deserializer
ISERDES_NODELAY	プリミティブ : Input SERIAL/DESerializer
KEEPER	プリミティブ : KEEPER Symbol
OBUF	プリミティブ : Output Buffer
OBUFDS	プリミティブ : Differential Signaling Output Buffer
OBUFT	プリミティブ : 3-State Output Buffer with Active Low Output Enable
OBUFTDS	プリミティブ : 3-State Output Buffer with Differential Signaling, Active-Low Output Enable
OSERDES	プリミティブ : Dedicated IOB Output Serializer
PULLDOWN	プリミティブ : Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs
PULLUP	プリミティブ : Resistor to VCC for Input PADS, Open-Drain, and 3-State Outputs

## プロセッサ

デザイン エLEMENT	説明
PPC440	プリミティブ : Power PC 440 CPU Core

## RAM/ROM

デザイン エLEMENT	説明
FIFO18	プリミティブ : 18kb FIFO (First In, First Out) Block RAM Memory
FIFO18_36	プリミティブ : 36-bit Wide by 512 Deep 18kb FIFO (First In, First Out) Block RAM Memory
FIFO36	プリミティブ : 36kb FIFO (First In, First Out) Block RAM Memory
FIFO36_72	プリミティブ : 72-Bit Wide by 512 Deep 36kb FIFO (First In, First Out) Block RAM Memory with ECC (Error Detection and Correction Circuitry)
RAM128X1D	プリミティブ : 128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)
RAM256X1S	プリミティブ : 256-Deep by 1-Wide Random Access Memory (Select RAM)
RAM32M	プリミティブ : 32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)
RAM32X1D	プリミティブ : 32-Deep by 1-Wide Static Dual Port Synchronous RAM
RAM32X1S	プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM
RAM32X1S_1	プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAM32X2S	プリミティブ : 32-Deep by 2-Wide Static Synchronous RAM
RAM64M	プリミティブ : 64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)
RAM64X1D	プリミティブ : 64-Deep by 1-Wide Dual Port Static Synchronous RAM
RAM64X1S	プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM
RAM64X1S_1	プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAMB18	プリミティブ : 18K-bit Configurable Synchronous True Dual Port Block RAM
RAMB18SDP	プリミティブ : 36-bit by 512 Deep, 18kb Synchronous Simple Dual Port Block RAM
RAMB36	プリミティブ : 36kb Configurable Synchronous True Dual Port Block RAM
RAMB36SDP	プリミティブ : 72-bit by 512 Deep, 36kb Synchronous Simple Dual Port Block RAM with ECC (Error Correction Circuitry)
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
FDCPE	プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Preset and 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
FDRSE	プリミティブ : D Flip-Flop with Synchronous Reset and Set and Clock Enable
FDSE	プリミティブ : D Flip-Flop with Clock Enable and Synchronous Set
LDCE	プリミティブ : Transparent Data Latch with Asynchronous Clear and Gate Enable
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
LDCPE	プリミティブ : Transparent Data Latch with Asynchronous Clear and Preset 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

## シフト レジスタ LUT

デザイン エLEMENT	説明
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

## スライス/CLB プリミティブ

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

デザイン エLEMENT	説明
LUT2_D	マクロ : 2-Bit Look-Up Table with Dual Output
LUT2_L	マクロ : 2-Bit Look-Up Table with Local Output
LUT3	マクロ : 3-Bit Look-Up Table with General Output
LUT3_D	マクロ : 3-Bit Look-Up Table with Dual Output
LUT3_L	マクロ : 3-Bit Look-Up Table with Local Output
LUT4	マクロ : 4-Bit Look-Up-Table with General Output
LUT4_D	マクロ : 4-Bit Look-Up Table with Dual Output
LUT4_L	マクロ : 4-Bit Look-Up Table with Local Output
LUT5	プリミティブ : 5-Input Lookup Table with General Output
LUT5_D	プリミティブ : 5-Input Lookup Table with General and Local Outputs
LUT5_L	プリミティブ : 5-Input Lookup Table with Local Output
LUT6	プリミティブ : 6-Input Lookup Table with General Output
LUT6_2	プリミティブ : Six-input, 2-output, Look-Up Table
LUT6_D	プリミティブ : 6-Input Lookup Table with General and Local Outputs
LUT6_L	プリミティブ : 6-Input Lookup Table with Local Output
MUXF7	プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output
MUXF7_D	プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF7_L	プリミティブ : 2-to-1 look-up table Multiplexer with Local Output
MUXF8	プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output
MUXF8_D	プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF8_L	プリミティブ : 2-to-1 Look-Up Table Multiplexer with Local Output

## デザイン エLEMENT

---

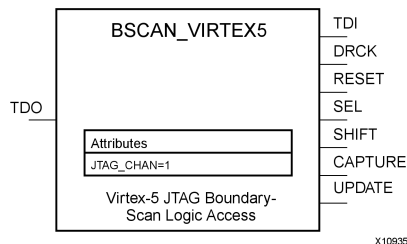
このセクションでは、Virtex®-5 デバイスで利用できるデザイン エLEMENTについて説明します。デザイン エLEMENTは、アルファベット順に並べられています。

各ライブラリ エLEMENTについて、次の情報を示します。

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

## BSCAN\_VIRTEX5

**プリミティブ：Virtex®-5 JTAG Boundary-Scan Logic Access Circuit**



### 概要

このデザイン エレメントを使用すると、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にアクセスできるようになり、内部実行デザインと FPGA の専用 JTAG ピン間の通信が可能になります。

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

**注記：** 各アーキテクチャのバウンダリ スキャンの詳細は、データシートを参照してください。

### ポートの説明

ポート名	方向	幅	機能
CAPTURE	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP コントローラーが CAPTURE-DR ステートになると High にアサートされます。
DRCK	出力	1	JTAG_CHAIN によって割り当てられた JTAG USER 命令が読み込まれ、JTAG TAP コントローラーが SHIFT-DR ステートまたは CAPTURE-DR ステートになると、FPGA への TCK ピンと同じ値を出力します。
RESET	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP コントローラーが TEST-LOGIC-RESET ステートになると High にアサートされます。
SEL	出力	1	JTAG 命令レジスタに USER 命令が読み込まれたことを示します。UPDATE-IR ステートになるとアクティブになり、新しい命令が読み込まれるまでアクティブのままになります。
SHIFT	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP コントローラーが SHIFT-DR ステートになると High にアサートされます。
TDI	出力	1	TDI ピンと同じ値を出力します。
UPDATE	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP コントローラが UPDATE-DR ステートになると High にアサートされます。
TDO	入力	1	USER 命令が読み込まれるとアクティブになり、外部 JTAG TDO ピンには、マクロの TDO1 ピンへのデータ入力の値が示されます。

### 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



## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- BSCAN_VIRTEX5: Boundary Scan primitive for connecting internal logic to
--                JTAG interface.
--                Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

BSCAN_VIRTEX5_inst : BSCAN_VIRTEX5
generic map (
    JTAG_CHAIN => 1) -- Value for USER command. Possible values: (1,2,3 or 4)
port map (
    CAPTURE => CAPTURE, -- CAPTURE output from TAP controller
    DRCK => DRCK,       -- Data register output for USER functions
    RESET => RESET,     -- Reset output from TAP controller
    SEL => SEL,         -- USER active output
    SHIFT => SHIFT,     -- SHIFT output from TAP controller
    TDI => TDI,         -- TDI output from TAP controller
    UPDATE => UPDATE,   -- UPDATE output from TAP controller
    TDO => TDO         -- Data input for USER function
);

-- End of BSCAN_VIRTEX5_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BSCAN_VIRTEX5: Boundary Scan primitive for connecting internal
//                logic to JTAG interface.
//                Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

BSCAN_VIRTEX5 #(
    .JTAG_CHAIN(1) // Value for USER command. Possible values: (1,2,3 or 4)
) BSCAN_VIRTEX5_inst (
    .CAPTURE(CAPTURE), // CAPTURE output from TAP controller
    .DRCK(DRCK),       // Data register output for USER function
    .RESET(RESET),     // Reset output from TAP controller
    .SEL(SEL),         // USER active output
    .SHIFT(SHIFT),     // SHIFT output from TAP controller
    .TDI(TDI),         // TDI output from TAP controller
    .UPDATE(UPDATE),   // UPDATE output from TAP controller
    .TDO(TDO)         // Data input for USER function
);

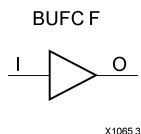
// End of BSCAN_VIRTEX5_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## BUFCF

### プリミティブ：Fast Connect Buffer



### 概要

このデザイン エレメントは、一部の専用ロジックと LUT の出力を別の LUT の入力に直接接続するために使用する、単一の高速結合バッファです。このバッファを使用すると、CLB パックも行われます。LUT は、4 つまで 1 つのグループとして接続できます。

### 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

### 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;

-- BUFCF: Fast connect buffer used to connect the outputs of the LUTs
--         and some dedicated logic directly to the input of another LUT.
--         Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

BUFCF_inst: BUFCF (
  port map (
    O => O, -- Connect to the output of a LUT
    I => I  -- Connect to the input of a LUT
  );

-- End of BUFCF_inst instantiation
```

### Verilog 記述（インスタンス化）

```
// BUFCF: Fast connect buffer used to connect the outputs of the LUTs
//         and some dedicated logic directly to the input of another LUT.
//         Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

BUFCF BUFCF_inst (
  .O(O), // Connect to the output of a LUT
  .I(I)  // Connect to the input of a LUT
);

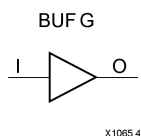
// End of BUFCF_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## BUFG

### プリミティブ：Global Clock Buffer



### 概要

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

### ポートの説明

ポート名	方向	幅	機能
I	入力	1	クロック バッファー入力
O	出力	1	クロック バッファー出力

### 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	不可
マクロのサポート	不可

### VHDL 記述（インスタンス化）

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

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

-- BUFG: Global Clock Buffer (source by an internal signal)
--      Virtex-5
--      Xilinx HDL Libraries Guide, version 14.5

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

-- End of BUFG_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFG: Global Clock Buffer (source by an internal signal)
//      All FPGAs
// Xilinx HDL Libraries Guide, version 14.5

BUFG BUFG_inst (
    .O(O),      // Clock buffer output
    .I(I)       // Clock buffer input
);

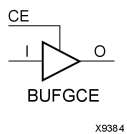
// End of BUFG_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## BUFGCE

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



### 概要

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

### 論理表

入力		出力
I	CE	O
X	0	0
I	1	I

### ポートの説明

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

### デザインの入力方法

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

## VHDL 記述 (インスタンス化)

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

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

-- BUFGCE: Global Clock Buffer with Clock Enable (active high)
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

BUFGCE_inst : BUFGCE
port map (
    O => O, -- Clock buffer output
    CE => CE, -- Clock enable input
    I => I -- Clock buffer input
);

-- End of BUFGCE_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGCE: Global Clock Buffer with Clock Enable (active high)
//          Virtex-5/6, Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 14.5

BUFGCE BUFGCE_inst (
    .O(O), // Clock buffer output
    .CE(CE), // Clock enable input
    .I(I) // Clock buffer input
);

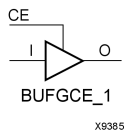
// End of BUFGCE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## BUFGCE\_1

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



### 概要

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

### 論理表

入力		出力
I	CE	O
X	0	1
I	1	I

### ポートの説明

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

### デザインの入力方法

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



## VHDL 記述 (インスタンス化)

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

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

-- BUFGCE_1: Global Clock Buffer with Clock Enable (active low)
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

BUFGCE_1_inst : BUFGCE_1
port map (
    O => O,  -- Clock buffer output
    CE => CE, -- Clock enable input
    I => I   -- Clock buffer input
);

-- End of BUFGCE_1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGCE_1: Global Clock Buffer with Clock Enable (active low)
//          Virtex-5/6, Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 14.5

BUFGCE_1 BUFGCE_1_inst (
    .O(O),    // Clock buffer output
    .CE(CE),  // Clock enable input
    .I(I)     // Clock buffer input
);

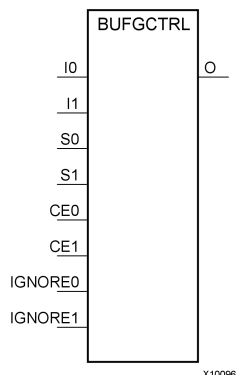
// End of BUFGCE_1_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート: DC 特性およびスイッチ特性』\(DS202\)](#)

## BUFGCTRL

### プリミティブ：Global Clock MUX Buffer



### 概要

BUFGCTRL は、2 つのクロック入力を持つ同期/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グローバル クロック バッファです。Virtex-4 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	機能
O	出力	1	クロック出力ピン
I0、I1	入力	1 (それぞれ)	クロック入力： I0：クロック入力ピン I1：クロック入力ピン
CE0、CE1	入力	1 (それぞれ)	クロック イネーブル入力。CE ピンは、各クロック入力ピンのクロック イネーブル入力で、クロック入力を選択するときに使用します。入力を選択するために CE ピンを使用する場合は、セットアップ/ホールド タイムを設定する必要があります。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。
S0、S1	入力	1 (それぞれ)	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。入力を選択するために S ピンを使用する場合は、セットアップおよびホールド タイム要件を満たす必要があります。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。
IGNORE0、IGNORE1	入力	1 (それぞれ)	クロック IGNORE 入力。IGNORE ピンは、BUFGCTRL により実行されるスイッチ アルゴリズムをバイパスする場合に使用します。

## 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
INIT_OUT	整数	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: Advanced Clock Primitive
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

BUFGCTRL_inst : BUFGCTRL
generic map (
  INIT_OUT => 0,           -- Initial value of 0 or 1 after configuration
  PRESELECT_I0 => FALSE, -- TRUE/FALSE set the I0 input after configuration
  PRESELECT_I1 => FALSE) -- TRUE/FALSE set the I1 input after configuration
port map (
  O => O,                 -- Clock MUX output
  CE0 => CE0,             -- Clock enable0 input
  CE1 => CE1,             -- Clock enable1 input
  I0 => I0,               -- Clock0 input
  I1 => I1,               -- Clock1 input
  IGNORE0 => IGNORE0,     -- Ignore clock select0 input
  IGNORE1 => IGNORE1,     -- Ignore clock select1 input
  S0 => S0,               -- Clock select0 input
  S1 => S1                -- Clock select1 input
);

-- End of BUFGCTRL_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFGCTRL: Advanced Clock MUX Primitive
//          Virtex-5/6
// Xilinx HDL Libraries Guide, version 14.5

BUFGCTRL #(
    .INIT_OUT(0), // Initial value of 0 or 1 after configuration
    .PRESELECT_I0("FALSE"), // "TRUE" or "FALSE" set the I0 input after configuration
    .PRESELECT_I1("FALSE") // "TRUE" or "FALSE" set the I1 input after configuration
) BUFGCTRL_inst (
    .O(0), // 1-bit output
    .CE0(CE0), // 1-bit clock enable 0
    .CE1(CE1), // 1-bit clock enable 1
    .I0(I0), // 1-bit clock 0 input
    .I1(I1), // 1-bit clock 1 input
    .IGNORE0(IGNORE0), // 1-bit ignore 0 input
    .IGNORE1(IGNORE1), // 1-bit ignore 1 input
    .S0(S0), // 1-bit select 0 input
    .S1(S1) // 1-bit select 1 input
);

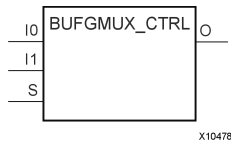
// End of BUFGCTRL_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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。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	機能
O	出力	1	クロック出力
I0	入力	1	2 つのクロック入力の 1 つ
I1	入力	1	2 つのクロック入力の 1 つ
S	入力	1	I0 (S=0) または I1 (S=1) クロック出力の選択

### 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;

-- BUFGMUX_CTRL: Global Clock Buffer 2-to-1 MUX
--               Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

BUFGMUX_CTRL_inst : BUFGMUX_CTRL
port map (
    O => O,      -- Clock MUX output
    I0 => I0,    -- Clock0 input
    I1 => I1,    -- Clock1 input
    S => S       -- Clock select input
);

-- End of BUFGMUX_CTRL_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFGMUX_CTRL: Global Clock Buffer 2-to-1 MUX
//                Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

BUFGMUX_CTRL BUFGMUX_CTRL_inst (
    .O(O),        // Clock MUX output
    .IO(I0),       // Clock0 input
    .I1(I1),       // Clock1 input
    .S(S)         // Clock select input
);

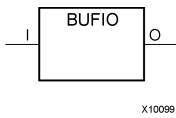
// End of BUFGMUX_CTRL_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## BUFIO

### プリミティブ：Local Clock Buffer for I/O



### 概要

このデザイン エLEMENTはクロック バッファです。単にクロック信号を入力し、出力します。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。これらのELEMENTを駆動できるのは、同じクロック領域内のクロック兼用 I/O のみです。BUFIO では、隣接する 2 つの I/O クロック ネット (最大 3 クロック領域まで) とリージョナル クロック バッファ (BUFR) を駆動できます。ただし、I/O クロック ネットワークは I/O 列までしか到達しないので、CLB やブロック RAM などのロジック リソースは駆動できません。

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	クロック出力
I	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

### 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: Clock in, clock out buffer
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

BUFIO_inst : BUFIO
port map (
    O => O,      -- Clock buffer output
    I => I       -- Clock buffer input
);

-- End of BUFIO_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFIO: Local Clock Buffer
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

BUFIO BUFIO_inst (
    .O(O),      // Clock buffer output
    .I(I)       // Clock buffer input
);

// End of BUFIO_inst instantiation
```

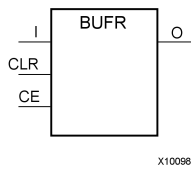
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## BUFR

### プリミティブ：Regional Clock Buffer for I/O and Logic Resources



## 概要

BUFR はクロック 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 は、同じ領域内のリージョナル クロック ネット 2 つと、隣接するクロック領域のクロック ネット 2 つを駆動できます (最大 3 クロック領域まで)。BUFIO と異なり、BUFR は I/O ロジックだけでなく、同じクロック領域および隣接する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、BUFIO の出力かローカル インターコネクトのどちらかで駆動されます。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。分周比は、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。中央列には BUFR は含まれません。

## ポートの説明

ポート名	方向	幅	機能
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、High になるとクロックが O ポートに出力されます。“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。BUFIO の出力またはローカル インターコネクトで駆動できます。
O	出力	1	クロック出力ポート。BUFR と同じクロック領域および 2 つの隣接するクロック領域 (最大 3 クロック領域) のクロック ネットを駆動できます。FPGA および IOB を駆動します。

## 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
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比を指定します。
SIM_DEVICE	文字列	"VIRTEX4"、 "VIRTEX5"、 "VIRTEX6"	"VIRTEX4"	BUFR の CE レイテンシを定義します。

## 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;

-- BUFR: Regional (Local) Clock Buffer /w Enable, Clear and Division Capabilities
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

BUFR_inst : BUFR
generic map (
    BUFR_DIVIDE => "BYPASS",    -- "BYPASS", "1", "2", "3", "4", "5", "6", "7", "8"
    SIM_DEVICE  => "VIRTEX5")   -- Specify target device, "VIRTEX4", "VIRTEX5", "VIRTEX6"
port map (
    O => O,      -- Clock buffer output
    CE => CE,    -- Clock enable input
    CLR => CLR,  -- Clock buffer reset input
    I => I       -- Clock buffer input
);

-- End of BUFR_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFR: Regional Clock Buffer w/ Enable, Clear and Division Capabilities
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

BUFR #(
    .BUFR_DIVIDE("BYPASS"), // "BYPASS", "1", "2", "3", "4", "5", "6", "7", "8"
    .SIM_DEVICE("VIRTEX5") // Specify target device, "VIRTEX4", "VIRTEX5", "VIRTEX6"
) BUFR_inst (
    .O(O),      // Clock buffer output
    .CE(CE),    // Clock enable input
    .CLR(CLR),  // Clock buffer reset input
    .I(I)       // Clock buffer input
);

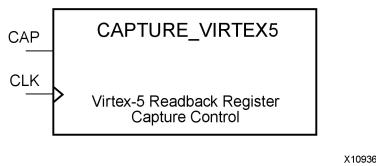
// End of BUFR_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## CAPTURE\_VIRTEX5

### プリミティブ：Virtex®-5 Readback Register Capture Control



### 概要

このデザイン エLEMENT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ELEMENT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。このELEMENTでは、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、各トリガー（CAP がアサートされているときの CLK の遷移）で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	機能
CAP	入力	1	リードバック キャプチャトリガー
CLK	入力	1	リードバック キャプチャ クロック

### 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

正しく動作するように、入力および出力をすべてデザインに接続します。

### 使用可能な属性

属性	データ型	値	デフォルト	説明
ONESHOT	ブール代数	TRUE、FALSE	TRUE	CAP 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;

-- CAPTURE_VIRTEX5: Register State Capture for Bitstream Readback
-- Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

CAPTURE_VIRTEX5_inst : CAPTURE_VIRTEX5
generic map (
    ONESHOT => TRUE) -- TRUE or FALSE
port map (
    CAP => CAP,      -- Capture input
    CLK => CLK       -- Clock input
);
-- End of CAPTURE_VIRTEX5_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// CAPTURE_VIRTEX5: Register State Capture for Bitstream Readback
// Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

CAPTURE_VIRTEX5 #(
    .ONESHOT("TRUE") // "TRUE" or "FALSE"
) CAPTURE_VIRTEX5_inst (
    .CAP(CAP),        // Capture input
    .CLK(CLK)         // Clock input
);

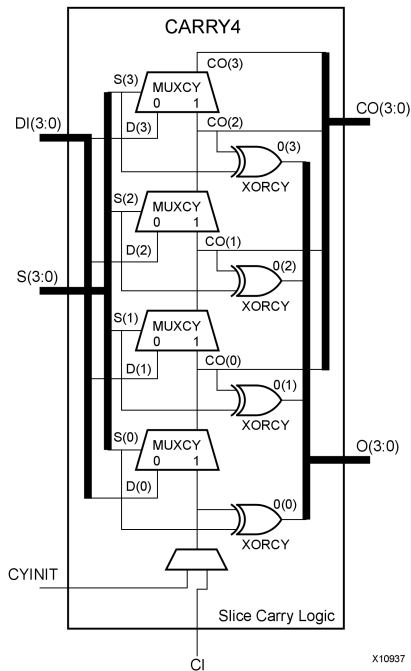
// End of CAPTURE_VIRTEX5_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## CARRY4

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



### 概要

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

### ポートの説明

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

## デザインの入力方法

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

## VHDL 記述 (インスタンス化)

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

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

-- CARRY4: Fast Carry Logic Component
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of CARRY4_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// CARRY4: Fast Carry Logic Component
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

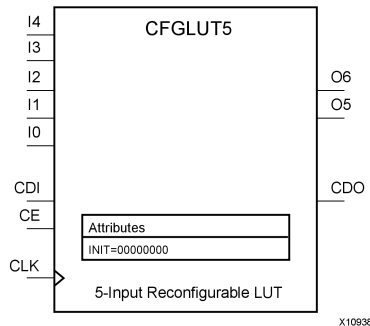
// End of CARRY4_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## CFGLUT5

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



### 概要

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

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

### ポートの説明

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

## デザインの入力方法

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

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

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

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

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

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

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

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

## 使用可能な属性

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



## VHDL 記述 (インスタンス化)

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

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

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

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

-- End of CFGLUT5_inst instantiation
```

## Verilog 記述 (インスタンス化)

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

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

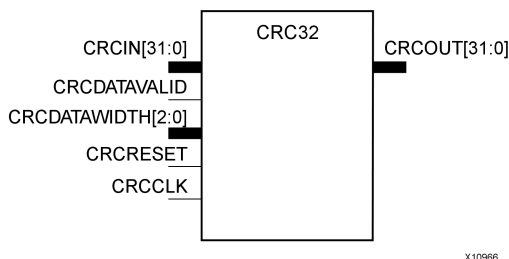
// End of CFGLUT5_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## CRC32

### プリミティブ : Cyclic Redundancy Check Calculator for 32 bits



## 概要

このデザイン エレメントは、フレームの内容を算出して、転送または格納前にフレームの最後に追加します。各 CRC ブロックでは、PCI EXPRESS®, ギガビット イーサネット、およびその他の汎用プロトコルに対して指定されている CRC-32 多項式を使用して、32 ビットの CRC が算出されます。32 ビットの CRC のプリミティブ CRC32 では、8、16、24、または 32 ビットの入力データを処理して 32 ビット CRC を生成できます。

## ポートの説明

ポート名	方向	幅	機能
CRCIN[31:0]	入力	32	CRC 入力データ。最大データ パス幅は 4 バイトです。
CRCDATAVALID	入力	1	CRCIN 入力のデータが有効であることを示します。 <ul style="list-style-type: none"> <li>1'b1 : データは有効です。</li> <li>1'b0 : データは無効です。</li> </ul> この信号をディアサートすると、ディアサートされているクロック サイクル間、CRC の値が保持されます。
CRCDATAWIDTH[2:0]	入力	3	有効な入力データ バイト数を示します。 <ul style="list-style-type: none"> <li>000 : CRCIN[31:24] の 8 ビット</li> <li>001 : CRCIN[31:16] の 16 ビット</li> <li>010 : CRCIN[31:8] の 24 ビット</li> <li>011 : CRCIN[31:0] の 32 ビット</li> </ul>
CRCRESET	入力	1	CRC レジスタの同期リセット。アサートされると、CRC ブロックが CRC_INIT の値に初期化されます。
CRCCLK	入力	1	CRC クロック
CRCOUT[31:0]	出力	32	32 ビット CRC 出力。バイトが反転されたビット反転 CRC 値で、直前のクロック サイクルの有効バイトおよび直前の CRC 値での CRC 計算に対応しています。CRCDATAVALID が 1 に設定されている必要があります。

## 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

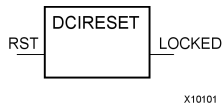
属性	データ型	値	デフォルト	説明
CRC_INIT[31:0]	16 進数	32 ビット値	0xFFFFFFFF	CRC の内部レジスタの初期値を設定します。LX30T および LX50T ES シリコンでは、値が 0xFFFFFFFF に固定されています。

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## DCIRESET

### プリミティブ：DCI State Machine Reset (After Configuration Has Been Completed)



### 概要

このデザイン エLEMENTは、コンフィギュレーション後に DCI ステート マシンをリセットするために使用します。

### ポートの説明

ポート名	方向	幅	機能
LOCKED	出力	1	DCIRESET LOCK ステータス出力
RST	入力	1	DCIRESET 非同期リセット入力

### 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

### 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: DCI reset component
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of DCIRESET_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// DCIRESET: Digital Controlled Impedance (DCI) Reset Component
//           Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

DCIRESET DCIRESET_inst (
    .LOCKED(LOCKED), // 1-bit DCI LOCKED Output
    .RST(RST)        // 1-bit DCI Reset Input
);

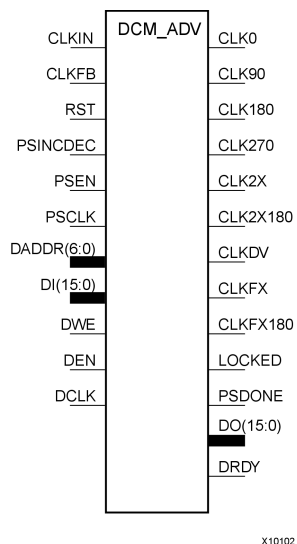
// End of DCIRESET_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## DCM\_ADV

### プリミティブ : Advanced Digital Clock Manager Circuit



### 概要

このデザイン エレメントは、位相制御機能および周波数合成制御機能を持つコンフィギュレーション/リコンフィギュレーション可能な DLL です。このコンポーネントは、システムで必要なさまざまなクロックを生成し、制御するため、多くの FPGA アプリケーションで使用されます。多くの FPGA アプリケーションで使用されます。ダイナミックリコンフィギュレーションが不要な場合は、DCM\_BASE または DCM\_PS コンポーネントを使用します。

### ポートの説明

ポート名	方向	幅	機能
クロック出力/入力			
CLK0	出力	1	CLKIN の有効周波数と同じ周波数のクロックを出力します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。CLKFB ピンを接続した場合、CLK0 は CLKIN の位相に揃えられます。
CLK90	出力	1	CLK0 と同じ周波数で位相を 90 度シフトしたクロックを出力します。
CLK180	出力	1	CLK0 と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLK270	出力	1	CLK0 と同じ周波数で位相を 270 度シフトしたクロックを出力します。
CLK2X	出力	1	デューティ サイクルが 50-50 に自動調整された、CLK0 と位相が同じで周波数が 2 倍のクロックを出力します。DCM がロック状態になるまでは、周波数が入力クロックの 1 倍で、デューティ サイクルが 25-75 のクロックが出力されます。これにより、DCM がソースクロックに対して正しいエッジでロック状態になります。
CLK2X180	出力	1	CLK2X と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKDV	出力	1	CLK0 と位相が同じで、CLKIN の有効周波数を分周したクロックを出力します。分周係数は、CLKDV_DIVIDE 属性で指定します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。

ポート名	方向	幅	機能
CLKFX	出力	1	<p>次の式で求められる周波数のクロックを出力します。</p> $\text{CLKFX の周波数} = (M/D) \times (\text{CLKIN の有効周波数})$ <p>この式で、M は CLKFX_MULTIPLY 属性で指定し、D は CLKFX_DIVIDE 属性で指定します。M および D の値の範囲、入力および出力の周波数範囲は、このアーキテクチャのデータシートを参照してください。CLKFX 出力の立ち上がりエッジは、フィードバックパス (CLKFB) が使用されるとき、CLK0、CLK2X、および CLKDV の立ち上がりエッジに揃えられます。M と D の値に公約数がない場合、位相は D 入力クロック サイクルごとに揃えられます。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。</p>
CLKFX180	出力	1	CLKFX と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKIN	入力	1	<p>DCM にソース クロックを供給します。CLKIN の周波数はこのアーキテクチャのデータシートで指定された範囲内にする必要があります。クロック入力信号は、次のいずれかのバッファから供給します。</p> <ul style="list-style-type: none"> <li>IBUFG：グローバル クロック入力バッファ。デバイス上で DCM と同じ側（上または下）にある IBUFG を使用すると、クロック入力バスが調整されます。</li> <li>BUFG/BUFGCTRL：内部グローバル クロック バッファ。専用グローバル配線を使用してデバイス上のどの DCM でも駆動できます。2 つの DCM を直列に接続する場合には、DCM の CLKIN ピンを駆動できます。</li> <li>IBUF：入力バッファ。IBUF で CLKIN 入力を駆動する場合、PAD から DCM 入力へのスキューは調整されず、ジッターが増加する可能性があります。このコンフィギュレーションは、使用しないでください。</li> </ul>
CLKFB	入力	1	<p>クロック出力の遅延を調整してクロック入力と位相を揃えるために参照するクロック信号です。DCM にフィードバックを供給するには、CLK0 出力のみを CLKFB 入力に接続しますが、内部フィードバックの場合は BUFG コンポーネントを介して、外部フィードバックの場合は OBUF および IBUFG を介します。CLK_FEEDBACK 属性を 1X に設定します。CLKFB ピンを接続すると、CLK0、CLKDV、および CLKFX が CLKIN の位相に揃えられます。CLKFB ピンを接続しない場合は、CLK_FEEDBACK を "NONE" に設定します。この場合、CLKFX および CLKFX180 出力は有効ですが、CLKIN の位相には揃えられません。</p>
ステータス出力/制御入力			
LOCKED	出力	1	位相アライメントが完了し、操作が開始可能であることを示す同期出力。
PSDONE	出力	1	ダイナミック CLKIN セレクト入力。High の場合は CLKIN1、Low の場合は CLKIN2 が選択されます。2 つのクロックを選択する必要がない場合は、この入力を 1 にします。

ポート名	方向	幅	機能
RST	入力	1	DCM 回路をリセットします。RST 信号は、アクティブ High の非同期リセットです。RST 信号をアサートすると、すべての DCM 出力 (LOCKED 信号、ステータス信号、出力クロック) がソースクロックの 4 サイクル以内に Low になります。リセットは非同期であるため、ディアサート中にクロックの最後のサイクルが短いパルスになったり、デューティサイクルが崩れたり、クロック間のスキューが調整される可能性があります。このため、デバイスをリコンフィギュレーションする場合や入力周波数を変更する場合は、RST ピンを使用する必要があります。RST 信号をディアサートすると、次の CLKIN サイクルに同期してクロックをロック状態にするプロセスが開始します。DCM がリセット後に正しくロックされるようにするには、CLKIN 信号が供給され、3 クロック サイクル以上安定するまで RST 信号をディアサートしておく必要があります。どのデザインでも、クロックが安定するまで DCM をリセットに保持する必要があります。コンフィギュレーションでは、GWE が解除されるまで DCM は自動的にリセット状態に保持されます。このクロックは、GSR が解放されると安定します。
PSCLK	入力	1	DCM 位相シフトのソースクロックを供給します。位相シフトクロック信号は、どのクロックソース (内部または外部) でも駆動できます。  PSCLK の周波数範囲は、PSCLK_FREQ_LF/HF で定義します (このアーキテクチャのデータシートを参照)。CLKOUT_PHASE_SHIFT 属性を "NONE" または "FIXED" に設定している場合は、この入力をグランドに接続する必要があります。
PSINCDEC	入力	1	PSINCDEC 入力は、PSCLK に同期しています。CLKOUT_PHASE_SHIFT が変数モードのいずれかに設定されているときに、位相シフト係数をインクリメント/デクリメントするために使用します。位相シフト係数をインクリメント/デクリメントすると、それに応じて出力クロックの位相がシフトします。PSINCDEC 信号が High の場合はインクリメント、Low の場合はデクリメントされます。CLKOUT_PHASE_SHIFT 属性を "NONE" または "FIXED" に設定している場合は、この入力をグランドに接続する必要があります。
PSEN	入力	1	PSEN 入力は、PSCLK に同期しています。CLKOUT_PHASE_SHIFT が変数モードに設定されているときに、この信号によって可変位相シフトを開始します。可変位相シフトを有効にするには、PSEN 信号を PSCLK の 1 サイクル分アクティブにする必要があります。位相の変更は、CLKIN の 100 周期分と PSCLK の 3 周期分を加えた時間以内に有効になり、PSDONE が High になることにより示されます。位相が変化する間、出力に突発的な変化やグリッチは発生しません。PSEN がイネーブルになってから PSDONE が High になるまでの間、DCM の出力クロックは元の位相からターゲットの位相に少しずつ移動していきます。PSDONE が High になったら、位相シフトは完了です。CLKOUT_PHASE_SHIFT 属性を "NONE" または "FIXED" に設定している場合は、この入力をグランドに接続する必要があります。
ダイナミック リコンフィギュレーション / DCM ステータス			
ダイナミック コンフィギュレーションの詳細は、該当デバイスの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。			



ポート名	方向	幅	機能
DO	出力	16	<p>ダイナミック リコンフィギュレーションを使用していない場合は DCM のステータス出力、使用している場合はリコンフィギュレーションのデータ出力になります。DCM ステータスが表示されている場合、次のマップが適用されます。</p> <ul style="list-style-type: none"> <li>DO[0]：位相シフト オーバーフロー</li> <li>DO[1]：CLKIN の停止</li> <li>DO[2]：CLKFX の停止</li> <li>DO[3]：CLKFB の停止</li> <li>DO[15:4]：割り当てなし</li> </ul>
DRDY	出力	1	ダイナミック リコンフィギュレーション機能が準備完了になったことを示します。
DI	入力	16	DI 入力バスは、ダイナミック リコンフィギュレーションのデータ入力です。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。
DADDR	入力	7	DADDR 入力バスは、ダイナミック リコンフィギュレーションのアドレス入力です。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。
DWE	入力	1	DI データの DADDR アドレスへの書き込みを制御するライト イネーブル信号です。使用しない場合は、Low に接続する必要があります。
DEN	入力	1	ダイナミック リコンフィギュレーション機能を使用するかどうかを制御する信号です。ダイナミック リコンフィギュレーションが使用されていないときに DO 出力バスに DCM ステータス信号を反映させるには、DEN を Low に設定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回路のソース クロックを供給します。DCLK には、CLKIN とは位相および周波数が非同期なクロックを使用できます。ダイナミック リコンフィギュレーション クロック信号は、どのクロック ソースでも駆動できます。DCLK の周波数範囲はこのアーキテクチャのデータシートに記載されています。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、この入力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	可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
CLK_FEEDBACK	文字列	"1X"、"NONE"	"1X"	クロック フィードバックを指定します。
CLKDV_DIVIDE	浮動小数点	1.5、2.0、2.5、3.0、3.5、4.0、4.5、5.0、5.5、6.0、6.5、7.0、7.5、8.0、9.0、10.0、11.0、12.0、13.0、14.0、15.0、16.0	2.0	CLKDLL、CLKDLLE、CLKDLLHF、DCM のクロック分周出力 CLKDV の分周比を指定します。
CLKFX_DIVIDE	整数	1 ～ 32	1	CLKFX 出力の分周比を指定します。
CLKFX_MULTIPLY	整数	2 ～ 32	4	CLKFX 出力の通倍比を指定します。

属性	データ型	値	デフォルト	説明
CLKIN_DIVIDE_BY_2	ブール代数	FALSE、TRUE	FALSE	DCM の入力クロック周波数の要件に合うように、必要に応じて入力クロック周波数を 2 で分周します。
CLKIN_PERIOD	浮動小数点	1.25 ~ 1000.00	10.0	入力クロックの周期を ns で指定します。
CLKOUT_PHASE_SHIFT	文字列	"NONE"、 "FIXED"、 "VARIABLE_POSITIVE"、 "VARIABLE_CENTER"、 "DIRECT"	"NONE"	位相シフト モードを指定します。
DCM_PERFORMANCE_MODE	文字列	"MAX_SPEED"、 "MAX_RANGE"	"MAX_SPEED"	DCM を最小ジッターで最大周波数のクロックを生成するよう最適化するか、位相シフト範囲が広い低周波数のクロックを生成するよう最適化するかを指定します。
DESKEW_ADJUST	文字列	"SOURCE_SYNCHRONOUS"、 "SYSTEM_SYNCHRONOUS"、 "0" ~ "15"	"SYSTEM_SYNCHRONOUS"	フィードバック パスの遅延の量を制御します。ソース同期のインターフェイスで使用する必要があります。
DFS_FREQUENCY_MODE	文字列	"LOW"、"HIGH"	"LOW"	周波数合成の周波数モードを指定します。
DLL_FREQUENCY_MODE	文字列	"LOW"、"HIGH"	"LOW"	DLL の周波数モードを指定します。
DUTY_CYCLE_CORRECTION	ブール代数	TRUE、FALSE	TRUE	CLK0、CLK90、CLK180、CLK270 の各出力のデューティ サイクルを修正します。
FACTORY_JF	16 進数	16 ビット値	F0F0	この属性は、DCM のジッター フィルター特性に影響します。ザイリックスからの指示なしにこのデフォルト値を変更しないでください。
PHASE_SHIFT	整数	-255 ~ 1023	0	位相シフト量を指定します。この値の範囲は CLKOUT_PHASE_SHIFT の指定によって異なります。
SIM_DEVICE	文字列	"VIRTEX4"、 "VIRTEX5"	"VIRTEX5"	デバイスを選択します。
STARTUP_WAIT	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、DCM がロック状態になるまでコンフィギュレーション スタートアップ シーケンスの指定したサイクルで待機します。

## 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;

-- DCM_ADV: Digital Clock Manager Circuit
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

DCM_ADV_inst : DCM_ADV
generic map (
    CLKDV_DIVIDE => 2.0,    -- Divide by: 1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
                           --      7.0,7.5,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0 or 16.0
    CLKFX_DIVIDE => 1,      -- Can be any integer from 1 to 32
    CLKFX_MULTIPLY => 4,    -- Can be any integer from 2 to 32
    CLKIN_DIVIDE_BY_2 => FALSE, -- TRUE/FALSE to enable CLKIN divide by two feature
    CLKIN_PERIOD => 10.0,   -- Specify period of input clock in ns from 1.25 to 1000.00
    CLKOUT_PHASE_SHIFT => "NONE", -- Specify phase shift mode of NONE, FIXED,
                                   --      VARIABLE POSITIVE, VARIABLE CENTER or DIRECT
    CLK_FEEDBACK => "1X",   -- Specify clock feedback of NONE or 1X
    DCM_PERFORMANCE_MODE => "MAX_SPEED", -- Can be MAX_SPEED or MAX_RANGE
    DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS", -- SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                   --      an integer from 0 to 15
    DFS_FREQUENCY_MODE => "LOW", -- HIGH or LOW frequency mode for frequency synthesis
    DLL_FREQUENCY_MODE => "LOW", -- LOW, HIGH, or HIGH_SER frequency mode for DLL
    DUTY_CYCLE_CORRECTION => TRUE, -- Duty cycle correction, TRUE or FALSE
    FACTORY_JF => X"F0F0", -- FACTORY JF Values Suggested to be set to X"F0F0"
    PHASE_SHIFT => 0, -- Amount of fixed phase shift from -255 to 1023
    SIM_DEVICE => "VIRTEX5", -- Set target device, "VIRTEX4" or "VIRTEX5"
    STARTUP_WAIT => FALSE) -- Delay configuration DONE until DCM LOCK, TRUE/FALSE
port map (
    CLK0 => CLK0,          -- 0 degree DCM CLK output
    CLK180 => CLK180,      -- 180 degree DCM CLK output
    CLK270 => CLK270,      -- 270 degree DCM CLK output
    CLK2X => CLK2X,        -- 2X DCM CLK output
    CLK2X180 => CLK2X180, -- 2X, 180 degree DCM CLK out
    CLK90 => CLK90,        -- 90 degree DCM CLK output
    CLKDV => CLKDV,        -- Divided DCM CLK out (CLKDV_DIVIDE)
    CLKFX => CLKFX,        -- DCM CLK synthesis out (M/D)
    CLKFX180 => CLKFX180, -- 180 degree CLK synthesis out
    DO => DO,              -- 16-bit data output for Dynamic Reconfiguration Port (DRP)
    DRDY => DRDY,          -- Ready output signal from the DRP
    LOCKED => LOCKED,      -- DCM LOCK status output
    PSDONE => PSDONE,      -- Dynamic phase adjust done output
    CLKFB => CLKFB,        -- DCM clock feedback
    CLKIN => CLKIN,        -- Clock input (from IBUFG, BUFG or DCM)
    DADDR => DADDR,        -- 7-bit address for the DRP
    DCLK => DCLK,          -- Clock for the DRP
    DEN => DEN,            -- Enable input for the DRP
    DI => DI,              -- 16-bit data input for the DRP
    DWE => DWE,            -- Active high allows for writing configuration memory
    PSCLK => PSCLK,        -- Dynamic phase adjust clock input
    PSEN => PSEN,          -- Dynamic phase adjust enable input
    PSINCDEC => PSINCDEC, -- Dynamic phase adjust increment/decrement
    RST => RST             -- DCM asynchronous reset input
);

-- End of DCM_ADV_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// DCM_ADV: Digital Clock Manager Circuit
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

DCM_ADV #(
    .CLKDV_DIVIDE(2.0), // Divide by: 1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
                        //      7.0,7.5,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0 or 16.0
    .CLKFX_DIVIDE(1),  // Can be any integer from 1 to 32
    .CLKFX_MULTIPLY(4), // Can be any integer from 2 to 32
    .CLKIN_DIVIDE_BY_2("FALSE"), // TRUE/FALSE to enable CLKIN divide by two feature
    .CLKIN_PERIOD(10.0), // Specify period of input clock in ns from 1.25 to 1000.00
    .CLKOUT_PHASE_SHIFT("NONE"), // Specify phase shift mode of NONE, FIXED,
                                // VARIABLE_POSITIVE, VARIABLE_CENTER or DIRECT
    .CLK_FEEDBACK("1X"), // Specify clock feedback of NONE or 1X
    .DCM_PERFORMANCE_MODE("MAX_SPEED"), // Can be MAX_SPEED or MAX_RANGE
    .DESKEW_ADJUST("SYSTEM_SYNCHRONOUS"), // SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                // an integer from 0 to 15
    .DFS_FREQUENCY_MODE("LOW"), // HIGH or LOW frequency mode for frequency synthesis
    .DLL_FREQUENCY_MODE("LOW"), // LOW, HIGH, or HIGH_SER frequency mode for DLL
    .DUTY_CYCLE_CORRECTION("TRUE"), // Duty cycle correction, "TRUE"/"FALSE"
    .FACTORY_JF(16'hf0f0), // FACTORY JF value suggested to be set to 16'hf0f0
    .PHASE_SHIFT(0), // Amount of fixed phase shift from -255 to 1023
    .SIM_DEVICE("VIRTEX5"), // Set target device, "VIRTEX4" or "VIRTEX5"
    .STARTUP_WAIT("FALSE") // Delay configuration DONE until DCM LOCK, "TRUE"/"FALSE"
) DCM_ADV_inst (
    .CLK0(CLK0), // 0 degree DCM CLK output
    .CLK180(CLK180), // 180 degree DCM CLK output
    .CLK270(CLK270), // 270 degree DCM CLK output
    .CLK2X(CLK2X), // 2X DCM CLK output
    .CLK2X180(CLK2X180), // 2X, 180 degree DCM CLK out
    .CLK90(CLK90), // 90 degree DCM CLK output
    .CLKDV(CLKDV), // Divided DCM CLK out (CLKDV_DIVIDE)
    .CLKFX(CLKFX), // DCM CLK synthesis out (M/D)
    .CLKFX180(CLKFX180), // 180 degree CLK synthesis out
    .DO(DO), // 16-bit data output for Dynamic Reconfiguration Port (DRP)
    .DRDY(DRDY), // Ready output signal from the DRP
    .LOCKED(LOCKED), // DCM LOCK status output
    .PSDONE(PSDONE), // Dynamic phase adjust done output
    .CLKFB(CLKFB), // DCM clock feedback
    .CLKIN(CLKIN), // Clock input (from IBUFG, BUFG or DCM)
    .DADDR(DADDR), // 7-bit address for the DRP
    .DCLK(DCLK), // Clock for the DRP
    .DEN(DEN), // Enable input for the DRP
    .DI(DI), // 16-bit data input for the DRP
    .DWE(DWE), // Active high allows for writing configuration memory
    .PSCLK(PSCLK), // Dynamic phase adjust clock input
    .PSEN(PSEN), // Dynamic phase adjust enable input
    .PSINCDEC(PSINCDEC), // Dynamic phase adjust increment/decrement
    .RST(RST) // DCM asynchronous reset input
);

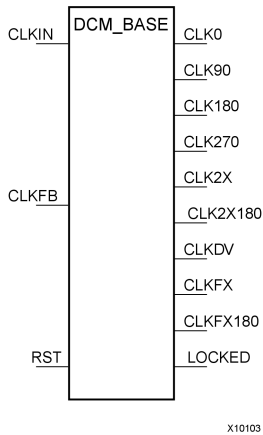
// End of DCM_ADV_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート: DC 特性およびスイッチ特性』\(DS202\)](#)

## DCM\_BASE

### プリミティブ：Base Digital Clock Manager Circuit



### 概要

このデザイン エLEMENTは、位相制御機能および周波数合成制御機能を持つコンフィギュレーション可能な DLL です。このコンポーネントは、システムに必要なさまざまなクロックを生成し、制御するため、多くの FPGA アプリケーションで使用されます。ダイナミック リコンフィギュレーションが必要な場合は DCM\_ADV コンポーネントを使用し、ダイナミック 位相シフトが必要な場合は DCM\_PS コンポーネントを使用します。

### ポートの説明

ポート名	方向	幅	機能
クロック出力/入力			
CLK0	出力	1	CLKIN の有効周波数と同じ周波数のクロックを出力します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。CLKFB ピンを接続した場合、CLK0 は CLKIN の位相に揃えられます。
CLK90	出力	1	CLK0 と同じ周波数で位相を 90 度シフトしたクロックを出力します。
CLK180	出力	1	CLK0 と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLK270	出力	1	CLK0 と同じ周波数で位相を 270 度シフトしたクロックを出力します。
CLK2X	出力	1	デューティ サイクルが 50-50 に自動調整された、CLK0 と位相が同じで周波数が 2 倍のクロックを出力します。DCM がロック状態になるまでは、周波数が入力クロックの 1 倍で、デューティ サイクルが 25-75 のクロックが CLK2X に出力されます。これにより、DCM がソース クロックに対して正しいエッジでロック状態になります。
CLK2X180	出力	1	CLK2X と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKDV	出力	1	CLK0 と位相が同じで、CLKIN の有効周波数を分周したクロックが出力されます。分周する係数は、CLKDV_DIVIDE 属性で指定します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。

ポート名	方向	幅	機能
CLKFX	出力	1	<p>次の式で求められる周波数のクロックを出力します。</p> $\text{CLKFX の周波数} = (M/D) \times (\text{CLKIN の有効周波数})$ <p>この式で、M は CLKFX_MULTIPLY 属性で指定し、D は CLKFX_DIVIDE 属性で指定します。M および D の値の範囲、入力および出力の周波数範囲は、このアーキテクチャのデータシートを参照してください。CLKFX 出力の立ち上がりエッジは、フィードバック パス (CLKFB) が使用されるとき、CLK0、CLK2X、および CLKDV の立ち上がりエッジに揃えられます。M と D の値に公約数がない場合、位相は D 入力クロック サイクルごとに揃えられます。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。</p>
CLKFX180	出力	1	CLKFX と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKIN	入力	1	<p>DCM にソース クロックを供給します。CLKIN の周波数はこのアーキテクチャのデータシートで指定された範囲内にする必要があります。クロック入力信号は、次のいずれかのバッファから供給します。</p> <ul style="list-style-type: none"> <li>IBUFG : グローバル クロック入力バッファ。デバイス上で DCM と同じ側 (上または下) にある IBUFG を使用すると、クロック入力パスが調整されます。</li> <li>BUFG/BUFGCTRL : 内部グローバル クロック バッファ。専用グローバル配線を使用してデバイス上のどの DCM でも駆動できます。2 つの DCM を直列に接続する場合には、DCM の CLKIN ピンを駆動できます。</li> <li>IBUF : 入力バッファ。IBUF で CLKIN 入力を駆動する場合、PAD から DCM 入力へのスキューは調整されず、ジッターが増加する可能性があります。このコンフィギュレーションは、使用しないでください。</li> </ul>
CLKFB	入力	1	<p>クロック出力の遅延を調整してクロック入力と位相を揃えるために参照するクロック信号です。DCM にフィードバックを供給するには、CLK0 出力のみを CLKFB 入力に接続しますが、内部フィードバックの場合は BUFG コンポーネントを介して、外部フィードバックの場合は OBUF および IBUFG を介します。CLK_FEEDBACK 属性を 1X に設定します。CLKFB ピンを接続すると、CLK0、CLKDV、および CLKFX が CLKIN の位相に揃えられます。CLKFB ピンを接続しない場合は、CLK_FEEDBACK を "NONE" に設定します。この場合、CLKFX および CLKFX180 出力は有効ですが、CLKIN の位相には揃えられません。</p>
ステータス出力/制御入力			
LOCKED	出力	1	位相アライメントが完了し、操作が開始可能であることを示す同期出力
RST	入力	1	<p>DCM 回路をリセットします。RST 信号は、アクティブ High の非同期リセットです。RST 信号をアサートすると、すべての DCM 出力 (LOCKED 信号、ステータス信号、出力クロック) がソース クロックの 4 サイクル以内に Low になります。リセットは非同期であるため、ディアサート中にクロックの最後のサイクルが短いパルスになったり、デューティ サイクルが崩れたり、クロック間のスキューが調整される可能性があります。このため、デバイスをリコンフィギュレーションする場合や入力周波数を変更する場合は、RST ピンを使用する必要があります。RST 信号をディアサートすると、次の CLKIN サイクルに同期してクロックをロック状態にするプロセスが開始します。DCM がリセット後に正しくロックされるようにするには、CLKIN 信号が供給され、3 クロック サイクル以上安定するまで RST 信号をディアサートしておく必要があります。どのデザインでも、クロックが安定するまで DCM をリセットに保持する必</p>

ポート名	方向	幅	機能
			必要があります。コンフィギュレーションでは、GWE が解除されるまで DCM は自動的にリセット状態に保持されます。このクロックは、GSR が解放されとる安定します。

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
CLK_FEEDBACK	文字列	"1X"、"2X"、 "NONE"	"1X"	DCM へのフィードバック入力を指定します (CLK0 または CLK2X)。
CLKDV_DIVIDE	浮動小数点	1.5、2.0、2.5、 3.0、3.5、4.0、4.5、 5.0、5.5、6.0、6.5、 7.0、7.5、8.0、9.0、 10.0、11.0、12.0、 13.0、14.0、15.0、 16.0	2.0	CLKDLL、CLKDLLE、CLKDLLHF、DCM のクロック分周器 (CLKDV 出力) の分周比を指定します。
CLKFX_DIVIDE	整数	1 ～ 32	1	CLKFX 出力の分周比を指定します。
CLKFX_MULTIPLY	整数	2 ～ 32	4	CLKFX 出力の通倍比を指定します。
CLKIN_DIVIDE_BY_2	ブール代数	FALSE、TRUE	FALSE	DCM の入力クロック周波数の要件に合うように、必要に応じて入力クロック周波数を 2 で分周します。
CLKIN_PERIOD	浮動小数点	1.25 ～ 1000.00	10.0	入力クロックの周期を ns で指定します。
CLKOUT_PHASE_SHIFT	文字列	"NONE"、 "FIXED"、 "VARIABLE_ POSITIVE"、 "VARIABLE_ CENTER"、 "DIRECT"	"NONE"	位相シフト モードを指定します。
DCM_PERFORMANCE_MODE	文字列	"MAX_SPEED"、 "MAX_RANGE"	"MAX_SPEED"	DCM を最小ジッターで最大周波数のクロックを生成するよう最適化するか、位相シフト範囲が広い低周波数のクロックを生成するよう最適化するかを指定します。
DESKEW_ADJUST	文字列	"SOURCE_ SYNCHRONOUS"、 "SYSTEM_ SYNCHRONOUS"、 "0" ～ "15"	"SYSTEM_ SYNCHRONOUS"	フィードバック パスの遅延量を制御します。ソース同期のインターフェイスで使用する必要があります。
DFS_FREQUENCY_MODE	文字列	"LOW"、"HIGH"	"LOW"	周波数合成の周波数モードを指定します。
DLL_FREQUENCY_MODE	文字列	"LOW"、"HIGH"	"LOW"	DLL の周波数モードを指定します。

属性	データ型	値	デフォルト	説明
DUTY_CYCLE_CORRECTION	ブール代数	TRUE、FALSE	TRUE	CLK0、CLK90、CLK180、CLK270 の各出力のデューティ サイクルを修正します。
FACTORY_JF	16 進数	16 ビット値	F0F0	この属性は、DCM のジッター フィルター特性に影響します。ザイリンクスからの指示なしにこのデフォルト値を変更しないでください。
PHASE_SHIFT	整数	-255 ~ 1023	0	位相シフト量を指定します。この値の範囲は CLKOUT_PHASE_SHIFT の指定によって異なります。
STARTUP_WAIT	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、DCM がロック状態になるまでコンフィギュレーション スタートアップ シーケンスの指定したサイクルで待機します。

## 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;

-- DCM_BASE: Base Digital Clock Manager Circuit
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

DCM_BASE_inst : DCM_BASE
generic map (
    CLKDV_DIVIDE => 2.0, -- Divide by: 1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
                        --          7.0,7.5,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0 or 16.0
    CLKFX_DIVIDE => 1,  -- Can be any integer from 1 to 32
    CLKFX_MULTIPLY => 4, -- Can be any integer from 2 to 32
    CLKIN_DIVIDE_BY_2 => FALSE, -- TRUE/FALSE to enable CLKIN divide by two feature
    CLKIN_PERIOD => 10.0, -- Specify period of input clock in ns from 1.25 to 1000.00
    CLKOUT_PHASE_SHIFT => "NONE", -- Specify phase shift mode of NONE or FIXED
    CLK_FEEDBACK => "1X", -- Specify clock feedback of NONE or 1X
    DCM_PERFORMANCE_MODE => "MAX_SPEED", -- Can be MAX_SPEED or MAX_RANGE
    DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS", -- SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                        --          an integer from 0 to 15
    DFS_FREQUENCY_MODE => "LOW", -- LOW or HIGH frequency mode for frequency synthesis
    DLL_FREQUENCY_MODE => "LOW", -- LOW, HIGH, or HIGH_SER frequency mode for DLL
    DUTY_CYCLE_CORRECTION => TRUE, -- Duty cycle correction, TRUE or FALSE
    FACTORY_JF => X"F0F0", -- FACTORY JF Values Suggested to be set to X"F0F0"
    PHASE_SHIFT => 0, -- Amount of fixed phase shift from -255 to 1023
    STARTUP_WAIT => FALSE) -- Delay configuration DONE until DCM LOCK, TRUE/FALSE
port map (
    CLK0 => CLK0, -- 0 degree DCM CLK ouptput
    CLK180 => CLK180, -- 180 degree DCM CLK output
    CLK270 => CLK270, -- 270 degree DCM CLK output
    CLK2X => CLK2X, -- 2X DCM CLK output
    CLK2X180 => CLK2X180, -- 2X, 180 degree DCM CLK out
    CLK90 => CLK90, -- 90 degree DCM CLK output
    CLKDV => CLKDV, -- Divided DCM CLK out (CLKDV_DIVIDE)
    CLKFX => CLKFX, -- DCM CLK synthesis out (M/D)
    CLKFX180 => CLKFX180, -- 180 degree CLK synthesis out
    LOCKED => LOCKED, -- DCM LOCK status output
    CLKFB => CLKFB, -- DCM clock feedback
    CLKIN => CLKIN, -- Clock input (from IBUFG, BUFG or DCM)
    RST => RST -- DCM asynchronous reset input
);

-- End of DCM_BASE_inst instantiation

```



## Verilog 記述 (インスタンス化)

```
// DCM_BASE: Base Digital Clock Manager Circuit
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

DCM_BASE #(
    .CLKDV_DIVIDE(2.0), // Divide by: 1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
                        // 7.0,7.5,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0 or 16.0
    .CLKFX_DIVIDE(1), // Can be any integer from 1 to 32
    .CLKFX_MULTIPLY(4), // Can be any integer from 2 to 32
    .CLKIN_DIVIDE_BY_2("FALSE"), // TRUE/FALSE to enable CLKIN divide by two feature
    .CLKIN_PERIOD(10.0), // Specify period of input clock in ns from 1.25 to 1000.00
    .CLKOUT_PHASE_SHIFT("NONE"), // Specify phase shift mode of NONE or FIXED
    .CLK_FEEDBACK("1X"), // Specify clock feedback of NONE or 1X
    .DCM_PERFORMANCE_MODE("MAX_SPEED"), // Can be MAX_SPEED or MAX_RANGE
    .DESKEW_ADJUST("SYSTEM_SYNCHRONOUS"), // SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                         // an integer from 0 to 15
    .DFS_FREQUENCY_MODE("LOW"), // LOW or HIGH frequency mode for frequency synthesis
    .DLL_FREQUENCY_MODE("LOW"), // LOW, HIGH, or HIGH_SER frequency mode for DLL
    .DUTY_CYCLE_CORRECTION("TRUE"), // Duty cycle correction, TRUE or FALSE
    .FACTORY_JF(16'hf0f0), // FACTORY JF value suggested to be set to 16'hf0f0
    .PHASE_SHIFT(0), // Amount of fixed phase shift from -255 to 1023
    .STARTUP_WAIT("FALSE") // Delay configuration DONE until DCM LOCK, TRUE/FALSE
) DCM_BASE_inst (
    .CLK0(CLK0), // 0 degree DCM CLK output
    .CLK180(CLK180), // 180 degree DCM CLK output
    .CLK270(CLK270), // 270 degree DCM CLK output
    .CLK2X(CLK2X), // 2X DCM CLK output
    .CLK2X180(CLK2X180), // 2X, 180 degree DCM CLK out
    .CLK90(CLK90), // 90 degree DCM CLK output
    .CLKDV(CLKDV), // Divided DCM CLK out (CLKDV_DIVIDE)
    .CLKFX(CLKFX), // DCM CLK synthesis out (M/D)
    .CLKFX180(CLKFX180), // 180 degree CLK synthesis out
    .LOCKED(LOCKED), // DCM LOCK status output
    .CLKFB(CLKFB), // DCM clock feedback
    .CLKIN(CLKIN), // Clock input (from IBUFG, BUFG or DCM)
    .RST(RST) // DCM asynchronous reset input
);

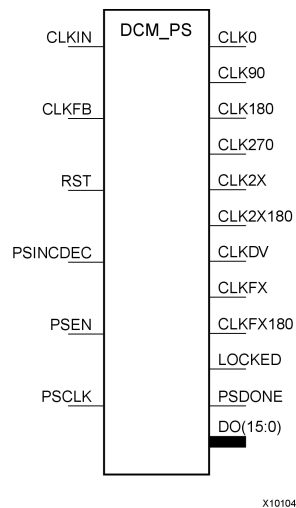
// End of DCM_BASE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## DCM\_PS

### プリミティブ：Digital Clock Manager with Basic and Phase Shift Features



## 概要

このデザイン エLEMENTは、位相制御機能および周波数合成制御機能を持つコンフィギュレーション可能な DLL です。このコンポーネントは、システムに必要なさまざまなクロックを生成し、制御するため、多くの FPGA アプリケーションで使用されます。ダイナミック リコンフィギュレーションが必要な場合は DCM\_ADV を使用し、ダイナミック位相シフトが不要な場合は DCM\_BASE を使用します。

## ポートの説明

ポート名	方向	幅	機能
クロック出力/入力			
CLK0	出力	1	CLKIN の有効周波数と同じ周波数のクロックを出力します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。CLKFB ピンを接続した場合、CLK0 は CLKIN の位相に揃えられます。
CLK90	出力	1	CLK0 と同じ周波数で位相を 90 度シフトしたクロックを出力します。
CLK180	出力	1	CLK0 と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLK270	出力	1	CLK0 と同じ周波数で位相を 270 度シフトしたクロックを出力します。
CLK2X	出力	1	デューティ サイクルが 50-50 に自動調整された、CLK0 と位相が同じで周波数が 2 倍のクロックを出力します。DCM がロック状態になるまでは、周波数が入力クロックの 1 倍で、デューティ サイクルが 25-75 のクロックが CLK2X に出力されます。これにより、DCM がソース クロックに対して正しいエッジでロック状態になります。
CLK2X180	出力	1	CLK2X と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKDV	出力	1	CLK0 と位相が同じで、CLKIN の有効周波数を分周したクロックが出力されます。分周する係数は、CLKDV_DIVIDE 属性で指定します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。

ポート名	方向	幅	機能
CLKFX	出力	1	<p>次の式で求められる周波数のクロックを出力します。</p> $\text{CLKFX の周波数} = (M/D) \times (\text{CLKIN の有効周波数})$ <p>この式で、M は CLKFX_MULTIPLY 属性で指定し、D は CLKFX_DIVIDE 属性で指定します。M および D の値の範囲、入力および出力の周波数範囲は、データシートを参照してください。CLKFX 出力の立ち上がりエッジは、フィードバックパス (CLKFB) が使用されるとき、CLK0、CLK2X、および CLKDV の立ち上がりエッジに揃えられます。M と D の値に公約数がない場合、位相は D 入力クロック サイクルごとに揃えられます。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。</p>
CLKFX180	出力	1	CLKFX と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKIN	入力	1	<p>DCM にソース クロックを供給します。CLKIN の周波数はデータシートで指定された範囲内にする必要があります。クロック入力信号は、次のいずれかのバッファから供給します。</p> <ul style="list-style-type: none"> <li>IBUFG : グローバル クロック入力バッファ。デバイス上で DCM と同じ側 (上または下) にある IBUFG を使用すると、クロック入力パスが調整されます。</li> <li>BUFG/BUFGCTRL : 内部グローバル クロック バッファ。専用グローバル配線を使用してデバイス上のどの DCM でも駆動できます。2 つの DCM を直列に接続する場合には、DCM の CLKIN ピンを駆動できます。</li> <li>IBUF : 入力バッファ。IBUF で CLKIN 入力を駆動する場合、PAD から DCM 入力へのスキューは調整されず、ジッタが増加する可能性があります。このコンフィギュレーションは、使用しないでください。</li> </ul>
CLKFB	入力	1	<p>クロック出力の遅延を調整してクロック入力と位相を揃えるために参照するクロック信号です。DCM にフィードバックを供給するには、CLK0 出力のみを CLKFB 入力に接続しますが、内部フィードバックの場合は BUFG コンポーネントを介して、外部フィードバックの場合は OBUF および IBUFG を介します。CLK_FEEDBACK 属性を 1X に設定します。CLKFB ピンを接続すると、CLK0、CLKDV、および CLKFX が CLKIN の位相に揃えられます。CLKFB ピンを接続しない場合は、CLK_FEEDBACK を "NONE" に設定します。この場合、CLKFX および CLKFX180 出力は有効ですが、CLKIN の位相には揃えられません。</p>
ステータス出力/制御入力			
LOCKED	出力	1	位相アライメントが完了し、操作が開始可能であることを示す同期出力
PSDONE	出力	1	<p>ダイナミック CLKIN セレクト入力。High の場合は CLKIN1、Low の場合は CLKIN2 が選択されます。2 つのクロックを選択する必要がない場合は、この入力を 1 にします。</p>

ポート名	方向	幅	機能
RST	入力	1	DCM 回路をリセットします。RST 信号は、アクティブ High の非同期リセットです。RST 信号をアサートすると、すべての DCM 出力 (LOCKED 信号、ステータス信号、出力クロック) がソースクロックの 4 サイクル以内に Low になります。リセットは非同期であるため、ディアサート中にクロックの最後のサイクルが短いパルスになったり、デューティサイクルが崩れたり、クロック間のスキューが調整されなる可能性があります。このため、デバイスをリコンフィギュレーションする場合や入力周波数を変更する場合は、RST ピンを使用する必要があります。RST 信号をディアサートすると、次の CLKIN サイクルに同期してクロックをロック状態にするプロセスが開始します。DCM がリセット後に正しくロックされるようにするには、CLKIN 信号が供給され、3 クロックサイクル以上安定するまで RST 信号をディアサートしておく必要があります。どのデザインでも、クロックが安定するまで DCM をリセットに保持する必要があります。コンフィギュレーションでは、GWE が解除されるまで DCM は自動的にリセット状態に保持されます。このクロックは、GSR が解放されると安定します。
PSCLK	入力	1	DCM 位相シフトのソースクロックを供給します。位相シフトクロック信号は、どのクロックソース (内部または外部) でも駆動できます。  PSCLK の周波数範囲は、PSCLK_FREQ_LF/HF で定義します (データシートを参照)。CLKOUT_PHASE_SHIFT 属性を "NONE" または "FIXED" に設定している場合は、この入力をグラウンドに接続する必要があります。
PSINCDEC	入力	1	PSINCDEC 入力は、PSCLK に同期しています。CLKOUT_PHASE_SHIFT が変数モードのいずれかに設定されているときに、位相シフト係数をインクリメント/デクリメントするために使用します。位相シフト係数をインクリメント/デクリメントすると、それに応じて出力クロックの位相がシフトします。PSINCDEC 信号が High の場合はインクリメント、Low の場合はデクリメントされます。CLKOUT_PHASE_SHIFT 属性を "NONE" または "FIXED" に設定している場合は、この入力をグラウンドに接続する必要があります。
PSEN	入力	1	PSEN 入力は、PSCLK に同期しています。CLKOUT_PHASE_SHIFT が変数モードに設定されているときに、この信号によって可変位相シフトを開始します。可変位相シフトを有効にするには、PSEN 信号を PSCLK の 1 サイクル分アクティブにする必要があります。位相の変更は、CLKIN の 100 周期分と PSCLK の 3 周期分を加えた時間以内に有効になり、PSDONE が High になることにより示されます。位相が変化する間、出力に突発的な変化やグリッチは発生しません。PSEN がイネーブルになってから PSDONE が High になるまでの間、DCM の出力クロックは元の位相からターゲットの位相に少しずつ移動していきます。PSDONE が High になったら、位相シフトは完了です。CLKOUT_PHASE_SHIFT 属性を "NONE" または "FIXED" に設定している場合は、この入力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	可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
CLK_FEEDBACK	文字列	"1X"、"2X"、 "NONE"	"1X"	クロック フィードバックを指定します。
CLKDV_DIVIDE	浮動小数点	1.5、2.0、2.5、3.0、 3.5、4.0、4.5、5.0、 5.5、6.0、6.5、7.0、 7.5、8.0、9.0、10.0、 11.0、12.0、13.0、 14.0、15.0、16.0	2.0	CLKDLL、CLKDLLE、CLKDLLHF、 DCM のクロック分周器 (CLKDV 出力) の分周比を指定します。
CLKFX_DIVIDE	整数	1 ~ 32	1	CLKFX 出力の分周比を指定します。
CLKFX_MULTIPLY	整数	2 ~ 32	4	CLKFX 出力の通倍比を指定します。
CLKIN_DIVIDE_BY_2	ブール代数	FALSE、TRUE	FALSE	DCM の入力クロック周波数の要件に合 うように、必要に応じて入力クロック周波 数を 2 で分周します。
CLKIN_PERIOD	浮動小数点	1.25 ~ 1000.00	10.0	入力クロックの周期を ns で指定します。
CLKOUT_PHASE_SHIFT	文字列	"NONE"、 "FIXED"、 "VARIABLE_ POSITIVE"、 "VARIABLE_ CENTER"、 "DIRECT"	"NONE"	位相シフト モードを指定します。
DESKEW_ADJUST	文字列	"SOURCE_ SYNCHRONOUS"、 "SYSTEM_ SYNCHRONOUS"、 "0" ~ "15"	"SYSTEM_ SYNCHRONOUS"	フィードバック パスの遅延量を制御しま す。ソース同期のインターフェイスで使 用する必要があります。
DFS_FREQUENCY_ MODE	文字列	"LOW"、"HIGH"	"LOW"	周波数合成の周波数モードを指定しま す。
DLL_FREQUENCY_ MODE	文字列	"LOW"、"HIGH"	"LOW"	DLL の周波数モードを指定します。
DUTY_CYCLE_ CORRECTION	ブール代数	TRUE、FALSE	TRUE	CLK0、CLK90、CLK180、CLK270 の 各出力のデューティサイクルを修正し ます。
FACTORY_JF	16 進数	16 ビット値	F0F0	この属性は、DCM のジッター フィルター 特性に影響します。ザイリンクスからの 指示なしにこのデフォルト値を変更しな いください。
PHASE_SHIFT	整数	-255 ~ 1023	0	位相シフト量を指定します。この値の範 囲は CLKOUT_PHASE_SHIFT の指定に よって異なります。
STARTUP_WAIT	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、DCM がロック状態 になるまでコンフィギュレーション スタ ートアップ シーケンスの指定したサイク ルで待機します。

## 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;

-- DCM_PS: Digital Clock Manager Circuit
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

DCM_PS_inst : DCM_PS
generic map (
  CLKDV_DIVIDE => 2.0, -- Divide by: 1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
                        --      7.0,7.5,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0 or 16.0
  CLKFX_DIVIDE => 1,  -- Can be any integer from 1 to 32
  CLKFX_MULTIPLY => 4, -- Can be any integer from 2 to 32
  CLKIN_DIVIDE_BY_2 => FALSE, -- TRUE/FALSE to enable CLKIN divide by two feature
  CLKIN_PERIOD => 10.0, -- Specify period of input clock in ns from 1.25 to 1000.00
  CLKOUT_PHASE_SHIFT => "NONE", -- Specify phase shift mode of NONE, FIXED,
                                --      VARIABLE POSITIVE, VARIABLE CENTER or DIRECT
  CLK_FEEDBACK => "1X", -- Specify clock feedback of NONE or 1X
  DCM_PERFORMANCE_MODE => "MAX_SPEED", -- Can be MAX_SPEED or MAX_RANGE
  DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS", -- SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                --      an integer from 0 to 15
  DFS_FREQUENCY_MODE => "LOW", -- HIGH or LOW frequency mode for frequency synthesis
  DLL_FREQUENCY_MODE => "LOW", -- LOW, HIGH, or HIGH_SER frequency mode for DLL
  DUTY_CYCLE_CORRECTION => TRUE, -- Duty cycle correction, TRUE or FALSE
  FACTORY_JF => X"F0F0", -- FACTORY JF Values Suggested to be set to X"F0F0"
  PHASE_SHIFT => 0, -- Amount of fixed phase shift from -255 to 1023
  STARTUP_WAIT => FALSE) -- Delay configuration DONE until DCM LOCK, TRUE/FALSE
port map (
  CLK0 => CLK0, -- 0 degree DCM CLK output
  CLK180 => CLK180, -- 180 degree DCM CLK output
  CLK270 => CLK270, -- 270 degree DCM CLK output
  CLK2X => CLK2X, -- 2X DCM CLK output
  CLK2X180 => CLK2X180, -- 2X, 180 degree DCM CLK out
  CLK90 => CLK90, -- 90 degree DCM CLK output
  CLKDV => CLKDV, -- Divided DCM CLK out (CLKDV_DIVIDE)
  CLKFX => CLKFX, -- DCM CLK synthesis out (M/D)
  CLKFX180 => CLKFX180, -- 180 degree CLK synthesis out
  DO => DO, -- 16-bit data output for Dynamic Reconfiguration Port (DRP)
  LOCKED => LOCKED, -- DCM LOCK status output
  PSDONE => PSDONE, -- Dynamic phase adjust done output
  CLKFB => CLKFB, -- DCM clock feedback
  CLKIN => CLKIN, -- Clock input (from IBUFG, BUFG or DCM)
  PSCLK => PSCLK, -- Dynamic phase adjust clock input
  PSEN => PSEN, -- Dynamic phase adjust enable input
  PSINCDEC => PSINCDEC, -- Dynamic phase adjust increment/decrement
  RST => RST -- DCM asynchronous reset input
);

-- End of DCM_PS_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// DCM_PS: Dynamic Phase Shift Digital Clock Manager Circuit
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

DCM_PS #(
    .CLKDV_DIVIDE(2.0), // Divide by: 1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
                          //      7.0,7.5,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0 or 16.0
    .CLKFX_DIVIDE(1), // Can be any integer from 1 to 32
    .CLKFX_MULTIPLY(4), // Can be any integer from 2 to 32
    .CLKIN_DIVIDE_BY_2("FALSE"), // TRUE/FALSE to enable CLKIN divide by two feature
    .CLKIN_PERIOD(10.0), // Specify period of input clock in ns from 1.25 to 1000.00
    .CLKOUT_PHASE_SHIFT("NONE"), // Specify phase shift mode of NONE, FIXED,
                                  //      VARIABLE_POSITIVE, VARIABLE_CENTER or DIRECT
    .CLK_FEEDBACK("1X"), // Specify clock feedback of NONE or 1X
    .DCM_PERFORMANCE_MODE("MAX_SPEED"), // Can be MAX_SPEED or MAX_RANGE
    .DESKEW_ADJUST("SYSTEM_SYNCHRONOUS"), // SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                          //      an integer from 0 to 15
    .DFS_FREQUENCY_MODE("LOW"), // HIGH or LOW frequency mode for frequency synthesis
    .DLL_FREQUENCY_MODE("LOW"), // LOW, HIGH, or HIGH_SER frequency mode for DLL
    .DUTY_CYCLE_CORRECTION("TRUE"), // Duty cycle correction, TRUE or FALSE
    .FACTORY_JF(16'hf0f0), // FACTORY JF value suggested to be set to 16'hf0f0
    .PHASE_SHIFT(0), // Amount of fixed phase shift from -255 to 1023
    .STARTUP_WAIT("FALSE") // Delay configuration DONE until DCM LOCK, TRUE/FALSE
) DCM_PS_inst (
    .CLK0(CLK0), // 0 degree DCM CLK output
    .CLK180(CLK180), // 180 degree DCM CLK output
    .CLK270(CLK270), // 270 degree DCM CLK output
    .CLK2X(CLK2X), // 2X DCM CLK output
    .CLK2X180(CLK2X180), // 2X, 180 degree DCM CLK out
    .CLK90(CLK90), // 90 degree DCM CLK output
    .CLKDV(CLKDV), // Divided DCM CLK out (CLKDV_DIVIDE)
    .CLKFX(CLKFX), // DCM CLK synthesis out (M/D)
    .CLKFX180(CLKFX180), // 180 degree CLK synthesis out
    .DO(DO), // 16-bit data output for Dynamic Reconfiguration Port (DRP)
    .LOCKED(LOCKED), // DCM LOCK status output
    .PSDONE(PSDONE), // Dynamic phase adjust done output
    .CLKFB(CLKFB), // DCM clock feedback
    .CLKIN(CLKIN), // Clock input (from IBUFG, BUFG or DCM)
    .PCLK(PCLK), // Dynamic phase adjust clock input
    .PSEN(PSEN), // Dynamic phase adjust enable input
    .PSINCDEC(PSINCDEC), // Dynamic phase adjust increment/decrement
    .RST(RST) // DCM asynchronous reset input
);

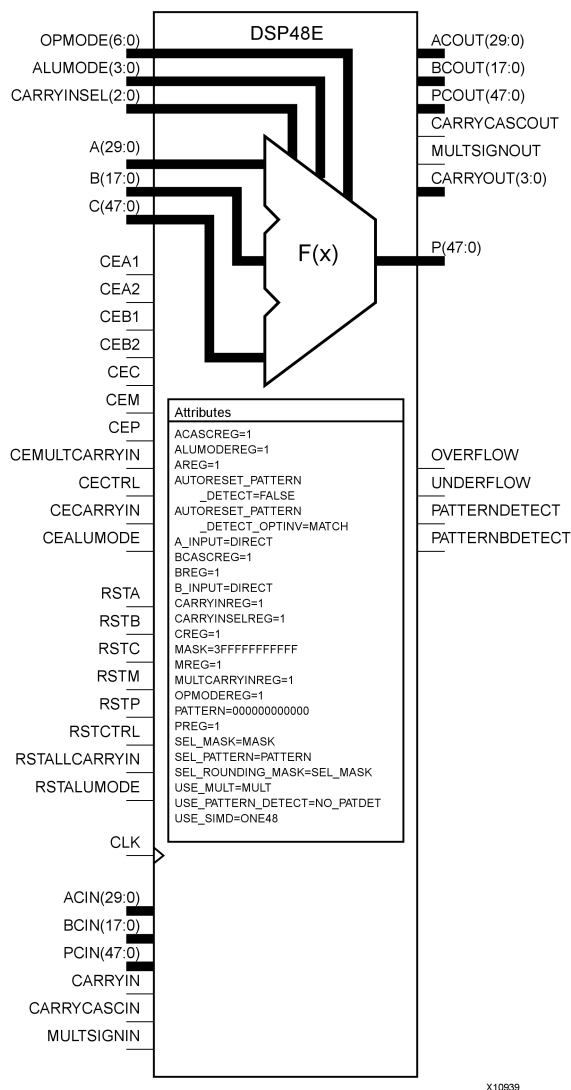
// End of DCM_PS_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## DSP48E

**プリミティブ：25x18 Two's Complement Multiplier with Integrated 48-Bit, 3-Input Adder/Subtractor/Accumulator or 2-Input Logic Unit**



X10939

## 概要

このデザイン エLEMENTは、柔軟性が高い多用途のハード IP ブロックで、多くの DSP アルゴリズムで見られる小型で高速な演算処理を作成できます。このブロックでは、乗算、加算、減算、累積、シフト、論理処理、およびパターン検出などが実行できます。

## ポートの説明

ポート名	方向	幅	機能
データ ポート			
A	入力	30	乗算器への 25 ビット データ入力または加算器/ロジック ユニット (LU) への 30 ビット MSB データ入力



ポート名	方向	幅	機能
B	入力	18	乗算器への 18 ビット データ入力または加算器/ロジック ユニット (LU) への 18 ビット LSB データ入力
C	入力	48	加算器/ロジック ユニット (LU) への 48 ビット データ入力および (または) パターン検出
CARRYIN	入力	1	加算器/ロジック ユニット (LU) への外部キャリー入力
P	出力	48	プライマリ データ出力
CARRYOUT	出力	4	<p>演算処理 (加算、減算など) のキャリー出力信号</p> <ul style="list-style-type: none"> <li>USE_SIMD が "FOUR12" の場合、CARRYOUT[3:0] は累積/加算/ロジック ユニット (LU) それぞれからの 12 ビットのキャリー出力を指します。</li> <li>USE_SIMD が "TWO24" の場合、CARRYOUT[3] および CARRYOUT[1] は累積/加算/ALU それぞれからの 24 ビットのキャリー出力を指します。</li> <li>USE_SIMD が "ONE48" の場合、CARRYOUT[3] は累積/加算/ロジック ユニット (LU) からの唯一の有効なキャリー出力です。</li> </ul>
制御入力/ステータス ビット			
CLK	入力	1	DSP48E のクロック入力
OPMODE	入力	7	ALUMODE と共に DSP48E の演算処理を選択する制御入力です。
ALUMODE	入力	4	加算および減算を含むロジック ユニット (LU) ファンクションを選択する制御入力です。
CARRYINSEL	入力	3	DSP48E へのキャリー入力ソースを選択します。
OVERFLOW	出力	1	パターン検出器が使用され PREG=1 のときに、加算器/累積でオーバーフローを検出するアクティブ High の出力です。
UNDERFLOW	出力	1	パターン検出器が使用され PREG=1 のときに、加算器/累積でアンダーフローを検出するアクティブ High の出力です。
PATTERNDETECT	出力	1	アクティブ High のパターン検出で、MASK 属性で指定した箇所で PATTERN の反転した値と P 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。
PATTERN BDETECT	出力	1	アクティブ High のパターン検出で、MASK 属性で指定した箇所で PATTERN の反転した値と P 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。
リセット/クロック イネーブル入力			
RSTA	入力	1	A ポートレジスタ (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。
RSTB	入力	1	B ポートレジスタ (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 ポートレジスタ (CREG=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。
RSTM	入力	1	乗算器レジスタ (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。
RSTP	入力	1	P 出力、UNDERFLOW 出力、OVERFLOW 出力、PATTERNDETECT 出力、PATTERNBDETECT 出力、および CARRYOUT 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。

ポート名	方向	幅	機能
RSTCTRL	入力	1	OPMODE および CARRYINSEL レジスタ (OPMODEREG=1 および 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。
RSTALLCARRYIN	入力	1	全キャリー入力レジスタ (CARRYINREG=1 または MULTCARRYINREG=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。
RSTALUMODE	入力	1	ALUMODE 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。
CEA1	入力	1	A ポートレジスタ (AREG=2) のアクティブ High のクロック イネーブルで、使用しない場合および AREG=2 の場合は論理 1 に、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 1 段目になります。
CEA2	入力	1	A ポートレジスタのアクティブ High のクロック イネーブルで、使用しない場合および 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 段目になります。
CEB1	入力	1	B ポートレジスタ (BREG=2) のアクティブ High のクロック イネーブルで、使用しない場合および 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 1 段目になります。
CEB2	入力	1	B ポートレジスタのアクティブ High のクロック イネーブルで、使用しない場合および 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 段目になります。
CEC	入力	1	C ポートレジスタ (CREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理 1 に接続します。
CEM	入力	1	乗算のレジスタ (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	出力ポートレジスタ (PREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理 1 に接続します。
CECTRL	入力	1	OPMODE およびキャリー入力セレクト レジスタ (CTRLREG=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	キャリー入力レジスタ (CARRYINREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理 1 に接続します。
CEMULTCARRYIN	入力	1	対称的な丸めを実行する乗算器の内部キャリー レジスタ (MULTCARRYINREG=1) のクロック イネーブルです。
CEALUMODE	入力	1	ALUMODE 入力レジスタ (ALUMODEREG=1) のクロック イネーブルです。
カスケード ポート			
ACIN	入力	30	ポート A のカスケード入力で、カスケード接続されている上位の DSP48E の ACOUT に接続します。使用しない場合は、ポートをすべて 0 にします。
BCIN	入力	18	ポート B のカスケード入力で、カスケード接続されている上位の DSP48E の BCOUT に接続します。使用しない場合は、ポートをすべて 0 にします。
PCIN	入力	48	ポート P のカスケード入力で、カスケード接続されている上位の DSP48E の PCOUT に接続します。使用しない場合は、ポートをすべて 0 にします。

ポート名	方向	幅	機能
CARRYCASCIN	入力	1	上位の DSP48E の CARRYOUT[2] とカスケード接続されます。
MULTSIGNIN	入力	1	48 ビット以上の出力が必要な場合に、カスケード接続されている DSP48E の乗算器の符号出力を使用して、加算器/アキュムレータの出力の符号を拡張します。MULTSIGNOUT 出力ピンにのみ接続します。
ACOUT	出力	30	ポート A のカスケード出力で、カスケード接続されている下位の DSP48E の ACIN に接続します。使用しない場合は未接続にします。
BCOUT	出力	18	ポート B のカスケード出力で、カスケード接続されている下位の DSP48E の BCIN に接続します。使用しない場合は未接続にします。
PCOUT	出力	48	ポート P のカスケード出力で、カスケード接続されている下位の DSP48E の PCIN に接続します。使用しない場合は未接続にします。
CARRYCASCOUT	出力	1	次の DSP48E にカスケード接続された CARRYOUT[3]
MULTSIGNOUT	出力	1	乗算器の符号出力を使用して、カスケード接続されている DSP48E で加算器/アキュムレータの符号を拡張します。MULTSIGNIN 入力ピンにのみ接続します。

## 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
ACASCREG	整数	0、1、2	1	AREG 属性と組み合わせて使用し、ACIN カスケード入力の A 入力レジスタの数を指定します。AREG の値以下にする必要があります。
AREG	整数	0、1、2	1	A 入力にレジスタを付けるかどうかを指定します。
ALUMODEREG	整数	0、1	1	ALUMODE 入力ピンにレジスタを付けるかどうかを指定します。
AUTORESET_PATTERN_DETECT	ブール代数	TRUE、FALSE	FALSE	AUTORESET_PATTERN_DETECT_OPTINV で定義されたパターン検出イベントがこのクロックサイクルで発生した場合、DSP48E の P レジスタ (累積値またはカウンタ値) を次のクロックサイクルで自動的にリセットします。
AUTORESET_PATTERN_DETECT_OPTINV	文字列	"MATCH"、 "NOT_MATCH"	"MATCH"	パターンが一致した場合、またはパターンは現在のクロックサイクルでは一致しないがその直前のクロックサイクルで一致していた場合に、AUTORESET_PATTERN_DETECT により P レジスタを次のクロックサイクルで自動的にリセットするかどうかを指定します。
A_INPUT	文字列	"DIRECT"、 "CASCADE"	"DIRECT"	A (DIRECT) または ACIN (CASCADE) のいずれかを選択します。

属性	データ型	値	デフォルト	説明
BCASCREG	整数	0、1、2	1	BREG 属性と組み合わせて使用し、BCIN カスケード入力の B 入力レジスタの数を指定します。
BREG	整数	0、1、2	1	B 入力にレジスタを付けるかどうかを指定します。
B_INPUT	文字列	"DIRECT"、 "CASCADE"	"DIRECT"	B (DIRECT) または BCIN (CASCADE) のいずれかを選択します。
CARRYINREG	整数	0、1	1	CARRYIN 入力にレジスタを付けるかどうかを指定します。
CARRYINSELREG	整数	0、1	1	CARRYINSEL 入力にレジスタを付けるかどうかを指定します。
CREG	整数	0、1	1	C 入力にレジスタを付けるかどうかを指定します。
MASK	16 進数	48 ビット値	3FFF	パターン検出器で使用されるマスクを指定します。
MREG	整数	0、1	1	乗算器の段にレジスタを付けるかを指定します。 イネーブル = 1/ディスエーブル = 0
MULTCARRYINREG	整数	0、1	1	内部キャリーのレジスタ数を選択します (対称的な丸めを実行する乗算でのみ使用)。
OPMODEREG	整数	0、1	1	OPMODE 入力にレジスタを付けるかどうかを指定します。
PATTERN	16 進数	48 ビット値	すべて 0	パターン検出器で使用されるパターンを指定します。
PREG	整数	0、1	1	P 入力にレジスタを付けるかどうかを指定します。
SEL_MASK	文字列	"MASK"、"C"	"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。
SEL_PATTERN	文字列	"PATTERN"、 "C"	"PATTERN"	パターン検出器のパターンに PATTERN を使用するか、C 入力を使用するかを指定します。
SEL_ROUNDING_MASK	文字列	"SEL_MASK"、 "MODE1"、 "MODE2"	"SEL_MASK"	パターン検出で対称的な丸めと偶数丸めに使用するマスクを選択します。MODE1 または MODE2 に設定すると、SEL_MASK 属性は無視されます。これらは、偶数丸めに使用されます。
SIM_MODE	文字列	"SAFE"、 "FAST"	"SAFE"	シミュレーションのみの属性です。"FAST" に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。
USE_MULT	文字列	"MULT"、 "MULT_S"、 "NONE"	"MULT_S"	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。MPEG が 0 の場合は MULT に、1 の場合は MULT_S に設定します。

属性	データ型	値	デフォルト	説明
USE_SIMD	文字列	"ONE48"、 "TWO24"、 "FOUR12"	"ONE48"	SIMD (Single Instruction Multiple Data) 加算器/ロジックユニットの使用方法を選択します。48 ビットのロジックユニット 1 個、24 ビットのロジックユニット 2 個、または 12 ビットのロジックユニット 4 個から選択します。12 ビットのロジックユニット 4 個では、同じ命令が実行されることに注意してください。つまり、すべてのロジックユニットで減算または加算が同サイクルで実行されます。これにより、計算量の比較的少ないアプリケーション向けに 48 ビットの加算器を小型の加算器に分割できます。SIMD は、加算、累積、減算などの演算処理にのみに影響し、論理処理には影響しません。
USE_PATTERN_DETECT	文字列	"PAT_DET"、 "NO_PAT_DET"	"NO_PATDET"	パターン検出をイネーブルにします。シミュレーションおよびスピード ファイルのみに実行されます。

## 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;

-- DSP48E: DSP Function Block
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

DSP48E_inst : DSP48E
generic map (
    ACASCREG => 1,          -- Number of pipeline registers between
                            -- A/ACIN input and ACOUT output, 0, 1, or 2
    ALUMODEREG => 1,        -- Number of pipeline registers on ALUMODE input, 0 or 1
    AREG => 1,              -- Number of pipeline registers on the A input, 0, 1 or 2
    AUTORESET_PATTERN_DETECT => FALSE, -- Auto-reset upon pattern detect, TRUE or FALSE
    AUTORESET_PATTERN_DETECT_OPTINV => "MATCH", -- Reset if "MATCH" or "NOMATCH"
    A_INPUT => "DIRECT",    -- Selects A input used, "DIRECT" (A port) or "CASCADE" (ACIN port)
    BCASCREG => 1,          -- Number of pipeline registers between B/BCIN input and BCOUT output, 0, 1, or 2
    BREG => 1,              -- Number of pipeline registers on the B input, 0, 1 or 2
    B_INPUT => "DIRECT",    -- Selects B input used, "DIRECT" (B port) or "CASCADE" (BCIN port)
    CARRYINREG => 1,        -- Number of pipeline registers for the CARRYIN input, 0 or 1
    CARRYINSELREG => 1,     -- Number of pipeline registers for the CARRYINSEL input, 0 or 1
    CREG => 1,              -- Number of pipeline registers on the C input, 0 or 1
    MASK => X"3FFFFFFF",   -- 48-bit Mask value for pattern detect
    MREG => 1,              -- Number of multiplier pipeline registers, 0 or 1
    MULTCARRYINREG => 1,    -- Number of pipeline registers for multiplier carry in bit, 0 or 1
    OPMODEREG => 1,         -- Number of pipeline registers on OPMODE input, 0 or 1
    PATTERN => X"000000000000", -- 48-bit Pattern match for pattern detect
    PREG => 1,              -- Number of pipeline registers on the P output, 0 or 1
    SIM_MODE => "SAFE",    -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
                            -- Design Guide" for details
    SEL_MASK => "MASK",    -- Select mask value between the "MASK" value or the value on the "C" port
    SEL_PATTERN => "PATTERN", -- Select pattern value between the "PATTERN" value or the value on the "C" port
    SEL_ROUNDING_MASK => "SEL_MASK", -- "SEL_MASK", "MODE1", "MODE2"
    USE_MULT => "MULT_S",  -- Select multiplier usage, "MULT" (MREG => 0),
                            -- "MULT_S" (MREG => 1), "NONE" (not using multiplier)
    USE_PATTERN_DETECT => "NO_PATDET", -- Enable pattern detect, "PATDET", "NO_PATDET"
    USE_SIMD => "ONE48") -- SIMD selection, "ONE48", "TWO24", "FOUR12"
port map (
    ACOUT => ACOUT,        -- 30-bit A port cascade output
    BCOUT => BCOUT,        -- 18-bit B port cascade output
    CARRYCASCOUT => CARRYCASCOUT, -- 1-bit cascade carry output
    CARRYOUT => CARRYOUT,  -- 4-bit carry output
    MULTSIGNOUT => MULTSIGNOUT, -- 1-bit multiplier sign cascade output

```

```

OVERFLOW => OVERFLOW, -- 1-bit overflow in add/acc output
P => P,                -- 48-bit output
PATTERNBDETECT => PATTERNBDETECT, -- 1-bit active high pattern bar detect output
PATTERNDETECT => PATTERNDETECT, -- 1-bit active high pattern detect output
PCOUT => PCOUT,        -- 48-bit cascade output
UNDERFLOW => UNDERFLOW, -- 1-bit active high underflow in add/acc output
A => A,                -- 30-bit A data input
ACIN => ACIN,          -- 30-bit A cascade data input
ALUMODE => ALUMODE,    -- 4-bit ALU control input
B => B,                -- 18-bit B data input
BCIN => BCIN,          -- 18-bit B cascade input
C => C,                -- 48-bit C data input
CARRYCASCIN => CARRYCASCIN, -- 1-bit cascade carry input
CARRYIN => CARRYIN,    -- 1-bit carry input signal
CARRYINSEL => CARRYINSEL, -- 3-bit carry select input
CEA1 => CEA1,          -- 1-bit active high clock enable input for 1st stage A registers
CEA2 => CEA2,          -- 1-bit active high clock enable input for 2nd stage A registers
CEALUMODE => CEALUMODE, -- 1-bit active high clock enable input for ALUMODE registers
CEB1 => CEB1,          -- 1-bit active high clock enable input for 1st stage B registers
CEB2 => CEB2,          -- 1-bit active high clock enable input for 2nd stage B registers
CEC => CEC,            -- 1-bit active high clock enable input for C registers
CECARRYIN => CECARRYIN, -- 1-bit active high clock enable input for CARRYIN register
CECTRL => CECTRL,      -- 1-bit active high clock enable input for OPMODE and carry registers
CEM => CEM,            -- 1-bit active high clock enable input for multiplier registers
CEMULTCARRYIN => CEMULTCARRYIN, -- 1-bit active high clock enable for multiplier carry in register
CEP => CEP,            -- 1-bit active high clock enable input for P registers
CLK => CLK,            -- Clock input
MULTSIGNIN => MULTSIGNIN, -- 1-bit multiplier sign input
OPMODE => OPMODE,      -- 7-bit operation mode input
PCIN => PCIN,          -- 48-bit P cascade input
RSTA => RSTA,          -- 1-bit reset input for A pipeline registers
RSTALLCARRYIN => RSTALLCARRYIN, -- 1-bit reset input for carry pipeline registers
RSTALUMODE => RSTALUMODE, -- 1-bit reset input for ALUMODE pipeline registers
RSTB => RSTB,          -- 1-bit reset input for B pipeline registers
RSTC => RSTC,          -- 1-bit reset input for C pipeline registers
RSTCTRL => RSTCTRL,    -- 1-bit reset input for OPMODE pipeline registers
RSTM => RSTM,          -- 1-bit reset input for multiplier registers
RSTP => RSTP,          -- 1-bit reset input for P pipeline registers
);

-- End of DSP48E_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// DSP48E: DSP Function Block
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

DSP48E #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .ACASCREG(1),      // Number of pipeline registers between A/ACIN input and ACOUT output, 0, 1, or 2
    .ALUMODEREG(1),    // Number of pipeline registers on ALUMODE input, 0 or 1
    .AREG(1),          // Number of pipeline registers on the A input, 0, 1 or 2
    .AUTORESET_PATTERN_DETECT("FALSE"), // Auto-reset upon pattern detect, "TRUE" or "FALSE"
    .AUTORESET_PATTERN_DETECT_OPTINV("MATCH"), // Reset if "MATCH" or "NOMATCH"
    .A_INPUT("DIRECT"), // Selects A input used, "DIRECT" (A port) or "CASCADE" (ACIN port)
    .BCASCREG(1),      // Number of pipeline registers between B/BCIN input and BCOUT output, 0, 1, or 2
    .BREG(1),          // Number of pipeline registers on the B input, 0, 1 or 2
    .B_INPUT("DIRECT"), // Selects B input used, "DIRECT" (B port) or "CASCADE" (BCIN port)
    .CARRYINREG(1),    // Number of pipeline registers for the CARRYIN input, 0 or 1
    .CARRYINSELREG(1), // Number of pipeline registers for the CARRYINSEL input, 0 or 1
    .CREG(1),          // Number of pipeline registers on the C input, 0 or 1
    .MASK(48'h3fffffff), // 48-bit Mask value for pattern detect
    .MREG(1),          // Number of multiplier pipeline registers, 0 or 1
    .MULTCARRYINREG(1), // Number of pipeline registers for multiplier carry in bit, 0 or 1
    .OPMODEREG(1),     // Number of pipeline registers on OPMODE input, 0 or 1
    .PATTERN(48'h000000000000), // 48-bit Pattern match for pattern detect
    .PREG(1),          // Number of pipeline registers on the P output, 0 or 1
    .SEL_MASK("MASK"), // Select mask value between the "MASK" value or the value on the "C" port
    .SEL_PATTERN("PATTERN"), // Select pattern value between the "PATTERN" value or the value on the "C" port
    .SEL_ROUNDING_MASK("SEL_MASK"), // "SEL_MASK", "MODE1", "MODE2"

```

```

.USE_MULT("MULT_S"), // Select multiplier usage, "MULT" (MREG => 0), "MULT_S" (MREG => 1), "NONE" (no multiplier)
.USE_PATTERN_DETECT("NO_PATDET"), // Enable pattern detect, "PATDET", "NO_PATDET"
.USE_SIMD("ONE48") // SIMD selection, "ONE48", "TWO24", "forUR12"
) DSP48E_inst (
.ACOUT(ACOUT), // 30-bit A port cascade output
.BCOUT(BCOUT), // 18-bit B port cascade output
.CARRYCASCOUT(CARRYCASCOUT), // 1-bit cascade carry output
.CARRYOUT(CARRYOUT), // 4-bit carry output
.MULTSIGNOUT(MULTSIGNOUT), // 1-bit multiplier sign cascade output
.OVERFLOW(OVERFLOW), // 1-bit overflow in add/acc output
.P(P), // 48-bit output
.PATTERNBDETECT(PATTERNBDETECT), // 1-bit active high pattern bar detect output
.PATTERNDETECT(PATTERNDETECT), // 1-bit active high pattern detect output
.PCOUT(PCOUT), // 48-bit cascade output
.UNDERFLOW(UNDERFLOW), // 1-bit active high underflow in add/acc output
.A(A), // 30-bit A data input
.ACIN(ACIN), // 30-bit A cascade data input
.ALUMODE(ALUMODE), // 4-bit ALU control input
.B(B), // 18-bit B data input
.BCIN(BCIN), // 18-bit B cascade input
.C(C), // 48-bit C data input
.CARRYCASCIN(CARRYCASCIN), // 1-bit cascade carry input
.CARRYIN(CARRYIN), // 1-bit carry input signal
.CARRYINSEL(CARRYINSEL), // 3-bit carry select input
.CEA1(CEA1), // 1-bit active high clock enable input for 1st stage A registers
.CEA2(CEA2), // 1-bit active high clock enable input for 2nd stage A registers
.CEALUMODE(CEALUMODE), // 1-bit active high clock enable input for ALUMODE registers
.CEB1(CEB1), // 1-bit active high clock enable input for 1st stage B registers
.CEB2(CEB2), // 1-bit active high clock enable input for 2nd stage B registers
.CEC(CEC), // 1-bit active high clock enable input for C registers
.CECARRYIN(CECARRYIN), // 1-bit active high clock enable input for CARRYIN register
.CECTRL(CECTRL), // 1-bit active high clock enable input for OPMODE and carry registers
.CEM(CEM), // 1-bit active high clock enable input for multiplier registers
.CEMULTCARRYIN(CEMULTCARRYIN), // 1-bit active high clock enable for multiplier carry in register
.CEP(CEP), // 1-bit active high clock enable input for P registers
.CLK(CLK), // Clock input
.MULTSIGNIN(MULTSIGNIN), // 1-bit multiplier sign input
.OPMODE(OPMODE), // 7-bit operation mode input
.PCIN(PCIN), // 48-bit P cascade input
.RSTA(RSTA), // 1-bit reset input for A pipeline registers
.RSTALLCARRYIN(RSTALLCARRYIN), // 1-bit reset input for carry pipeline registers
.RSTALUMODE(RSTALUMODE), // 1-bit reset input for ALUMODE pipeline registers
.RSTB(RSTB), // 1-bit reset input for B pipeline registers
.RSTC(RSTC), // 1-bit reset input for C pipeline registers
.RSTCTRL(RSTCTRL), // 1-bit reset input for OPMODE pipeline registers
.RSTM(RSTM), // 1-bit reset input for multiplier registers
.RSTP(RSTP) // 1-bit reset input for P pipeline registers
);

// End of DSP48E_inst instantiation

```

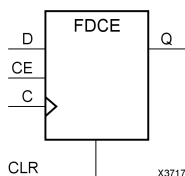
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## FDCE

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



## 概要

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

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

## 論理表

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

## デザインの入力方法

インスタンシエーション	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	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;

-- FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
--       Clock Enable (posedge clk).
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of FDCE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

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
);

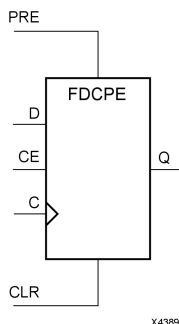
// End of FDCE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## FDCPE

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



### 概要

このデザイン エレメントは、データ (D)、クロック イネーブル (CE)、非同期プリセット (PRE)、非同期クリア (CLR) 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。非同期 PRE が High になると Q 出力が High にセットされ、CLR が High になると出力が Low にリセットされます (CLR 入力の方が優先されます)。PRE と CLR が Low で 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、以前の値が保持されます。FDCPE は通常、スライスまたは IOB レジスタとしてインプリメントされます。

FPGA では電源が投入されると、INIT 属性を使用して指定した初期値に設定されます。GSR (グローバル セット/リセット) をアサートすると、INIT で指定した初期値に非同期で設定されます。

**注記 :** 非同期のセットおよびリセットの使用がサポートされていますが、これらの使用は通常お勧めしません。非同期信号を使用するとタイミングの問題を検出および制御しにくく、ロジックの最適化に悪影響を及ぼし、同期セットまたはリセットを使用した場合に比べて消費電力が大きくなる場合があります。

### 論理表

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

### ポートの説明

ポート名	方向	幅	機能
Q	出力	1	データ出力
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
CLR	入力	1	非同期クリア入力
D	入力	1	データ入力
PRE	入力	1	非同期セット入力

## デザインの入力方法

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

## 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および GSR 入力時の Q 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;

-- FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
--       Clock Enable (posedge clk).
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

FDCPE_inst : FDCPE
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
  PRE => PRE   -- Asynchronous set input
);

-- End of FDCPE_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
//       Clock Enable (posedge clk).
//       Virtex-5, Spartan-3/3E/3A/3A DSP
// Xilinx HDL Libraries Guide, version 14.5

FDCPE #(
  .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDCPE_inst (
  .Q(Q),      // Data output
  .C(C),      // Clock input
  .CE(CE),    // Clock enable input
  .CLR(CLR),  // Asynchronous clear input
  .D(D),      // Data input
  .PRE(PRE)   // Asynchronous set input
);

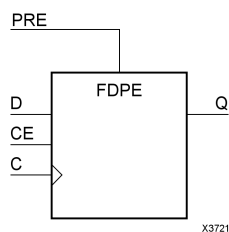
// End of FDCPE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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\_architecture シンボルの GSR 入力の前にインバーターを追加するとアクティブ Low にできます。

## 論理表

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

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数	0、1	1	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).
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of FDPE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

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
);

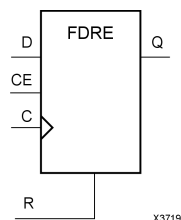
// End of FDPE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## FDRE

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



## 概要

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

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

## 論理表

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

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数	0、1	0	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).
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of FDRE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

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
```

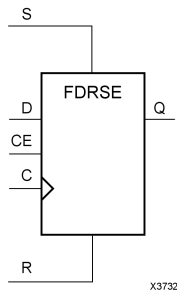
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## FDRSE

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



## 概要

FDRSE は、同期リセット (R)、同期セット (S)、クロック イネーブル (CE) 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 に切り替わる時に出力 (Q) が Low にリセットされます (リセットがセットよりも優先される)。セット入力 (S) が High、R が Low の場合、クロック (C) が Low から High に切り替わる時にフリップフロップがセットされ、出力が High になります。R と S が Low、CE が High の場合、クロックが 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。

電源が投入されると、INIT 属性を使用して指定した初期値に設定されます。GSR (グローバル セット/リセット) をアサートすると、INIT で指定した初期値に非同期で設定されます。

## 論理表

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

## 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および GSR 入力時の Q 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;

-- FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
--       Clock Enable (posedge clk).
--       Virtex-5
--       Xilinx HDL Libraries Guide, version 14.5

FDRSE_inst : FDRSE
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
  D => D,      -- Data input
  R => R,      -- Synchronous reset input
  S => S      -- Synchronous set input
);

-- End of FDRSE_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//       Clock Enable (posedge clk).
//       Virtex-5, Spartan-3/3E/3A/3A DSP
//       Xilinx HDL Libraries Guide, version 14.5

FDRSE #(
  .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDRSE_inst (
  .Q(Q),      // Data output
  .C(C),      // Clock input
  .CE(CE),    // Clock enable input
  .D(D),      // Data input
  .R(R),      // Synchronous reset input
  .S(S)       // Synchronous set input
);

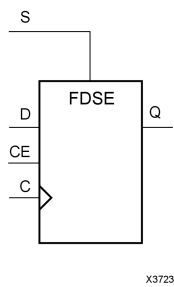
// End of FDRSE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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 High 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 に切り替わるときに Q 出力が High 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。

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

### 論理表

入力				出力
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).
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of FDSE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

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
);

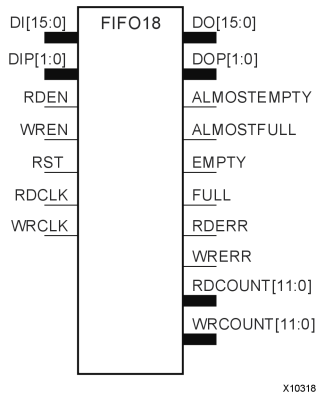
// End of FDSE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## FIFO18

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



### 概要

Virtex®-5 以降の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、自動誤り訂正 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。FIFO18 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 ワード、9 ビット X 2K ワード、または 18 ビット X 1K ワードのコンフィギュレーションで使用できます。また、このプリミティブは関連するすべての FIFO フラグおよびステータス信号と共に、同期モードまたはマルチレート (非同期) モードのいずれにも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 モデルではアーキテクチャのユーザー 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 を使用します。さらにワード数が多く、データ幅の広いコンフィギュレーションには、FIFO 36 または FIFO36\_72 を使用します。誤り訂正回路が必要な場合は、FIFO36\_72 を使用します。

### ポートの説明

ポート名	方向	幅	機能
DO	出力	4、8、16	FIFO データ出力バス
DOP	出力	0、1、2	FIFO パリティ データ出力バス
FULL	出力	1	FIFO がフルであることを示します。
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。
EMPTY	出力	1	FIFO が空であることを示します。
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。
WRERR、RDERR	出力	1	WRERR は FIFO がフルの間に書き込みが実行されたことを示し、RDERR は FIFO が空の間に読み出しが実行されたことを示します。
WRCOUNT、RDCOUNT	出力	12	FIFO 書き込み/読み出しポインター
DI	入力	4、8、16	FIFO データ入力バス

ポート名	方向	幅	機能
DIP	入力	0、1、2	FIFO パリティ データ入力バス
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
RST	入力	1	FIFO 機能、フラグ、ポインターの非同期リセット (アクティブ High)。RESET は 3 クロック サイクル間アサートする必要があります。
WRCLK、RDCLK	入力	1	FIFO リード クロックおよびライト クロック (立ち上がりエッジで動作)

## 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	推奨

4 ビット幅でコンフィギュレーションされたプリミティブをインスタンス化するには、DIP ポートを論理 0 にし、DOP ポートは未接続のままにします。DI[3:0] および DO[3:0] は適切な入力および出力信号に、DI[15:4] は論理 0 に接続し、DO[15:4] は未接続のままにします。

9 ビット幅でコンフィギュレーションするときは、DIP[0] ポートを適切なデータ入力に接続し、DIP[1] は論理 0 に接続します。DOP[0] は適切なデータ出力に接続し、DOP[1] は未接続のままにします。DI[7:0] および DO[7:0] は適切な入力および出力信号に、DI[15:8] は論理 0 に接続し、DO[15:8] は未接続のままにします。

18 ビット幅をコンフィギュレーションするときは、DI、DIP、DO、および DOP 信号すべてを接続する必要があります。

いずれのコンフィギュレーションでも、未使用の DI または DIP 入力は論理 0 に、未使用の DO または DOP ピンは未接続のままにする必要があります。EN\_SYM 属性を TRUE に設定して FIFO を同期に設定するときは、同じクロックソースを WRCLK および RDCLK に接続する必要があります。非同期モード (EN\_SYN = FALSE) のときは、個別のクロック信号を使用できます。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、ALMOSTEMPTY フラグと 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 モデルではアーキテクチャのユーザー 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。

FIFO は電源投入後にリセットされる必要があります。FULL、ALMOSTFULL、EMPTY、および ALMOSTEMPTY 出力フラグは、適切なデスティネーション ロジックに接続するか、使用しない場合は未接続のままにする必要があります。WRERR、RDERR、WRCOUNT、および RDCOUNT はオプションの出力で、使用しない場合は未接続のままにできます。インスタンス化コードに含まれるジェネリック (VHDL) またはインライン defparam (Verilog) を変更することですべての属性を設定し、目的どおりに FIFO が動作するようにします。

## 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	12 ビット値	すべて 0	ALMOST_FULL 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	12 ビット値	すべて 0	ALMOST_EMPTY フラグをトリガーする RAM のデータ量を指定します。
FIRST_WORD_FALL_THROUGH	ブール代数	TRUE、FALSE	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。
DATA_WIDTH	整数	4、9、18	4	FIFO のデータ幅を指定します。

属性	データ型	値	デフォルト	説明
EN_SYN	ブール代数	TRUE、FALSE	FALSE	FIFO が非同期 (独立した 2 つのクロック) または同期 (1 つのクロック) のいずれで動作しているかを示します。マルチレートの場合は DO_REG=1 を設定する必要があります。
DO_REG	整数	0、1	1	EN_SYN のデータ パイプライン レジスタ
SIM_MODE	文字列	"SAFE"、"FAST"	"SAFE"	シミュレーションのみの属性です。"FAST" に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。  注記: "FAST" に設定した場合、一部の機能がサポートされません。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。

## 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;

-- FIFO18: 16k+2k Parity Synchronous/Asynchronous BlockRAM FIFO BlockRAM Memory
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

FIFO18_inst : FIFO18
generic map (
    ALMOST_FULL_OFFSET => X"080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"080", -- Sets the almost empty threshold
    DATA_WIDTH => 4, -- Sets data width to 4, 9, or 18
    DO_REG => 1, -- Enable output register ( 0 or 1)
    -- Must be 1 if the EN_SYN = FALSE
    EN_SYN => FALSE, -- Specified FIFO as Asynchronous (FALSE) or
    -- Synchronous (TRUE)
    FIRST_WORD_FALL_THROUGH => FALSE, -- Sets the FIFO FWFT to TRUE or FALSE
    SIM_MODE => "SAFE") -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
    -- Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    DO => DO, -- 32-bit data output
    DOP => DOP, -- 2-bit parity data output
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 12-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 12-bit write count output
    WRERR => WRERR, -- 1-bit write error
    DI => DI, -- 16-bit data input
    DIP => DIP, -- 2-bit parity input
    RDCLK => RDCLK, -- 1-bit read clock input
    RDEN => RDEN, -- 1-bit read enable input
    RST => RST, -- 1-bit reset input
    WRCLK => WRCLK, -- 1-bit write clock input
    WREN => WREN -- 1-bit write enable input
);

-- End of FIFO18_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// FIFO18: 16k+2k Parity Synchronous/Asynchronous BlockRAM FIFO
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

FIFO18 #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .ALMOST_FULL_OFFSET(12'h080), // Sets almost full threshold
    .ALMOST_EMPTY_OFFSET(12'h080), // Sets the almost empty threshold
    .DATA_WIDTH(4), // Sets data width to 4, 9 or 18
    .DO_REG(1), // Enable output register (0 or 1)
    // Must be 1 if EN_SYN = "FALSE"
    .EN_SYN("FALSE"), // Specifies FIFO as Asynchronous ("FALSE")
    // or Synchronous ("TRUE")
    .FIRST_WORD_FALL_THROUGH("FALSE") // Sets the FIFO FWFT to "TRUE" or "FALSE"
) FIFO18_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit almost empty output flag
    .ALMOSTFULL(ALMOSTFULL), // 1-bit almost full output flag
    .DO(DO), // 16-bit data output
    .DOP(DOP), // 2-bit parity data output
    .EMPTY(EMPTY), // 1-bit empty output flag
    .FULL(FULL), // 1-bit full output flag
    .RDCOUNT(RDCOUNT), // 12-bit read count output
    .RDERR(RDERR), // 1-bit read error output
    .WRCOUNT(WRCOUNT), // 12-bit write count output
    .WRERR(WRERR), // 1-bit write error
    .DI(DI), // 16-bit data input
    .DIP(DIP), // 2-bit parity input
    .RDCLK(RDCLK), // 1-bit read clock input
    .RDEN(RDEN), // 1-bit read enable input
    .RST(RST), // 1-bit reset input
    .WRCLK(WRCLK), // 1-bit write clock input
    .WREN(WREN) // 1-bit write enable input
);

// End of FIFO18_inst instantiation
```

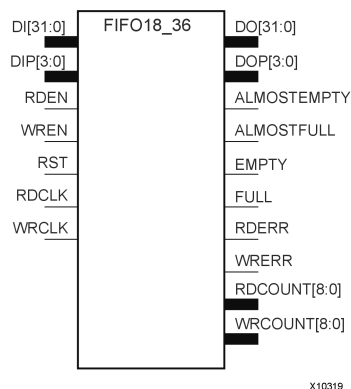
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## FIFO18\_36

**プリミティブ：36-bit Wide by 512 Deep 18kb FIFO (First In, First Out) Block RAM Memory**



### 概要

Virtex®-5 デバイスにはブロック RAM が数個含まれ、FIFO、自動誤り訂正 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。FIFO18\_36 を使用すると、幅の広いデータ バスが必要なときにブロック RAM を 18Kb FIFO コンフィギュレーションで使用できます。このコンポーネントは、36 ビット X 512 ワードの同期または非同期 FIFO RAM としてコンフィギュレーションできます。また、この FIFO RAM では関連するすべての FIFO フラグおよびステータス信号も提供されます。

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 モデルではアーキテクチャのユーザー 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：** ワード数が多くデータ幅が狭いコンフィギュレーションには、代わりに FIFO18 を使用し、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、FIFO 36 または FIFO36\_72 を使用します。誤り訂正回路が必要な場合は FIFO36\_72 を使用します。

### ポートの説明

ポート名	方向	幅	機能
DO	出力	32	FIFO データ出力バス
DOP	出力	4	FIFO パリティ データ出力バス
FULL	出力	1	FIFO がフルであることを示します。
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。
EMPTY	出力	1	FIFO が空であることを示します。
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。
WRERR、RDERR	出力	1	WRERR は FIFO がフルの間に書き込みが実行されたことを示し、RDERR は FIFO が空の間に読み出しが実行されたことを示します。
WRCOUNT、RDCOUNT	出力	9	FIFO 書き込み/読み出しポインター

ポート名	方向	幅	機能
DI	入力	32	FIFO データ入力バス
DIP	入力	4	FIFO パリティ データ入力バス
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
RST	入力	1	FIFO 機能、フラグ、ポインタの非同期リセット (アクティブ High)。RESET は 3 クロック サイクル間アサートする必要があります。
WRCLK、RDCLK	入力	1	FIFO リード クロックおよびライト クロック (立ち上がりエッジで動作)

## 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	推奨

DI、DIP、DO、および DOP ピンは、対応する入力および出力データソースに接続する必要があります。36 ビット以下を使用するときは、未使用の DI または DIP 入力を論理 0 に、未使用の DO または DOP ピンは未接続のままにする必要があります。EN\_SYM 属性を TRUE に設定して FIFO を同期に設定するときは、同じクロックソースを WRCLK および RDCLK に接続する必要があります。非同期モード (EN\_SYN = FALSE) のときは、個別のクロック信号を使用します。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、ALMOSTEMPTY フラグと 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 モデルではアーキテクチャのユーザー 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。WREN および RDEN は、対応するライト イネーブルおよびリード イネーブル信号/ロジックに接続し、RST は適切なリセット信号/ロジックに接続するか、使用しない場合は論理 0 に接続する必要があります。

FULL、ALMOSTFULL、EMPTY、および ALMOSTEMPTY 出力フラグは、適切なデスティネーション ロジックに接続するか、使用しない場合は未接続のままにする必要があります。WRERR、RDERR、WRCOUNT、および RDCOUNT はオプションの出力で、使用しない場合は未接続のままにできます。インスタンス化コードに含まれるジェネリック (VHDL) またはインライン defparam (Verilog) を変更することですべての属性を設定し、目的どおりに FIFO が動作するようにします。

## 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	9 ビット値	すべて 0	ALMOST_FULL 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	9 ビット値	すべて 0	ALMOST_EMPTY フラグをトリガーする RAM のデータ量を指定します。
FIRST_WORD_FALL_THROUGH	ブール代数	TRUE、FALSE	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。
EN_SYN	ブール代数	TRUE、FALSE	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。
DO_REG	整数	0、1	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

属性	データ型	値	デフォルト	説明
				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。
SIM_MODE	文字列	"SAFE"、"FAST"	"SAFE"	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。

## 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;

-- FIFO18_36: 36x18k Synchronous/Asynchronous BlockRAM FIFO
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

FIFO18_36_inst : FIFO18_36
generic map (
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DO_REG => 1, -- Enable output register (0 or 1)
                -- Must be 1 if EN_SYN = FALSE
    EN_SYN => FALSE, -- Specifies FIFO as Asynchronous (FALSE)
                -- or Synchronous (TRUE)
    FIRST_WORD_FALL_THROUGH => FALSE, -- Sets the FIFO FWFT to TRUE or FALSE
    SIM_MODE => "SAFE") -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
                -- Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    DO => DO, -- 32-bit data output
    DOP => DOP, -- 4-bit parity data output
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 9-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 9-bit write count output
    WRERR => WRERR, -- 1-bit write error
    DI => DI, -- 32-bit data input
    DIP => DIP, -- 4-bit parity input
    RDCLK => RDCLK, -- 1-bit read clock input
    RDEN => RDEN, -- 1-bit read enable input
    RST => RST, -- 1-bit reset input
    WRCLK => WRCLK, -- 1-bit write clock input
    WREN => WREN -- 1-bit write enable input
);

-- End of FIFO18_36_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```
// FIFO18_36: 36x18k Synchronous/Asynchronous BlockRAM FIFO
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

FIFO18_36 #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .ALMOST_FULL_OFFSET(9'h080), // Sets almost full threshold
    .ALMOST_EMPTY_OFFSET(9'h080), // Sets the almost empty threshold
    .DO_REG(1), // Enable output register (0 or 1)
    // Must be 1 if EN_SYN = "FALSE"
    .EN_SYN("FALSE"), // Specifies FIFO as Asynchronous ("FALSE")
    // or Synchronous ("TRUE")
    .FIRST_WORD_FALL_THROUGH("FALSE") // Sets the FIFO FWFT to "TRUE" or "FALSE"
) FIFO18_36_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit almost empty output flag
    .ALMOSTFULL(ALMOSTFULL), // 1-bit almost full output flag
    .DO(DO), // 32-bit data output
    .DOP(DOP), // 4-bit parity data output
    .EMPTY(EMPTY), // 1-bit empty output flag
    .FULL(FULL), // 1-bit full output flag
    .RDCOUNT(RDCOUNT), // 9-bit read count output
    .RDERR(RDERR), // 1-bit read error output
    .WRCOUNT(WRCOUNT), // 9-bit write count output
    .WRERR(WRERR), // 1-bit write error
    .DI(DI), // 32-bit data input
    .DIP(DIP), // 4-bit parity input
    .RDCLK(RDCLK), // 1-bit read clock input
    .RDEN(RDEN), // 1-bit read enable input
    .RST(RST), // 1-bit reset input
    .WRCLK(WRCLK), // 1-bit write clock input
    .WREN(WREN) // 1-bit write enable input
);

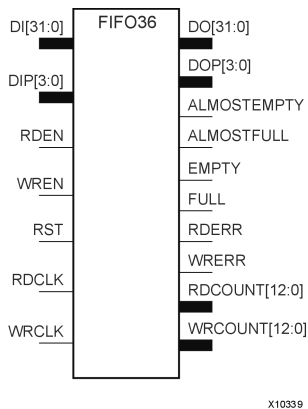
// End of FIFO18_36_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## FIFO36

### プリミティブ：36kb FIFO (First In, First Out) Block RAM Memory



## 概要

Virtex®-5 デバイスにはブロック RAM が数個含まれ、FIFO、自動誤り訂正 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。FIFO36 を使用すると、36Kb の FIFO のブロック RAM にアクセスできます。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 ワードの同期またはマルチレート (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 モデルではアーキテクチャのユーザー ガイドに示されているデ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 を使用します。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、FIFO18 または FIFO18\_36 を使用します。誤り訂正回路が必要な場合は、FIFO36\_72 を使用します。

## ポートの説明

ポート名	方向	幅	機能
DO	出力	4、8、16、32	FIFO データ出力バス
DOP	出力	0、1、2、4	FIFO パリティ データ出力バス
FULL	出力	1	FIFO がフルであることを示します。
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。
EMPTY	出力	1	FIFO が空であることを示します。
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。
WRERR、RDERR	出力	1	WRERR は FIFO がフルの間に書き込みが実行されたことを示し、RDERR は FIFO が空の間に読み出しが実行されたことを示します。
WRCOUNT、RDCOUNT	出力	13	FIFO 書き込み/読み出しポインター

ポート名	方向	幅	機能
DI	入力	4、8、16、32	FIFO データ入力バス
DIP	入力	0、1、2、4	FIFO パリティ データ バス
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
RST	入力	1	FIFO 機能、フラグ、ポインタの非同期リセット (アクティブ High)。RESET は 3 クロック サイクル間アサートする必要があります。
WRCLK、RDCLK	入力	1	FIFO リード クロックおよびライト クロック (立ち上がりエッジで動作)

## 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	推奨

4 ビット幅でコンフィギュレーションされたプリミティブをインスタンス化するには、DIP ポートを論理 0 にし、DOP ポートは未接続のままにします。DI[3:0] および DO[3:0] は適切な入力および出力信号に、DI[31:4] は論理 0 に接続し、DO[31:4] は未接続のままにします。

9 ビット幅でコンフィギュレーションする場合は、DIP[0] ポートを適切なデータ入力に、DIP[3:1] を論理 0 に接続します。DOP[0] は適切なデータ出力に接続し、DOP[3:1] は未接続のままにします。DI[7:0] および DO[7:0] は適切な入力および出力信号に、DI[31:8] は論理 0 に接続し、DO[31:8] は未接続のままにします。

18 ビット幅でコンフィギュレーションする場合は、DIP[1:0] ポートを適切なデータ入力に、DIP[3:2] を論理 0 に接続します。DOP[1:0] は適切なデータ出力に接続し、DOP[3:2] は未接続のままにします。DI[15:0] および DO[15:0] は適切な入力および出力信号に、DI[31:16] は論理 0 に接続し、DO[31:16] は未接続のままにします。

36 ビット幅をコンフィギュレーションする場合は、DI、DIP、DO、および DOP 信号すべてを接続します。

いずれのコンフィギュレーションでも、未使用の DI または DIP 入力は論理 0 に接続し、未使用の DO または DOP ピンは未接続のままにする必要があります。EN\_SYM 属性を TRUE に設定して FIFO を同期に設定するときは、同じクロックソースを WRCLK および RDCLK に接続する必要があります。

非同期モード (EN\_SYN = FALSE) のときは、個別のクロック信号を使用します。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、ALMOSTEMPTY フラグと ALMOSTFULL フラグが 1 サイクル後にディASSERT 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 モデルではアーキテクチャのユーザー ガイドに示されているディASSERT レイテンシ サイクルのみが反映されます。WREN および RDEN は、対応するライト イネーブルおよびリード イネーブル信号/ロジックに接続し、RST は適切なリセット信号/ロジックに接続するか、使用しない場合は論理 0 に接続する必要があります。FULL、ALMOSTFULL、EMPTY、および ALMOSTEMPTY 出力フラグは、適切なデスティネーション ロジックに接続するか、使用しない場合は未接続のままにする必要があります。WRERR、RDERR、WRCOUNT、および RDCOUNT はオプションの出力で、使用しない場合は未接続のままにできます。インスタンス化コードに含まれるジェネリック (VHDL) またはインライン defparam (Verilog) を変更することですべての属性を設定し、目的どおりに FIFO が動作するようにします。

## 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 0	ALMOST_FULL 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 ビット値	すべて 0	ALMOST_EMPTY フラグをトリガーする RAM のデータ量を指定します。
FIRST_WORD_FALL_THROUGH	ブール代数	TRUE、FALSE	FALSE	TRUE に設定すると、RDEN がアサートされずに FIFO に最初書き込まれた値が DO に出力されます。
DATA_WIDTH	整数	4 ～ 36	4	FIFO のデータ幅を指定します。
EN_SYN	ブール代数	TRUE、FALSE	FALSE	FIFO が非同期 (独立した 2 つのクロック) または同期 (1 つのクロック) のいずれで動作しているかを示します。マルチレートの場合は DO_REG=1 を設定する必要があります。
DO_REG	整数	0、1	1	EN_SYN のデータパイプラインレジスタ
SIM_MODE	文字列	"SAFE"、"FAST"	"SAFE"	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。

## 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;

-- FIFO36: 32k+4k Parity Synchronous/Asynchronous BlockRAM FIFO BlockRAM Memory
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

FIFO36_inst : FIFO36
generic map (
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DATA_WIDTH => 4, -- Sets data width to 4, 9, 18, or 36
    DO_REG => 1, -- Enable output register ( 0 or 1)
    EN_SYN => FALSE, -- Must be 1 if the EN_SYN = FALSE
    FIRST_WORD_FALL_THROUGH => FALSE, -- Specified FIFO as Asynchronous (FALSE) or
    SIM_MODE => "SAFE", -- Synchronous (TRUE)
    -- Sets the FIFO FWFT to TRUE or FALSE
    -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
    -- Design Guide" for details
)
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    DO => DO, -- 32-bit data output
    DOP => DOP, -- 4-bit parity data output
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 13-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 13-bit write count output
    WRERR => WRERR, -- 1-bit write error
    DI => DI, -- 32-bit data input
    DIP => DIP, -- 4-bit parity input
    RDCLK => RDCLK, -- 1-bit read clock input
    RDEN => RDEN, -- 1-bit read enable input
    RST => RST, -- 1-bit reset input

```

```

    WRCLK => WRCLK,          -- 1-bit write clock input
    WREN => WREN             -- 1-bit write enable input
);

-- End of FIFO36_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// FIFO36: 32k+4k Parity Synchronous/Asynchronous BlockRAM FIFO
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

FIFO36 #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .ALMOST_FULL_OFFSET(13'h0080), // Sets almost full threshold
    .ALMOST_EMPTY_OFFSET(13'h0080), // Sets the almost empty threshold
    .DATA_WIDTH(4), // Sets data width to 4, 9, 18 or 36
    .DO_REG(1), // Enable output register (0 or 1)
    // Must be 1 if EN_SYN = "FALSE"
    .EN_SYN("FALSE"), // Specifies FIFO as Asynchronous ("FALSE")
    // or Synchronous ("TRUE")
    .FIRST_WORD_FALL_THROUGH("FALSE") // Sets the FIFO FWFT to "TRUE" or "FALSE"
) FIFO36_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit almost empty output flag
    .ALMOSTFULL(ALMOSTFULL), // 1-bit almost full output flag
    .DO(DO), // 32-bit data output
    .DOP(DOP), // 4-bit parity data output
    .EMPTY(EMPTY), // 1-bit empty output flag
    .FULL(FULL), // 1-bit full output flag
    .RDCOUNT(RDCOUNT), // 13-bit read count output
    .RDERR(RDERR), // 1-bit read error output
    .WRCOUNT(WRCOUNT), // 13-bit write count output
    .WRERR(WRERR), // 1-bit write error
    .DI(DI), // 32-bit data input
    .DIP(DIP), // 4-bit parity input
    .RDCLK(RDCLK), // 1-bit read clock input
    .RDEN(RDEN), // 1-bit read enable input
    .RST(RST), // 1-bit reset input
    .WRCLK(WRCLK), // 1-bit write clock input
    .WREN(WREN) // 1-bit write enable input
);

// End of FIFO36_inst instantiation

```

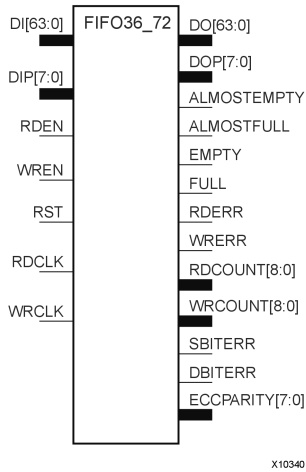
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## FIFO36\_72

**プリミティブ：72-Bit Wide by 512 Deep 36kb FIFO (First In, First Out) Block RAM Memory with ECC (Error Detection and Correction Circuitry)**



### 概要

Virtex®-5 デバイスにはブロック RAM が数個含まれ、FIFO、自動誤り訂正 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。このエレメントを使用すると、幅の広いデータパスが必要なときにブロック RAM を 36Kb FIFO コンフィギュレーションで使用できます。このコンポーネントは、72 ビット X 512 ワードの同期または非同期 FIFO RAM としてコンフィギュレーションできます。誤り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。また、この FIFO RAM では関連するすべての FIFO フラグおよびステータス信号も提供されます。

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 モデルではアーキテクチャのユーザー ガイドに示されているデアサート レイテンシ サイクルのみが反映されます。

**注記：** これよりワード数が多くデータ幅が狭いコンフィギュレーションには FIFO36 を使用し、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、FIFO18 または FIFO18\_36 を使用します。

### ポートの説明

ポート名	方向	幅	機能
DO	出力	64	FIFO データ出力バス
DOP	出力	8	FIFO パリティ データ出力バス
FULL	出力	1	FIFO がフルであることを示します。
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。
EMPTY	出力	1	FIFO が空であることを示します。
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。
WRERR, RDERR	出力	1	WRERR は FIFO がフルの間に書き込みが実行されたことを示し、RDERR は FIFO が空の間に読み出しが実行されたことを示します。

ポート名	方向	幅	機能
WRCOUNT、 RDCOUNT	出力	9	FIFO 書き込み/読み出しポインター
SBITTERR	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 ファンクションからのステータス出力。使用する場合は、EN_ECC_READ を TRUE にする必要があります。
DBITTERR	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。使用する場合は、EN_ECC_READ を TRUE にする必要があります。
ECCPARITY	出力	8	ECC デコーダーでメモリの誤り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
DI	入力	64	FIFO データ入力バス
DIP	入力	8	FIFO パリティ データ入力バス
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
RST	入力	1	FIFO 機能、フラグ、ポインターの非同期リセット (アクティブ High)。RESET は 3 クロック サイクル間アサートする必要があります。
WRCLK、 RDCLK	入力	1	FIFO リード クロックおよびライト クロック (立ち上がりエッジで動作)

## 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	推奨

DI、DIP、DO、および DOP ピンは、FIFO が ECC モードで動作していない場合は、対応する入力および出力データソースに接続する必要があります。ECC モードで動作している場合は、パリティビットが ECC の機能に必要なため、DI および DO ポートのみを使用する必要があります。使用可能なビット数以下を使用するときは、未使用の DI または DIP 入力を論理 0 に接続し、未使用の DO または DOP ピンは未接続のままにする必要があります。EN\_SYM 属性を TRUE に設定して FIFO を同期に設定するときは、同じクロックソースを WRCLK および RDCLK に接続する必要があります。

非同期モード (EN\_SYN = FALSE) のときは、個別のクロック信号を使用します。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、ALMOSTEMPTY フラグと ALMOSTFULL フラグが 1 サイクル後にディASSERT 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 モデルではアーキテクチャのユーザー ガイドに示されているディASSERT レイテンシ サイクルのみが反映されます。WREN および RDEN は、対応するライト イネーブルおよびリード イネーブル信号/ロジックに接続し、RST は適切なリセット信号/ロジックに接続するか、使用しない場合は論理 0 に接続する必要があります。

FULL、ALMOSTFULL、EMPTY、および ALMOSTEMPTY 出力フラグは、適切なデスティネーション ロジックに接続するか、使用しない場合は未接続のままにする必要があります。WRERR、RDERR、WRCOUNT、および RDCOUNT はオプションの出力で、使用しない場合は未接続のままにできます。ECC 機能を使用するには、EN\_ECC\_READ および EN\_ECC\_WRITE 属性を TRUE に設定する必要があります。誤り検出回路の動作を監視する場合は、SBITTERR、DBITTERR、および ECCPARITY 信号を適切なロジックに接続します。インスタンス化 コードに含まれるジェネリック (VHDL) またはインライン defparam (Verilog) を変更することですべての属性を設定し、目的どおりに FIFO が動作するようにします。

## 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	9 ビット値	080	ALMOST_FULL 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	9 ビット値	080	ALMOST_EMPTY フラグをトリガーする RAM のデータ量を指定します。
FIRST_WORD_FALL_THROUGH	ブール代数	TRUE、FALSE	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。
EN_SYN	ブール代数	TRUE、FALSE	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。
DO_REG	整数	0、1	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	TRUE、FALSE	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	TRUE、FALSE	FALSE	ECC エンコーダー回路をイネーブルにします。
SIM_MODE	文字列	"SAFE"、"FAST"	"SAFE"	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。

## 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;

-- FIFO36_72: 72x36k Synchronous/Asynchronous BlockRAM FIFO /w ECC
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

FIFO36_72_inst : FIFO36_72
generic map (
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DO_REG => 1,
    EN_ECC_READ => FALSE, -- Enable output register (0 or 1)
    EN_ECC_WRITE => FALSE, -- Must be 1 if EN_SYN = FALSE
    EN_SYN => FALSE, -- Enable ECC decoder, TRUE or FALSE
    FIRST_WORD_FALL_THROUGH => FALSE, -- Enable ECC encoder, TRUE or FALSE
    SIM_MODE => "SAFE") -- Specifies FIFO as Asynchronous (FALSE)
    -- or Synchronous (TRUE)
    -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
    -- Design Guide" for details

port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    DBITERR => DBITERR, -- 1-bit double bit error status output
    DO => DO, -- 64-bit data output
    DOP => DOP, -- 4-bit parity data output
    ECCPARITY => ECCPARITY, -- 8-bit generated error correction parity
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 9-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 9-bit write count output
    WRERR => WRERR, -- 1-bit write error
    DI => DI, -- 64-bit data input

```

```

DIP => DIP,           -- 4-bit parity input
RDCLK => RDCLK,       -- 1-bit read clock input
RDEN => RDEN,         -- 1-bit read enable input
RST => RST,           -- 1-bit reset input
WRCLK => WRCLK,       -- 1-bit write clock input
WREN => WREN          -- 1-bit write enable input
);

-- End of FIFO36_72_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// FIFO36_72: 72x36k Synchronous/Asynchronous BlockRAM FIFO w/ ECC
//           Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

FIFO36_72 #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .ALMOST_FULL_OFFSET(9'h080), // Sets almost full threshold
    .ALMOST_EMPTY_OFFSET(9'h080), // Sets the almost empty threshold
    .DO_REG(1), // Enable output register (0 or 1)
    // Must be 1 if EN_SYN = "FALSE"
    .EN_ECC_READ("FALSE"), // Enable ECC decoder, "TRUE" or "FALSE"
    .EN_ECC_WRITE("FALSE"), // Enable ECC encoder, "TRUE" or "FALSE"
    .EN_SYN("FALSE"), // Specifies FIFO as Asynchronous ("FALSE")
    // or Synchronous ("TRUE")
    .FIRST_WORD_FALL_THROUGH("FALSE") // Sets the FIFO FWFT to "TRUE" or "FALSE"
) FIFO36_72_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit almost empty output flag
    .ALMOSTFULL(ALMOSTFULL), // 1-bit almost full output flag
    .DBITERR(DBITERR), // 1-bit double bit error status output
    .DO(DO), // 64-bit data output
    .DOP(DOP), // 4-bit parity data output
    .ECCPARITY(ECCPARITY), // 8-bit generated error correction parity
    .EMPTY(EMPTY), // 1-bit empty output flag
    .FULL(FULL), // 1-bit full output flag
    .RDCOUNT(RDCOUNT), // 9-bit read count output
    .RDERR(RDERR), // 1-bit read error output
    .SBITERR(SBITERR), // 1-bit single bit error status output
    .WRCOUNT(WRCOUNT), // 9-bit write count output
    .WRERR(WRERR), // 1-bit write error
    .DI(DI), // 64-bit data input
    .DIP(DIP), // 4-bit parity input
    .RDCLK(RDCLK), // 1-bit read clock input
    .RDEN(RDEN), // 1-bit read enable input
    .RST(RST), // 1-bit reset input
    .WRCLK(WRCLK), // 1-bit write clock input
    .WREN(WREN) // 1-bit write enable input
);

// End of FIFO36_72_inst instantiation

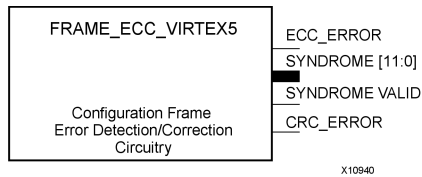
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## FRAME\_ECC\_VIRTEX5

プリミティブ：Virtex®-5 Configuration Frame Error Detection and Correction Circuitry



### 概要

このデザイン エLEMENTは、FPGA のコンフィギュレーション メモリ用の専用ビルトイン ECC (エラー検出および修正回路) をイネーブルにします。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	機能
ECCERROR	出力	1	フレーム ECC エラーが見つかったことを示します。SYNDROME が 0 以外 のときは 1、SYNDROME がすべて 0 のときは 0 になり、エラーが検出さ れなかったことを示します。
SYNDROME	出力	12	フレーム ECC エラーの発生場所を示します。 <ul style="list-style-type: none"> <li>エラーなし：SYNDROME[11:0]=0</li> <li>1 ビット エラー：SYNDROME[11]=0、SYNDROME[10:0]= &lt;フレーム でのエラーの位置&gt;</li> <li>2 ビット エラー：SYNDROME[11]=1、SYNDROME[10:0]= &lt;ドントケア&gt;</li> <li>2 ビット以上のエラー：不明な出力</li> </ul>
SYNDROMEVALID	出力	1	SYNDROME の値が有効であることを示します。
CRCERROR	出力	1	リードバック CRC エラーを示します。

### 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;

-- FRAME_ECC_VIRTEX5: Configuration Frame Error Correction Circuitry
--                               Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

FRAME_ECC_VIRTEX5_inst : FRAME_ECC_VIRTEX5
port map (
    CRCERROR => CRCERROR,    -- 1-bit output indicating a CRC error
    ECCERROR => ECCERROR,    -- 1-bit output indicating an ECC error
    SYNDROME => SYNDROME,    -- 12-bit output location of erroneous bit
    SYNDROMEVALID => SYNDROMEVALID -- 1-bit output indicating the
                                -- SYNDROME output is valid
);

-- End of FRAME_ECC_VIRTEX5_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// FRAME_ECC_VIRTEX5: Configuration Frame Error Correction Circuitry
//                               Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

FRAME_ECC_VIRTEX5 FRAME_ECC_VIRTEX5_inst (
    .CRCERROR(CRCERROR), // 1-bit output indicating a CRC error
    .ECCERROR(ECCERROR), // 1-bit output indicating an ECC error
    .SYNDROME(SYNDROME), // 12-bit output location of erroneous bit
    .SYNDROMEVALID(SYNDROMEVALID) // 1-bit output indicating the
                                // SYNDROME output is valid
);

// End of FRAME_ECC_VIRTEX5_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## GTP\_DUAL

プリミティブ：Dual Gigabit Transceiver

### 概要

このエレメントは、Virtex®-5 FPGA 用の消費電力を抑えたトランシーバーです。高度なコンフィギュレーションが可能で、FPGA のプログラマブル ロジック リソースに完全に統合されています。

### 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	推奨
マクロのサポート	不可

### 詳細情報

- ・ [『Virtex-5 FPGA RocketIO GTP トランシーバー ユーザー ガイド』\(UG196\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)
- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)

## GTX\_DUAL

### プリミティブ：Dual Gigabit Transceiver

#### 概要

このエレメントは、Virtex®-5 FPGA 用の消費電力を抑えたトランシーバーです。高度なコンフィギュレーションが可能で、FPGA のプログラマブル ロジック リソースに完全に統合されています。

#### デザインの入力方法

インスタンシエーション	不可
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

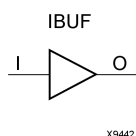
#### 詳細情報

- ・ [『Virtex-5 FPGA RocketIO GTX トランシーバー ユーザー ガイド』\(UG198\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)
- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)



## IBUF

プリミティブ：Input Buffer



## 概要

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

## ポートの説明

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

## デザインの入力方法

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

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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;

-- IBUF: Single-ended Input Buffer
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

IBUF_inst : IBUF
generic map (
    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 (インスタンス化)

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

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

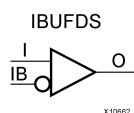
// End of IBUF_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## IBUFDS

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



### 概要

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

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- IBUFDS: Differential Input Buffer
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

IBUFDS_inst : IBUFDS
generic map (
    DIFF_TERM => FALSE, -- Differential Termination
    IOSTANDARD => "DEFAULT")
port map (
    O => O, -- Buffer output
    I => I, -- Diff_p buffer input (connect directly to top-level port)
    IB => IB -- Diff_n buffer input (connect directly to top-level port)
);

-- End of IBUFDS_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IBUFDS: Differential Input Buffer
//      Virtex-5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 14.5

IBUFDS #(
    .DIFF_TERM("FALSE"), // Differential Termination (Virtex-5, Spartan-3E/3A)
    .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)
);

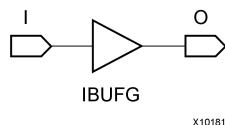
// End of IBUFDS_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## IBUFG

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



### 概要

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

### ポートの説明

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

### デザインの入力方法

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

### 使用可能な属性

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

### VHDL 記述 (インスタンス化)

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

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

-- IBUFG: Single-ended global clock input buffer
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

IBUFG_inst : IBUFG
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O, -- Clock buffer output
    I => I -- Clock buffer input (connect directly to top-level port)
);
```

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

## Verilog 記述 (インスタンス化)

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

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

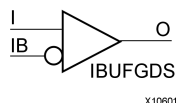
// End of IBUFG_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## IBUFGDS

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



### 概要

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

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスエーション)

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
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

IBUFGDS_inst : IBUFGDS
generic map (
    DIFF_TERM => FALSE, -- Differential Termination
    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
//      Virtex-5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 14.5

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

// End of IBUFGDS_inst instantiation
```

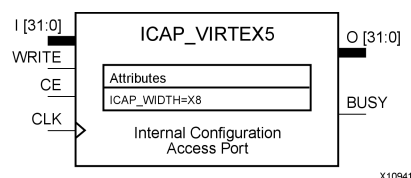
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## ICAP\_VIRTEX5

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



### 概要

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

### ポートの説明

ポート名	方向	幅	機能
O	出力	32	コンフィギュレーション データ出力バス
Busy	出力	1	Busy/Ready 出力
I	入力	32	コンフィギュレーション データ入力バス
WRITE	入力	1	アクティブ Low のライト入力
CE	入力	1	アクティブ Low のイネーブル入力
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

パラレル バスのビット順の詳細は、『Virtex-5 FPGA コンフィギュレーション ユーザー ガイド』を参照してください。

### 使用可能な属性

属性	データ型	値	デフォルト	説明
ICAP_WIDTH	文字列	"X8"、"X16"、 "X32"	"X8"	ICAP_VIRTEX5 で使用する入力および出力データ幅を指定します。

## 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;

-- ICAP_VIRTEX5: Internal Configuration Access Port
-- Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

ICAP_VIRTEX5_inst : ICAP_VIRTEX5
generic map (
    ICAP_WIDTH => "X8") -- "X8", "X16" or "X32"
port map (
    BUSY => BUSY,      -- Busy output
    O => O,             -- 32-bit data output
    CE => CE,          -- Clock enable input
    CLK => CLK,         -- Clock input
    I => I,             -- 32-bit data input
    WRITE => WRITE     -- Write input
);

-- End of ICAP_VIRTEX5_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// ICAP_VIRTEX5: Internal Configuration Access Port
// Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

ICAP_VIRTEX5 #(
    .ICAP_WIDTH("X8") // "X8", "X16" or "X32"
) ICAP_VIRTEX5_inst (
    .BUSY(BUSY),      // Busy output
    .O(O),            // 32-bit data output
    .CE(CE),          // Clock enable input
    .CLK(CLK),        // Clock input
    .I(I),            // 32-bit data input
    .WRITE(WRITE)     // Write input
);

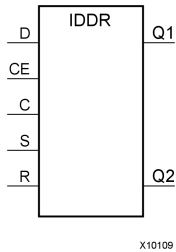
// End of ICAP_VIRTEX5_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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 で外部デュアル データレート (DDR) 信号を受信するための専用入力レジスタです。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 IODELAY 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	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	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 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。
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	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"、 "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"、"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: Double Data Rate Input Register with Set, Reset
--       and Clock Enable.
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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 "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 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.
//      Virtex-5/6
// Xilinx HDL Libraries Guide, version 14.5

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
);

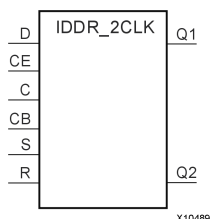
// End of IDDR_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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 で外部デュアル データレート (DDR) 信号を受信するための専用入力レジスタです。IDDR\_2CLK プリミティブは、DDR アプリケーションの立ち上がりデータと立ち下がりデータをキャプチャするのに 2 つのクロックが必要なアプリケーションでのみ使用してください。

- 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 の各立ち上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、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 IODELAY 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。
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、ローカル反転ではなく、グローバル配線 (DCM/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、DCM/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.
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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.
//           Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

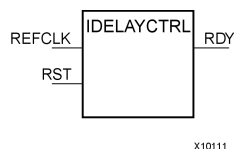
IDDR_2CLK #(
  .DDR_CLK_EDGE("OPPOSITE_EDGE"), // "OPPOSITE_EDGE", "SAME_EDGE"
                                   // or "SAME_EDGE_PIPELINED"
  .INIT_Q1(1'b0), // Initial value of Q1: 1'b0 or 1'b1
  .INIT_Q2(1'b0), // Initial value of Q2: 1'b0 or 1'b1
  .SRTYPE("SYNC") // Set/Reset type: "SYNC" or "ASYN"
) 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
```



## IDELAYCTRL

### プリミティブ：IDELAY Tap Delay Value Control



### 概要

このエレメントは、IODELAYE1 を使用する場合にインスタンス化する必要があります。これは、IDELAY または ISERDES プリミティブがインスタンス化されており、IOBDelay\_Type 属性が FIXED または VARIABLE に設定されている場合です。このモジュールは、一定の周波数の基準クロック REFCLK を使用して、プロセス、電圧、および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。これにより、正確な遅延調整が可能になります。

### ポートの説明

ポート名	方向	幅	機能
RDY	出力	1	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。
REFCLK	入力	1	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。
RST	入力	1	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。

**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、RDY 信号が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	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;

-- IDELAYCTRL : Input Delay Element Control
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

IDELAYCTRL_inst : IDELAYCTRL
port map (
    RDY => RDY,           -- 1-bit output indicates validity of the REFCLK
    REFCLK => REFCLK,     -- 1-bit reference clock input
    RST => RST            -- 1-bit reset input
);

-- End of IDELAYCTRL_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IDELAYCTRL: Input Delay Control Element (Must be used in conjunction with the IDELAY
//           when used in FIXED or VARIABLE tap-delay mode)
//           Virtex-5/6
// Xilinx HDL Libraries Guide, version 14.5

(* IODELAY_GROUP = "<iodelay_group_name>" *) // Specifies group name for associated IODELAYs and IDELAYCTRL
IDELAYCTRL IDELAYCTRL_inst (
    .RDY(RDY),           // 1-bit ready output
    .REFCLK(REFCLK),    // 1-bit reference clock input
    .RST(RST)           // 1-bit reset input
);

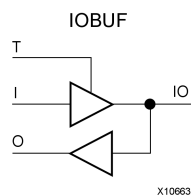
// End of IDELAYCTRL_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## IOBUF

プリミティブ：Bi-Directional Buffer



## 概要

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

## 論理表

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

## ポートの説明

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

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	I/O 規格として LVTTTL、LVCMOS12、LVCMOS15、LVCMOS18、LVCMOS25 または LVCMOS33 を使用する SelectIO™ バッファの出力の駆動電流 (mA) を選択します。
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。
SLEW	文字列	"SLOW"、"FAST"、 "QUIETIO"	"SLOW"	出力の立ち上がり時間と立ち下がり時間を設定します。この属性の最適な設定方法は、データシートを参照してください。

## VHDL 記述 (インスタンスレーション)

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
--      Virtex-5
--      Xilinx HDL Libraries Guide, version 14.5

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

-- End of IOBUF_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

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

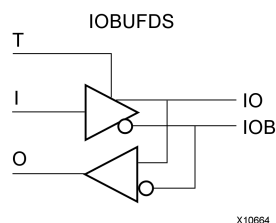
// End of IOBUF_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

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

## 論理表

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

## ポートの説明

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- IOBUFDS: Differential Bi-directional Buffer
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of IOBUFDS_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IOBUFDS: Differential Bi-directional Buffer
//      Virtex-5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 14.5

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

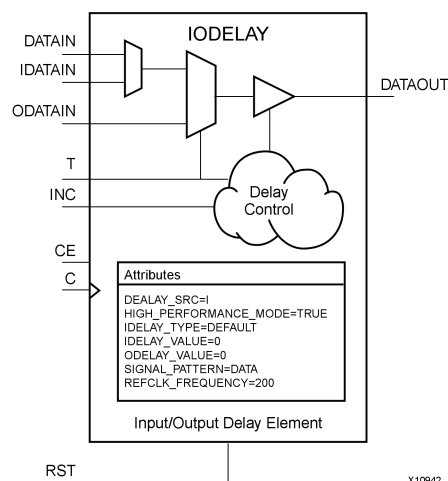
// End of IOBUFDS_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## IODELAY

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



### 概要

このデザイン エLEMENTは、FPGA の入力パスに固定遅延または可変遅延を、出力パスに固定遅延を追加します。この遅延は、チップへ入力されるデータ、またはチップから出力されるデータをアライメントするのに有効で、プロセス、温度、電圧が変動した場合のデータアライメントを監視できます。IODELAY は FPGA の I/O すべてに使用でき、IDELAYCTRL 回路と共に使用すると、正確に増分された遅延を追加できます。VARIABLE モードでは、入力パスを調整して遅延の量を増分または減分できます。出力遅延パスは、固定遅延でのみ使用できます。IODELAY を使用して、FPGA の内部パスに一定の遅延または可変遅延を追加することも可能です。ただし、IODELAY をこのように使用する場合、入力パス遅延または出力パス遅延に関連する I/O には使用できなくなります。

### ポートの説明

ポート名	方向	幅	機能
DATAOUT	出力	1	入力ポートで遅延されたデータ出力 (入力データ パス ロジックに接続)
IDATAIN	入力	1	I/O からのデータ入力 (ポートの I/O バッファに直接接続)。IDATAIN を使用する場合、DATAIN は論理 0 (グランド) に接続する必要があります。
ODATAIN	入力	1	出力データ パスのデータ入力 (出力データ ソースに接続)。ODATAIN を使用する場合、DATAIN は論理 0 (グランド) に接続する必要があります。
DATAIN	入力	1	内部データ パス遅延のデータ入力。DATAIN を使用する場合、IDATAIN および ODATAIN は論理 0 (グランド) に接続する必要があります。
T	入力	1	トライステート入力制御ピン。入力のみまたは内部遅延の場合は High に、出力のみの場合は Low にします。
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
INC	入力	1	インクリメント/デクリメント タップ遅延



ポート名	方向	幅	機能
C	入力	1	クロック入力 (必ず VARIABLE モードに設定)
RST	入力	1	アクティブ High の同期リセット。遅延チェーンを IDELAY_VALUE/ODELAY_VALUE タップにリセットします。値が指定されていない場合、デフォルトは 0 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

入力遅延処理の場合は、最上位の I/O ポート、入力バッファ、または I/O バッファのいずれかに直接 IDATAIN ピンを接続します。出力遅延処理の場合は、遅延させる出力データを供給するロジックに ODATAIN 入力を接続します。内部パス遅延処理の場合は、FPGA 内の適切なソースおよびデスティネーション ロジックに DATAIN ピンを接続します。IODELAY を内部信号遅延に使用する場合は、IDATAIN および ODATAIN を論理 0 (グラウンド) に接続する必要があります。

すべての場合に、DATAOUT は遅延させたデータが供給される I/O またはロジックに接続する必要があります。IODELAY を使用する場合、トリステート出力にするために T ピンを制御信号に接続します。出力遅延のみに IODELAY を使用する場合は、T ピンを論理 0 (グラウンド) に接続します。入力のみ IODELAY を使用する場合は、または内部信号を遅延させる場合は、T ピンを論理 1 (VCC) に接続します。IODELAY を VARIABLE モードに設定する場合は、CE ピン、INC ピン、C ピン、および RST ピンを適切な遅延制御信号に接続します。FIXED モードでは、これらのピンは論理 0 (グラウンド) に接続する必要があります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
HIGH_PERFORMANCE_MODE	ブール代数	TRUE、FALSE	FALSE	"TRUE" に設定すると、出力ジッターが減少します。
DELAY_SRC	文字列	"I"、"O"、"IO"、 "DATAIN"	"I"	IODELAY コンポーネントのソースを指定します。"I" に設定すると入力ポートまたは IBUF (入力モード)、"O" に設定すると出力ポートまたは OBUF (出力モード)、"IO" に設定するとポートに接続され、"DATAIN" に設定するとポートには接続されません (内部モード)。
IDELAY_TYPE	文字列	"DEFAULT"、 "FIXED"、 "VARIABLE"	"DEFAULT"	入力遅延タイプを固定 ("FIXED")、変動 ("VARIABLE")、またはホールド タイムを除去するデフォルト ("DEFAULT") のいずれかに設定します。
IDELAY_VALUE	整数	0 ～ 63	0	"FIXED" モードでは入力パス遅延のタップ数、"VARIABLE" モードでは初期遅延タップ値を指定します。
ODELAY_VALUE	整数	0 ～ 63	0	出力パス遅延のタップ数を指定します。

属性	データ型	値	デフォルト	説明
REFCLK_FREQUENCY	実数	190.00 ~ 210.00	200.00	IDELAYCTRL を使用する場合、 IODELAY の入力基準周波数を指定します。
SIGNAL_PATTERN	文字列	"CLOCK"、"DATA"	"DATA"	ディレイ カリキュレーターで使用 され、IODELAY ブロックを介した 伝搬遅延を指定します。"DATA" に設定すると、タップあたりの遅延 およびジッターが追加されます。ク ロック同様の信号では、ジッターは 増加されません。

## 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;

-- IDELAY: Input and/or Output Fixed/Variable Delay Element
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

IDELAY_inst : IDELAY
generic map (
    DELAY_SRC => "I", -- Specify which input port to be used
                    -- "I"=IDATAIN, "O"=ODATAIN, "DATAIN"=DATAIN, "IO"=Bi-directional
    HIGH_PERFORMANCE_MODE => TRUE, -- TRUE specifies lower jitter
                    -- at expense of more power
    IDELAY_TYPE => "FIXED", -- "FIXED" or "VARIABLE"
    IDELAY_VALUE => 0, -- 0 to 63 tap values
    ODELAY_VALUE => 0, -- 0 to 63 tap values
    REFCLK_FREQUENCY => 200.0, -- Frequency used for IDELAYCTRL
                    -- 175.0 to 225.0
    SIGNAL_PATTERN => "DATA") -- Input signal type, "CLOCK" or "DATA"
port map (
    DATAOUT => DATAOUT, -- 1-bit delayed data output
    C => C, -- 1-bit clock input
    CE => CE, -- 1-bit clock enable input
    DATAIN => DATAIN, -- 1-bit internal data input
    IDATAIN => IDATAIN, -- 1-bit input data input (connect to port)
    INC => INC, -- 1-bit increment/decrement input
    ODATAIN => ODATAIN, -- 1-bit output data input
    RST => RST, -- 1-bit active high, synch reset input
    T => T -- 1-bit 3-state control input
);

-- End of IDELAY_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// IODELAY: Input and/or Output Fixed/variable Delay Element
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

(* IODELAY_GROUP = "<iodelay_group_name>" *) // Specifies group name for associated IODELAYS and IDELAYCTRL
IODELAY # (
    .DELAY_SRC("I"), // Specify which input port to be used, "I"=IDATAIN,
                      // "O"=ODATAIN, "DATAIN"=DATAIN, "IO"=Bi-directional
    .HIGH_PERFORMANCE_MODE("TRUE"), // "TRUE" specifies lower jitter
                                      // at expense of more power
    .IDELAY_TYPE("FIXED"), // "FIXED" or "VARIABLE"
    .IDELAY_VALUE(0),      // 0 to 63 tap values
    .ODELAY_VALUE(0),      // 0 to 63 tap values
    .REFCLK_FREQUENCY(200.0), // Frequency used for IDELAYCTRL
                              // 175.0 to 225.0
    .SIGNAL_PATTERN("DATA") // Input signal type, "CLOCK" or "DATA"
) IODELAY_INST (
    .DATAOUT(DATAOUT), // 1-bit delayed data output
    .C(C),             // 1-bit clock input
    .CE(CE),           // 1-bit clock enable input
    .DATAIN(DATAIN),   // 1-bit internal data input
    .IDATAIN(IDATAIN), // 1-bit input data input (connect to port)
    .INC(INC), // 1-bit increment/decrement input
    .ODATAIN(ODATAIN), // 1-bit output data input
    .RST(RST), // 1-bit active high, synch reset input
    .T(T)      // 1-bit 3-state control input
);

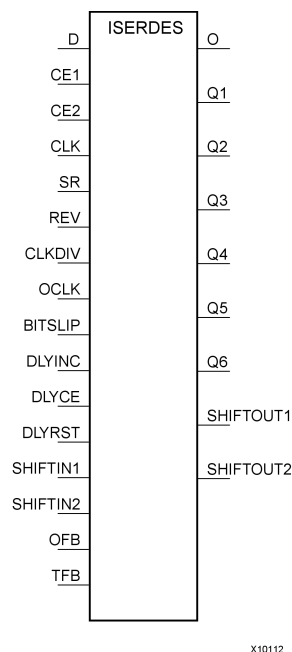
// End of IODELAY_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## ISERDES

### プリミティブ：Dedicated I/O Buffer Input Deserializer



## 概要

このデザイン エLEMENTを使用することにより、同期ソリューションを簡単にインプリメントできます。ISERDES は専用のソース同期 I/O アーキテクチャです。このモジュールを使用すると、ソース同期アプリケーションで FPGA のロジックリソースを節約でき、タイミングが複雑になるのを防ぎます。

このモジュールには、シリアル/パラレル コンバーター、シリアル遅延チェーン、ワード アライメント ユニット (BITSLLIP)、クロック イネーブル (CE) モジュールが含まれています。また、さまざまなアプリケーションに対応した複数のクロック入力があり、SelectIO™ 機能と共に使用できます。SERDES のサブモジュールの詳細は次のとおりです。

### 遅延チェーン モジュール

遅延チェーン モジュールは、入力データと転送されたクロックのタイミング関係を調整するための専用アーキテクチャです。このタイミング関係の調整は、ISERDES モジュール内に遅延を配置し、入力のスキューを調整することによって達成します。入力遅延チェーンは、あらかじめ設定しておくか (固定)、動作中に変更できます (可変)。また、このモジュールは IDELAYCTRL プリミティブと共に使用します。

遅延チェーン モジュールを使用するには、次の属性が必要です。

- ・ IOBDelay\_VALUE
- ・ IOBDelay
- ・ IOBDelay\_TYPE

IOBDelay\_VALUE は、使用する遅延タップの数を指定します。指定可能な値は 0 ～ 63 で、デフォルト値は 0 です。

IOBDelay 属性を IBUF、IFD、または BOTH に設定すると、遅延チェーンをそれぞれ組み合わせ出力 (O)、レジスタ付き出力 (Q1 ～ Q6)、または両方に使用できます。IOBDelay 属性を NONE に設定すると、遅延チェーンモジュールがバイパスされます。

IOBDELAY\_TYPE は、“DEFAULT”、“FIXED”、または “VARIABLE” に設定できます。“DEFAULT” に設定すると、ホールド タイムが 0 になります。“FIXED” に設定すると、遅延タップ値は IOBDELAY\_VALUE で設定した値になります。このモードでは、デバイスがプログラムされた後に値を変更することはできません。“VARIABLE” に設定すると、遅延タップの初期値が IOBDELAY\_VALUE の値に設定され、デバイスをプログラムした後に変更可能です。

遅延チェーン モードは、DLYRST、DLYCE、および DLYINC ピンを使用して制御します。これらのピンによる動作は、CLKDIV クロック信号に同期しています。DLYRST を High にアサートすると、遅延チェーンのタップ値が IOBDELAY\_VALUE で設定した値にリセットされます。遅延タップ値をインクリメント/デクリメントするには、DLYCE および DLYINC を使用します。遅延タップ値を変更するには、DLYCE を High にアサートする必要があります。DLYINC を 1 に設定するとインクリメントされ、0 に設定するとデクリメントされます。

次の表に、遅延チェーン制御ピンの値による動作を示します。

動作	DLYRST	DLYCE	DLYINC
IOBDELAY_VALUE の値にリセット	1	X	X
タップ値をインクリメント	0	1	1
タップ値をデクリメント	0	1	0
変化なし	0	0	X

注記：遅延チェーンのすべての動作は、CLKDIV に同期しています。

### シリアル/パラレル コンバーター

ISERDES モジュールのシリアル/パラレル コンバーターは、シリアル データを取り込み、2 ～ 6 のデータ幅に変換します。2 つの ISERDES モジュールをカスケード接続すると、データ入力幅を 7、8、または 10 ビットに拡張できます。データ幅を拡張するには、1 つの ISERDES をマスター モード、もう 1 つの ISERDES をスレーブ モードに設定し、スレーブの SHIFTIN ポートをマスターの SHIFTOUT ポートに接続します。スレーブでは、出力として Q3 ～ Q6 ポートのみを使用します。シリアル/パラレル コンバーターは、SDR または DDR モードの両方で使用できます。

このモジュールは、主に CLK および CLKDIV クロックで制御されます。次の表に、SDR および DDR の異なるモードにおける CLK と CLKDIV の関係を示します。

次の表に、シリアル/パラレル コンバーターの CLK と CLKDIV の関係を示します。

SDR のデータ幅	DDR のデータ幅	CLK	CLKDIV
2	4	2X	X
3	6	3X	X
4	8	4X	X
5	10	5X	X
6	–	6X	X
7	–	7X	X
8	–	8X	X

### CE モジュール

CE モジュールは、基本的には 2:1 パラレル/シリアル コンバーターです。このモジュールは CLKDIV クロック入力で制御され、シリアル/パラレル コンバーター モジュールのクロック イネーブル ポートを制御するのに使用されます。

## BITSLLIP モジュール

BITSLLIP モジュールは、出力シーケンスを並べ替えるバレル シフターのような機能を実行します。BITSLLIP を起動するたびに、出力パターンが変化します。BITSLLIP による並べ替えの最大回数は、パターンに含まれるビット数から 1 を引いた数になります (DATA\_WIDTH - 1)。BITSLLIP は、SDR および DDR の両方でサポートされます。SDR と DDR の出力の並べ替えは、異なるので注意してください。

BITSLLIP を使用するには BITSLLIP\_ENABLE 属性をオンに設定する必要があります。この属性をオフに設定すると、BITSLLIP モジュールはバイパスされます。

BITSLLIP の動作は、CLKDIV クロック入力に同期しています。BITSLLIP モジュールを起動するには、BITSLLIP ポートを CLKDIV の 1 サイクル分だけ High にアサートします。BITSLLIP ポートが CLKDIV の 1 サイクル分 High にアサートされると、BITSLLIP の動作が完了します。DDR モードでは、CLKDIV の 2 サイクル分経過するまで BITSLLIP の動作は安定しません。BITSLLIP の出力は、すべてレジスタ付き出力ポート (Q1 ~ Q6) に出力されます。

## その他の機能

## データ幅の拡張

ISERDES モジュールでは、7 ビット幅以上のデータ幅を回復できます。この機能を使用するには、2 つの ISERDES モジュールをインスタンス化する必要があります。この 2 つの ISERDES は、隣接したマスター/スレーブ ペアである必要があります。ISERDES\_MODE 属性を "MASTER" または "SLAVE" に設定し、ISERDES のペアを区別する必要があります。また、マスターの SHIFTOUT ポートをスレーブの SHIFTIN ポートに接続します。SDR および DDR モードでは、データ幅 7、8、10 がサポートされています。次の表に、SDR および DDR モードで使用可能なデータ幅を示します。

モード	幅
SDR	2、3、4、5、6、7、8
DDR	4、6、8、10

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	ISERDES モジュールのレジスタを介さない組み合わせ出力で、遅延チェーンの出力です。この出力ポートは、ISERDES モジュール内のすべてのサブモジュールをバイパスするようにコンフィギュレーションすることも可能です。BUFIO を駆動できます。
Q1 ~ Q6	出力	1 (それぞれ)	ISERDES モジュールのレジスタ付き出力です。これらの出力を使用すると、ISERDES のサブモジュール バスの次の組み合わせを入力として使用できます。 <ul style="list-style-type: none"> <li>遅延チェーン → シリアル/パラレル コンバーター → BITSLLIP モジュール</li> <li>遅延チェーン → シリアル/パラレル コンバーター</li> </ul> これらのポートは、2 ~ 6 ビットに設定できます。データ幅拡張モードでは、10 ビットまで拡張できます。
SHIFTOUT1、SHIFTOUT2	出力	1 (それぞれ)	データ入力を拡張するためのキャリー出力。スレーブの SHIFTIN1、SHIFTIN2 に接続します。
BITSLLIP	入力	1	BITSLLIP 動作を起動します。このピンを High にすると、BITSLLIP モジュールがイネーブルになります。
CE1、CE2	入力	1 (それぞれ)	CE モジュールに供給するクロック イネーブル入力

ポート名	方向	幅	機能
CLK	入力	1	<p>高速転送クロック入力。シリアル/パラレル コンバーターおよび BITSLLIP モジュールを駆動するのに使用するクロック入力です。CLK ポートは、次のいずれかのクロックリソースで駆動します。</p> <ul style="list-style-type: none"> <li>・ クロック領域内の 8 個のグローバル クロック ライン</li> <li>・ 2 個のリージョナル クロック ライン</li> <li>・ 6 個のクロック兼用 I/O (隣接したクロック領域内)</li> <li>・ FPGA (バイパスを介す)</li> </ul>
CLKDIV	入力	1	<p>高速転送分周クロック入力。シリアル/パラレル コンバーター、遅延チェーン、BITSLLIP モジュール、および CE モジュールを駆動するのに使用するクロック入力です。CLK ポートに接続されたクロックよりも低周波数のクロックを入力する必要があります。CLKDIV ポートは、次のいずれかのクロックリソースで駆動します。</p> <ul style="list-style-type: none"> <li>・ クロック領域内の 8 個のグローバル クロック ライン</li> <li>・ 2 個のリージョナル クロック ライン</li> </ul>
D	入力	1	<p>IOB からのシリアル入力データ。ISERDES モジュールにデータを入力するポートです。SelectIO 機能を使用して、I/O 規格を指定できます。</p>
DLYCE	入力	1	<p>遅延チェーンのインクリメント/デクリメントをイネーブルにします。</p>
DLYINC	入力	1	<p>遅延チェーン インクリメント/デクリメントピン。DLYCE ピンを High にアサートすると、DLYINC ピンの値に応じて遅延チェーンのタップ値がインクリメント/デクリメントされます。DLYINC ピンが High の場合はタップ値がインクリメントされ、Low の場合はデクリメントされます。</p>
DLYRST	入力	1	<p>遅延チェーン リセットピン。遅延チェーンのタップ数を IOBDelay_Value で設定された値にリセットします。値が設定されていない場合は、0 にリセットします。</p>
OCLK	入力	1	<p>メモリ インターフェイス アプリケーション用高速クロック。シリアル/パラレル コンバーターを駆動するのに使用するクロック入力です。OCLK ポートは、次のいずれかのクロックリソースで駆動します。</p> <ul style="list-style-type: none"> <li>・ クロック領域内の 8 個のグローバル クロック ライン</li> <li>・ 2 個のリージョナル クロック ライン</li> <li>・ 6 個のクロック兼用 I/O (隣接したクロック領域内)</li> <li>・ FPGA (バイパスを介す)</li> </ul> <p>このクロックは、ストロブ信号が必要なメモリ インターフェイスに適しています。</p>
REV	入力	1	<p>SR を反転します。内部テスト用です。SR を使用した場合、REV ピンによりストレージ エLEMENTが反対のステートになります。リセットがセットよりも優先されます。REV ピンは、ISERDES ではサポートされていません。</p>

ポート名	方向	幅	機能
SR	入力	1	セット/リセット入力。ストレージ エLEMENTの状態をユーザー制約ファイル (UCF) で設定した SRVAL 属性で設定された状態にします。SRVAL = 1 の場合は 1、SRVAL = 0 の場合は 0 になります。SR を使用した場合、REV ピンによりストレージ ELEMENTが反対のステートになります。リセットがセットよりも優先されます。ISERDES コンポーネントのすべてのレジスタに対し、SR ピンはアクティブ High に非同期リセットします。
SHIFTIN1、SHIFTIN2	入力	1 (それぞれ)	データ入力を拡張するためのキャリー入力です。マスターの SHIFTOUT1、SHIFTOUT2 に接続します。

## 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
BITSLIP_ENABLE	ブール代数	FALSE、TRUE	FALSE	BITSLIP コントローラーをイネーブルにします。
DATA_RATE	文字列	"SDR"、"DDR"	"DDR"	データ レートを指定します。
DATA_WIDTH	文字列	DATA_RATE = "DDR" の場合は "4"、"6"、"8"、"10"、 DATA_RATE = "SDR" の場合は "2"、"3"、"4"、"5"、"6"、"7"、"8"	"4"	シリアル/パラレル コンバーターの幅を指定します。SDR と DDR のどちらを選択するかで値が異なります。
INTERFACE_TYPE	文字列	"MEMORY"、 "NETWORKING"	"MEMORY"	ISERDES の使用モデルを指定します。
IOBDelay	文字列	"NONE"、"IBUF"、"IFD"、 "BOTH"	"NONE"	遅延チェーンを使用する ISERDES の出力を指定します。
IOBDelay_TYPE	文字列	"DEFAULT"、 "FIXED"、 "VARIABLE"	"DEFAULT"	遅延チェーンを固定モードまたは可変モードに指定します。
IOBDelay_VALUE	整数	0 ~ 63	0	タップ遅延の初期値を指定します。
NUM_CE	整数	1、2	2	クロック イネーブルの数を指定します。



## 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;

-- ISERDES: Input SERDES
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

ISERDES_inst : ISERDES
generic map (
    BITSLLIP_ENABLE => FALSE, -- TRUE/FALSE to enable bitslip controller
                                -- Must be "FALSE" in interface type is "MEMORY"
    DATA_RATE => "DDR", -- Specify data rate of "DDR" or "SDR"
    DATA_WIDTH => 4, -- Specify data width - For DDR 4,6,8, or 10
                                -- For SDR 2,3,4,5,6,7, or 8
    INTERFACE_TYPE => "MEMORY", -- Use model - "MEMORY" or "NETWORKING"
    IOBDELAY => "NONE", -- Specify outputs where delay chain will be applied
                                -- "NONE", "IBUF", "IFD", or "BOTH"
    IOBDELAY_TYPE => "DEFAULT", -- Set tap delay "DEFAULT", "FIXED", or "VARIABLE"
    IOBDELAY_VALUE => 0, -- Set initial tap delay to an integer from 0 to 63
    NUM_CE => 2, -- Define number of clock enables to an integer of 1 or 2
    SERDES_MODE => "MASTER") --Set SERDES mode to "MASTER" or "SLAVE"
port map (
    O => O, -- 1-bit output
    Q1 => Q1, -- 1-bit output
    Q2 => Q2, -- 1-bit output
    Q3 => Q3, -- 1-bit output
    Q4 => Q4, -- 1-bit output
    Q5 => Q5, -- 1-bit output
    Q6 => Q6, -- 1-bit output
    SHIFTOUT1 => SHIFTOUT1, -- 1-bit output
    SHIFTOUT2 => SHIFTOUT2, -- 1-bit output
    BITSLLIP => BITSLLIP, -- 1-bit input
    CE1 => CE1, -- 1-bit input
    CE2 => CE2, -- 1-bit input
    CLK => CLK, -- 1-bit input
    CLKDIV => CLKDIV, -- 1-bit input
    D => D, -- 1-bit input
    DLYCE => DLYCE, -- 1-bit input
    DLYINC => DLYINC, -- 1-bit input
    DLYRST => DLYRST, -- 1-bit input
    OCLK => OCLK, -- 1-bit input
    REV => '0', -- Must be tied to logic zero
    SHIF TIN1 => SHIF TIN1, -- 1-bit input
    SHIF TIN2 => SHIF TIN2, -- 1-bit input
    SR => SR -- 1-bit input
);

-- End of ISERDES_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ISERDES: Source Synchronous Input Deserializer
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

ISERDES #(
    .BITSLLIP_ENABLE("FALSE"), // "TRUE"/"FALSE" to enable bitslip controller
                                //      Must be "FALSE" if INTERFACE_TYPE set to "MEMORY"
    .DATA_RATE("DDR"), // Specify data rate of "DDR" or "SDR"
    .DATA_WIDTH(4), // Specify data width - for DDR 4,6,8, or 10
                    //      for SDR 2,3,4,5,6,7, or 8
    .INTERFACE_TYPE("MEMORY"), // Use model - "MEMORY" or "NETWORKING"
    .IOBDelay("NONE"), // Specify outputs where delay chain will be applied
                    //      "NONE", "IBUF", "IFD", or "BOTH"
    .IOBDelay_Type("DEFAULT"), // Set tap delay "DEFAULT", "FIXED", or "VARIABLE"
    .IOBDelay_Value(0), // Set initial tap delay to an integer from 0 to 63
    .NUM_CE(2), // Define number or clock enables to an integer of 1 or 2
    .SERDES_MODE("MASTER") // Set SERDES mode to "MASTER" or "SLAVE"
) ISERDES_inst (
    .O(0), // 1-bit combinatorial output
    .Q1(Q1), // 1-bit registered output
    .Q2(Q2), // 1-bit registered output
    .Q3(Q3), // 1-bit registered output
    .Q4(Q4), // 1-bit registered output
    .Q5(Q5), // 1-bit registered output
    .Q6(Q6), // 1-bit registered output
    .SHIFTOUT1(SHIFTOUT1), // 1-bit carry output
    .SHIFTOUT2(SHIFTOUT2), // 1-bit carry output
    .BITSLLIP(BITSLLIP), // 1-bit Bitslip input
    .CE1(CE1), // 1-bit clock enable input
    .CE2(CE2), // 1-bit clock enable input
    .CLK(CLK), // 1-bit clock input
    .CLKDIV(CLKDIV), // 1-bit divided clock input
    .D(D), // 1-bit serial data input
    .DLYCE(DLYCE), // 1-bit delay chain enable input
    .DLYINC(DLYINC), // 1-bit delay increment/decrement input
    .DLYRST(DLYRST), // 1-bit delay chain reset input
    .OCLK(OCLK), // 1-bit high-speed clock input
    .REV(1'b0), // Must be tied to logic zero
    .SHIFTIN1(SHIFTIN1), // 1-bit carry input
    .SHIFTIN2(SHIFTIN2), // 1-bit carry input
    .SR(SR) // 1-bit set/reset input
);

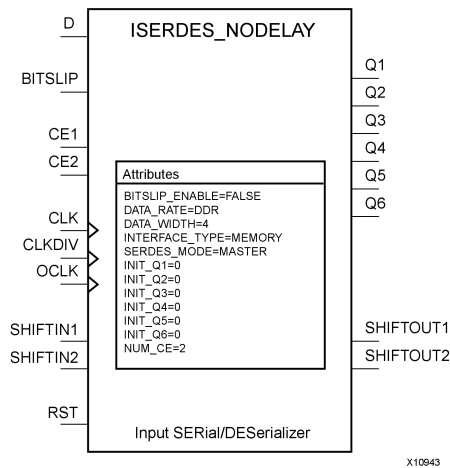
// End of ISERDES_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## ISERDES\_NODELAY

### プリミティブ：Input SERIAL/DESerializer



### 概要

ISERDES\_NODELAY は、シリアル/パラレル コンバーターで、高速のソース同期シリアル データを取り込みます。ISERDES\_NODELAY には、クロック管理およびデータ アライメントを実行するロジックが含まれており、シングル データ レート (SDR) またはダブル データ レート (DDR) のデータを、単一のインスタンス (MASTER) では 2 ～ 6 ビットのデータ幅に、カスケード接続された 2 つの ISERDES\_NODELAY (MASTER/SLAVE) では 7 ～ 10 ビットのデータ幅に変換できます。ISERDES\_NODELAY はメモリに使用でき、さまざまなタイプのデータ インターフェイス アプリケーションに接続できます。ISERDES\_NODELAY は IODELAY と共に使用して、入力シリアル データのアライメントを実行できます。DDR モードでは、データを取り込むために ISERDES\_NODELAY に 1 つのクロックまたは 2 つのクロックからクロックを供給できます。2 クロック モードを使用すると、パフォーマンスが向上する場合がありますが、必要なクロック リソース数および消費電力が増加し、配置配線で制限が発生する可能性があります。高速の I/O パフォーマンスが不要な場合は、1 クロック モードを使用してください。

### ポートの説明

ポート名	方向	幅	機能
Q1 ～ Q6	出力	1	レジスタ付きパラレル入力データ
SHIFTOUT1/SHIFTOUT2	出力	1	ISERDES_MODE を "MASTER" に設定しており、2 つの ISERDES_NODELAY をカスケード接続している場合に、スレーブの SHIFTIN1 と SHIFTIN2 入力に接続します。
D	入力	1	追加の入力遅延が必要な場合に、デザイン最上位の入力ポート、I/O ポート、または IODELAY に直接接続する入力データ
BITSLIP	入力	1	入力データ BITSLIP イネーブル
CE1/CE2	入力	1	入力データレジスタのクロック イネーブル
CLK	入力	1	プライマリ クロック入力
CLKB	入力	1	OSERDES の入力でのビット順は、ISERDES_NODELAY ブロックの出力でのものとは逆になります。詳細は、該当デバイスのユーザー ガイドを参照してください。

ポート名	方向	幅	機能
CLKDIV	入力	1	パラレル データに使用する分周クロック
OCLK	入力	1	通常メモリ インターフェイスに使用される高速の出力クロック
SHIFTIN1/SHIFTIN2	入力	1	ISERDES_MODE が "SLAVE" の場合、マスターの SHIFTOUT1 と SHIFTOUT2 出力に接続します。このピンはグラウンドに接続する必要があります。
RST	入力	1	SERDES のレジスタのアクティブ High の非同期リセット

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
BITSLIP_ENABLE	ブール代数	TRUE、FALSE	FALSE	BITSLIP をイネーブルにします。 "NETWORKING" モードでのみ使用できます。
DATA_RATE	文字列	"SDR"、"DDR"	"DDR"	シングル データレートまたはダブル データレートを指定します。
DATA_WIDTH	整数	DATA_RATE="DDR" の場合は 4、6、8、10、 DATA_RATE="SDR" の場合は 2、3、4、5、6、7、8	4	パラレル データ幅を選択します。
INTERFACE_TYPE	文字列	"MEMORY"、 "NETWORKING"	"MEMORY"	メモリ インターフェイスまたはネットワーク インターフェイスを指定します。
SERDES_MODE	文字列	"MASTER"、 "SLAVE"	"MASTER"	カスケード接続してデータ幅を拡張する場合に、ISERDES をマスター モードにするかスレーブ モードにするかを指定します。
NUM_CE	整数	1、2	2	使用するクロック イネーブルの数を指定します。

## 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;

-- ISERDES_NODELAY: Input SERIAL / DESerializer
--                               Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

ISERDES_NODELAY_inst : ISERDES_NODELAY
generic map (
    BITSLLIP_ENABLE => FALSE, -- TRUE/FALSE to enable bitslip controller
                                -- Must be "FALSE" in interface type is "MEMORY"
    DATA_RATE => "DDR", -- Specify data rate of "DDR" or "SDR"
    DATA_WIDTH => 4, -- Specify data width -
                        -- NETWORKING SDR: 2, 3, 4, 5, 6, 7, 8 : DDR 4, 6, 8, 10
                        -- MEMORY SDR N/A : DDR 4
    INTERFACE_TYPE => "MEMORY", -- Use model - "MEMORY" or "NETWORKING"
    NUM_CE => 2, -- Define number or clock enables to an integer of 1 or 2
    SERDES_MODE => "MASTER") --Set SERDES mode to "MASTER" or "SLAVE"
port map (
    Q1 => Q1, -- 1-bit registered SERDES output
    Q2 => Q2, -- 1-bit registered SERDES output
    Q3 => Q3, -- 1-bit registered SERDES output
    Q4 => Q4, -- 1-bit registered SERDES output
    Q5 => Q5, -- 1-bit registered SERDES output
    Q6 => Q6, -- 1-bit registered SERDES output
    SHIFTOUT1 => SHIFTOUT1, -- 1-bit cascade Master/Slave output
    SHIFTOUT2 => SHIFTOUT2, -- 1-bit cascade Master/Slave output
    BITSLLIP => BITSLLIP, -- 1-bit Bitslip enable input
    CE1 => CE1, -- 1-bit clock enable input
    CE2 => CE2, -- 1-bit clock enable input
    CLK => CLK, -- 1-bit master clock input
    CLKB => CLKB, -- 1-bit secondary clock input for DATA_RATE=DDR
    CLKDIV => CLKDIV, -- 1-bit divided clock input
    D => D, -- 1-bit data input, connects to IODELAY or input buffer
    OCLK => OCLK, -- 1-bit fast output clock input
    RST => RST, -- 1-bit asynchronous reset input
    SHIFTIN1 => SHIFTIN1, -- 1-bit cascade Master/Slave input
    SHIFTIN2 => SHIFTIN2 -- 1-bit cascade Master/Slave input
);

-- End of ISERDES_NODELAY_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ISERDES_NODELAY: Input SERIAL / DESerializer
//                               Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

ISERDES_NODELAY #(
    .BITSLIP_ENABLE("FALSE"), // "TRUE"/"FALSE" to enable bitflip controller
                                // Must be "FALSE" if INTERFACE_TYPE set to "MEMORY"
    .DATA_RATE("DDR"),        // Specify data rate of "DDR" or "SDR"
    .DATA_WIDTH(4),           // Specify data width -
                                // NETWORKING SDR: 2, 3, 4, 5, 6, 7, 8 : DDR 4, 6, 8, 10
                                // MEMORY SDR N/A : DDR 4
    .INTERFACE_TYPE("MEMORY"), // Use model - "MEMORY" or "NETWORKING"
    .NUM_CE(2),                // Number of clock enables used, 1 or 2
    .SERDES_MODE("MASTER")    // Set SERDES mode to "MASTER" or "SLAVE"
) ISERDES_NODELAY_inst (
    .Q1(Q1), // 1-bit registered SERDES output
    .Q2(Q2), // 1-bit registered SERDES output
    .Q3(Q3), // 1-bit registered SERDES output
    .Q4(Q4), // 1-bit registered SERDES output
    .Q5(Q5), // 1-bit registered SERDES output
    .Q6(Q6), // 1-bit registered SERDES output
    .SHIFTOUT1(SHIFTOUT1), // 1-bit cascade Master/Slave output
    .SHIFTOUT2(SHIFTOUT2), // 1-bit cascade Master/Slave output
    .BITSLIP(BITSLIP),     // 1-bit Bitflip enable input
    .CE1(CE1),              // 1-bit clock enable input
    .CE2(CE2),              // 1-bit clock enable input
    .CLK(CLK),              // 1-bit master clock input
    .CLKB(CLKB),            // 1-bit secondary clock input for DATA_RATE=DDR
    .CLKDIV(CLKDIV),        // 1-bit divided clock input
    .D(D),                  // 1-bit data input, connects to IODELAY or input buffer
    .OCLK(OCLK),            // 1-bit fast output clock input
    .RST(RST),              // 1-bit asynchronous reset input
    .SHIFTIN1(SHIFTIN1),    // 1-bit cascade Master/Slave input
    .SHIFTIN2(SHIFTIN2)     // 1-bit cascade Master/Slave input
);

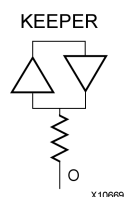
// End of ISERDES_NODELAY_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## KEEPER

### プリミティブ：KEEPER Symbol



### 概要

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

### ポートの説明

ポート名	方向	幅	機能
O	出力	1 ビット	キーパー出力

### デザインの入力方法

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

### VHDL 記述（インスタンス化）

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

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

-- KEEPER: I/O Buffer Weak Keeper
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of KEEPER_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// KEEPER: I/O Buffer Weak Keeper
//           All FPGA, CoolRunner-II
// Xilinx HDL Libraries Guide, version 14.5

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

// End of KEEPER_inst instantiation
```

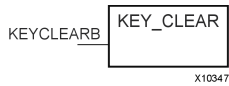
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## KEY\_CLEAR

**プリミティブ：Virtex-5 Configuration Encryption Key Erase**



### 概要

このデザイン エLEMENTを使用すると、内部ロジックからコンフィギュレーション暗号回路キー レジスタの内容を消去できます。

### ポートの説明

ポート名	方向	幅	機能
KEYCLEARB	入力	1	アクティブ Low の入力で、コンフィギュレーション暗号キーを消去します。

### 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;

-- KEY_CLEAR: Startup primitive for GSR, GTS or startup sequence control
--             Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

KEY_CLEAR_inst : KEY_CLEAR
  port map (
    KEYCLEARB => KEYCLEARB -- Active low key reset 1-bit input
  );

-- End of KEY_CLEAR_inst instantiation
```

### Verilog 記述（インスタンス化）

```
// KEY_CLEAR: Startup primitive for GSR, GTS or startup sequence control
//             Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

KEY_CLEAR KEY_CLEAR_inst (
  .KEYCLEARB(KEYCLEARB) // Active low key reset 1-bit input
);

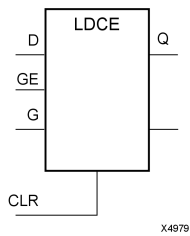
// End of KEY_CLEAR_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LDCE

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



### 概要

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

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

### 論理表

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

### デザインの入力方法

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

### 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LDCE: Transparent latch with Asynchronous Reset and
--       Gate Enable.
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LDCE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

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
);

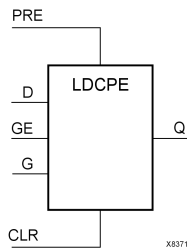
// End of LDCE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LDCPE

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



### 概要

このデザイン エLEMENTは、データ (D)、非同期クリア (CLR)、非同期プリセット (PRE)、ゲート イネーブル (GE) がある透過データ ラッチです。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。PRE が High、CLR が Low の場合、データ出力 (Q) は High にプリセットされます。ゲート入力 (G) と GE が High で CLR と PRE が Low の場合、Q にはデータ入力 (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。

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

### 論理表

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

### ポートの説明

ポート名	方向	幅	機能
Q	出力	1	データ出力
CLR	入力	1	非同期クリア/リセット入力
D	入力	1	データ入力
G	入力	1	ゲート入力
GE	入力	1	ゲート イネーブル入力
PRE	入力	1	非同期プリセット/セット入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	整数	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;

-- LDCPE: Transparent latch with Asynchronous Reset, Preset and
--       Gate Enable.
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

LDCPE_inst : LDCPE
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
  PRE => PRE       -- Asynchronous preset/set input
);

-- End of LDCPE_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LDCPE: Transparent latch with Asynchronous Reset, Preset and
//       Gate Enable.
//       Virtex-5, Spartan-3/3E/3A/3A DSP
// Xilinx HDL Libraries Guide, version 14.5

LDCPE #(
  .INIT(1'b0) // Initial value of latch (1'b0 or 1'b1)
) LDCPE_inst (
  .Q(Q),      // Data output
  .CLR(CLR),  // Asynchronous clear/reset input
  .D(D),      // Data input
  .G(G),      // Gate input
  .GE(GE),    // Gate enable input
  .PRE(PRE)   // Asynchronous preset/set input
);

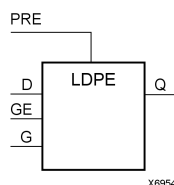
// End of LDCPE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LDPE

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



## 概要

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

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

## 論理表

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

## デザインの入力方法

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

## 使用可能な属性

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



## VHDL 記述 (インスタンスレーション)

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

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

-- LDPE: Transparent latch with Asynchronous Set and
--       Gate Enable.
--       Virtex-5
--       Xilinx HDL Libraries Guide, version 14.5

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

-- End of LDPE_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

LDPE #(
    .INIT(1'b0) // 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
);

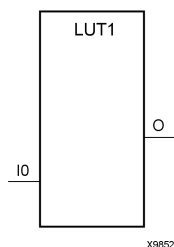
// End of LDPE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT1

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



## 概要

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

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

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

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

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

## 論理表

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT1: 1-input Look-Up Table with general output
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT1_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT1: 1-input Look-Up Table with general output
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

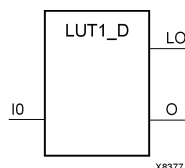
// End of LUT1_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT1\_D

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



### 概要

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

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

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

- ・ **論理表を使用する方法**：LUT の INIT 値を決定する一般的な方法。すべての入力をリストした 2 進数の論理表を作成し、出力の論理値を指定して、これらの出力値から INIT 文字列を作成します。
- ・ **論理式を使用する方法**：リストされた論理表の値に対応する LUT の各入力にパラメータを定義し、パラメータ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を指定する必要があります。

### 論理表

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

### デザインの入力方法

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

### 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT1_D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT1_D: 1-input Look-Up Table with general and local outputs
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

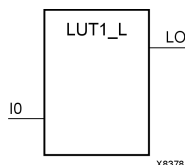
// End of LUT1_D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート : DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT1\_L

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



### 概要

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

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

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

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

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

### 論理表

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

### デザインの入力方法

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

### 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT1_L: 1-input Look-Up Table with local output
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT1_L_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT1_L: 1-input Look-Up Table with local output
//       Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

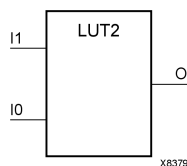
// End of LUT1_L_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT2

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



### 概要

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

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

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

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

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

### 論理表

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

### デザインの入力方法

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



## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT2: 2-input Look-Up Table with general output
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT2_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT2: 2-input Look-Up Table with general output
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

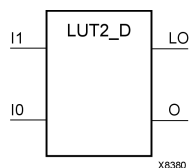
// End of LUT2_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT2\_D

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



### 概要

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

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

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

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

### 論理表

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

### デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT2_D: 2-input Look-Up Table with general and local outputs
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT2_D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT2_D: 2-input Look-Up Table with general and local outputs
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

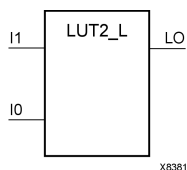
// End of LUT2_D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT2\_L

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



### 概要

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

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

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

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

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

### 論理表

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

### デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT2_L: 2-input Look-Up Table with local output
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT2_L_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT2_L: 2-input Look-Up Table with local output
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

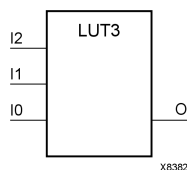
// End of LUT2_L_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT3

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



### 概要

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

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

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

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

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

### 論理表

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- LUT3: 3-input Look-Up Table with general output
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT3_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT3: 3-input Look-Up Table with general output
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

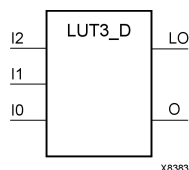
// End of LUT3_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT3\_D

### マクロ：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 スライス内の別の入力または高速バッファーに接続します。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

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

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

### 論理表

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



## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- LUT3_D: 3-input Look-Up Table with general and local outputs
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT3_D_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT3_D: 3-input Look-Up Table with general and local outputs
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

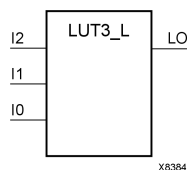
// End of LUT3_D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT3\_L

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

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

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

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

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

### 論理表

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- LUT3_L: 3-input Look-Up Table with local output
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT3_L_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT3_L: 3-input Look-Up Table with local output
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

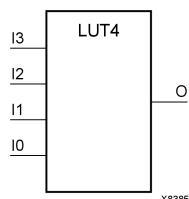
// End of LUT3_L_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT4

### マクロ：4-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 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ELEMENTは、バッファまたはインバーターの機能を果たします。これらのELEMENTは基本ブロックで、各 CLB スライスに 2 つ、各 CLB に 4 つずつあります。LUT には複数のバリエーションがあり、異なるタイミング モデルでレイアウト前のタイミングをより正確に予測する必要がある場合に使用できます。

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

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

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

### 論理表

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

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4: 4-input Look-Up Table with general output
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT4_inst instantiation

```

## Verilog 記述 (インスタンス化)

```
// LUT4: 4-input Look-Up Table with general output
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

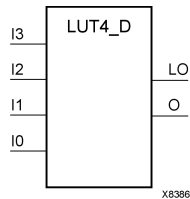
// End of LUT4_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT4\_D

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



### 概要

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

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

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

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

### 論理表

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

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_D: 4-input Look-Up Table with general and local outputs
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT4_D_inst instantiation

```



## Verilog 記述 (インスタンス化)

```
// LUT4_D: 4-input Look-Up Table with general and local outputs
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

LUT4_D #(
    .INIT(16'h0000) // Specify LUT Contents
) LUT4_D_inst (
    .LO(LO), // LUT local output
    .O(0),   // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3)  // LUT input
);

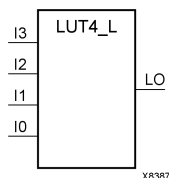
// End of LUT4_D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT4\_L

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



### 概要

このデザイン エレメント 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。

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

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

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

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

### 論理表

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

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_L: 4-input Look-Up Table with local output
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of LUT4_L_inst instantiation

```

## Verilog 記述（インスタンスレーション）

```
// LUT4_L: 4-input Look-Up Table with local output
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

LUT4_L #(
    .INIT(16'h0000) // Specify LUT Contents
) LUT4_L_inst (
    .LO(LO), // LUT local output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3)  // LUT input
);

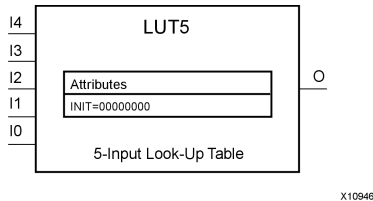
// End of LUT4_L_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT5

### プリミティブ：5-Input Lookup Table with General Output



### 概要

このデザイン エLEMENTは、入力 5 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 32 ビット ROM (5 ビットのアドレス指定) または 5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT5 が 1 つの場合はスライス内の LUT6 に、2 つの場合は多少の制限はありますが 1 つの LUT6 にパックできます。LUT5、LUT5\_L、および LUT5\_D の機能は同じですが、LUT5\_L および LUT5\_D では、LUT5 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT5\_L では LUT5 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT5\_D では LUT5 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT5 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 32 ビットの 16 進数を設定する必要があります。入力が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を 32'h80000000 (VHDL では X"80000000") に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (5 入力の AND ゲート)。また、Verilog で INIT 値を 32'hffffff (VHDL では X"FFFFFFFE") に設定すると、入力がすべて 0 の場合以外は出力は 1 になります (5 入力 OR ゲート)。

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

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

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

### 論理表

入力					出力
I4	I3	I2	I1	I0	LO
0	0	0	0	0	INIT[0]
0	0	0	0	1	INIT[1]
0	0	0	1	0	INIT[2]
0	0	0	1	1	INIT[3]
0	0	1	0	0	INIT[4]
0	0	1	0	1	INIT[5]

入力					出力
I4	I3	I2	I1	I0	LO
0	0	1	1	0	INIT[6]
0	0	1	1	1	INIT[7]
0	1	0	0	0	INIT[8]
0	1	0	0	1	INIT[9]
0	1	0	1	0	INIT[10]
0	1	0	1	1	INIT[11]
0	1	1	0	0	INIT[12]
0	1	1	0	1	INIT[13]
0	1	1	1	0	INIT[14]
0	1	1	1	1	INIT[15]
1	0	0	0	0	INIT[16]
1	0	0	0	1	INIT[17]
1	0	0	1	0	INIT[18]
1	0	0	1	1	INIT[19]
1	0	1	0	0	INIT[20]
1	0	1	0	1	INIT[21]
1	0	1	1	0	INIT[22]
1	0	1	1	1	INIT[23]
1	1	0	0	0	INIT[24]
1	1	0	0	1	INIT[25]
1	1	0	1	0	INIT[26]
1	1	0	1	1	INIT[27]
1	1	1	0	0	INIT[28]
1	1	1	0	1	INIT[29]
1	1	1	1	0	INIT[30]
1	1	1	1	1	INIT[31]

INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述（インスタンス化）

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

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

-- LUT5: 5-input Look-Up Table with general output
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

LUT5_inst : LUT5
generic map (
  INIT => X"00000000") -- Specify LUT Contents
port map (
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4 -- LUT input
);

-- End of LUT5_inst instantiation
```

## Verilog 記述（インスタンス化）

```
// LUT5: 5-input Look-Up Table with general output
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

LUT5 #(
  .INIT(32'h00000000) // Specify LUT Contents
) LUT5_inst (
  .O(O), // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4) // LUT input
);

// End of LUT5_inst instantiation
```

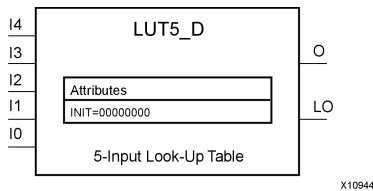
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## LUT5\_D

### プリミティブ：5-Input Lookup Table with General and Local Outputs



### 概要

このデザイン エLEMENTは、入力 5 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 32 ビット ROM (5 ビットのアドレス指定) または 5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT5 が 1 つの場合はスライス内の LUT6 に、2 つの場合は多少の制限はありますが 1 つの LUT6 にパックできます。LUT5、LUT5\_L、および LUT5\_D の機能は同じですが、LUT5\_L および LUT5\_D では、LUT5 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT5\_L では LUT5 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT5\_D では LUT5 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT5 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 32 ビットの 16 進数を設定する必要があります。入力が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を 32'h80000000 (VHDL では X"80000000") に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (5 入力の AND ゲート)。また、Verilog で INIT 値を 32'hffffff (VHDL では X"FFFFFFFE") に設定すると、入力がすべて 0 の場合以外は出力は 1 になります (5 入力 OR ゲート)。

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

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

### 論理表

入力					出力	
I4	I3	I2	I1	I0	O	LO
0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	1	1	INIT[3]	INIT[3]
0	0	1	0	0	INIT[4]	INIT[4]
0	0	1	0	1	INIT[5]	INIT[5]

入力					出力	
I4	I3	I2	I1	I0	O	LO
0	0	1	1	0	INIT[6]	INIT[6]
0	0	1	1	1	INIT[7]	INIT[7]
0	1	0	0	0	INIT[8]	INIT[8]
0	1	0	0	1	INIT[9]	INIT[9]
0	1	0	1	0	INIT[10]	INIT[10]
0	1	0	1	1	INIT[11]	INIT[11]
0	1	1	0	0	INIT[12]	INIT[12]
0	1	1	0	1	INIT[13]	INIT[13]
0	1	1	1	0	INIT[14]	INIT[14]
0	1	1	1	1	INIT[15]	INIT[15]
1	0	0	0	0	INIT[16]	INIT[16]
1	0	0	0	1	INIT[17]	INIT[17]
1	0	0	1	0	INIT[18]	INIT[18]
1	0	0	1	1	INIT[19]	INIT[19]
1	0	1	0	0	INIT[20]	INIT[20]
1	0	1	0	1	INIT[21]	INIT[21]
1	0	1	1	0	INIT[22]	INIT[22]
1	0	1	1	1	INIT[23]	INIT[23]
1	1	0	0	0	INIT[24]	INIT[24]
1	1	0	0	1	INIT[25]	INIT[25]
1	1	0	1	0	INIT[26]	INIT[26]
1	1	0	1	1	INIT[27]	INIT[27]
1	1	1	0	0	INIT[28]	INIT[28]
1	1	1	0	1	INIT[29]	INIT[29]
1	1	1	1	0	INIT[30]	INIT[30]
1	1	1	1	1	INIT[31]	INIT[31]

INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	5 入力 LUT 出力
LO	出力	1	内部 CLB 接続用の 5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- LUT5_D: 5-input Look-Up Table with general and local outputs
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

LUT5_D_inst : LUT5_D
generic map (
  INIT => X"00000000") -- Specify LUT contents
port map (
  LO => LO, -- LUT local output
  O  => O,  -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4  -- LUT input
);

-- End of LUT5_D_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT5_D: 5-input Look-Up Table with general and local outputs
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

LUT5_D #(
  .INIT(32'h00000000) // Specify LUT Contents
) LUT5_D_inst (
  .LO(LO), // LUT local output
  .O(O),   // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4)  // LUT input
);

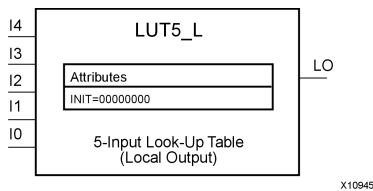
// End of LUT5_D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT5\_L

### プリミティブ：5-Input Lookup Table with Local Output



### 概要

このデザイン エLEMENTは、入力 5 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 32 ビット ROM (5 ビットのアドレス指定) または 5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT5 が 1 つの場合はスライス内の LUT6 に、2 つの場合は多少の制限はありますが 1 つの LUT6 にパックできます。LUT5、LUT5\_L、および LUT5\_D の機能は同じですが、LUT5\_L および LUT5\_D では、LUT5 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT5\_L では LUT5 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT5\_D では LUT5 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT5 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 32 ビットの 16 進数を設定する必要があります。入力が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を 32'h80000000 (VHDL では X"80000000") に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (5 入力の AND ゲート)。また、Verilog で INIT 値を 32'hffffff (VHDL では X"FFFFFFFE") に設定すると、入力がすべて 0 の場合以外は出力は 1 になります (5 入力 OR ゲート)。

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

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

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

### 論理表

入力					出力
I4	I3	I2	I1	I0	LO
0	0	0	0	0	INIT[0]
0	0	0	0	1	INIT[1]
0	0	0	1	0	INIT[2]
0	0	0	1	1	INIT[3]
0	0	1	0	0	INIT[4]
0	0	1	0	1	INIT[5]

入力					出力
I4	I3	I2	I1	I0	LO
0	0	1	1	0	INIT[6]
0	0	1	1	1	INIT[7]
0	1	0	0	0	INIT[8]
0	1	0	0	1	INIT[9]
0	1	0	1	0	INIT[10]
0	1	0	1	1	INIT[11]
0	1	1	0	0	INIT[12]
0	1	1	0	1	INIT[13]
0	1	1	1	0	INIT[14]
0	1	1	1	1	INIT[15]
1	0	0	0	0	INIT[16]
1	0	0	0	1	INIT[17]
1	0	0	1	0	INIT[18]
1	0	0	1	1	INIT[19]
1	0	1	0	0	INIT[20]
1	0	1	0	1	INIT[21]
1	0	1	1	0	INIT[22]
1	0	1	1	1	INIT[23]
1	1	0	0	0	INIT[24]
1	1	0	0	1	INIT[25]
1	1	0	1	0	INIT[26]
1	1	0	1	1	INIT[27]
1	1	1	0	0	INIT[28]
1	1	1	0	1	INIT[29]
1	1	1	1	0	INIT[30]
1	1	1	1	1	INIT[31]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値					

## ポートの説明

ポート名	方向	幅	機能
L0	出力	1	内部 CLB 接続用の 6/5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- LUT5_L: 5-input Look-Up Table with local output
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

LUT5_L_inst : LUT5_L
generic map (
  INIT => X"00000000") -- Specify LUT Contents
port map (
  LO => LO, -- LUT local output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4  -- LUT input
);

-- End of LUT5_L_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT5_L: 5-input Look-Up Table with local output
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

LUT5_L #(
  .INIT(32'h00000000) // Specify LUT Contents
) LUT5_L_inst (
  .LO(LO), // LUT local output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4)  // LUT input
);

// End of LUT5_L_inst instantiation
```

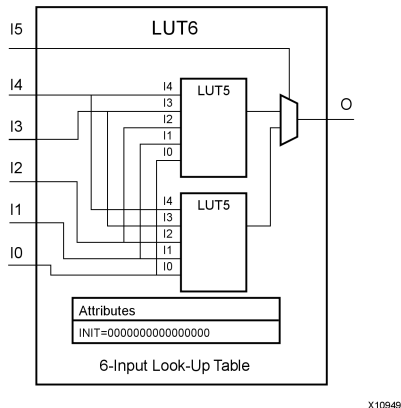
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## 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、入力がすべて 0 の場合以外は出力は 1 になります (6 入力 OR ゲート)。

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

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

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

### 論理表

入力						出力
I5	I4	I3	I2	I1	I0	O
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]
0	0	0	0	1	0	INIT[2]

入力						出力
I5	I4	I3	I2	I1	I0	O
0	0	0	0	1	1	INIT[3]
0	0	0	1	0	0	INIT[4]
0	0	0	1	0	1	INIT[5]
0	0	0	1	1	0	INIT[6]
0	0	0	1	1	1	INIT[7]
0	0	1	0	0	0	INIT[8]
0	0	1	0	0	1	INIT[9]
0	0	1	0	1	0	INIT[10]
0	0	1	0	1	1	INIT[11]
0	0	1	1	0	0	INIT[12]
0	0	1	1	0	1	INIT[13]
0	0	1	1	1	0	INIT[14]
0	0	1	1	1	1	INIT[15]
0	1	0	0	0	0	INIT[16]
0	1	0	0	0	1	INIT[17]
0	1	0	0	1	0	INIT[18]
0	1	0	0	1	1	INIT[19]
0	1	0	1	0	0	INIT[20]
0	1	0	1	0	1	INIT[21]
0	1	0	1	1	0	INIT[22]
0	1	0	1	1	1	INIT[23]
0	1	1	0	0	0	INIT[24]
0	1	1	0	0	1	INIT[25]
0	1	1	0	1	0	INIT[26]
0	1	1	0	1	1	INIT[27]
0	1	1	1	0	0	INIT[28]
0	1	1	1	0	1	INIT[29]
0	1	1	1	1	0	INIT[30]
0	1	1	1	1	1	INIT[31]
1	0	0	0	0	0	INIT[32]
1	0	0	0	0	1	INIT[33]
1	0	0	0	1	0	INIT[34]
1	0	0	0	1	1	INIT[35]
1	0	0	1	0	0	INIT[36]
1	0	0	1	0	1	INIT[37]

入力						出力
I5	I4	I3	I2	I1	I0	O
1	0	0	1	1	0	INIT[38]
1	0	0	1	1	1	INIT[39]
1	0	1	0	0	0	INIT[40]
1	0	1	0	0	1	INIT[41]
1	0	1	0	1	0	INIT[42]
1	0	1	0	1	1	INIT[43]
1	0	1	1	0	0	INIT[44]
1	0	1	1	0	1	INIT[45]
1	0	1	1	1	0	INIT[46]
1	0	1	1	1	1	INIT[47]
1	1	0	0	0	0	INIT[48]
1	1	0	0	0	1	INIT[49]
1	1	0	0	1	0	INIT[50]
1	1	0	0	1	1	INIT[51]
1	1	0	1	0	0	INIT[52]
1	1	0	1	0	1	INIT[53]
1	1	0	1	1	0	INIT[54]
1	1	0	1	1	1	INIT[55]
1	1	1	0	0	0	INIT[56]
1	1	1	0	0	1	INIT[57]
1	1	1	0	1	0	INIT[58]
1	1	1	0	1	1	INIT[59]
1	1	1	1	0	0	INIT[60]
1	1	1	1	0	1	INIT[61]
1	1	1	1	1	0	INIT[62]
1	1	1	1	1	1	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値						

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	6/5 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- LUT6: 6-input Look-Up Table with general output
-- Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

LUT6_inst : LUT6
generic map (
  INIT => X"0000000000000000") -- Specify LUT Contents
port map (
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4, -- LUT input
  I5 => I5 -- LUT input
);

-- End of LUT6_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT6: 6-input Look-Up Table with general output
// Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

LUT6 #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_inst (
  .O(O), // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4), // LUT input
  .I5(I5) // LUT input
);

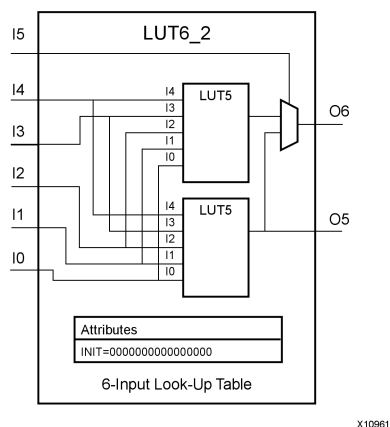
// End of LUT6_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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、または入力と論理値を共有する 6 入力および 5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT6\_2 は、スライスに含まれる 4 個のルックアップ テーブル (LUT) のいずれかにマップされます。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数を設定する必要があります。入力が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば、Verilog で INIT 値を `64'hfffffffffe` (VHDL では `X"FFFFFFFFFFFFFFFE"`) に設定すると、入力がすべて 0 の場合以外は O6 出力は 1 になり、I[4:0] がすべて 0 の場合以外は O5 出力は 1 になります (5 または 6 入力の OR ゲート)。INIT 値の下位半分 (ビット 31:0) は O5 出力のロジック ファンクションに適用されます。

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

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

### 論理表

入力						出力	
I5	I4	I3	I2	I1	I0	O5	O6
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	0	1	1	INIT[3]	INIT[3]
0	0	0	1	0	0	INIT[4]	INIT[4]

入力						出力	
0	0	0	1	0	1	INIT[5]	INIT[5]
0	0	0	1	1	0	INIT[6]	INIT[6]
0	0	0	1	1	1	INIT[7]	INIT[7]
0	0	1	0	0	0	INIT[8]	INIT[8]
0	0	1	0	0	1	INIT[9]	INIT[9]
0	0	1	0	1	0	INIT[10]	INIT[10]
0	0	1	0	1	1	INIT[11]	INIT[11]
0	0	1	1	0	0	INIT[12]	INIT[12]
0	0	1	1	0	1	INIT[13]	INIT[13]
0	0	1	1	1	0	INIT[14]	INIT[14]
0	0	1	1	1	1	INIT[15]	INIT[15]
0	1	0	0	0	0	INIT[16]	INIT[16]
0	1	0	0	0	1	INIT[17]	INIT[17]
0	1	0	0	1	0	INIT[18]	INIT[18]
0	1	0	0	1	1	INIT[19]	INIT[19]
0	1	0	1	0	0	INIT[20]	INIT[20]
0	1	0	1	0	1	INIT[21]	INIT[21]
0	1	0	1	1	0	INIT[22]	INIT[22]
0	1	0	1	1	1	INIT[23]	INIT[23]
0	1	1	0	0	0	INIT[24]	INIT[24]
0	1	1	0	0	1	INIT[25]	INIT[25]
0	1	1	0	1	0	INIT[26]	INIT[26]
0	1	1	0	1	1	INIT[27]	INIT[27]
0	1	1	1	0	0	INIT[28]	INIT[28]
0	1	1	1	0	1	INIT[29]	INIT[29]
0	1	1	1	1	0	INIT[30]	INIT[30]
0	1	1	1	1	1	INIT[31]	INIT[31]
1	0	0	0	0	0	INIT[0]	INIT[32]
1	0	0	0	0	1	INIT[1]	INIT[33]
1	0	0	0	1	0	INIT[2]	INIT[34]
1	0	0	0	1	1	INIT[3]	INIT[35]
1	0	0	1	0	0	INIT[4]	INIT[36]
1	0	0	1	0	1	INIT[5]	INIT[37]
1	0	0	1	1	0	INIT[6]	INIT[38]
1	0	0	1	1	1	INIT[7]	INIT[39]
1	0	1	0	0	0	INIT[8]	INIT[40]

入力						出力	
1	0	1	0	0	1	INIT[9]	INIT[41]
1	0	1	0	1	0	INIT[10]	INIT[42]
1	0	1	0	1	1	INIT[11]	INIT[43]
1	0	1	1	0	0	INIT[12]	INIT[44]
1	0	1	1	0	1	INIT[13]	INIT[45]
1	0	1	1	1	0	INIT[14]	INIT[46]
1	0	1	1	1	1	INIT[15]	INIT[47]
1	1	0	0	0	0	INIT[16]	INIT[48]
1	1	0	0	0	1	INIT[17]	INIT[49]
1	1	0	0	1	0	INIT[18]	INIT[50]
1	1	0	0	1	1	INIT[19]	INIT[51]
1	1	0	1	0	0	INIT[20]	INIT[52]
1	1	0	1	0	1	INIT[21]	INIT[53]
1	1	0	1	1	0	INIT[22]	INIT[54]
1	1	0	1	1	1	INIT[23]	INIT[55]
1	1	1	0	0	0	INIT[24]	INIT[56]
1	1	1	0	0	1	INIT[25]	INIT[57]
1	1	1	0	1	0	INIT[26]	INIT[58]
1	1	1	0	1	1	INIT[27]	INIT[59]
1	1	1	1	0	0	INIT[28]	INIT[60]
1	1	1	1	0	1	INIT[29]	INIT[61]
1	1	1	1	1	0	INIT[30]	INIT[62]
1	1	1	1	1	1	INIT[31]	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値							

## ポートの説明

ポート名	方向	幅	機能
O6	出力	1	6/5 LUT 出力
O5	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

## デザインの入力方法

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



## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	LUT5/6 の出力ファンクションを指定します。

## VHDL 記述 (インスタンス化)

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

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

-- LUT6_2: 6-input 2 output Look-Up Table
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

LUT6_2_inst : LUT6_2
generic map (
  INIT => X"0000000000000000" -- Specify LUT Contents
port map (
  O6 => O6, -- 6/5-LUT output (1-bit)
  O5 => O5, -- 5-LUT output (1-bit)
  I0 => I0, -- LUT input (1-bit)
  I1 => I1, -- LUT input (1-bit)
  I2 => I2, -- LUT input (1-bit)
  I3 => I3, -- LUT input (1-bit)
  I4 => I4, -- LUT input (1-bit)
  I5 => I5  -- LUT input (1-bit)
);

-- End of LUT6_2_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT6_2: 6-input, 2 output Look-Up Table
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

LUT6_2 #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_2_inst (
  .O6(O6), // 1-bit LUT6 output
  .O5(O5), // 1-bit lower LUT5 output
  .I0(I0), // 1-bit LUT input
  .I1(I1), // 1-bit LUT input
  .I2(I2), // 1-bit LUT input
  .I3(I3), // 1-bit LUT input
  .I4(I4), // 1-bit LUT input
  .I5(I5)  // 1-bit LUT input (fast MUX select only available to O6 output)
);

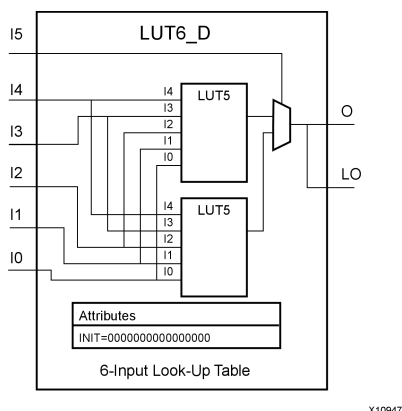
// End of LUT6_2_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT6\_D

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



### 概要

このデザイン エLEMENTは、入力 6 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 64 ビット ROM (6 ビットのアドレス指定) または 6 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT6 は、4 個のルックアップ テーブル (LUT) のいずれかにマップされます。LUT6、LUT6\_L、および LUT6\_D の機能は同じですが、LUT6\_L および LUT6\_D では、LUT6 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT6\_L では LUT6 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT6\_D では LUT6 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数値を設定する必要があります。入力 が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を 64'h8000000000000000 (VHDL では X"8000000000000000") に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値を 64'hffffffffffffff (VHDL では X"FFFFFFFFFFFFFFFF") に設定すると、入力すべてが 0 の場合以外は出力は 1 になります (6 入力 OR ゲート)。

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

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

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

### 論理表

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	0	1	0	INIT[2]	INIT[2]

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
0	0	0	0	1	1	INIT[3]	INIT[3]
0	0	0	1	0	0	INIT[4]	INIT[4]
0	0	0	1	0	1	INIT[5]	INIT[5]
0	0	0	1	1	0	INIT[6]	INIT[6]
0	0	0	1	1	1	INIT[7]	INIT[7]
0	0	1	0	0	0	INIT[8]	INIT[8]
0	0	1	0	0	1	INIT[9]	INIT[9]
0	0	1	0	1	0	INIT[10]	INIT[10]
0	0	1	0	1	1	INIT[11]	INIT[11]
0	0	1	1	0	0	INIT[12]	INIT[12]
0	0	1	1	0	1	INIT[13]	INIT[13]
0	0	1	1	1	0	INIT[14]	INIT[14]
0	0	1	1	1	1	INIT[15]	INIT[15]
0	1	0	0	0	0	INIT[16]	INIT[16]
0	1	0	0	0	1	INIT[17]	INIT[17]
0	1	0	0	1	0	INIT[18]	INIT[18]
0	1	0	0	1	1	INIT[19]	INIT[19]
0	1	0	1	0	0	INIT[20]	INIT[20]
0	1	0	1	0	1	INIT[21]	INIT[21]
0	1	0	1	1	0	INIT[22]	INIT[22]
0	1	0	1	1	1	INIT[23]	INIT[23]
0	1	1	0	0	0	INIT[24]	INIT[24]
0	1	1	0	0	1	INIT[25]	INIT[25]
0	1	1	0	1	0	INIT[26]	INIT[26]
0	1	1	0	1	1	INIT[27]	INIT[27]
0	1	1	1	0	0	INIT[28]	INIT[28]
0	1	1	1	0	1	INIT[29]	INIT[29]
0	1	1	1	1	0	INIT[30]	INIT[30]
0	1	1	1	1	1	INIT[31]	INIT[31]
1	0	0	0	0	0	INIT[32]	INIT[32]
1	0	0	0	0	1	INIT[33]	INIT[33]
1	0	0	0	1	0	INIT[34]	INIT[34]
1	0	0	0	1	1	INIT[35]	INIT[35]
1	0	0	1	0	0	INIT[36]	INIT[36]
1	0	0	1	0	1	INIT[37]	INIT[37]

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
1	0	0	1	1	0	INIT[38]	INIT[38]
1	0	0	1	1	1	INIT[39]	INIT[39]
1	0	1	0	0	0	INIT[40]	INIT[40]
1	0	1	0	0	1	INIT[41]	INIT[41]
1	0	1	0	1	0	INIT[42]	INIT[42]
1	0	1	0	1	1	INIT[43]	INIT[43]
1	0	1	1	0	0	INIT[44]	INIT[44]
1	0	1	1	0	1	INIT[45]	INIT[45]
1	0	1	1	1	0	INIT[46]	INIT[46]
1	0	1	1	1	1	INIT[47]	INIT[47]
1	1	0	0	0	0	INIT[48]	INIT[48]
1	1	0	0	0	1	INIT[49]	INIT[49]
1	1	0	0	1	0	INIT[50]	INIT[50]
1	1	0	0	1	1	INIT[51]	INIT[51]
1	1	0	1	0	0	INIT[52]	INIT[52]
1	1	0	1	0	1	INIT[53]	INIT[53]
1	1	0	1	1	0	INIT[54]	INIT[54]
1	1	0	1	1	1	INIT[55]	INIT[55]
1	1	1	0	0	0	INIT[56]	INIT[56]
1	1	1	0	0	1	INIT[57]	INIT[57]
1	1	1	0	1	0	INIT[58]	INIT[58]
1	1	1	0	1	1	INIT[59]	INIT[59]
1	1	1	1	0	0	INIT[60]	INIT[60]
1	1	1	1	0	1	INIT[61]	INIT[61]
1	1	1	1	1	0	INIT[62]	INIT[62]
1	1	1	1	1	1	INIT[63]	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値							

## ポートの説明

ポート名	方向	幅	機能
O6	出力	1	6/5 LUT 出力
O5	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- LUT6_D: 6-input Look-Up Table with general and local outputs
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

LUT6_D_inst : LUT6_D
generic map (
  INIT => X"0000000000000000") -- Specify LUT contents
port map (
  LO => LO, -- LUT local output
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4, -- LUT input
  I5 => I5 -- LUT input
);

-- End of LUT6_D_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT6_D: 6-input Look-Up Table with general and local outputs
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

LUT6_D #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_D_inst (
  .LO(LO), // LUT local output
  .O(O), // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4), // LUT input
  .I5(I5) // LUT input
);

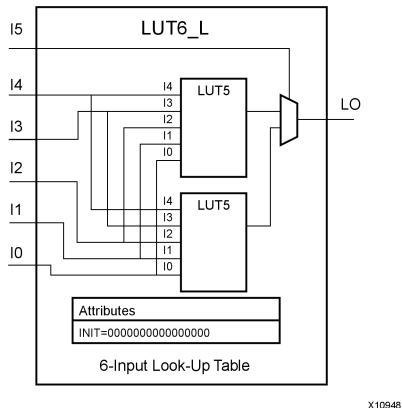
// End of LUT6_D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## LUT6\_L

**プリミティブ：6-Input Lookup Table with Local Output**



### 概要

このデザイン エLEMENTは、入力 6 個、出力 1 個を持つルックアップ テーブル (LUT) で、非同期 64 ビット ROM (6 ビットのアドレス指定) または 6 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT6 は、4 個のルックアップ テーブル (LUT) のいずれかにマップされます。LUT6、LUT6\_L、および LUT6\_D の機能は同じですが、LUT6\_L および LUT6\_D では、LUT6 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続できます。LUT6\_L では LUT6 からの接続が 1 つのスライスまたは CLB 内に制限されるのに対し、LUT6\_D では LUT6 の出力を内部スライス/CLB ロジックおよび外部ロジックの両方に接続できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数値を設定する必要があります。入力 が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を `64'h8000000000000000` (VHDL では `X"8000000000000000"`) に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値を `64'hfffffffffffffffe` (VHDL では `X"FFFFFFFFFFFFFFFFFFFE"`) に設定すると、入力すべてが 0 の場合以外は出力は 1 になります (6 入力 OR ゲート)。

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

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

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

### 論理表

入力						出力
I5	I4	I3	I2	I1	I0	LO
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]
0	0	0	0	1	0	INIT[2]

入力						出力
I5	I4	I3	I2	I1	I0	LO
0	0	0	0	1	1	INIT[3]
0	0	0	1	0	0	INIT[4]
0	0	0	1	0	1	INIT[5]
0	0	0	1	1	0	INIT[6]
0	0	0	1	1	1	INIT[7]
0	0	1	0	0	0	INIT[8]
0	0	1	0	0	1	INIT[9]
0	0	1	0	1	0	INIT[10]
0	0	1	0	1	1	INIT[11]
0	0	1	1	0	0	INIT[12]
0	0	1	1	0	1	INIT[13]
0	0	1	1	1	0	INIT[14]
0	0	1	1	1	1	INIT[15]
0	1	0	0	0	0	INIT[16]
0	1	0	0	0	1	INIT[17]
0	1	0	0	1	0	INIT[18]
0	1	0	0	1	1	INIT[19]
0	1	0	1	0	0	INIT[20]
0	1	0	1	0	1	INIT[21]
0	1	0	1	1	0	INIT[22]
0	1	0	1	1	1	INIT[23]
0	1	1	0	0	0	INIT[24]
0	1	1	0	0	1	INIT[25]
0	1	1	0	1	0	INIT[26]
0	1	1	0	1	1	INIT[27]
0	1	1	1	0	0	INIT[28]
0	1	1	1	0	1	INIT[29]
0	1	1	1	1	0	INIT[30]
0	1	1	1	1	1	INIT[31]
1	0	0	0	0	0	INIT[32]
1	0	0	0	0	1	INIT[33]
1	0	0	0	1	0	INIT[34]
1	0	0	0	1	1	INIT[35]
1	0	0	1	0	0	INIT[36]
1	0	0	1	0	1	INIT[37]



入力						出力
I5	I4	I3	I2	I1	I0	LO
1	0	0	1	1	0	INIT[38]
1	0	0	1	1	1	INIT[39]
1	0	1	0	0	0	INIT[40]
1	0	1	0	0	1	INIT[41]
1	0	1	0	1	0	INIT[42]
1	0	1	0	1	1	INIT[43]
1	0	1	1	0	0	INIT[44]
1	0	1	1	0	1	INIT[45]
1	0	1	1	1	0	INIT[46]
1	0	1	1	1	1	INIT[47]
1	1	0	0	0	0	INIT[48]
1	1	0	0	0	1	INIT[49]
1	1	0	0	1	0	INIT[50]
1	1	0	0	1	1	INIT[51]
1	1	0	1	0	0	INIT[52]
1	1	0	1	0	1	INIT[53]
1	1	0	1	1	0	INIT[54]
1	1	0	1	1	1	INIT[55]
1	1	1	0	0	0	INIT[56]
1	1	1	0	0	1	INIT[57]
1	1	1	0	1	0	INIT[58]
1	1	1	0	1	1	INIT[59]
1	1	1	1	0	0	INIT[60]
1	1	1	1	0	1	INIT[61]
1	1	1	1	1	0	INIT[62]
1	1	1	1	1	1	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値						

## ポートの説明

ポート名	方向	幅	機能
LO	出力	1	6/5 入力 LUT 出力または内部 CLB 接続
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- LUT6_L: 6-input Look-Up Table with local output
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

LUT6_L_inst : LUT6_L
generic map (
  INIT => X"0000000000000000") -- Specify LUT Contents
port map (
  LO => LO, -- LUT local output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4, -- LUT input
  I5 => I5 -- LUT input
);

-- End of LUT6_L_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// LUT6_L: 6-input Look-Up Table with local output
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

LUT6_L #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_L_inst (
  .LO(LO), // LUT local output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4), // LUT input
  .I5(I5) // LUT input
);

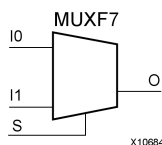
// End of LUT6_L_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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 エレメントは、2 つの LUT6 ルックアップ テーブルを組み合わせ、7 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、LUT6 の 2 つのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O は汎用インターコネクトです。

このほか、ローカル出力を持つ MUXF7\_D および MUXF7\_L があり、異なるタイミング モデルでレイアウト前のタイミングをより正確に予測する必要がある場合に使用できます。

### 論理表

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

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
I0	入力	1	入力 (MUXF6 LO 出力に接続)
I1	入力	1	入力 (MUXF6 LO 出力に接続)
S	入力	1	MUX への入力セレクト

### デザインの入力方法

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

## VHDL 記述 (インスタンスレーション)

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

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

-- MUXF7: CLB MUX to tie two LUT6's together with general output
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

MUXF7_inst : MUXF7
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,     -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,     -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S       -- Input select to MUX
);

-- End of MUXF7_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF7: CLB MUX to tie two LUT6's together with general output
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

MUXF7 MUXF7_inst (
    .O(O),       // Output of MUX to general routing
    .I0(I0),     // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .I1(I1),     // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .S(S)        // Input select to MUX
);

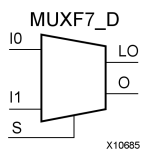
// End of MUXF7_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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 エレメントは、2 つの LUT6 ルックアップ テーブルを組み合わせて、7 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、LUT6 の 2 つのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は汎用インターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

「MUXF7」および「MUXF7\_L」も参照してください。

### 論理表

入力			出力	
S	I0	I1	O	LO
0	I0	X	I0	I0
1	X	I1	I1	I1
X	0	0	0	0
X	1	1	1	1

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF6 LO 出力に接続)
I1	入力	1	入力 (MUXF6 LO 出力に接続)
S	入力	1	MUX への入力セレクト

### デザインの入力方法

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

## VHDL 記述 (インスタンスレーション)

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

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

-- MUXF7_D: CLB MUX to tie two LUT6's together with general and local outputs
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

MUXF7_D_inst : MUXF7_D
port map (
    LO => LO,  -- Output of MUX to local routing
    O  => O,   -- Output of MUX to general routing
    IO => IO,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S  => S    -- Input select to MUX
);

-- End of MUXF7_D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF7_D: CLB MUX to tie two LUT6's together with general and local outputs
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

MUXF7_D MUXF7_D_inst (
    .LO(LO),  // Output of MUX to local routing
    .O(O),   // Output of MUX to general routing
    .IO(IO), // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .I1(I1), // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .S(S)    // Input select to MUX
);

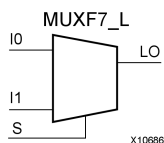
// End of MUXF7_D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## MUXF7\_L

プリミティブ：2-to-1 look-up table Multiplexer with Local Output



### 概要

このデザイン エLEMENTは、2 つの LUT6 ルックアップ テーブルを組み合わせ、7 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、LUT6 の 2 つのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

「MUXF7」および「MUXF7\_D」も参照してください。

### 論理表

入力			出力
S	I0	I1	LO
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

### ポートの説明

ポート名	方向	幅	機能
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力
I1	入力	1	入力
S	入力	1	MUX への入力セレクト

### デザインの入力方法

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



## VHDL 記述 (インスタンス化)

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_L: CLB MUX to tie two LUT6's together with local output
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

MUXF7_L_inst : MUXF7_L
port map (
    LO => LO, -- Output of MUX to local routing
    IO => IO, -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1, -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S    -- Input select to MUX
);

-- End of MUXF7_L_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// MUXF7_L: CLB MUX to tie two LUT6's together with local output
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

MUXF7_L MUXF7_L_inst (
    .LO(IO), // Output of MUX to local routing
    .IO(IO), // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .I1(I1), // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .S(S)    // Input select to MUX
);

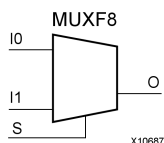
// End of MUXF7_L_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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、ルックアップ テーブル、MUXF5、MUXF6、および MUXF7 と組み合わせて、スライス 8 個に 8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサーを作成するためのマルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

### 論理表

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

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

### デザインの入力方法

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

## VHDL 記述 (インスタンスレーション)

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

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

-- MUXF8: CLB MUX to tie two MUXF7's together with general output
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

MUXF8_inst : MUXF8
port map (
    O => O,      -- Output of MUX to general routing
    IO => IO,    -- Input (tie to MUXF7 LO out)
    I1 => I1,    -- Input (tie to MUXF7 LO out)
    S => S       -- Input select to MUX
);

-- End of MUXF8_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF8: CLB MUX to tie two MUXF7's together with general output
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

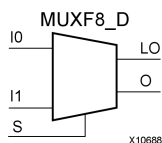
// End of MUXF8_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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、ルックアップ テーブル、MUXF5、MUXF6、および MUXF7 を組み合わせて、スライス 8 個に 8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサーを作成するためのマルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は汎用インターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

### 論理表

入力			出力	
S	I0	I1	O	LO
0	I0	X	I0	I0
1	X	I1	I1	I1
X	0	0	0	0
X	1	1	1	1

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

### デザインの入力方法

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

## VHDL 記述 (インスタンスレーション)

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

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

-- MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

MUXF8_D_inst : MUXF8_D
port map (
    LO => LO,  -- Output of MUX to local routing
    O  => O,   -- Output of MUX to general routing
    IO => IO,  -- Input (tie to MUXF7 LO out)
    I1 => I1,  -- Input (tie to MUXF7 LO out)
    S  => S    -- Input select to MUX
);

-- End of MUXF8_D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

MUXF8_D MUXF8_D_inst (
    .LO(LO),  // Output of MUX to local routing
    .O(O),   // Output of MUX to general routing
    .IO(IO), // Input (tie to MUXF7 LO out)
    .I1(I1), // Input (tie to MUXF7 LO out)
    .S(S)    // Input select to MUX
);

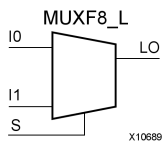
// End of MUXF8_D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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 エレメントは、ルックアップ テーブル、MUXF5、MUXF6、および MUXF7 を組み合わせて、スライス 8 個に 8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサーを作成するためのマルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

### 論理表

入力			出力
S	I0	I1	LO
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

### ポートの説明

ポート名	方向	幅	機能
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

### デザインの入力方法

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

## VHDL 記述 (インスタンスレーション)

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

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

-- MUXF8_L: CLB MUX to tie two MUXF7's together with local output
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

MUXF8_L_inst : MUXF8_L
port map (
    LO => LO,  -- Output of MUX to local routing
    IO => IO,  -- Input (tie to MUXF7 LO out)
    I1 => I1,  -- Input (tie to MUXF7 LO out)
    S => S     -- Input select to MUX
);

-- End of MUXF8_L_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF8_L: CLB MUX to tie two MUXF7's together with local output
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

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

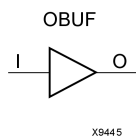
// End of MUXF8_L_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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、トライステートにならない (常に駆動される) FPGA デバイス ピンへの出力信号を駆動するために使用します。デザインのすべての出力ポートに OBUF、OBUFT、OBUFDS、OBUFTDS のいずれかを接続する必要があります。

この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、LVTTL 規格が使用され、DRIVE 制約と SLOW または FAST 制約を使用して駆動電流とスルー レートを選択できます。デフォルトでは、DRIVE は 12mA、スルー レートは SLOW に設定されています。

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	最上位出力ポートに直接接続される OBUF の出力
I	入力	1	OBUF の入力。出力ポートを駆動するロジックに接続

### デザインの入力方法

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

### 使用可能な属性

属性	データ型	値	デフォルト	説明
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;

-- OBUF: Single-ended Output Buffer
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

OBUF_inst : OBUF
generic map (
    DRIVE => 12,
    IOSTANDARD => "DEFAULT",
    SLEW => "SLOW")
port map (
    O => O,      -- Buffer output (connect directly to top-level port)
    I => I       -- Buffer input
);

-- End of OBUF_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// OBUF: Single-ended Output Buffer
//      All devices
// Xilinx HDL Libraries Guide, version 14.5

OBUF #(
    .DRIVE(12), // Specify the output drive strength
    .IOSTANDARD("DEFAULT"), // Specify the output I/O standard
    .SLEW("SLOW") // Specify the output slew rate
) OBUF_inst (
    .O(O), // Buffer output (connect directly to top-level port)
    .I(I) // Buffer input
);

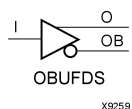
// End of OBUF_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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、低電圧の差動信号 (1.8V CMOS) をサポートする単一の出力バッファです。内部回路を外部から分離し、チップから出力する信号の駆動電流を供給します。出力には 2 つの異なるポート (O および OB) があり、これらのポートをそれぞれ「マスター」および「スレーブ」と呼びます。マスターとスレーブは MYNET と MYNETB のように、同じ論理信号の反対の状態を示します。

### 論理表

入力	出力	
I	O	OB
0	0	1
1	1	0

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	Diff_p 出力 (最上位ポートに直接接続)
OB	出力	1	Diff_n 出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力

### デザインの入力方法

インスタンスエーション	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。

## 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
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

OBUFDS_inst : OBUFDS
generic map (
  IOSTANDARD => "DEFAULT")
port map (
  O => O,      -- Diff_p output (connect directly to top-level port)
  OB => OB,    -- Diff_n output (connect directly to top-level port)
  I => I       -- Buffer input
);

-- End of OBUFDS_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// OBUFDS: Differential Output Buffer
//       Virtex-5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 14.5

OBUFDS #(
  .IOSTANDARD("DEFAULT") // Specify the output I/O standard
) OBUFDS_inst (
  .O(O),      // Diff_p output (connect directly to top-level port)
  .OB(OB),    // Diff_n output (connect directly to top-level port)
  .I(I)       // Buffer input
);

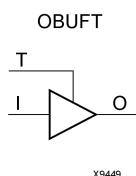
// End of OBUFDS_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート : DC 特性およびスイッチ特性』\(DS202\)](#)

## OBUFT

### プリミティブ：3-State Output Buffer with Active Low Output Enable



## 概要

このデザイン エLEMENTは、入力 (I)、出力 (O)、アクティブ Low 出力イネーブル (T) を持つ単一のトライステート出力バッファです。このELEMENTでは、LVTTTL 規格が使用され、DRIVE 制約と SLOW または FAST 制約を使用して駆動電流とスルー レートを選択できます。デフォルトでは、DRIVE=12mA、スルー レートは SLOW に設定されています。

T が Low の場合、バッファに入力された値が対応する出力に送られます。T が High の場合は、出力がハイインピーダンス (オフまたは Z ステート) になります。OBUFT は、双方向 I/O を作成するなど、トライステート機能にシングルエンド出力を使用する必要がある場合に使用します。

## 論理表

入力		出力
T	I	O
1	X	Z
0	1	1
0	0	0

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファ出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力
T	入力	1	トライステート イネーブル入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	出力の駆動電流を指定します。デザインの駆動電流およびタイミング要件で許容される最も低い値を使用してください。
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。
SLEW	文字列	"SLOW"、"FAST"	"SLOW"	出力ドライバーのスルー レートを指定します。この属性の最適な設定方法は、データシートを参照してください。

## VHDL 記述 (インスタンス化)

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

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

-- OBUFT: Single-ended 3-state Output Buffer
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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

-- End of OBUFT_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// OBUFT: Single-ended 3-state Output Buffer
//      All devices
// Xilinx HDL Libraries Guide, version 14.5

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

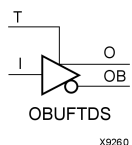
// End of OBUFT_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## OBUFTDS

**プリミティブ：3-State Output Buffer with Differential Signaling, Active-Low Output Enable**



### 概要

このデザイン エLEMENTは、低電圧差動信号をサポートする出力バッファです。OBUFTDS では、デザイン レベルのインターフェイス信号は、一方がマスターで、もう一方がスレーブとなる 2 つの異なるポート (O、OB) で表されます。マスターとスレーブは、MYNET\_P と MYNET\_N のように、同じ論理信号の反対の状態を示します。

### 論理表

入力		出力	
I	T	O	OB
X	1	Z	Z
0	0	0	1
1	0	1	0

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	Diff_p 出力 (最上位ポートに直接接続)
OB	出力	1	Diff_n 出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力
T	入力	1	トライステート イネーブル入力

### デザインの入力方法

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

### 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- OBUFTDS: Differential 3-state Output Buffer
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

OBUFTDS_inst : OBUFTDS
generic map (
  IOSTANDARD => "DEFAULT")
port map (
  O => O,      -- Diff_p output (connect directly to top-level port)
  OB => OB,    -- Diff_n output (connect directly to top-level port)
  I => I,      -- Buffer input
  T => T       -- 3-state enable input
);

-- End of OBUFTDS_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// OBUFTDS: Differential 3-state Output Buffer
//           Virtex-5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 14.5

OBUFTDS #(
  .IOSTANDARD("DEFAULT") // Specify the output I/O standard
) OBUFTDS_inst (
  .O(O),      // Diff_p output (connect directly to top-level port)
  .OB(OB),    // Diff_n output (connect directly to top-level port)
  .I(I),      // Buffer input
  .T(T)       // 3-state enable input
);

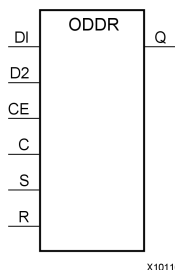
// End of OBUFTDS_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート: DC 特性およびスイッチ特性』\(DS202\)](#)

## 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 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。また、ODDR は SelectIO™ 機能と組み合わせて使用されます。

#### 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。クロック入力ピンです。
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	セット。アクティブ High の非同期セットピンです。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	2 進数	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.
--       Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

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.
//       Virtex-5/6
// Xilinx HDL Libraries Guide, version 14.5

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
);

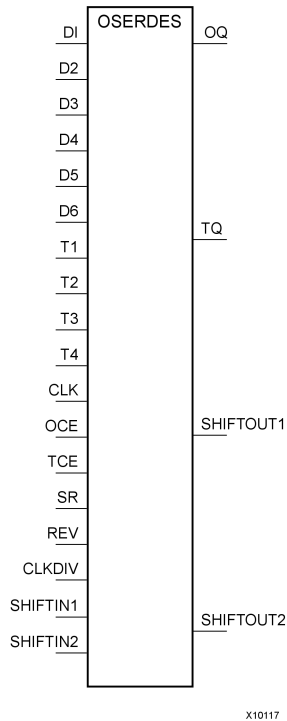
// End of ODDR_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## OSERDES

### プリミティブ：Dedicated IOB Output Serializer



### 概要

このプリミティブを使用すると、同期インターフェイスを簡単にインプリメントできます。FPGA のロジックリソースを節約でき、タイミングが複雑になるのを防ぎます。また、さまざまなアプリケーションに対応した複数のクロック入力があり、SelectIO™ 機能と共に使用できます。

### ポートの説明

ポート名	方向	幅	機能
OQ	出力	1	データパス出力。OSERDES モジュールのデータ出力です。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 パッドのデータ入力を接続します。また、OSERDES モジュール内のすべてのサブモジュールをバイパスするようにコンフィギュレーションすることも可能です。
SHIFTOUT1、SHIFTOUT2	出力	1 (それぞれ)	データ入力を拡張するためのキャリー出力です。マスターの SHIFTIN1、SHIFTIN2 に接続します。
TQ	出力	1	トライステートパス出力。OSERDES モジュールのトライステート出力です。このポートは、トライステート パラレル/シリアル コンバーターの出力と IOB パッドの制御入力を接続します。
CLK	入力	1	高速クロック入力。パラレル/シリアル コンバーターを駆動します。CLK ポートは、次のいずれかのクロックリソースで駆動します。 <ul style="list-style-type: none"> <li>・ クロック領域内の 10 個のグローバル クロック ライン</li> <li>・ 4 個のリージョナル クロック ライン</li> <li>・ 4 個のクロック兼用 I/O (隣接したクロック領域内)</li> </ul>

ポート名	方向	幅	機能
			・ FPGA (バイパスを介す)
CLKDIV	入力	1	高速分周クロック入力。パラレル/シリアル コンバーターを駆動するのに使用するクロック入力です。CLK ポートに接続されたクロックよりも低周波数に分周したクロックを入力する必要があります。CLKDIV のソースには、次のクロックリソースのいずれかを使用できます。  ・ クロック領域内の 10 個のグローバル クロック ライン  ・ 4 個のリージョナル クロック ライン
D1 ~ D6	入力	1	パラレル データ入力。OSERDES モジュールにパラレル データが入力されるポートです。このポートは FPGA に接続され、2 ~ 6 ビットにコンフィギュレーションできます。データ幅拡張モードでは、10 ビットまで拡張できます。
OCE	入力	1	パラレル/シリアル コンバーター (データ) クロック イネーブル。High の場合、データ パラレル/シリアル コンバーターの出力がイネーブルになります。
SR	入力	1	セット/リセット入力。ストレージ エLEMENT のステートを SRVAL 属性で設定したステートにします。SRVAL = 1 の場合は 1、SRVAL = 0 の場合は 0 になります。リセットがセットよりも優先されます。
SHIFTIN1、SHIFTIN2	入力	1 (それぞれ)	データ入力を拡張するためのキャリー入力です。スレーブの SHIFTOUT1、SHIFTOUT2 に接続します。
T1 ~ T4	入力	1 (それぞれ)	パラレル トライステート入力。OSERDES モジュールにパラレル トライステート信号が入力されるポートです。このポートは FPGA に接続され、1 ~ 4 ビットにコンフィギュレーションできます。この機能は、データ幅拡張モードではサポートされません。
TCE	入力	1	パラレル/シリアル コンバーター (トライステート) クロック イネーブル。High の場合、トライステート信号パラレル/シリアル コンバーターの出力がイネーブルになります。

## 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

OSERDES モジュールのデータ パラレル/シリアル コンバーターは、2 ~ 6 ビットのパラレル データを取り込み、シリアル データに変換します。2 つの OSERDES モジュールをカスケード接続すると、データ入力幅を 7、8、または 10 ビットに拡張できます。データ幅を拡張するには、1 つの OSERDES をマスター モード、もう 1 つの OSERDES をスレーブ モードに設定し、スレーブの SHIFTOUT ポートをマスターの SHIFTIN ポートに接続します。スレーブでは、入力として D3 ~ D6 ポートのみを使用します。パラレル/シリアル コンバーターは、SDR または DDR モードの両方で使用できます。

D1 ポートのデータ入力が、最初の出力ビットになります。このモジュールは、CLK および CLKDIV クロックで制御されます。次の表に、SDR および DDR の異なるモードにおける CLK と CLKDIV の関係を示します。

SDR のデータ幅	DDR のデータ幅	CLK	CLKDIV
2	4	2X	X
3	6	3X	X
4	8	4X	X
5	10	5X	X
6	–	6X	X
7	–	7X	X
8	–	8X	X

このブロックの出力は、FPGA の IOB パッドのデータ入力に接続されます。この IOB パッドには、SelectIO を使用して信号規格を設定できます。

#### パラレル/シリアル コンバーター (トリステスト)

OSERDES モジュールのトリステスト パラレル/シリアル コンバーターは、4 ビットのパラレル トリステスト信号を取り込み、シリアル トリステスト信号に変換します。データ パラレル/シリアル コンバーターとは異なり、トリステスト パラレル/シリアル コンバーターは信号幅を 5 ビット以上には拡張できません。このモジュールは、主に CLK および CLKDIV クロックで制御されます。このモードを使用するには、DATA\_RATE\_TQ および TRISTATE\_WIDTH 属性を設定する必要があります。場合によっては、DATA\_RATE\_OQ および DATA\_WIDTH を設定することも必要です。次の表に、使用する機能と属性の値を示します。

機能	DATA_RATE_TQ	TRISTATE_WIDTH
4 ビット DDR*	DDR	4
1 ビット SDR	SDR	1
バッファ	BUF	1

このブロックの出力は、FPGA の IOB パッドのトリステスト入力に接続されます。この IOB パッドには、SelectIO を使用して信号規格を設定できます。

#### データ幅の拡張

このエレメントでは、7 ビット幅以上のパラレル データを送信できます。ただし、トリステスト出力の信号幅は拡張できません。7 ビット幅以上のデータを送信するには、エレメントを 2 つインスタンス化する必要があります。この 2 つは、隣接したマスター/スレーブ ペアである必要があります。OSERDES\_MODE 属性を "MASTER" または "SLAVE" に設定し、OSERDES のペアを区別する必要があります。また、マスターの SHIFTIN ポートをスレーブの SHIFTOUT ポートに接続します。SDR および DDR モードでは、データ幅 7、8、10 がサポートされています。次の表に、SDR および DDR モードで使用可能なデータ幅を示します。

モード	幅
SDR	2、3、4、5、6、7、8
DDR	4、6、8、10

## 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	"SDR"、"DDR"	"DDR"	データを CLK の各エッジで変化させるか、各立ち上がりエッジで変化させるかを指定します。
DATA_RATE_TQ	文字列	"BUF"、"SDR"、 "DDR"	"DDR"	トライステート CLK の各エッジで変化させるか、各立ち上がりエッジで変化させるか、バッファのコンフィギュレーションで変化させるかを指定します。
DATA_WIDTH	整数	2、3、4、5、6、7、8、10	4	設定可能な値は、DATA_RATE_OQ = "DDR" の場合は 4、6、8、または 10、DATA_RATE_OQ = "SDR" の場合は 2、3、4、5、6、7、または 8 です。
INIT_OQ	2 進数	0、1	0	OQ 出力の初期値を指定します。
INIT_TQ	2 進数	0、1	0	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かスレーブかを指定します。
SRVAL_OQ	2 進数	0、1	0	リセット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	0、1	0	リセットをアサートした場合の TQ 出力の値を指定します。
TRISTATE_WIDTH	整数	1、2、4	4	設定可能な値は、DATA_RATE_TQ = "DDR" の場合は 2 または 4、DATA_RATE_TQ = "SDR" または "BUF" の場合は 1 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;

-- OSERDES: Output SERDES
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

OSERDES_inst : OSERDES
generic map (
  DATA_RATE_OQ => "DDR", -- Specify data rate to "DDR" or "SDR"
  DATA_RATE_TQ => "DDR", -- Specify data rate to "DDR", "SDR", or "BUF"
  DATA_WIDTH => 4, -- Specify data width - For DDR: 4,6,8, or 10
  -- For SDR or BUF: 2,3,4,5,6,7, or 8
  INIT_OQ => '0', -- INIT for Q1 register - '1' or '0'
  INIT_TQ => '0', -- INIT for Q2 register - '1' or '0'
  SERDES_MODE => "MASTER", --Set SERDES mode to "MASTER" or "SLAVE"
  SRVAL_OQ => '0', -- Define Q1 output value upon SR assertion - '1' or '0'
  SRVAL_TQ => '0', -- Define Q1 output value upon SR assertion - '1' or '0'
  TRISTATE_WIDTH => 4) -- Specify parallel to serial converter width
  -- When DATA_RATE_TQ = DDR: 2 or 4
  -- When DATA_RATE_TQ = SDR or BUF: 1 "
port map (
  OQ => OQ, -- 1-bit output
  SHIFTOUT1 => SHIFTOUT1, -- 1-bit data expansion output
  SHIFTOUT2 => SHIFTOUT2, -- 1-bit data expansion output
  TQ => TQ, -- 1-bit 3-state control output
  CLK => CLK, -- 1-bit clock input

```

```

CLKDIV => CLKDIV, -- 1-bit divided clock input
D1 => D1, -- 1-bit parallel data input
D2 => D2, -- 1-bit parallel data input
D3 => D3, -- 1-bit parallel data input
D4 => D4, -- 1-bit parallel data input
D5 => D5, -- 1-bit parallel data input
D6 => D6, -- 1-bit parallel data input
OCE => OCE, -- 1-bit clcok enable input
REV => '0', -- Must be tied to logic zero
SHIFTIN1 => SHIFTIN1, -- 1-bit data expansion input
SHIFTIN2 => SHIFTIN2, -- 1-bit data expansion input
SR => SR, -- 1-bit set/reset input
T1 => T1, -- 1-bit parallel 3-state input
T2 => T2, -- 1-bit parallel 3-state input
T3 => T3, -- 1-bit parallel 3-state input
T4 => T4, -- 1-bit parallel 3-state input
TCE => TCE -- 1-bit 3-state signal clock enable input
);

-- End of OSERDES_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// OSERDES: Source Synchronous Output Serializer
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

OSERDES #(
    .DATA_RATE_OQ("DDR"), // Specify data rate to "DDR" or "SDR"
    .DATA_RATE_TQ("DDR"), // Specify data rate to "DDR", "SDR", or "BUF"
    .DATA_WIDTH(4), // Specify data width - for DDR: 4,6,8, or 10
                      //      for SDR or BUF: 2,3,4,5,6,7, or 8
    .INIT_OQ(1'b0), // INIT for OQ register - 1'b1 or 1'b0
    .INIT_TQ(1'b0), // INIT for OQ register - 1'b1 or 1'b0
    .SERDES_MODE("MASTER"), // Set SERDES mode to "MASTER" or "SLAVE"
    .SRVAL_OQ(1'b0), // Define OQ output value upon SR assertion - 1'b1 or 1'b0
    .SRVAL_TQ(1'b0), // Define TQ output value upon SR assertion - 1'b1 or 1'b0
    .TRISTATE_WIDTH(4) // Specify parallel to serial converter width
                      //      When DATA_RATE_TQ = DDR: 2 or 4
                      //      When DATA_RATE_TQ = SDR or BUF: 1
) OSERDES_inst (
    .OQ(OQ), // 1-bit data path output
    .SHIFTOUT1(SHIFTOUT1), // 1-bit data expansion output
    .SHIFTOUT2(SHIFTOUT2), // 1-bit data expansion output
    .TQ(TQ), // 1-bit 3-state control output
    .CLK(CLK), // 1-bit clock input
    .CLKDIV(CLKDIV), // 1-bit divided clock input
    .D1(D1), // 1-bit parallel data input
    .D2(D2), // 1-bit parallel data input
    .D3(D3), // 1-bit parallel data input
    .D4(D4), // 1-bit parallel data input
    .D5(D5), // 1-bit parallel data input
    .D6(D6), // 1-bit parallel data input
    .OCE(OCE), // 1-bit clock enable input
    .REV(1'b0), // Must be tied to logic zero
    .SHIFTIN1(SHIFTIN1), // 1-bit data expansion input
    .SHIFTIN2(SHIFTIN2), // 1-bit data expansion input
    .SR(SR), // 1-bit set/reset input
    .T1(T1), // 1-bit parallel 3-state input
    .T2(T2), // 1-bit parallel 3-state input
    .T3(T3), // 1-bit parallel 3-state input
    .T4(T4), // 1-bit parallel 3-state input
    .TCE(TCE) // 1-bit 3-state signal clock enable input
);

// End of OSERDES_inst instantiation

```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## PCIE\_EP

プリミティブ：PCI Express

### 概要

このデザイン エLEMENTは、Virtex®-5 デバイスに組み込まれた Integrated Endpoint ブロックです。これにより、PCI™ および PCI-X™ パラレル バス規格の次世代版である PCI EXPRESS® (PCIe®) の機能を使用できます。高パフォーマンスでパケット ベースのポイント ツー ポイント シリアル インターフェイスで、PCI および PCI-X コンフィギュレーション、デバイスドライバおよびアプリケーション ソフトウェアとも下位互換性があります。

### デザインの入力方法

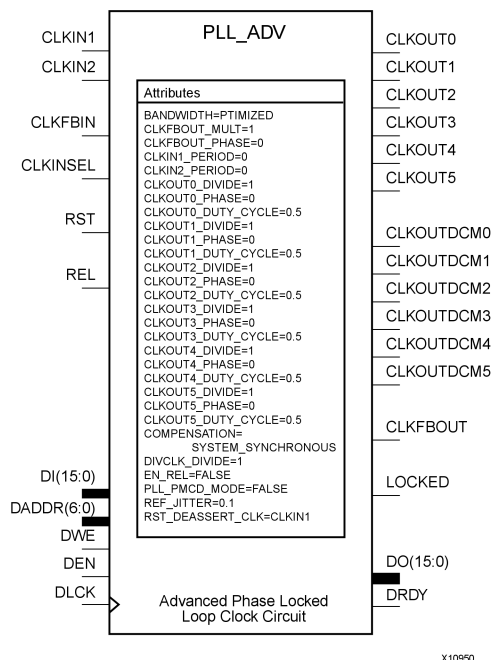
インスタンス化	不可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

### 詳細情報

- ・ [『PCI Express 用 Virtex-5 統合エンドポイントブロック デザイン ユーザー ガイド』\(UG197\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)
- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)

## PLL\_ADV

### プリミティブ：Advanced Phase Locked Loop Clock Circuit



## 概要

PLL\_ADV プリミティブを使用すると、PLL\_BASE のすべての機能にアクセスできます。このエレメントは参照用によりのみ掲載されており、ターゲット デバイスを変更する場合に使用できます。ほとんどのデザインでは、PLL\_BASE プリミティブまたは Clocking Wizard を使用してください。

## ポートの説明

ポート名	方向	幅	機能
CLKFBDCM	出力	1	再ターゲットに使用する PLL_ADV ピン。DCM を駆動する場合に調整に使用する PLL フィードバックです。CLKFBOUT ピンをこの目的で使用する場合、ツールにより自動的に正しいポートにマップされます。
CLKFBIN	入力	1	クロック フィードバック入力
CLKFBOUT	出力	1	専用 PLL フィードバック出力
CLKINSEL	入力	1	再ターゲットに使用する PLL_ADV ピン。スタティック High またはスタティック Low に接続して PLL_ADV のクロック入力を選択します。 <ul style="list-style-type: none"> <li>High = CLKIN1</li> <li>Low = CLKIN2</li> </ul>
CLKIN1	入力	1	再ターゲットに使用する PLL_ADV ピン。汎用クロック入力
CLKIN2	入力	1	再ターゲットに使用する PLL_ADV ピン。セカンダリクロック入力です。

ポート名	方向	幅	機能
CLKOUTDCM0 ~ CLKOUTDCM5	出力	1	再ターゲットに使用する PLL_ADV ピン。コンフィギュレーション可能なクロック (0 ~ 5) で、PLL と同じ CMT 内の DCM にのみ接続できます。
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 (0 ~ 5) 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。入力クロックと出力クロックは、位相が揃っています。
DADDR[4:0]	入力	5	ダイナミックリコンフィギュレーションのアドレスを指定します。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が使用されていない場合は、DEN が Low に接続されます。DEN が Low の場合 DO 出力がステータス信号を反映します。
DI[15:0]	入力	16	リコンフィギュレーション データを提供。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 ダイナミックリコンフィギュレーション用に DEN 信号への応答を提供します。
DWE	入力	1	DI データの DADDR アドレスへの書き込みを制御するライトイネーブル信号です。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	PLL からの非同期出力で、PLL で、位相アライメントが指定範囲内で達成され、周波数が指定 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起きた場合 (入力クロックの位相シフトなど)、LOCKED はディアサートされます。LOCKED がディアサートされた場合は、PLL をリセットする必要があります。
REL	入力	1	Virtex®-4 PMCD コンポーネントを移行する場合に使用します。その他の場合には使用しないことをお勧めします。
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 PLL 特性に影響する PLL プログラム アルゴリズムを指定します。
CLKFBOUT_ DESKEW_ ADJUST	文字列	"NONE"、 "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"	"NONE"	一部の IP コアで、PPC440 などのブロックのクロック挿入遅延を調整するために使用されます。適切な位相アライメントが実行されるようにするため、特に指示がない限り、"NONE" のままにしてください。
CLKFBOUT_MULT	整数	1 ~ 64	1	すべての CLKOUT クロック出力を逡倍する値を指定します。この値と、CLKOUT#_DIVIDE 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKFBOUT_PHASE	1 上位ビット浮動 小数点	0.0 ~ 360.0	0.0	クロック フィードバック出力の位相オフセットを度数で指定します。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	3 上位ビット浮動 小数点	ns の単位で指定された実数 (精度 (ps) は小数 3 桁まで)	0.0	CLKIN1 の入力周期を ns で指定します。精度は ps です。CLKIN1 クロック入力を使用する場合は、この値を必ず設定する必要があります。
CLKIN2_PERIOD	3 上位ビット浮動 小数点	ns の単位で指定された実数 (精度 (ps) は小数 3 桁まで)	0.0	CLKIN2 の入力周期を ns で指定します。精度は ps です。CLKIN2 クロック入力を使用する場合は、この値を必ず設定する必要があります。
CLKOUT0_DESKEW_ ADJUST ~ CLKOUT5_DESKEW_ ADJUST	文字列	"NONE"、 "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"	"NONE"	PPC440 デザインでのみ使用されます。詳細は、エンベデッド プロセッサ ブロック ユーザー ガイドのクロック挿入遅延および PLL の使用に関するセクションを参照してください。
CLKOUT0_DIVIDE ~ CLKOUT5_DIVIDE	整数	1 ~ 128	1	CLKOUT クロック出力を分周する値を指定します。この値と、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 ~ CLKOUT5_DUTY_ CYCLE	2 上位ビット浮動 小数点	0.01 ~ 0.99	0.50	CLKOUT クロック出力のデューティサイクルをパーセントで指定します。0.50 に設定すると、デューティサイクルは 50% になります。

属性	データ型	値	デフォルト	説明
CLKOUT0_PHASE ~ CLKOUT5_PHASE	1 上位ビット浮動 小数点	0.0 ~ 360.0	0.0	CLKOUT クロック出力との位相オフセットを度数で指定します。90 は 90 度 (4 分の 1 サイクル) の位相オフセット、180 は 180 度 (2 分の 1 サイクル) の位相オフセットを示します。
COMPENSATION	文字列	"SYSTEM_SYNCHRONOUS"、 "SOURCE_SYNCHRONOUS"、 "INTERNAL"、 "EXTERNAL"、 "DCM2PLL"、 "PLL2DCM"	"SYSTEM_SYNCHRONOUS"	入力クロックの PLL 位相調整を指定します。"SYSTEM_SYNCHRONOUS" に設定すると、ホールドタイム 0 ですべてのクロック遅延の調整が試みられます。"SOURCE_SYNCHRONOUS" は、クロックがデータと共に供給されており、クロックと位相が揃っている場合に使用します。その他の設定 ("INTERNAL"、"EXTERNAL"、"DCM2PLL"、"PLL2DCM") は、ISE ツールで自動的に選択されます。
DIVCLK_DIVIDE	整数	1 ~ 52	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、
EN_REL	ブール代数	FALSE、TRUE	FALSE	PMCD モードの場合 (PLL_PMCD_MODE = TRUE)、REL 入力ピンがアサートされたときに分周クロック出力が開始するように設定します。
PLL_PMCD_MODE	ブール代数	FALSE、TRUE	FALSE	PLL が PMCD として動作するよう指定します。
REF_JITTER	3 上位ビット浮動 小数点	0.000 ~ 1.000	0.100	PLL パフォーマンスを最適化するため、基準クロック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、値が既知でないときに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。
RESET_ON_LOSS_OF_LOCK	ブール代数	FALSE、TRUE	FALSE	FALSE に設定する必要があります。シリコンではサポートされていません。
RST_DEASSERT_CLK	文字列	"CLKIN1"、 "CLKFBIN"	"CLKIN1"	選択した PMCD 入力クロックに同期して、RST 信号のディASSERTを指定
SIM_DEVICE	文字列	"VIRTEX5"、 "SPARTAN6"	"VIRTEX5"	コンポーネントを正しくシミュレーションするため、ターゲット デバイスを指定します。Virtex®-5 をターゲットとする場合は、VIRTEX5 に設定する必要があります。

## 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;
```

```

-- PLL_ADV: Phase-Lock Loop Clock Circuit
-- Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

PLL_ADV_inst : PLL_ADV
generic map (
    BANDWIDTH => "OPTIMIZED", -- "HIGH", "LOW" or "OPTIMIZED"
    CLKFBOUT_MULT => 1, -- Multiplication factor for all output clocks
    CLKFBOUT_PHASE => 0.0, -- Phase shift (degrees) of all output clocks
    CLKIN1_PERIOD => 0.000, -- Clock period (ns) of input clock on CLKIN1
    CLKIN2_PERIOD => 0.000, -- Clock period (ns) of input clock on CLKIN2
    CLKOUT0_DIVIDE => 1, -- Division factor for CLKOUT0 (1 to 128)
    CLKOUT0_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT0 (0.01 to 0.99)
    CLKOUT0_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT0 (0.0 to 360.0)
    CLKOUT1_DIVIDE => 1, -- Division factor for CLKOUT1 (1 to 128)
    CLKOUT1_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT1 (0.01 to 0.99)
    CLKOUT1_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT1 (0.0 to 360.0)
    CLKOUT2_DIVIDE => 1, -- Division factor for CLKOUT2 (1 to 128)
    CLKOUT2_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT2 (0.01 to 0.99)
    CLKOUT2_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT2 (0.0 to 360.0)
    CLKOUT3_DIVIDE => 1, -- Division factor for CLKOUT3 (1 to 128)
    CLKOUT3_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT3 (0.01 to 0.99)
    CLKOUT3_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT3 (0.0 to 360.0)
    CLKOUT4_DIVIDE => 1, -- Division factor for CLKOUT4 (1 to 128)
    CLKOUT4_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT4 (0.01 to 0.99)
    CLKOUT4_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT4 (0.0 to 360.0)
    CLKOUT5_DIVIDE => 1, -- Division factor for CLKOUT5 (1 to 128)
    CLKOUT5_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT5 (0.01 to 0.99)
    CLKOUT5_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT5 (0.0 to 360.0)
    COMPENSATION => "SYSTEM_SYNCHRONOUS", -- "SYSTEM_SYNCHRONOUS",
                                           -- "SOURCE_SYNCHRONOUS", "INTERNAL",
                                           -- "EXTERNAL", "DCM2PLL", "PLL2DCM"
    DIVCLK_DIVIDE => 1, -- Division factor for all clocks (1 to 52)
    EN_REL => FALSE, -- Enable release (PMCD mode only)
    PLL_PMCD_MODE => FALSE, -- PMCD Mode, TRUE/FALSE
    REF_JITTER => 0.100, -- Input reference jitter (0.000 to 0.999 UI%)
    RST_DEASSERT_CLK => "CLKIN1") -- In PMCD mode, clock to synchronize RST release
port map (
    CLKFBDCM => CLKFBDCM, -- Output feedback signal used when PLL feeds a DCM
    CLKFBOUT => CLKFBOUT, -- General output feedback signal
    CLKOUT0 => CLKOUT0, -- One of six general clock output signals
    CLKOUT1 => CLKOUT1, -- One of six general clock output signals
    CLKOUT2 => CLKOUT2, -- One of six general clock output signals
    CLKOUT3 => CLKOUT3, -- One of six general clock output signals
    CLKOUT4 => CLKOUT4, -- One of six general clock output signals
    CLKOUT5 => CLKOUT5, -- One of six general clock output signals
    CLKOUTDCM0 => CLKOUTDCM0, -- One of six clock outputs to connect to the DCM
    CLKOUTDCM1 => CLKOUTDCM1, -- One of six clock outputs to connect to the DCM
    CLKOUTDCM2 => CLKOUTDCM2, -- One of six clock outputs to connect to the DCM
    CLKOUTDCM3 => CLKOUTDCM3, -- One of six clock outputs to connect to the DCM
    CLKOUTDCM4 => CLKOUTDCM4, -- One of six clock outputs to connect to the DCM
    CLKOUTDCM5 => CLKOUTDCM5, -- One of six clock outputs to connect to the DCM
    DO => DO, -- Dynamic reconfig data output (16-bits)
    DRDY => DRDY, -- Dynamic reconfig ready output
    LOCKED => LOCKED, -- Active high PLL lock signal
    CLKFBIN => CLKFBIN, -- Clock feedback input
    CLKIN1 => CLKIN1, -- Primary clock input
    CLKIN2 => CLKIN2, -- Secondary clock input
    CLKINSEL => CLKINSEL, -- Selects CLKIN1 or CLKIN2
    DADDR => DADDR, -- Dynamic reconfig address input (5-bits)
    DCLK => DCLK, -- Dynamic reconfig clock input
    DEN => DEN, -- Dynamic reconfig enable input
    DI => DI, -- Dynamic reconfig data input (16-bits)
    DWE => DWE, -- Dynamic reconfig write enable input
    REL => REL, -- Clock release input (PMCD mode only)
    RST => RST -- Asynchronous PLL reset
);

-- End of PLL_ADV_inst instantiation

```

## Verilog 記述 (インスタンス化)

```
// PLL_ADV: Phase-Lock Loop Clock Circuit
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

PLL_ADV #(
    .BANDWIDTH("OPTIMIZED"), // "HIGH", "LOW" or "OPTIMIZED"
    .CLKFBOUT_MULT(1),        // Multiplication factor for all output clocks
    .CLKFBOUT_PHASE(0.0),     // Phase shift (degrees) of all output clocks
    .CLKIN1_PERIOD(0.000),    // Clock period (ns) of input clock on CLKIN1
    .CLKIN2_PERIOD(0.000),    // Clock period (ns) of input clock on CLKIN2
    .CLKOUT0_DIVIDE(1),       // Division factor for CLKOUT0 (1 to 128)
    .CLKOUT0_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT0 (0.01 to 0.99)
    .CLKOUT0_PHASE(0.0),      // Phase shift (degrees) for CLKOUT0 (0.0 to 360.0)
    .CLKOUT1_DIVIDE(1),       // Division factor for CLKOUT1 (1 to 128)
    .CLKOUT1_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT1 (0.01 to 0.99)
    .CLKOUT1_PHASE(0.0),      // Phase shift (degrees) for CLKOUT1 (0.0 to 360.0)
    .CLKOUT2_DIVIDE(1),       // Division factor for CLKOUT2 (1 to 128)
    .CLKOUT2_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT2 (0.01 to 0.99)
    .CLKOUT2_PHASE(0.0),      // Phase shift (degrees) for CLKOUT2 (0.0 to 360.0)
    .CLKOUT3_DIVIDE(1),       // Division factor for CLKOUT3 (1 to 128)
    .CLKOUT3_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT3 (0.01 to 0.99)
    .CLKOUT3_PHASE(0.0),      // Phase shift (degrees) for CLKOUT3 (0.0 to 360.0)
    .CLKOUT4_DIVIDE(1),       // Division factor for CLKOUT4 (1 to 128)
    .CLKOUT4_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT4 (0.01 to 0.99)
    .CLKOUT4_PHASE(0.0),      // Phase shift (degrees) for CLKOUT4 (0.0 to 360.0)
    .CLKOUT5_DIVIDE(1),       // Division factor for CLKOUT5 (1 to 128)
    .CLKOUT5_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT5 (0.01 to 0.99)
    .CLKOUT5_PHASE(0.0),      // Phase shift (degrees) for CLKOUT5 (0.0 to 360.0)
    .COMPENSATION("SYSTEM_SYNCHRONOUS"), // "SYSTEM_SYNCHRONOUS",
                                         // "SOURCE_SYNCHRONOUS", "INTERNAL", "EXTERNAL",
                                         // "DCM2PLL", "PLL2DCM"
    .DIVCLK_DIVIDE(1),        // Division factor for all clocks (1 to 52)
    .EN_REL("FALSE"),         // Enable release (PMCD mode only)
    .PLL_PMCD_MODE("FALSE"),   // PMCD Mode, TRUE/FALSE
    .REF_JITTER(0.100),       // Input reference jitter (0.000 to 0.999 UI%)
    .RST_DEASSERT_CLK("CLKIN1") // In PMCD mode, clock to synchronize RST release
) PLL_ADV_inst (
    .CLKFBDCM(CLKFBDCM),      // Output feedback signal used when PLL feeds a DCM
    .CLKFBOUT(CLKFBOUT),      // General output feedback signal
    .CLKOUT0(CLKOUT0),         // One of six general clock output signals
    .CLKOUT1(CLKOUT1),         // One of six general clock output signals
    .CLKOUT2(CLKOUT2),         // One of six general clock output signals
    .CLKOUT3(CLKOUT3),         // One of six general clock output signals
    .CLKOUT4(CLKOUT4),         // One of six general clock output signals
    .CLKOUT5(CLKOUT5),         // One of six general clock output signals
    .CLKOUTDCM0(CLKOUTDCM0),   // One of six clock outputs to connect to the DCM
    .CLKOUTDCM1(CLKOUTDCM1),   // One of six clock outputs to connect to the DCM
    .CLKOUTDCM2(CLKOUTDCM2),   // One of six clock outputs to connect to the DCM
    .CLKOUTDCM3(CLKOUTDCM3),   // One of six clock outputs to connect to the DCM
    .CLKOUTDCM4(CLKOUTDCM4),   // One of six clock outputs to connect to the DCM
    .CLKOUTDCM5(CLKOUTDCM5),   // One of six clock outputs to connect to the DCM
    .DO(DO),                   // Dynamic reconfig data output (16-bits)
    .DRDY(DRDY),               // Dynamic reconfig ready output
    .LOCKED(LOCKED),           // Active high PLL lock signal
    .CLKFBIN(CLKFBIN),         // Clock feedback input
    .CLKIN1(CLKIN1),           // Primary clock input
    .CLKIN2(CLKIN2),           // Secondary clock input
    .CLKINSEL(CLKINSEL),       // Selects '1' = CLKIN1, '0' = CLKIN2
    .DADDR(DADDR),             // Dynamic reconfig address input (5-bits)
    .DCLK(DCLK),               // Dynamic reconfig clock input
    .DEN(DEN),                 // Dynamic reconfig enable input
    .DI(DI),                   // Dynamic reconfig data input (16-bits)
    .DWE(DWE),                 // Dynamic reconfig write enable input
    .REL(REL),                 // Clock release input (PMCD mode only)
    .RST(RST)                  // Asynchronous PLL reset
);

// End of PLL_ADV_inst instantiation
```

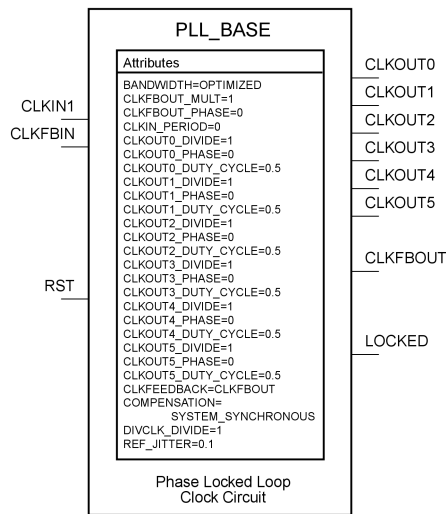
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## PLL\_BASE

### プリミティブ：Basic Phase Locked Loop Clock Circuit



X10951

## 概要

このデザイン エLEMENTは、FPGA 内部および外部回路の両方でクロック合成およびクロック管理機能を提供するためのエンベデッド位相ロック ループ クロック回路で、PLL\_ADV デザイン エLEMENTのサブセットです。PLL\_BASE を使用すると、ほとんどの PLL クロック回路を簡単に統合できます。PLL のすべての機能は備わっていませんが、入力クロックを位相シフト、通倍、分周でき、またデューティ サイクルを変更したり、ジッターをフィルター処理する機能などがあります。

## ポートの説明

ポート名	方向	幅	機能
クロック出力/入力			
CLKOUT0 ~ 5	出力	1	位相シフトされる 6 個の出力クロック
CLKFBOUT	出力	1	クロック ネットワークの遅延調整方法を指定するために使用する専用 PLL フィードバック出力。調整方法によって、この出力を接続する場合と接続する必要のない場合があります。
CLKIN	入力	1	PLL のクロック ソース入力。FPGA の専用クロックピン、DCM 出力クロックピン、または BUFG 出力ピンで駆動できます。
CLKFBIN	入力	1	クロック フィードバック入力。CLKFBOUT ポートでのみ駆動できます。
ステータス出力/制御入力			
LOCKED	出力	1	位相アライメントが完了し、操作が開始可能であることを示す非同期出力
RST	入力	1	非同期リセット

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
COMPENSATION	文字列	"SYSTEM_SYNCHRONOUS"、 "SOURCE_SYNCHRONOUS"	"SYSTEM_SYNCHRONOUS"	入力クロックの PLL 位相調整を指定します。すべてのクロック遅延を調整する場合は "SYSTEM_SYNCHRONOUS" を、クロックがデータと共に供給されて位相が揃っているときには "SOURCE_SYNCHRONOUS" を使用します。
BANDWIDTH	文字列	"HIGH"、"LOW"、 "OPTIMIZED"	"OPTIMIZED"	ジッター、位相マージンなどの PLL 特性に影響する PLL プログラム アルゴリズムを指定します。
CLKOUT0_DIVIDE、 CLKOUT1_DIVIDE、 CLKOUT2_DIVIDE、 CLKOUT3_DIVIDE、 CLKOUT4_DIVIDE、 CLKOUT5_DIVIDE	整数	1 ~ 128	1	CLKOUT クロック出力を分周する値を指定します。この値と FBCLKOUT_MULT 値により出力周波数が決まります。
CLKOUT0_PHASE、 CLKOUT1_PHASE、 CLKOUT2_PHASE、 CLKOUT3_PHASE、 CLKOUT4_PHASE、 CLKOUT5_PHASE	実数	0.0 ~ 360.0	0.0	CLKOUT クロック出力との位相オフセットを度数で指定します。90 は 90 度 (4 分の 1 サイクル) の位相オフセット、180 は 180 度 (2 分の 1 サイクル) の位相オフセットを示します。
CLKOUT0_DUTY_CYCLE、 CLKOUT1_DUTY_CYCLE、 CLKOUT2_DUTY_CYCLE、 CLKOUT3_DUTY_CYCLE、 CLKOUT4_DUTY_CYCLE、 CLKOUT5_DUTY_CYCLE	実数	0.01 ~ 0.99	0.50	CLKOUT クロック出力のデューティサイクルをパーセントで指定します。0.50 に設定すると、デューティサイクルは 50% になります。
CLKFBOUT_MULT	整数	1 ~ 64	1	すべての CLKOUT クロック出力を逡倍する値を指定します。この値と CLKOUT#_DIVIDE 値により出力周波数が決まります。
DIVCLK_DIVIDE	整数	1 ~ 52	1	すべての出力クロックの分周比を指定します。
CLKFBOUT_PHASE	実数	0.0 ~ 360	0.0	クロック フィードバック出力の位相オフセットを度数で指定します。

属性	データ型	値	デフォルト	説明
REF_JITTER	実数	0 ～ 0.999	0.100	基準クロック ジッターを基準クロックの割合を示す UI (ユニット インターバル) で指定します。この値は、入力クロックの最大ピークトゥピーク値にします。
CLKIN_PERIOD	実数	1.000 ～ 52.630	なし	CLKIN の入力周期を ns で指定します。

## VHDL 記述 (インスタンス化)

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;

-- PLL_BASE: Phase-Lock Loop Clock Circuit
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

PLL_BASE_inst : PLL_BASE
generic map (
    BANDWIDTH => "OPTIMIZED", -- "HIGH", "LOW" or "OPTIMIZED"
    CLKFBOUT_MULT => 1, -- Multiplication factor for all output clocks
    CLKFBOUT_PHASE => 0.0, -- Phase shift (degrees) of all output clocks
    CLKIN_PERIOD => 0.000, -- Clock period (ns) of input clock on CLKIN
    CLKOUT0_DIVIDE => 1, -- Division factor for CLKOUT0 (1 to 128)
    CLKOUT0_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT0 (0.01 to 0.99)
    CLKOUT0_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT0 (0.0 to 360.0)
    CLKOUT1_DIVIDE => 1, -- Division factor for CLKOUT1 (1 to 128)
    CLKOUT1_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT1 (0.01 to 0.99)
    CLKOUT1_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT1 (0.0 to 360.0)
    CLKOUT2_DIVIDE => 1, -- Division factor for CLKOUT2 (1 to 128)
    CLKOUT2_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT2 (0.01 to 0.99)
    CLKOUT2_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT2 (0.0 to 360.0)
    CLKOUT3_DIVIDE => 1, -- Division factor for CLKOUT3 (1 to 128)
    CLKOUT3_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT3 (0.01 to 0.99)
    CLKOUT3_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT3 (0.0 to 360.0)
    CLKOUT4_DIVIDE => 1, -- Division factor for CLKOUT4 (1 to 128)
    CLKOUT4_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT4 (0.01 to 0.99)
    CLKOUT4_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT4 (0.0 to 360.0)
    CLKOUT5_DIVIDE => 1, -- Division factor for CLKOUT5 (1 to 128)
    CLKOUT5_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT5 (0.01 to 0.99)
    CLKOUT5_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT5 (0.0 to 360.0)
    COMPENSATION => "SYSTEM_SYNCHRONOUS", -- "SYSTEM_SYNCHRONOUS",
                                           -- "SOURCE_SYNCHRONOUS", "INTERNAL",
                                           -- "EXTERNAL", "DCM2PLL", "PLL2DCM"
    DIVCLK_DIVIDE => 1, -- Division factor for all clocks (1 to 52)
    REF_JITTER => 0.100) -- Input reference jitter (0.000 to 0.999 UI%)
port map (
    CLKFBOUT => CLKFBOUT, -- General output feedback signal
    CLKOUT0 => CLKOUT0, -- One of six general clock output signals
    CLKOUT1 => CLKOUT1, -- One of six general clock output signals
    CLKOUT2 => CLKOUT2, -- One of six general clock output signals
    CLKOUT3 => CLKOUT3, -- One of six general clock output signals
    CLKOUT4 => CLKOUT4, -- One of six general clock output signals
    CLKOUT5 => CLKOUT5, -- One of six general clock output signals
    LOCKED => LOCKED, -- Active high PLL lock signal
    CLKFBIN => CLKFBIN, -- Clock feedback input
    CLKIN => CLKIN, -- Clock input
    RST => RST -- Asynchronous PLL reset
);

-- End of PLL_BASE_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// PLL_BASE: Phase-Lock Loop Clock Circuit
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

PLL_BASE #(
    .BANDWIDTH("OPTIMIZED"), // "HIGH", "LOW" or "OPTIMIZED"
    .CLKFBOUT_MULT(1),       // Multiplication factor for all output clocks
    .CLKFBOUT_PHASE(0.0),    // Phase shift (degrees) of all output clocks
    .CLKIN_PERIOD(0.000),    // Clock period (ns) of input clock on CLKIN
    .CLKOUT0_DIVIDE(1),      // Division factor for CLKOUT0 (1 to 128)
    .CLKOUT0_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT0 (0.01 to 0.99)
    .CLKOUT0_PHASE(0.0),     // Phase shift (degrees) for CLKOUT0 (0.0 to 360.0)
    .CLKOUT1_DIVIDE(1),      // Division factor for CLKOUT1 (1 to 128)
    .CLKOUT1_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT1 (0.01 to 0.99)
    .CLKOUT1_PHASE(0.0),     // Phase shift (degrees) for CLKOUT1 (0.0 to 360.0)
    .CLKOUT2_DIVIDE(1),      // Division factor for CLKOUT2 (1 to 128)
    .CLKOUT2_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT2 (0.01 to 0.99)
    .CLKOUT2_PHASE(0.0),     // Phase shift (degrees) for CLKOUT2 (0.0 to 360.0)
    .CLKOUT3_DIVIDE(1),      // Division factor for CLKOUT3 (1 to 128)
    .CLKOUT3_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT3 (0.01 to 0.99)
    .CLKOUT3_PHASE(0.0),     // Phase shift (degrees) for CLKOUT3 (0.0 to 360.0)
    .CLKOUT4_DIVIDE(1),      // Division factor for CLKOUT4 (1 to 128)
    .CLKOUT4_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT4 (0.01 to 0.99)
    .CLKOUT4_PHASE(0.0),     // Phase shift (degrees) for CLKOUT4 (0.0 to 360.0)
    .CLKOUT5_DIVIDE(1),      // Division factor for CLKOUT5 (1 to 128)
    .CLKOUT5_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT5 (0.01 to 0.99)
    .CLKOUT5_PHASE(0.0),     // Phase shift (degrees) for CLKOUT5 (0.0 to 360.0)
    .COMPENSATION("SYSTEM_SYNCHRONOUS"), // "SYSTEM_SYNCHRONOUS",
                                         // "SOURCE_SYNCHRONOUS", "INTERNAL", "EXTERNAL",
                                         // "DCM2PLL", "PLL2DCM"
    .DIVCLK_DIVIDE(1),       // Division factor for all clocks (1 to 52)
    .REF_JITTER(0.100)      // Input reference jitter (0.000 to 0.999 UI%)
) PLL_BASE_inst (
    .CLKFBOUT(CLKFBOUT),    // General output feedback signal
    .CLKOUT0(CLKOUT0),      // One of six general clock output signals
    .CLKOUT1(CLKOUT1),      // One of six general clock output signals
    .CLKOUT2(CLKOUT2),      // One of six general clock output signals
    .CLKOUT3(CLKOUT3),      // One of six general clock output signals
    .CLKOUT4(CLKOUT4),      // One of six general clock output signals
    .CLKOUT5(CLKOUT5),      // One of six general clock output signals
    .LOCKED(LOCKED),        // Active high PLL lock signal
    .CLKFBIN(CLKFBIN),      // Clock feedback input
    .CLKIN(CLKIN),          // Clock input
    .RST(RST)               // Asynchronous PLL reset
);

// End of PLL_BASE_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート : DC 特性およびスイッチ特性』\(DS202\)](#)

## PPC440

プリミティブ : Power PC 440 CPU Core

### 概要

このELEMENTは 2 命令同時発行可能なスーパースカラー プロセッサで、以前の PowerPC® 405 と同じ命令セットアーキテクチャのインプリメントしますが、パフォーマンスが大幅に改善されています。

### デザインの入力方法

インスタンシエーション	不可
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

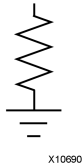
### 詳細情報

- ・ [『IBM PPC440x5 CPU Core User's Manual』](#)
- ・ [『Virtex-5 FPGA データシート : DC 特性およびスイッチ特性』\(DS202\)](#)
- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)

## 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
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

PULLDOWN_inst : PULLDOWN
port map (
  O => O      -- Pulldown output (connect directly to top-level port)
);

-- End of PULLDOWN_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// PULLDOWN: I/O Buffer Weak Pull-down
//           All FPGA
// Xilinx HDL Libraries Guide, version 14.5

PULLDOWN PULLDOWN_inst (
    .O(0)      // Pulldown output (connect directly to top-level port)
);

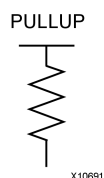
// End of PULLDOWN_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## PULLUP

**プリミティブ : Resistor to VCC for Input PADs, Open-Drain, and 3-State Outputs**



### 概要

このデザイン エLEMENTは、入力、トライステート出力、または双方向ポートが内部または外部ソースで駆動されないときに、弱い High で駆動します。すべてのドライバーがオフのときに、オープンドレイン エLEMENTおよびマクロのロジックレベルを High にします。

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	プルアップ出力 (最上位ポートに直接接続)

### デザインの入力方法

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

### VHDL 記述 (インスタンス化)

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

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

-- PULLUP: I/O Buffer Weak Pull-up
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

PULLUP_inst : PULLUP
port map (
  O => O      -- Pullup output (connect directly to top-level port)
);

-- End of PULLUP_inst instantiation
```



## Verilog 記述 (インスタンス化)

```
// PULLUP: I/O Buffer Weak Pull-up
//           All FPGA, CoolRunner-II
// Xilinx HDL Libraries Guide, version 14.5

PULLUP PULLUP_inst (
    .O(0)      // Pullup output (connect directly to top-level port)
);

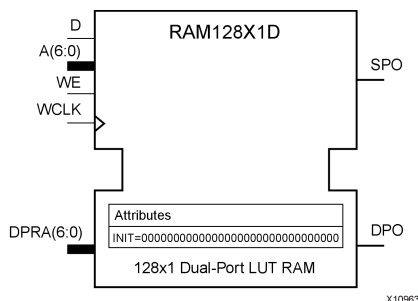
// End of PULLUP_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAM128X1D

### プリミティブ：128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)



### 概要

このデザイン エLEMENTは 128 ワード X 1 ビットの RAM で読み出し/書き込みポートがあり、ライト イネーブル (WE) が High のときにアドレス バス A で指定されたロケーションに D 入力データピンの値が書き込まれます。この書き込みは WCLK の立ち上がりエッジの直後に実行され、同じ値が SPO に出力されます。WE が Low のときは非同期読み出しが実行され、アドレス バス A で指定されたメモリ ロケーションの値が SPO に非同期で出力されます。アドレス バス DPRA の値を変更することにより、読み出しポートでは非同期読み出しを実行できます。DPO にその値が出力されます。

### ポートの説明

ポート名	方向	幅	機能
SPO	出力	1	アドレス バス A で指定される読み出し/書き込みポートのデータ出力
DPO	出力	1	アドレス バス DPRA で指定される読み出しポートのデータ出力
D	入力	1	アドレス バス A で指定される書き込みデータ入力
A	入力	7	読み出し/書き込みポートのアドレス バス
DPRA	入力	7	読み出しポートのアドレス バス
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

インスタンスエートする場合は、このコンポーネントを次のように接続します。

- WCLK 入力をクロックソースに、D 入力を格納するデータソースに、DPO 出力を FDCE の D 入力などの適切なデスティネーションに接続します。
- オプションで、SPO 出力を適切なデスティネーションに接続するか、または未接続にすることもできます。
- クロック イネーブル ピン (WE) は、適切なライト イネーブル ソースに接続します。
- 7 ビット バス A は読み出し/書き込みアドレスに、7 ビット バス DPRA は読み出しアドレスに接続する必要があります。
- 128 ビットの 16 進数で構成される INIT 属性で、RAM の初期値を指定できます。

指定しない場合は、初期値はすべて 0 になります。

## デザインの入力方法

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

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	128 ビット値	すべて 0	RAM の初期値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- RAM128X1D: 128-deep by 1-wide positive edge write, asynchronous read
--             dual-port distributed LUT RAM
--             Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM128X1D_inst : RAM128X1D
generic map (
  INIT => X"00000000000000000000000000000000"
)
port map (
  DPO => DPO,      -- Read/Write port 1-bit output
  SPO => SPO,      -- Read port 1-bit output
  A => A,          -- Read/Write port 7-bit address input
  D => D,          -- RAM data input
  DPRA => DPRA,    -- Read port 7-bit address input
  WCLK => WCLK,    -- Write clock input
  WE => WE         -- RAM data input
);

-- End of RAM128X1D_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAM128X1D: 128-deep by 1-wide positive edge write, asynchronous read
//             dual-port distributed LUT RAM
//             Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

RAM128X1D #(
  .INIT(128'h00000000000000000000000000000000)
) RAM128X1D_inst (
  .DPO(DPO),      // Read port 1-bit output
  .SPO(SPO),      // Read/Write port 1-bit output
  .A(A),          // Read/Write port 7-bit address input
  .D(D),          // RAM data input
  .DPRA(DPRA),    // Read port 7-bit address input
  .WCLK(WCLK),    // Write clock input
  .WE(WE)         // Write enable input
);

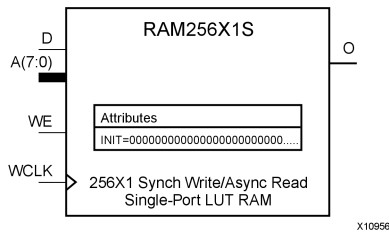
// End of RAM128X1D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAM256X1S

### プリミティブ：256-Deep by 1-Wide Random Access Memory (Select RAM)



## 概要

このデザイン エLEMENTは、256 ワード X 1 ビットの RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM と呼ばれる) を使用してインプリメントされるため、ブロック RAM リソースは使用しません。同期読み出しを行う場合は、出力にレジスタを付けて同じスライスに配置できます。この場合、RAM とレジスタで同じクロックを使用する必要があります。アクティブ High のライト イネーブル WE が High になると、WCLK ピンの立ち上がりエッジで D 入力データピンの値がメモリ アレイに書き込まれます。出力 O は、WE の値にかかわらず、アドレス バス A で指定されたメモリ ロケーションの値を出力します。書き込みが実行されると、その直後に出力の値が新しい値に更新されます。

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	アドレス バス A で指定される読み出し/書き込みポートのデータ出力
D	入力	1	アドレス バス A で指定される書き込みデータ入力
A	入力	8	読み出し/書き込みポートのアドレス バス
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

## デザインの入力方法

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

インスタンス化する場合、このコンポーネントを次のように接続します。

- ・ WCLK 入力をクロック ソースに、D 入力を格納するデータ ソースに、O 出力を FDCE の D 入力などの適切なデスティネーションに接続します。
- ・ クロック イネーブル ピン WE は、適切なライト イネーブル ソースに接続します。
- ・ 8 ビット バス A は、読み出し/書き込みのソースに接続します。
- ・ 256 ビットの 16 進数で構成される INIT 属性で、RAM の初期値を指定できます。

指定しない場合は、初期値はすべて 0 になります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	256 ビット値	すべて 0	RAM の初期値を指定します。

## VHDL 記述 (インスタンスレーション)

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

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

-- RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
--             single-port distributed LUT RAM
--             Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM256X1S_inst : RAM256X1S
generic map (
  INIT => X"0000000000000000000000000000000000000000000000000000000000000000"
) port map (
  O => O, -- Read/Write port 1-bit output
  A => A, -- Read/Write port 8-bit address input
  D => D, -- RAM data input
  WCLK => WCLK, -- Write clock input
  WE => WE -- Write enable input
);

-- End of RAM256X1S_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
//             single-port distributed LUT RAM
//             Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

RAM256X1S #(
  .INIT(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAM256X1S_inst (
  .O(O), // Readw/rite port 1-bit output
  .A(A), // Readw/rite port 8-bit address input
  .WE(WE), // Write enable input
  .WCLK(WCLK), // Write clock input
  .D(D) // RAM data input
);

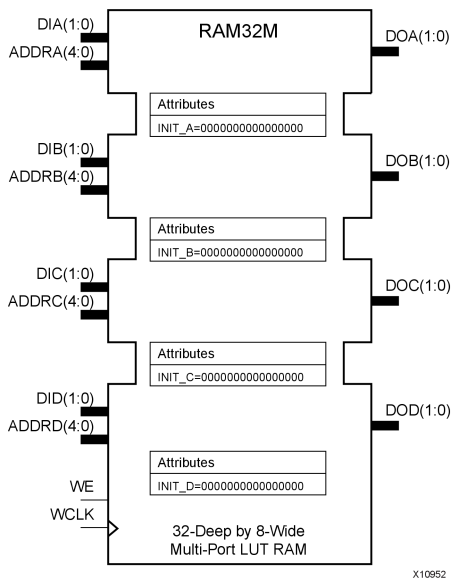
// End of RAM256X1S_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAM32M

**プリミティブ：32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)**



X10952

## 概要

このデザイン エLEMENTは、32 ワード X 8 ビットのマルチポート RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM™) を使用してインプリメントされるため、デバイスのブロック RAM リソースを使用しません。RAM32M コンポーネントは 1 つのスライスにインプリメントされ、8 ビット書き込みポート 1 つ、2 ビット読み出しポート 1 つ、および同じメモリからの 2 ビット読み出しポート 3 つから構成されています。これにより、RAM のバイト幅の書き込みと独立した 2 ビットの読み出しが可能です。DIA、DIB、DIC、および DID 入力をすべて同じデータ入力に接続すると、読み出し/書き込みポート 1 つ、独立した読み出しポート 3 つの 32x2 クワッドポートメモリになります。DID をグランドに接続した場合、DOD は使用されません。ADDRA、ADDRb、ADDRc を同じアドレスに接続すると、32x6 のシンプルデュアルポート RAM になります。ADDRd を ADDRA、ADDRb、ADDRc に接続すると、32x8 のシングルポート RAM になります。この RAM には、ほかにも可能なコンフィギュレーションがあります。

## ポートの説明

ポート名	方向	幅	機能
DOA	出力	2	アドレス バス ADDRA で指定された読み出しポートのデータ出力
DOB	出力	2	アドレス バス ADDRb で指定された読み出しポートのデータ出力
DOC	出力	2	アドレス バス ADDRc で指定された読み出しポートのデータ出力
DOD	出力	2	アドレス バス ADDRd で指定された読み出し/書き込みポートのデータ出力
DIA	入力	2	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRA で指定)
DIB	入力	2	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRb で指定)

ポート名	方向	幅	機能
DIC	入力	2	ADDRD で指定された書き込みデータ入力 (読み出し出力は ADDRRC で指定)
DID	入力	2	アドレス バス ADDRD で指定された書き込みデータ入力
ADDRA	入力	5	読み出しアドレス バス A
ADDRB	入力	5	読み出しアドレス バス B
ADDRC	入力	5	読み出しアドレス バス C
ADDRD	入力	5	8 ビットのデータ書き込みポート、2 ビットのデータ読み出しポートのアドレス バス D
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

## デザインの入力方法

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

このエレメントは、同期書き込みと非同期読み出し機能を備えた RAM を記述することにより、一部の合成ツールで推論できます。RAM の推論およびコード例の詳細は、合成ツールのマニュアルを参照してください。RAM32M のインスタンス化は、RAM ファンクションを暗示的に指定する必要がある場合、コンポーネントを手動でまたは相対的に配置する必要がある場合に実行することをお勧めします。同期読み出しが必要な場合は、RAM32M の出力を FDRSE (非同期セット/リセットが必要な場合は FDCPE) に接続してファンクションの出力タイミングを向上させることも可能ですが、RAM の正しい動作には不要です。

インバーターをこのコンポーネントのクロック入力に追加すると、クロックの立ち下がりエッジでデータを入力できます。このインバーターはブロック内に組み込まれ、クロックの立ち下がりエッジで RAM への書き込みを実行できます。

インスタンス化する場合、このコンポーネントは、次のように接続します。WCLK 入力をクロックソースに、DIA、DIB、DIC、DID 入力を格納するデータソースに、DOA、DOB、DOC、DOD 出力を FDCE の D 入力などの適切なデスティネーションに接続するか、使用しない場合は未接続のままにします。クロック イネーブルピン WE は、適切なライト イネーブルソースに接続します。5 ビット バス ADDRD は読み出し/書き込みアドレスに、5 ビット バス ADDRA、ADDRB、ADDRC は読み出しアドレスに接続する必要があります。オプションで INIT\_A、INIT\_B、INIT\_C、INIT\_D 属性を使用すると、各ポートの初期メモリ内容を 64 ビット (16 進数) で指定できます。RAM の INIT 値は、 $ADDRy[z] = INIT_y[2*z+1:2*z]$  で計算されます。たとえば、RAM の ADDRRC ポートが 00001 の場合、INIT\_C[3:2] 値がそのアドレスで最初の書き込みが行われる前の DOC ポートの初期値になります。指定しない場合は、初期値はすべて 0 になります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_A	16 進数	64 ビット値	すべて 0	A ポートの RAM の初期値を指定します。
INIT_B	16 進数	64 ビット値	すべて 0	B ポートの RAM の初期値を指定します。
INIT_C	16 進数	64 ビット値	すべて 0	C ポートの RAM の初期値を指定します。
INIT_D	16 進数	64 ビット値	すべて 0	D ポートの RAM の初期値を指定します。



## VHDL 記述 (インスタンス化)

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

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

-- RAM32M: 32-deep by 8-wide Multi Port LUT RAM
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM32M_inst : RAM32M
generic map (
  INIT_A => X"0000000000000000",  -- Initial contents of A port
  INIT_B => X"0000000000000000",  -- Initial contents of B port
  INIT_C => X"0000000000000000",  -- Initial contents of C port
  INIT_D => X"0000000000000000")  -- Initial contents of D port
port map (
  DOA => DOA, -- Read port A 2-bit output
  DOB => DOB, -- Read port B 2-bit output
  DOC => DOC, -- Read port C 2-bit output
  DOD => DOD, -- Read/Write port D 2-bit output
  ADDRA => ADDRA,  -- Read port A 5-bit address input
  ADDR_B => ADDR_B,  -- Read port B 5-bit address input
  ADDR_C => ADDR_C,  -- Read port C 5-bit address input
  ADDR_D => ADDR_D,  -- Read/Write port D 5-bit address input
  DIA => DIA, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDRA
  DIB => DIB, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_B
  DIC => DIC, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_C
  DID => DID, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_D
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);
-- End of RAM32M_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM32M: 32-deep by 8-wide Multi Port LUT RAM
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

RAM32M #(
    .INIT_A(64'h0000000000000000), // Initial contents of A Port
    .INIT_B(64'h0000000000000000), // Initial contents of B Port
    .INIT_C(64'h0000000000000000), // Initial contents of C Port
    .INIT_D(64'h0000000000000000) // Initial contents of D Port
) RAM32M_inst (
    .DOA(DOA), // Read port A 2-bit output
    .DOB(DOB), // Read port B 2-bit output
    .DOC(DOC), // Read port C 2-bit output
    .DOD(DOD), // Readw/rite port D 2-bit output
    .ADDRA(ADDRA), // Read port A 5-bit address input
    .ADDRB(ADDRB), // Read port B 5-bit address input
    .ADDRC(ADDRC), // Read port C 5-bit address input
    .ADDRD(ADDRD), // Readw/rite port D 5-bit address input
    .DIA(DIA), // RAM 2-bit data write input addressed by ADDRD,
                // read addressed by ADDRA
    .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
);

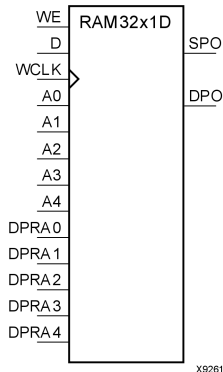
// End of RAM32M_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAM32X1D

### プリミティブ：32-Deep by 1-Wide Static Dual Port Synchronous RAM



### 概要

このデザイン エLEMENTは 32 ワード X 1 ビットのデュアル ポート SRAM で、同期書き込み機能を備えています。デバイスには、読み出しアドレス (DPRA4 ~ DPRA0) と書き込みアドレス (A4 ~ A0) の独立した 2 種類のアドレス ポートがあります。この 2 種類のアドレス ポートは完全に非同期です。読み出しアドレスによって出力ピン (DPO) に出力される値が指定され、書き込みアドレスによって書き込みを行う位置が指定されます。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High の場合、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 5 ビットの書き込みアドレスで選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。INIT 属性を使用すると、コンフィギュレーション中に RAM32X1D を初期化できます。モード選択を次の論理表に示します。

SPO 出力には、A4 ~ A0 で指定されたメモリ セルの値が出力されます。DPO 出力には、DPRA4 ~ DPRA0 で指定されたメモリ セルの値が出力されます。書き込み処理は、読み出しアドレス ポートのアドレスには影響されません。

### 論理表

入力			出力	
WE (モード)	WCLK	D	SPO	DPO
0 (読み出し)	X	X	data_a	data_d
1 (読み出し)	0	X	data_a	data_d
1 (読み出し)	1	X	data_a	data_d
1 (書き込み)	↑	D	D	data_d
1 (読み出し)	↓	X	data_a	data_d

### デザインの入力方法

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

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	ROM、RAM、レジスタ、LUT の初期値を指定します。

## VHDL 記述 (インスタンスエーション)

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

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

-- RAM32X1D: 32 x 1 positive edge write, asynchronous read
--           dual-port distributed RAM
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM32X1D_inst : RAM32X1D
generic map (
  INIT => X"00000000") -- Initial contents of RAM
port map (
  DPO => DPO,      -- Read-only 1-bit data output
  SPO => SPO,      -- R/W 1-bit data output
  A0 => A0,        -- R/W address[0] input bit
  A1 => A1,        -- R/W address[1] input bit
  A2 => A2,        -- R/W address[2] input bit
  A3 => A3,        -- R/W address[3] input bit
  A4 => A4,        -- R/W address[4] input bit
  D => D,          -- Write 1-bit data input
  DPRA0 => DPRA0,  -- Read-only address[0] input bit
  DPRA1 => DPRA1,  -- Read-only address[1] input bit
  DPRA2 => DPRA2,  -- Read-only address[2] input bit
  DPRA3 => DPRA3,  -- Read-only address[3] input bit
  DPRA4 => DPRA4,  -- Read-only address[4] input bit
  WCLK => WCLK,    -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM32X1D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM32X1D: 32 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

RAM32X1D #(
    .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1D_inst (
    .DPO(DPO),          // Read-only 1-bit data output
    .SPO(SPO),          // Rw/ 1-bit data output
    .A0(A0),            // Rw/ address[0] input bit
    .A1(A1),            // Rw/ address[1] input bit
    .A2(A2),            // Rw/ address[2] input bit
    .A3(A3),            // Rw/ address[3] input bit
    .A4(A4),            // Rw/ address[4] input bit
    .D(D),              // Write 1-bit data input
    .DPRA0(DPRA0),      // Read-only address[0] input bit
    .DPRA1(DPRA1),      // Read-only address[1] input bit
    .DPRA2(DPRA2),      // Read-only address[2] input bit
    .DPRA3(DPRA3),      // Read-only address[3] input bit
    .DPRA4(DPRA4),      // Read-only address[4] input bit
    .WCLK(WCLK),        // Write clock input
    .WE(WE)             // Write enable input
);

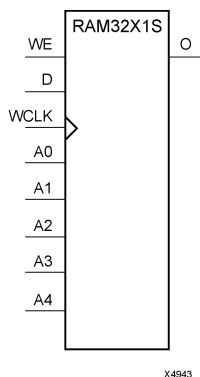
// End of RAM32X1D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAM32X1S

プリミティブ：32-Deep by 1-Wide Static 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 SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。INIT 属性を使用すると、コンフィギュレーション中に RAM32X1S を初期化できます。

### 論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D	D
1 (読み出し)	↓	X	データ

### デザインの入力方法

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

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	RAM の初期値を指定します。

## VHDL 記述 (インスタンスレーション)

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
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM32X1S_inst : RAM32X1S
generic map (
  INIT => X"00000000")
port map (
  O => O,      -- RAM output
  A0 => A0,     -- RAM address[0] input
  A1 => A1,     -- RAM address[1] input
  A2 => A2,     -- RAM address[2] input
  A3 => A3,     -- RAM address[3] input
  A4 => A4,     -- RAM address[4] input
  D => D,       -- RAM data input
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);

-- End of RAM32X1S_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 14.5

RAM32X1S #(
  .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1S_inst (
  .O(O),              // RAM output
  .A0(A0),            // RAM address[0] input
  .A1(A1),            // RAM address[1] input
  .A2(A2),            // RAM address[2] input
  .A3(A3),            // RAM address[3] input
  .A4(A4),            // RAM address[4] input
  .D(D),              // RAM data input
  .WCLK(WCLK),        // Write clock input
  .WE(WE)             // Write enable input
);

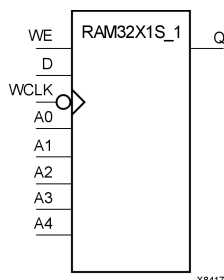
// End of RAM32X1S_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	0	RAM、レジスタ、LUT の初期値を指定します。

## VHDL 記述 (インスタンスレーション)

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
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM32X1S_1_inst : RAM32X1S_1
generic map (
  INIT => X"00000000")
port map (
  O => O,      -- RAM output
  A0 => A0,     -- RAM address[0] input
  A1 => A1,     -- RAM address[1] input
  A2 => A2,     -- RAM address[2] input
  A3 => A3,     -- RAM address[3] input
  A4 => A4,     -- RAM address[4] input
  D => D,       -- RAM data input
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);

-- End of RAM32X1S_1_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 14.5

RAM32X1S_1 #(
  .INIT(32'h00000000) // Initial contents of RAM
)RAM32X1S_1_inst (
  .O(O),              // RAM output
  .A0(A0),            // RAM address[0] input
  .A1(A1),            // RAM address[1] input
  .A2(A2),            // RAM address[2] input
  .A3(A3),            // RAM address[3] input
  .A4(A4),            // RAM address[4] input
  .D(D),              // RAM data input
  .WCLK(WCLK),        // Write clock input
  .WE(WE)             // Write enable input
);

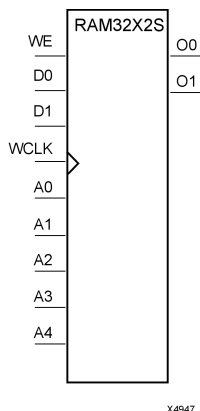
// End of RAM32X1S_1_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAM32X2S

プリミティブ：32-Deep by 2-Wide Static Synchronous RAM



### 概要

このデザイン エLEMENTは 32 ワード X 2 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、入力 (D1 ~ D0) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。出力ピン (O1 ~ O0) に出力される値は、アドレスピンで指定された RAM 内の位置に格納されている値です。

INIT\_00 および INIT\_01 属性を使用して RAM32X2S の初期値を指定できます。

### 論理表

入力			出力
WE (モード)	WCLK	D	O0 ~ O1
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D1 ~ D0	D1 ~ D0
1 (読み出し)	↓	X	データ
データ = A4 ~ A0 で指定されたワード			

### デザインの入力方法

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

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT_00	16 進数	32 ビット値	すべて 0	RAM のビット 0 の初期値を指定します。
INIT_01	16 進数	32 ビット値	すべて 0	RAM のビット 1 の初期値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM32X2S_inst : RAM32X2S
generic map (
    INIT_00 => X"00000000", -- INIT for bit 0 of RAM
    INIT_01 => X"00000000") -- INIT for bit 1 of RAM
port map (
    O0 => O0,      -- RAM data[0] output
    O1 => O1,      -- RAM data[1] output
    A0 => A0,      -- RAM address[0] input
    A1 => A1,      -- RAM address[1] input
    A2 => A2,      -- RAM address[2] input
    A3 => A3,      -- RAM address[3] input
    A4 => A4,      -- RAM address[4] input
    D0 => D0,      -- RAM data[0] input
    D1 => D1,      -- RAM data[1] input
    WCLK => WCLK,  -- Write clock input
    WE => WE       -- Write enable input
);

-- End of RAM32X2S_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 14.5

RAM32X2S #(
    .INIT_00(32'h00000000), // INIT for bit 0 of RAM
    .INIT_01(32'h00000000) // INIT for bit 1 of RAM
) RAM32X2S_inst (
    .O0(O0),      // RAM data[0] output
    .O1(O1),      // RAM data[1] output
    .A0(A0),      // RAM address[0] input
    .A1(A1),      // RAM address[1] input
    .A2(A2),      // RAM address[2] input
    .A3(A3),      // RAM address[3] input
    .A4(A4),      // RAM address[4] input
    .D0(D0),      // RAM data[0] input
    .D1(D1),      // RAM data[1] input
    .WCLK(WCLK),  // Write clock input
    .WE(WE)       // Write enable input
);

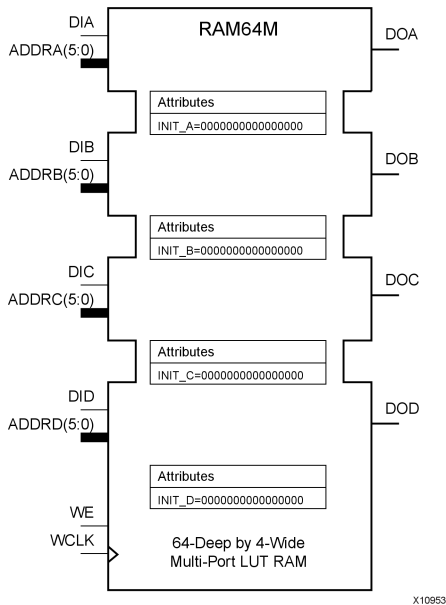
// End of RAM32X2S_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAM64M

**プリミティブ：64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)**



### 概要

このデザイン エLEMENTは、64 ワード X 4 ビットのマルチポート RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM™ と呼ばれる) を使用してインプリメントされるため、ブロック RAM リソースを使用しません。RAM64M コンポーネントは 1 つのスライスにインプリメントされます。4 ビット書き込みポート 1 つ、1 ビット読み出しのポート 1 つ、および同じメモリからの 1 ビット読み出しポート 3 つから構成されており、RAM の 4 ビット書き込みおよび個別ビット読み出しが可能です。DIA、DIB、DIC、および DID 入力をすべて同じデータ入力に接続すると、読み出し/書き込みポート 1 つ、独立した読み出しポート 3 つの 64x1 クワッド ポートメモリになります。DID をグラウンドに接続した場合、DOD は使用されません。ADDRA、ADDRB、ADDRC を同じアドレスに接続すると、64x3 のシングル デュアル ポート RAM になります。ADDRD を ADDRA、ADDRB、ADDRC に接続すると、64x4 のシングル ポート RAM になります。この RAM には、ほかにも可能なコンフィギュレーションがあります。

### ポートの説明

ポート名	方向	幅	機能
DOA	出力	1	アドレス バス ADDRA で指定される読み出しポートのデータ出力
DOB	出力	1	アドレス バス ADDR B 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 ADDR C 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 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	1	ADDRD で指定される書き込みデータ入力 (読み出し出力は ADDRA で指定)
DIB	入力	1	ADDRD で指定される書き込みデータ入力 (読み出し出力は ADDR B で指定)

ポート名	方向	幅	機能
DIC	入力	1	ADDRD で指定される書き込みデータ入力 (読み出し出力は ADDRRC で指定)
DID	入力	1	アドレス バス ADDRD で指定される書き込みデータ入力
ADDRA	入力	6	読み出しアドレス バス A
ADDRB	入力	6	読み出しアドレス バス B
ADDRC	入力	6	読み出しアドレス バス C
ADDRD	入力	6	4 ビットのデータ書き込みポート、1 ビットのデータ読み出しポートのアドレス バス D
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

## デザインの入力方法

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

このエレメントは、同期書き込みと非同期読み出し機能を備えた RAM を記述することにより、一部の合成ツールで推論できます。RAM の推論およびコード例の詳細は、合成ツールのマニュアルを参照してください。RAM64M のインスタンス化は、RAM ファンクションを暗示的に指定する必要がある場合、コンポーネントを手動でまたは相対的に配置する必要がある場合に実行することをお勧めします。同期読み出しが必要な場合は、RAM64M の出力を FDRSE (非同期セット/リセットが必要な場合は FDCPE) に接続してファンクションの出力タイミングを向上させることも可能ですが、RAM の正しい動作には不要です。インバーターをこのコンポーネントのクロック入力に追加すると、クロックの立ち下がりエッジでデータを入力できます。このインバーターはブロック内に組み込まれ、クロックの立ち下がりエッジで RAM への書き込みを実行できます。

インスタンス化する場合、このコンポーネントは、次のように接続します。

- WCLK 入力をクロック ソースに、DIA、DIB、DIC、DID 入力を格納するデータ ソースに、DOA、DOB、DOC、DOD 出力を FDCE の D 入力などの適切なデスティネーションに接続するか、使用しない場合は未接続のままにします。
- クロック イネーブル ピン (WE) は、適切なライト イネーブル ソースに接続します。
- 5 ビット バス ADDRD は読み出し/書き込みアドレスに、5 ビット バス ADDRA、ADDRB、ADDRC は読み出しアドレスに接続する必要があります。
- オプションで INIT\_A、INIT\_B、INIT\_C、INIT\_D 属性を使用すると、各ポートの初期メモリ内容を 64 ビット (16 進数) で指定できます。RAM の INIT 値は、 $\text{ADDRy}[z] = \text{INIT}_y[z]$  で計算されます。

たとえば、RAM の ADDRRC ポートが 00001 の場合、INIT\_C[1] 値がそのアドレスで最初の書き込みが行われる前の DOC ポートの初期値になります。指定しない場合は、初期値はすべて 0 になります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_A	16 進数	64 ビット値	すべて 0	A ポートの RAM の初期値を指定します。
INIT_B	16 進数	64 ビット値	すべて 0	B ポートの RAM の初期値を指定します。
INIT_C	16 進数	64 ビット値	すべて 0	C ポートの RAM の初期値を指定します。
INIT_D	16 進数	64 ビット値	すべて 0	D ポートの RAM の初期値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- RAM64M: 64-deep by 4-wide Multi Port LUT RAM
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM64M_inst : RAM64M
generic map (
  INIT_A => X"0000000000000000", -- Initial contents of A port
  INIT_B => X"0000000000000000", -- Initial contents of B port
  INIT_C => X"0000000000000000", -- Initial contents of C port
  INIT_D => X"0000000000000000"  -- Initial contents of D port
) port map (
  DOA => DOA, -- Read port A 1-bit output
  DOB => DOB, -- Read port B 1-bit output
  DOC => DOC, -- Read port C 1-bit output
  DOD => DOD, -- Read/Write port D 1-bit output
  ADDRA => ADDRA, -- Read port A 6-bit address input
  ADDR_B => ADDR_B, -- Read port B 6-bit address input
  ADDR_C => ADDR_C, -- Read port C 6-bit address input
  ADDR_D => ADDR_D, -- Read/Write port D 6-bit address input
  DIA => DIA, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_A
  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
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

RAM64M #(
  .INIT_A(64'h0000000000000000), // Initial contents of A Port
  .INIT_B(64'h0000000000000000), // Initial contents of B Port
  .INIT_C(64'h0000000000000000), // Initial contents of C Port
  .INIT_D(64'h0000000000000000) // Initial contents of D Port
) RAM64M_inst (
  .DOA(DOA), // Read port A 1-bit output
  .DOB(DOB), // Read port B 1-bit output
  .DOC(DOC), // Read port C 1-bit output
  .DOD(DOD), // Read/Write port D 1-bit output
```

```
.DIA(DIA),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRdA  
.DIB(DIB),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRdB  
.DIC(DIC),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRdC  
.DID(DID),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRdD  
.ADDRd(ADDRd), // Read port A 6-bit address input  
.ADDRB(ADDRB), // Read port B 6-bit address input  
.ADDRc(ADDRc), // Read port C 6-bit address input  
.ADDRd(ADDRd), // Readw/rite port D 6-bit address input  
.WE(WE),        // Write enable input  
.WCLK(WCLK)     // Write clock input  
);  
  
// End of RAM64M_inst instantiation
```

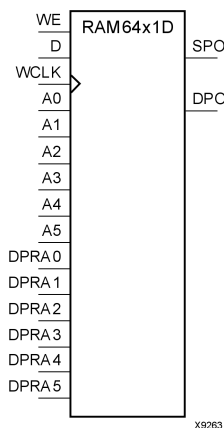
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## RAM64X1D

**プリミティブ：64-Deep by 1-Wide Dual Port Static Synchronous RAM**



### 概要

このデザイン エLEMENTは 64 ワード X 1 ビットのデュアル ポート SRAM で、同期書き込み機能を備えています。デバイスには、読み出しアドレス (DPRA5 ～ DPRA0) と書き込みアドレス (A5 ～ A0) の独立した 2 種類のアドレス ポートがあります。この 2 種類のアドレス ポートは完全に非同期です。読み出しアドレスによって出力ピン (DPO) に出力される値が指定され、書き込みアドレスによって書き込みを行う位置が指定されます。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。

WE が High になると、WCLK が Low から High に切り替わる時に、データ入力 (D) の値が 6 ビットの書き込みアドレス (A0 ～ A5) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

SPO 出力には、A5 ～ A0 で指定されたメモリ セルの値が出力されます。DPO 出力には、DPRA5 ～ DPRA0 で指定されたメモリ セルの値が出力されます。

**注記：** 書き込み処理は、読み出しアドレス ポートのアドレスには影響されません。

### 論理表

入力			出力	
WE (モード)	WCLK	D	SPO	DPO
0 (読み出し)	X	X	data_a	data_d
1 (読み出し)	0	X	data_a	data_d
1 (読み出し)	1	X	data_a	data_d
1 (書き込み)	↑	D	D	data_d
1 (読み出し)	↓	X	data_a	data_d
data_a = A5 ～ A0 で指定されたワード				
data_d = DPRA5 ～ DPRA0 で指定されたワード				

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	RAM の初期値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- RAM64X1D: 64 x 1 negative edge write, asynchronous read
--           dual-port distributed RAM
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM64X1D_1_inst : RAM64X1D_1
generic map (
  INIT => X"0000000000000000") -- Initial contents of RAM
port map (
  DPO => DPO,      -- Read-only 1-bit data output
  SPO => SPO,      -- R/W 1-bit data output
  A0 => A0,        -- R/W address[0] input bit
  A1 => A1,        -- R/W address[1] input bit
  A2 => A2,        -- R/W address[2] input bit
  A3 => A3,        -- R/W address[3] input bit
  A4 => A4,        -- R/W address[4] input bit
  A5 => A5,        -- R/W address[5] input bit
  D => D,          -- Write 1-bit data input
  DPRA0 => DPRA0,  -- Read-only address[0] input bit
  DPRA1 => DPRA1,  -- Read-only address[1] input bit
  DPRA2 => DPRA2,  -- Read-only address[2] input bit
  DPRA3 => DPRA3,  -- Read-only address[3] input bit
  DPRA4 => DPRA4,  -- Read-only address[4] input bit
  DPRA5 => DPRA5,  -- Read-only address[5] input bit
  WCLK => WCLK,    -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM64X1D_1_inst instantiation
```

## Verilog 記述 (インスタンスエーション)

```
// RAM64X1D: 64 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

RAM64X1D #(
    .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1D_inst (
    .DPO(DPO),           // Read-only 1-bit data output
    .SPO(SPO),           // Rw/ 1-bit data output
    .A0(A0),             // Rw/ address[0] input bit
    .A1(A1),             // Rw/ address[1] input bit
    .A2(A2),             // Rw/ address[2] input bit
    .A3(A3),             // Rw/ address[3] input bit
    .A4(A4),             // Rw/ address[4] input bit
    .A5(A5),             // Rw/ address[5] input bit
    .D(D),               // Write 1-bit data input
    .DPRA0(DPRA0),       // Read-only address[0] input bit
    .DPRA1(DPRA1),       // Read-only address[1] input bit
    .DPRA2(DPRA2),       // Read-only address[2] input bit
    .DPRA3(DPRA3),       // Read-only address[3] input bit
    .DPRA4(DPRA4),       // Read-only address[4] input bit
    .DPRA5(DPRA5),       // Read-only address[5] input bit
    .WCLK(WCLK),         // Write clock input
    .WE(WE)              // Write enable input
);

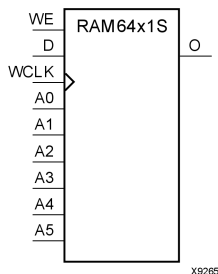
// End of RAM64X1D_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## 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 0	ROM、RAM、レジスタ、LUT の初期値を指定します。

## VHDL 記述 (インスタンスレーション)

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

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

-- RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM64X1S_inst : RAM64X1S
generic map (
  INIT => X"0000000000000000")
port map (
  O => O,          -- 1-bit data output
  A0 => A0,         -- Address[0] input bit
  A1 => A1,         -- Address[1] input bit
  A2 => A2,         -- Address[2] input bit
  A3 => A3,         -- Address[3] input bit
  A4 => A4,         -- Address[4] input bit
  A5 => A5,         -- Address[5] input bit
  D => D,          -- 1-bit data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM64X1S_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 14.5

RAM64X1S #(
  .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_inst (
  .O(O),           // 1-bit data output
  .A0(A0),         // Address[0] input bit
  .A1(A1),         // Address[1] input bit
  .A2(A2),         // Address[2] input bit
  .A3(A3),         // Address[3] input bit
  .A4(A4),         // Address[4] input bit
  .A5(A5),         // Address[5] input bit
  .D(D),           // 1-bit data input
  .WCLK(WCLK),     // Write clock input
  .WE(WE)          // Write enable input
);

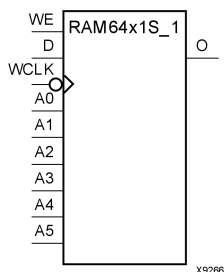
// End of RAM64X1S_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAM64X1S\_1

プリミティブ：64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



### 概要

このデザイン エLEMENTは 64 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が High から Low に切り替わるときに、データ入力 (D) の値が 6 ビットのアドレス (A5 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が High から Low に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ Low ですが、インバーターを使用してアクティブ High にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。

INIT 属性を使用すると、コンフィギュレーション中にこのELEMENTを初期化できます。

### 論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↓	D	D
1 (読み出し)	↑	X	データ
データ = A5 ~ A0 で指定されたワード			

### デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ROM、RAM、レジスタ、LUT の初期値を指定します。

## VHDL 記述 (インスタンスレーション)

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

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

-- RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAM64X1S_1_inst : RAM64X1S_1
generic map (
  INIT => X"0000000000000000")
port map (
  O => O,          -- 1-bit data output
  A0 => A0,         -- Address[0] input bit
  A1 => A1,         -- Address[1] input bit
  A2 => A2,         -- Address[2] input bit
  A3 => A3,         -- Address[3] input bit
  A4 => A4,         -- Address[4] input bit
  A5 => A5,         -- Address[5] input bit
  D => D,          -- 1-bit data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM64X1S_1_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 14.5

RAM64X1S_1 #(
  .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_1_inst (
  .O(O),          // 1-bit data output
  .A0(A0),        // Address[0] input bit
  .A1(A1),        // Address[1] input bit
  .A2(A2),        // Address[2] input bit
  .A3(A3),        // Address[3] input bit
  .A4(A4),        // Address[4] input bit
  .A5(A5),        // Address[5] input bit
  .D(D),          // 1-bit data input
  .WCLK(WCLK),    // Write clock input
  .WE(WE)         // Write enable input
);

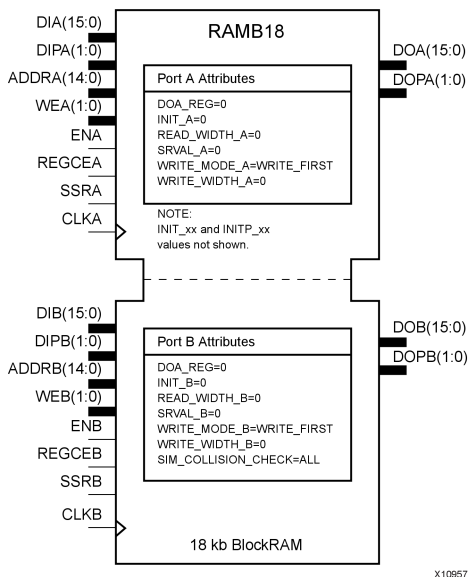
// End of RAM64X1S_1_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAMB18

### プリミティブ：18K-bit Configurable Synchronous True Dual Port Block RAM



## 概要

Virtex®-5 デバイスにはブロック RAM が数個含まれ、FIFO、自動誤り訂正 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。RAMB18 を使用すると、18kb 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 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。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	機能
DOA, DOB	出力	1、2、4、8、16	ポート A/B のデータ出力バス
DOPA, DOPB	出力	0、1、2	ポート A/B のパリティ データ出力バス
DIA, DIB	入力	1、2、4、8、16	ポート A/B のデータ入力バス
DIPA, DIPB	入力	0、1、2	ポート A/B のパリティ データ入力バス
ADDRA, ADDRb	入力	14	ポート A/B のアドレス入力バス
WEA	入力	2	ポート A のバイト幅ライト イネーブル
WEB	入力	2	ポート B のバイト幅ライト イネーブル
ENA, ENB	入力	1	ポート A/B のイネーブル
SSRA, SSRB	入力	1	ポート A/B の出力レジスタの同期セット/リセット。High になると、関連ポートのプリセット/リセットが SRVAL_A/SRVAL_B で指定した値になります。



ポート名	方向	幅	機能
REGCEA、REGCEB	入力	1	ポート A/B の出力レジスタのクロック イネーブル入力
CLKA、CLKB	入力	1	ポート A/B のクロック入力

## 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	可
マクロのサポート	可

次の表に、さまざまなポート幅のデータ、アドレスおよびライトイネーブル接続を、ポート A またはポート B の DATA\_WIDTH の値ごとに示します。同じポートの読み出しと書き込みで異なる幅を使用する場合は、アドレス接続を選択できるようにこの表のデータ幅が大きい方の 2 つを使用してください。

特定のコンフィギュレーションで必要にならないデータ ポートおよびアドレス ポートは、次の例外を除き、すべて未接続にするかまたはグランドに接続する必要があります。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1	DI[0]	ADDR[14:0]	WE[1:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[14:1]	WE[1:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[14:2]	WE[1:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[14:3]	WE[1:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]
18	DI[15:0]、DIP[1:0]	ADDR[14:4]	WE[1:0] および WE[1] をユーザー WE 信号に接続	DO[15:0]、DOP[1:0]

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DOA_REG、DOB_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロック サイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能ですが、clock-to-out タイムが長くなります。
INIT_A	16 進数	18 ビット値	すべて 0	コンフィギュレーション後のポート A の出力の初期値を指定します。
INIT_B	16 進数	18 ビット値	すべて 0	コンフィギュレーション後のポート B の出力の初期値を指定します。
READ_WIDTH_A	整数	0、1、2、4、9、18	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)。0 の場合、ポートは使用されません。

属性	データ型	値	デフォルト	説明
READ_WIDTH_B	整数	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)。0 の場合、ポートは使用されません。
SIM_COLLISION_CHECK	文字列	"ALL"、 "WARNING_ONLY"、 "GENERATE_X_ONLY"、 "NONE"	"ALL"	<p>メモリの競合が発生した場合のシミュレーションの動作を指定します。出力は次のようになります。</p> <ul style="list-style-type: none"> <li>・ "ALL" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。</li> <li>・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。</li> </ul> <p><b>注記 :</b> "ALL" 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SIM_MODE	文字列	"SAFE"、"FAST"	"SAFE"	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。
SRVAL_A	16 進数	18 ビット値	すべて 0	同期リセット信号 (SSRA) がアサートされたときのポート A の出力値を指定します。
SRVAL_B	16 進数	18 ビット値	すべて 0	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みが実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> <li>・ "WRITE_FIRST" : 書き込まれた値が出力ポートに出力されます。</li> <li>・ "READ_FIRST" : 書き込み前にそのメモリロケーションに格納されていた値が出力ポートに出力されます。</li> <li>・ "NO_CHANGE" : 出力ポートの以前の値が保持されます。</li> </ul>
WRITE_WIDTH_A	整数	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)。0 の場合、ポートは使用されません。

属性	データ型	値	デフォルト	説明
WRITE_WIDTH_B	整数	0、1、2、4、9、18	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)。0 の場合、ポートは使用されません。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべて 0	16kb のデータ メモリ アレイの初期値を指定します。
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべて 0	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。

### INIT\_A、INIT\_B、SRVAL\_A、SRVAL\_B のマップ

INIT\_A、INIT\_B、SRVAL\_A および SRVAL\_B の属性はすべて 18 ビットですが、特定のポートで READ\_WIDTH が 18 ビット未満の値に設定されていると、そのビットのサブセットのみが使用されます。

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB18: 16k+2k Parity Parameterizable True Dual-Port BlockRAM
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAMB18_inst : RAMB18
generic map (
    DOA_REG => 0, -- Optional output register on A port (0 or 1)
    DOB_REG => 0, -- Optional output register on B port (0 or 1)
    INIT_A => X"00000", -- Initial values on A output port
    INIT_B => X"00000", -- Initial values on B output port
    READ_WIDTH_A => 0, -- Valid values are 1, 2, 4, 9, or 18
    READ_WIDTH_B => 0, -- Valid values are 1, 2, 4, 9, or 18
    SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
    -- "GENERATE_X_ONLY" or "NONE"
    SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
    -- Design Guide" for details
    SRVAL_A => X"00000", -- Set/Reset value for A port output
    SRVAL_B => X"00000", -- Set/Reset value for B port 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, 2, 4, 9, or 18
    WRITE_WIDTH_B => 0, -- Valid values are 1, 2, 4, 9, or 18
    -- 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",
    INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",

```

```
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- 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",
port map (
  DOA => DOA, -- 16-bit A port data output
  DOB => DOB, -- 16-bit B port data output
  DOPA => DOPA, -- 2-bit A port parity data output
  DOPB => DOPB, -- 2-bit B port parity data output
  ADDRA => ADDRA, -- 14-bit A port address input
  ADDRb => ADDRb, -- 14-bit B port address input
  CLKA => CLKA, -- 1-bit A port clock input
  CLKB => CLKB, -- 1 bit B port clock input
  DIA => DIA, -- 16-bit A port data input
  DIB => DIB, -- 16-bit B port data input
  DIPA => DIPA, -- 2-bit A port parity data input
  DIPB => DIPB, -- 2-bit B port parity data input
  ENA => ENA, -- 1-bit A port enable input
  ENB => ENB, -- 1-bit B port enable input
  REGCEA => REGCEA, -- 1-bit A port register enable input
  REGCEB => REGCEB, -- 1-bit B port register enable input
  SSRA => SSRA, -- 1-bit A port set/reset input
  SSRB => SSRB, -- 1-bit B port set/reset input
  WEA => WEA, -- 2-bit A port write enable input
  WEB => WEB -- 2-bit B port write enable input
```

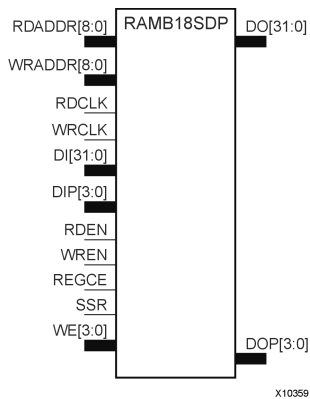


## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAMB18SDP

**プリミティブ：36-bit by 512 Deep, 18kb Synchronous Simple Dual Port Block RAM**



### 概要

このデザイン エLEMENTはブロック RAM メモリの 1 つで、FIFO、自動誤り訂正 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。RAMB18SDP を使用すると、18kb コンフィギュレーションでブロック RAM にアクセスできます。このコンポーネントは、36 ビット幅 X 512 ワードのシングル デュアル ポート RAM に設定されています。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。バイト イネーブルの書き込みが可能で、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

**注記：** このELEMENT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	32	RDADDR で指定されたデータ出力バス
DOP	出力	4	RDADDR で指定されたデータ パリティ出力バス
DI	入力	32	WRADDR で指定されたデータ入力バス
DIP	入力	4	WRADDR で指定されたデータ パリティ入力バス
WRDDRA、RDDDRB	入力	9	書き込み/読み出しアドレス入力バス
WE	入力	4	ライト イネーブル
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

## 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
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、1 クロック サイクルで読み出しが可能ですが、clock-to-out タイムが長くなります。
INIT	16 進数	36 ビット値	すべて 0	コンフィギュレーション後の出力の初期値を指定します。
SIM_COLLISION_CHECK	文字列	"ALL"、"WARNING_ONLY"、"GENERATE_X_ONLY"、"NONE"	"ALL"	<p>メモリの競合が発生した場合のシミュレーションの動作を指定します。出力は次のようになります。</p> <ul style="list-style-type: none"> <li>・ "ALL"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。</li> <li>・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。</li> </ul> <p><b>注記：</b> "ALL" 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SIM_MODE	文字列	"SAFE"、"FAST"	"SAFE"	シミュレーションのみの属性です。"FAST" に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。
SRVAL	16 進数	36 ビット値	すべて 0	同期リセット信号 (SSR) 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_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
port map (
DO => DO,          -- 32-bit Data Output
DOP => DOP,         -- 4-bit Parity Output
RDCLK => RDCLK,     -- 1-bit read port clock
RDEN => RDEN,       -- 1-bit read port enable
REGCE => REGCE,     -- 1-bit register enable input
SSR => SSR,         -- 1-bit synchronous output set/reset input
WRCLK => WRCLK,     -- 1-bit write port clock
WREN => WREN,       -- 1-bit write port enable
WRADDR => WRADDR,   -- 9-bit write port address input
RDADDR => RDADDR,   -- 9-bit read port address input
DI => DI,           -- 32-bit data input
DIP => DIP,         -- 4-bit parity data input
WE => WE            -- 4-bit write enable input
);

-- End of RAMB18SDP_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// RAMB18SDP: 36x512 Simple Dual-Port BlockRAM
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

RAMB18SDP #(
.SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
.DO_REG(0), // Optional output register (0 or 1)
.INIT(36'h00000000), // Initial values on output port
.SIM_COLLISION_CHECK("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
// "GENERATE_X_ONLY" or "NONE"
.SRVAL(36'h00000000), // Set/Reset value for port output

// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_01(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),

```

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

```
.DO(DO),           // 32-bit data output
.DOP(DOP),         // 4-bit parity data output
.RDCLK(RDCLK),     // 1-bit read port clock
.RDEN(RDEN),       // 1-bit read port enable
.REGCE(REGCE),     // 1-bit register enable input
.SSR(SSR),         // 1-bit synchronous output set/reset input
.WRCLK(WRCLK),     // 1-bit write port clock
.WREN(WREN),       // 1-bit write port enable
.WRADDR(WRADDR),   // 9-bit write port address input
.RDADDR(RDADDR),   // 9-bit read port address input
.DI(DI),           // 32-bit data input
.DIP(DIP),         // 4-bit parity data input
.WE(WE)            // 4-bit write enable input
);

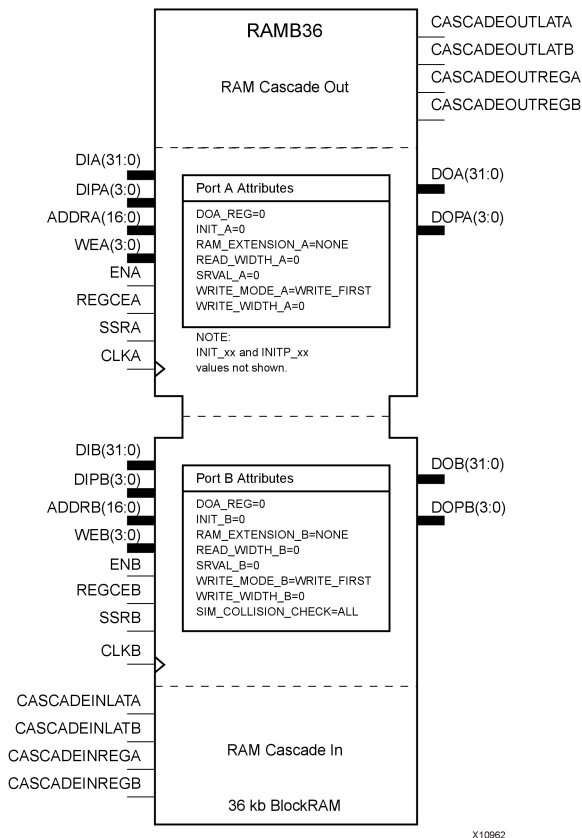
// End of RAMB18SDP_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAMB36

### プリミティブ：36kb Configurable Synchronous True Dual Port Block RAM



X10962

## 概要

このデザイン エLEMENTはブロック RAM メモリの 1 つで、FIFO、自動誤り訂正 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。このデザイン エLEMENT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。このコンポーネントは、1 ビット X 32K ワード～ 36 ビット 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。読み出しと書き込みは、コンポーネントに供給されるクロックに同期して実行されますが、ポート A とポート B 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が、レイテンシは増加します。

このデザイン エLEMENTはシンプル デュアル ポート (72 ビット幅 X 512 ワード、オプションで ECC スクラブ機能あり) に使用してください。このELEMENTは、カスケード接続された RAMB18 を使用して作成できます。

RAMB36 に配置可能なELEMENTの組み合わせを次に示します。

- ・ RAMB18/RAMB18
- ・ RAMB18/FIFO18
- ・ RAMB18SDP/RAMB18SDP
- ・ RAMB18SDP/FIFO18\_36

## ポートの説明

ポート名	方向	幅	機能
DOA	出力	1、2、4、8、16、32	ポート A のデータ出力バス
DOB	出力	1、2、4、8、16、32	ポート B のデータ出力バス
DOPA、DOPB	出力	0、1、2、4	ポート A/B のパリティ データ出力バス
CASCADEOUTLATA、 CASCADEOUTLATB	出力	1	2 つのブロック RAM をカスケード接続して 64K ワード X 1 ビットのメモリを作成するときに使用するポート A とポート B の出力（下位にある RAMB36 の CASCADEINLATA、CASCADEINLATB に接続。64K X 1 RAM を作成しない場合または RAM_EXTENSION_A/B が LOWER に設定されていない場合は未接続）
CASCADEOUTREGA、 CASCADEOUTREGB	出力	1	2 つのブロック RAM をカスケード接続して 64K ワード X 1 ビットのメモリを作成するときに使用するポート A とポート B の出力（下位にある RAMB36 の CASCADEINREGA、CASCADEINREGB に接続。64K X 1 RAM を作成しない場合または RAM_EXTENSION_A/B が LOWER に設定されていない場合は未接続）
CASCADEINLATA、 CASCADEINLATB	入力	1	2 つのブロック RAM をカスケード接続して 64K ワード X 1 ビットのメモリを作成するときに使用するポート A とポート B の入力（上位にある RAMB36 の CASCADEOUTLATA、CASCADEOUTLATB に接続。64K X 1 RAM を作成しない場合または RAM_EXTENSION_A/B が UPPER に設定されていない場合は未接続）
CASCADEINREGA、 CASCADEINREGB	入力	1	2 つのブロック RAM をカスケード接続して 64K ワード X 1 ビットのメモリを作成するときに使用するポート A とポート B の入力（上位にある RAMB36 の CASCADEOUTREGA、CASCADEOUTREGB に接続。64K X 1 RAM を作成しない場合または RAM_EXTENSION_A/B が UPPER に設定されていない場合は未接続）
DIA	入力	1、2、4、8、16、32	ポート A のデータ入力バス
DIB	入力	1、2、4、8、16、32	ポート B のデータ入力バス
DIPA、DIPB	入力	0、1、2、4	ポート A/B のパリティ データ入力バス
ADDRA、ADDRB	入力	16	ポート A/B のアドレス入力バス。CASC モードの場合は 16。
WEA	入力	4	ポート A のバイト幅ライト イネーブル
WEB	入力	4	ポート B のバイト幅ライト イネーブル
ENA、ENB	入力	1	ポート A/B のイネーブル。High になると関連するポートの読み出しまたは書き込みが実行されます。
SSRA、SSRB	入力	1	ポート A/B の出力レジスタの同期セット/リセット。High になると、関連ポートのプリセット/リセットが SRVAL_A/SRVAL_B で指定した値になります。
REGCEA、REGCEB	入力	1	ポート A/B の出力レジスタのクロック イネーブル入力。High になると、関連するポートの出力レジスタが有効になります。

## 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	可
マクロのサポート	可

次の表に、さまざまなポート幅のデータ、アドレスおよびライトイネーブル接続を、ポート A またはポート B の DATA\_WIDTH の値ごとに示します。同じポートの読み出しと書き込みで異なる幅を使用する場合は、アドレス接続を選択できるようにこの表のデータ幅が大きい方の 2 つを使用してください。

特定のコンフィギュレーションで必要にならないデータ ポートおよびアドレス ポートは、次の例外を除き、すべて未接続にするかまたはグランドに接続する必要があります。

- ・ アドレス ビット 15 はカスケード可能なブロック RAM でのみ使用します。カスケードしないブロック RAM の場合は、High に接続します。
- ・ ADDR ピンは 16 ビット幅である必要があります。ただし、カスケード可能でない RAM の有効アドレスはピン 14 (15 - アドレス幅) のみです。残りのピンは、ピン 15 を含め High に接続しておく必要があります。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1 (カスケードあり)	DI[0]	ADDR[15:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
1 (カスケードなし)	DI[0]	ADDR[14:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[14:1]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[14:2]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[14:3]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]
18	DI[15:0]、DIP[1:0]	ADDR[14:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続	DO[15:0]、DOP[1:0]
36	DI[31:0]、DIP[3:0]	ADDR[14:5]	各 WE[3:0] 信号を関連したバイトライト イネーブルに接続	DO[31:0]、DOP[3:0]

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DOA_REG、 DOB_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、1 クロック サイクルで読み出しが可能です。clock-to-out タイムが長くなります。
INIT_A	16 進数	36 ビット値	すべて 0	コンフィギュレーション後のポート A の出力の初期値を指定します。
INIT_B	16 進数	36 ビット値	すべて 0	コンフィギュレーション後のポート B の出力の初期値を指定します。

属性	データ型	値	デフォルト	説明
READ_WIDTH_A	整数	0、1、4、9、18、36	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 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、必要なポート幅に設定してください。
READ_WIDTH_B	整数	0、1、4、9、18、36	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、必要なポート幅に設定してください。
SIM_COLLISION_CHECK	文字列	"ALL"、 "WARNING_ONLY"、 "GENERATE_X_ONLY"、 "NONE"	"ALL"	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> <li>・ "ALL"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。</li> <li>・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。</li> </ul> <p><b>注記：</b>"ALL" 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SIM_MODE	文字列	"SAFE"、"FAST"	"SAFE"	シミュレーションのみの属性です。 "FAST" に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。
SRVAL_A	16 進数	36 ビット値	すべて 0	同期リセット信号 (SSRA) がアサートされたときのポート A の出力値を指定します。
SRVAL_B	16 進数	36 ビット値	すべて 0	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。



属性	データ型	値	デフォルト	説明
WRITE_MODE_A、 WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	書き込みが実行されるときのパートの動作を指定します。  ・ "WRITE_FIRST": 書き込まれた値が出力ポートに出力されます。  ・ "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。  ・ "NO_CHANGE": 出力ポートの以前の値が保持されます。
WRITE_WIDTH_A	整数	0、1、2、4、9、18、 36	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、任意のデータ幅に設定してください。
WRITE_WIDTH_B	整数	0、1、2、4、9、18、 36	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、任意のデータ幅に設定してください。
RAM_EXTENSION_A、 RAM_EXTENSION_B	文字列	"UPPER"、 "LOWER"、 "NONE"	"NONE"	2 つのブロック RAM をカスケード接続して 72K 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。
INIT_00 ~ INIT_7F	16 進数	256 ビット値	すべて 0	72kb のデータ メモリ アレイの初期値を指定します。
INITP_00 ~ INITP_0F	16 進数	256 ビット値	すべて 0	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;

-- RAMB36: 32k+4k Parity Parameterizable True Dual-Port BlockRAM
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

RAMB36_inst : RAMB36
generic map (
  DOA_REG => 0, -- Optional output register on A port (0 or 1)
  DOB_REG => 0, -- Optional output register on B port (0 or 1)
  INIT_A => X"0000000000", -- Initial values on A output port
  INIT_B => X"0000000000", -- Initial values on B output port
  RAM_EXTENSION_A => "NONE", -- "UPPER", "LOWER" or "NONE" when cascaded
  RAM_EXTENSION_B => "NONE", -- "UPPER", "LOWER" or "NONE" when cascaded
  READ_WIDTH_A => 0, -- Valid values are 1, 2, 4, 9, 18, or 36
  READ_WIDTH_B => 0, -- Valid values are 1, 2, 4, 9, 18, or 36
  SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
  -- "GENERATE_X_ONLY" or "NONE"
```

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

```

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 (
  CASCADEOUTLATA => CASCADEOUTLATA, -- 1-bit cascade A latch output
  CASCADEOUTLATB => CASCADEOUTLATB, -- 1-bit cascade B latch output
  CASCADEOUTREGA => CASCADEOUTREGA, -- 1-bit cascade A register output
  CASCADEOUTREGB => CASCADEOUTREGB, -- 1-bit cascade B register output
  DOA => DOA, -- 32-bit A port data output
  DOB => DOB, -- 32-bit B port data output
  DOPA => DOPA, -- 4-bit A port parity data output
  DOPB => DOPB, -- 4-bit B port parity data output
  ADDRA => ADDRA, -- 16-bit A port address input
  ADDRb => ADDRb, -- 16-bit B port address input
  CASCADEINLATA => CASCADEINLATA, -- 1-bit cascade A latch input
  CASCADEINLATB => CASCADEINLATB, -- 1-bit cascade B latch input
  CASCADEINREGA => CASCADEINREGA, -- 1-bit cascade A register input
  CASCADEINREGB => CASCADEINREGB, -- 1-bit cascade B register input
  CLKA => CLKA, -- 1-bit A port clock input
  CLKB => CLKB, -- 1 bit B port clock input
  DIA => DIA, -- 32-bit A port data input
  DIB => DIB, -- 32-bit B port data input
  DIPa => DIPa, -- 4-bit A port parity data input
  DIPB => DIPB, -- 4-bit B port parity data input
  ENA => ENA, -- 1-bit A port enable input
  ENB => ENB, -- 1-bit B port enable input
  REGCEA => REGCEA, -- 1-bit A port register enable input
  REGCEB => REGCEB, -- 1-bit B port register enable input
  SSRA => SSRA, -- 1-bit A port set/reset input
  SSRB => SSRB, -- 1-bit B port set/reset input
  WEA => WEA, -- 4-bit A port write enable input
  WEB => WEB -- 4-bit B port write enable input
);

-- End of RAMB36_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// RAMB36: 32k+4k Parity parameterizable True Dual-Port BlockRAM
// Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

RAMB36 #(
  .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
  .DOA_REG(0), // Optional output registers on A port (0 or 1)
  .DOB_REG(0), // Optional output registers on B port (0 or 1)
  .INIT_A(36'h000000000), // Initial values on A output port
  .INIT_B(36'h000000000), // Initial values on B output port
  .RAM_EXTENSION_A("NONE"), // "UPPER", "LOWER" or "NONE" when cascaded
  .RAM_EXTENSION_B("NONE"), // "UPPER", "LOWER" or "NONE" when cascaded
  .READ_WIDTH_A(0), // Valid values are 1, 2, 4, 9, 18, or 36
  .READ_WIDTH_B(0), // Valid values are 1, 2, 4, 9, 18, or 36
  .SIM_COLLISION_CHECK("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
  // "GENERATE_X_ONLY" or "NONE"
  .SRVAL_A(36'h000000000), // Set/Reset value for A port output
  .SRVAL_B(36'h000000000), // Set/Reset value for B port 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, 2, 4, 9, 18, or 36
  .WRITE_WIDTH_B(0), // Valid values are 1, 2, 4, 9, 18, or 36

  // The following INIT xx declarations specify the initial contents of the RAM
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日

```

.CASCADEOUTREGB(CASCADEOUTREGB), // 1-bit cascade B register output
.DOA(DOA), // 32-bit A port data output
.DOB(DOB), // 32-bit B port data output
.DOPA(DOPA), // 4-bit A port parity data output
.DOPB(DOPB), // 4-bit B port parity data output
.ADDRA(ADDRB), // 16-bit A port address input
.ADDRB(ADDRB), // 16-bit B port address input
.CASCADEINLATA(CASCADEINLATA), // 1-bit cascade A latch input
.CASCADEINLATB(CASCADEINLATB), // 1-bit cascade B latch input
.CASCADEINREGA(CASCADEINREGA), // 1-bit cascade A register input
.CASCADEINREGB(CASCADEINREGB), // 1-bit cascade B register input
.CLKA(CLKA), // 1-bit A port clock input
.CLKB(CLKB), // 1-bit B port clock input
.DIA(DIA), // 32-bit A port data input
.DIB(DIB), // 32-bit B port data input
.DIPA(DIPA), // 4-bit A port parity data input
.DIPB(DIPB), // 4-bit B port parity data input
.ENA(ENA), // 1-bit A port enable input
.ENB(ENB), // 1-bit B port enable input
.REGCEA(REGCEA), // 1-bit A port register enable input
.REGCEB(REGCEB), // 1-bit B port register enable input
.SSRA(SSRA), // 1-bit A port set/reset input
.SSRB(SSRB), // 1-bit B port set/reset input
.WEA(WEA), // 4-bit A port write enable input
.WEB(WEB) // 4-bit B port write enable input
);

// End of RAMB36_inst instantiation

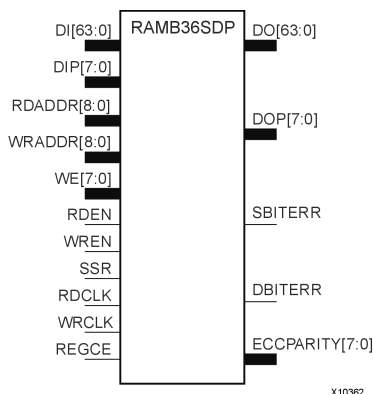
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## RAMB36SDP

**プリミティブ：72-bit by 512 Deep, 36kb Synchronous Simple Dual Port Block RAM with ECC (Error Correction Circuitry)**



### 概要

このデザイン エLEMENTはブロック RAM メモリの 1 つで、FIFO、自動誤り訂正 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。RAMB36SDP 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。このコンポーネントは、72 ビット幅 X 512 ワードのシンプル デュアル ポート RAM に設定されています。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。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。誤り検出と訂正回路をイネーブルにすると、メモリの破損を検出し、訂正することもできます。

**注記：** このELEMENT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	64	RDADDR で指定されたデータ出力バス
DOP	出力	8	RDADDR で指定されたデータ パリティ出力バス
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 ファンクションからのステータス出力。使用する場合は、EN_ECC_READ を TRUE にする必要があります。
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。使用する場合は、EN_ECC_READ を TRUE にする必要があります。
ECCPARITY	出力	8	ECC デコーダーでメモリの誤りを検出および訂正するために使用される、ECC エンコーダーから生成された 8 ビット データ
DI	入力	64	WRADDR で指定されたデータ入力バス
DIP	入力	8	WRADDR で指定されたデータ パリティ入力バス
WRADDR, RDADDR	入力	9	書き込み/読み出しアドレス入力バス



ポート名	方向	幅	機能
WE	入力	8	ライト イネーブル
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

## 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
DO_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロックサイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能です。clock-to-out タイムが長くなります。
INIT	16 進数	72 ビット値	すべて 0	コンフィギュレーション後の出力の初期値を指定します。
EN_ECC_READ	ブール代数	TRUE、FALSE	FALSE	ECC デコーダ回路をイネーブルにします。
EN_ECC_WRITE	ブール代数	TRUE、FALSE	FALSE	ECC エンコーダ回路をイネーブルにします。
EN_ECC_SCRUB	ブール代数	TRUE、FALSE	FALSE	RAM の内容の ECC スクラブ機能をイネーブルにします。
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>

## VHDL 記述 (インスタンス化)

[illegible]

355

```
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",
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 (
  DBITERR => DBITERR, -- 1-bit double bit error status output
  SBITERR => SBITERR, -- 1-bit single bit error status output
  DO => DO,           -- 64-bit Data Output
  DOP => DOP,         -- 8-bit Parity Output
  ECCPARITY => ECCPARITY, -- 8-bit generated error correction parity
  RDCLK => RDCLK,     -- 1-bit read port clock
  RDEN => RDEN,       -- 1-bit read port enable
  REGCE => REGCE,     -- 1-bit register enable input
  SSR => SSR,         -- 1-bit synchronous output set/reset input
  WRCLK => WRCLK,     -- 1-bit write port clock
  WREN => WREN,       -- 1-bit write port enable
  WRADDR => WRADDR,   -- 9-bit write port address input
  RDADDR => RDADDR,   -- 9-bit read port address input
  DI => DI,           -- 64-bit data input
  DIP => DIP,         -- 8-bit parity data input
  WE => WE            -- 8-bit write enable input
);

-- End of RAMB36SDP_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAMB36SDP: 72x512 Simple Dual-Port BlockRAM w/ ECC
//          Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

RAMB36SDP #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .DO_REG(0), // Optional output register (0 or 1)
    .EN_ECC_READ("FALSE"), // Enable ECC decoder, "TRUE" or "FALSE"
    .EN_ECC_WRITE("FALSE"), // Enable ECC encoder, "TRUE" or "FALSE"
    .INIT(72'h00000000000000000000), // Initial values on output port
    .SIM_COLLISION_CHECK("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
    // "GENERATE_X_ONLY" or "NONE"
    .SRVAL(72'h00000000000000000000), // Set/Reset value for port output

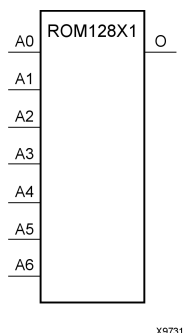
    // The following INIT_xx declarations specify the initial contents of the RAM
    .INIT_00(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_01(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_02(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_03(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_04(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_05(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_06(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_07(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_08(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_09(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_0A(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_0B(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_0C(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_0D(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_0E(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_0F(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_10(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_11(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_12(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_13(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_14(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_15(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_16(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_17(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_18(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_19(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_1A(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_1B(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_1C(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_1D(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_1E(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_1F(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_20(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_21(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_22(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_23(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_24(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_25(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_26(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_27(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_28(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_29(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_2A(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_2B(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_2C(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_2D(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_2E(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_2F(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_30(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_31(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_32(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_33(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_34(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_35(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
    .INIT_36(256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000),
```

Virtex-5 ライブラリ ガイド (HDL 用)  
UG621 (v14.5) 2013 年 3 月 20 日



## ROM128X1

### プリミティブ：128-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 128 ワード X 1 ビットの ROM です。データ出力 (O) には、7 ビットのアドレス (A6 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 32 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)



## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	128 ビット値	すべて 0	ROM の値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

ROM128X1_inst : ROM128X1
generic map (
  INIT => X"00000000000000000000000000000000")
port map (
  O => O,    -- ROM output
  A0 => A0,  -- ROM address[0]
  A1 => A1,  -- ROM address[1]
  A2 => A2,  -- ROM address[2]
  A3 => A3,  -- ROM address[3]
  A4 => A4,  -- ROM address[4]
  A5 => A5,  -- ROM address[5]
  A6 => A6,  -- ROM address[6]
);

-- End of ROM128X1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
//           Virtex-5/6, Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 14.5

ROM128X1 #(
  .INIT(128'h00000000000000000000000000000000) // Contents of ROM
) ROM128X1_inst (
  .O(O),    // ROM output
  .A0(A0), // ROM address[0]
  .A1(A1), // ROM address[1]
  .A2(A2), // ROM address[2]
  .A3(A3), // ROM address[3]
  .A4(A4), // ROM address[4]
  .A5(A5), // ROM address[5]
  .A6(A6)  // ROM address[6]
);

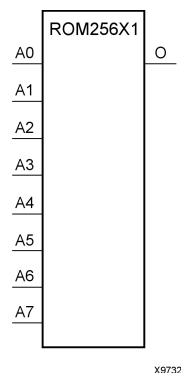
// End of ROM128X1_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## ROM256X1

プリミティブ：256-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 256 ワード X 1 ビットの ROM です。データ出力 (O) には、8 ビットのアドレス (A7 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 64 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。

INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	256 ビット値	すべて 0	ROM の値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

ROM256X1_inst : ROM256X1
generic map (
  INIT => X"0000000000000000000000000000000000000000000000000000000000000000"
)
port map (
  O => O,    -- ROM output
  A0 => A0,   -- ROM address[0]
  A1 => A1,   -- ROM address[1]
  A2 => A2,   -- ROM address[2]
  A3 => A3,   -- ROM address[3]
  A4 => A4,   -- ROM address[4]
  A5 => A5,   -- ROM address[5]
  A6 => A6,   -- ROM address[6]
  A7 => A7,   -- ROM address[7]
);

-- End of ROM256X1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
//           Virtex-5/6, Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 14.5

ROM256X1 #(
  .INIT(256'h0000000000000000000000000000000000000000000000000000000000000000) // Contents of ROM
) ROM256X1_inst (
  .O(O),    // ROM output
  .A0(A0),  // ROM address[0]
  .A1(A1),  // ROM address[1]
  .A2(A2),  // ROM address[2]
  .A3(A3),  // ROM address[3]
  .A4(A4),  // ROM address[4]
  .A5(A5),  // ROM address[5]
  .A6(A6),  // ROM address[6]
  .A7(A7)   // ROM address[7]
);

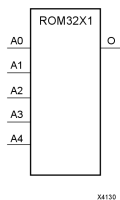
// End of ROM256X1_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## ROM32X1

プリミティブ：32-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 32 ワード X 1 ビットの ROM です。データ出力 (O) には、5 ビットのアドレス (A4 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 8 桁の 16 進数で、ROM には最上位ビット A=1FH から最下位ビット A=00H の順に書き込まれます。

たとえば INIT=10A78F39 と指定すると、「0001 0000 1010 0111 1000 1111 0011」というデータストリームが生成されます。INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## デザインの入力方法

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

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	ROM の値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

ROM32X1_inst : ROM32X1
generic map (
  INIT => X"00000000")
port map (
  O => O,    -- ROM output
  A0 => A0,  -- ROM address[0]
  A1 => A1,  -- ROM address[1]
  A2 => A2,  -- ROM address[2]
  A3 => A3,  -- ROM address[3]
  A4 => A4   -- ROM address[4]
);
-- End of ROM32X1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
//          All FPGAs
// Xilinx HDL Libraries Guide, version 14.5

ROM32X1 #(
  .INIT(32'h00000000) // Contents of ROM
) ROM32X1_inst (
  .O(O),    // ROM output
  .A0(A0),  // ROM address[0]
  .A1(A1),  // ROM address[1]
  .A2(A2),  // ROM address[2]
  .A3(A3),  // ROM address[3]
  .A4(A4)   // ROM address[4]
);

// End of ROM32X1_inst instantiation
```

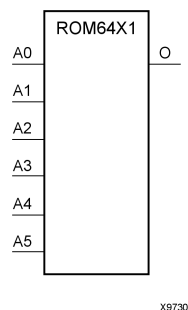
## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)



## ROM64X1

プリミティブ：64-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 64 ワード X 1 ビットの ROM です。データ出力 (O) には、6 ビットのアドレス (A5 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 16 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべて 0	ROM の値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

ROM64X1_inst : ROM64X1
generic map (
  INIT => X"0000000000000000")
port map (
  O => O,    -- ROM output
  A0 => A0,  -- ROM address[0]
  A1 => A1,  -- ROM address[1]
  A2 => A2,  -- ROM address[2]
  A3 => A3,  -- ROM address[3]
  A4 => A4,  -- ROM address[4]
  A5 => A5   -- ROM address[5]
);

-- End of ROM64X1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
//          Virtex-5/6, Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 14.5

ROM64X1 #(
  .INIT(64'h0000000000000000) // Contents of ROM
) ROM64X1_inst (
  .O(O),    // ROM output
  .A0(A0),  // ROM address[0]
  .A1(A1),  // ROM address[1]
  .A2(A2),  // ROM address[2]
  .A3(A3),  // ROM address[3]
  .A4(A4),  // ROM address[4]
  .A5(A5)   // ROM address[5]
);

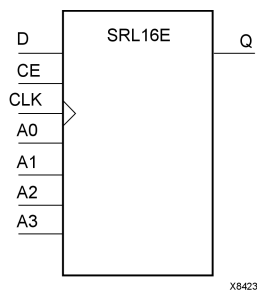
// End of ROM64X1_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## SRL16E

### プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Clock Enable



## 概要

このデザイン エLEMENTは、シフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには：** 入力 A3 ～ A0 の値を一定にします。シフトレジスタは 1 ～ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$  という式で算出できます。A3、A2、A1、A0 がすべて 0 の場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには：** 入力 A3 ～ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ～ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中に 0000 にクリアされます。

CE が High の場合、クロック (CLK) が Low から High に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが Low から High に切り替わるときに CE が High の場合、シフトレジスタの値は次の高位ビットにシフトされ、新しい値がロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。CE が Low の場合、クロック遷移は無視されます。

## 論理表

入力				出力
Am	CE	CLK	D	Q
Am	0	X	X	Q(Am)
Am	1	↑	D	Q(Am - 1)
m = 0、1、2、3				

## ポートの説明

ポート名	方向	幅	機能
Q	出力	1	シフトレジスタ データ出力
D	入力	1	シフトレジスタ データ入力
CLK	入力	1	クロック
CE	入力	1	アクティブ High のクロック イネーブル
A	入力	4	SRL のワード数のダイナミック選択 ・ A=0000 ==> 1 ビットシフト長 ・ A=1111 ==> 16 ビットシフト長

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべて 0	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
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

SRL16E_inst : SRL16E
generic map (
  INIT => X"0000")
port map (
  Q => Q,          -- SRL data output
  A0 => A0,         -- Select[0] input
  A1 => A1,         -- Select[1] input
  A2 => A2,         -- Select[2] input
  A3 => A3,         -- Select[3] input
  CE => CE,         -- Clock enable input
  CLK => CLK,       -- Clock input
  D => D           -- SRL data input
);

-- End of SRL16E_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//      All FPGAs
// Xilinx HDL Libraries Guide, version 14.5

SRL16E #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRL16E_inst (
    .Q(Q),           // SRL data output
    .A0(A0),         // Select[0] input
    .A1(A1),         // Select[1] input
    .A2(A2),         // Select[2] input
    .A3(A3),         // Select[3] input
    .CE(CE),         // Clock enable input
    .CLK(CLK),       // Clock input
    .D(D)            // SRL data input
);

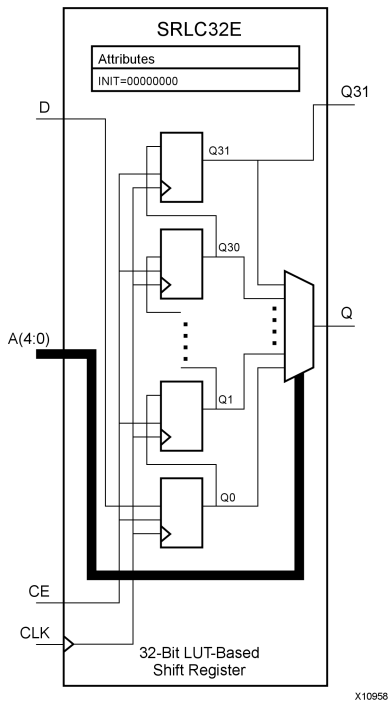
// End of SRL16E_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## SRLC32E

プリミティブ：32 Clock Cycle, Variable Length Shift Register Look-Up Table (LUT) with Clock Enable



### 概要

このデザイン エLEMENTは、1 つのルックアップ テーブル (LUT) にインプリメントされる、可変長で 1 ～ 32 クロック サイクルのシフトレジスタです。シフトレジスタの長さは、固定することも、変動させることもできます。このELEMENTは、アクティブ High のクロック イネーブルおよびカスケード機能も備えているため、複数の SRLC32E をカスケード接続でき、より大型のシフトレジスタを作成できます。

### ポートの説明

ポート名	方向	幅	機能
Q	出力	1	シフトレジスタ データ出力
Q31	出力	1	シフトレジスタ カスケード出力 (後続 SRLC32E の D 入力に接続)
D	入力	1	シフトレジスタ データ入力
CLK	入力	1	クロック
CE	入力	1	アクティブ High のクロック イネーブル
A	入力	5	SRL の長さのダイナミック選択 A=00000 ==> 1 ビット A=11111 ==> 32 ビット

## デザインの入力方法

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

インスタンス化する場合、このコンポーネントを次のように接続します。

- ・ CLK 入力を適切なクロック ソースに、D 入力をシフト/格納するデータ ソースに、Q 出力を FDCPE 入力または FDRSE 入力などの適切なデスティネーションに接続します。
- ・ クロック イネーブル ピン (CE) はクロック イネーブル信号に接続するか、使用しない場合は論理 1 にします。
- ・ 5 ビット バス A は、一定の値 (0 ~ 31) にしてシフトレジスタの長さを 1 ~ 32 ビットに固定するか、または適切な論理値にしてシフトレジスタの長さを 1 ~ 32 ビットの範囲で変更できます。
- ・ シフトレジスタの長さを 32 ビットより長くする場合は、Q31 出力ピンを後続の SRLC32E の D 入力に接続します。
- ・ Q31 出力は、別の SRLC32E 以外には接続できません。
- ・ Q 出力は、カスケード モードでも使用できます。
- ・ 32 ビットの 16 進数の INIT 属性で、シフトレジスタの初期シフトパターンを指定できます。
- ・ シフトアウトされる最初の値は INIT[0] です。

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべて 0	SRLC32E の初期シフトパターンを指定します。

## VHDL 記述 (インスタンス化)

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

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

-- SRLC32E: 32-bit variable length shift register LUT
--           with clock enable
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

SRLC32E_inst : SRLC32E
generic map (
  INIT => X"00000000")
port map (
  Q => Q,           -- SRL data output
  Q31 => Q31,       -- SRL cascade output pin
  A => A,           -- 5-bit shift depth select input
  CE => CE,         -- Clock enable input
  CLK => CLK,       -- Clock input
  D => D,           -- SRL data input
);

-- End of SRLC32E_inst instantiation
```



## Verilog 記述 (インスタンス化)

```
// SRLC32E: 32-bit variable length cascadable shift register LUT
//           with clock enable
//           Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

SRLC32E #(
    .INIT(32'h00000000) // Initial Value of Shift Register
) SRLC32E_inst (
    .Q(Q),           // SRL data output
    .Q31(Q31),       // SRL cascade output pin
    .A(A),           // 5-bit shift depth select input
    .CE(CE),         // Clock enable input
    .CLK(CLK),       // Clock input
    .D(D)            // SRL data input
);

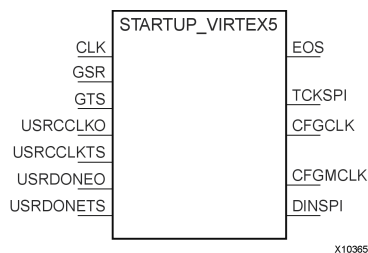
// End of SRLC32E_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## STARTUP\_VIRTEX5

### プリミティブ : Virtex®-5 Configuration Start-Up Sequence Interface



### 概要

このデザイン エLEMENTは、グローバル非同期セット/リセット (GSR) 信号、グローバルトライステート (GTS) 専用配線、内部コンフィギュレーション信号、SPI PROM が使用される場合は SPI PROM の入力ピンなどへのロジックとデバイス ピンの接続に使用されます。デバイスのコンフィギュレーションの最後にスタートアップ シーケンスで別のクロックを使用するよう指定したり、コンフィギュレーション クロックにアクセスするためにも使用できます。

### ポートの説明

ポート名	方向	幅	機能
EOS	出力	1	スタートアップの完了を示すアクティブ High の信号
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
USRCCLKO	入力	1	内部ユーザー CCLK。USRCCLKTS が Low の場合、この入力に供給される信号が CCLK ピンに現れます。USRCCLKTS が High から Low に遷移した場合、CCLK がアクティブドライバーになる前に USRCCLKO が Low から High に遷移する必要があります。
USRCCLKTS	入力	1	ユーザー CCLK トライステート イネーブル。この入力が High の場合、CCLK ピンがトライステート状態になります。Low の場合、CCLK ピンは USRCCLKO 入力を反映します。USRCCLKTS が High から Low に遷移した場合、CCLK がアクティブドライバーになる前に USRCCLKO が Low から High に遷移する必要があります。
USRDONEO	入力	1	内部ユーザー DONE ピンの出力を制御
USRDONETS	入力	1	DONE ピンに対するユーザー DONE トライステート イネーブル。この入力が High の場合、DONE はトライステート状態になります。
TCKSPI	出力	1	TCK コンフィギュレーション ピンへの直接アクセス
DINSPI	出力	1	DIN コンフィギュレーション ピンへの直接アクセス
GSR	入力	1	アクティブ High のグローバル トライステート (GTS) 信号。この入力がアサートされると、すべてのフリップフロップがビットストリームで指定された初期値にリセットされます。

ポート名	方向	幅	機能
GTS	入力	1	アクティブ High のグローバルトライステート (GTS) 信号。この入力のアサートされると、すべてのユーザー I/O がトライステート状態になります。
CLK	入力	1	ユーザー スタートアップ クロック

## デザインの入力方法

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

専用のグローバルトライステートが使用される場合は、適切なソース ピンまたはロジックをこのプリミティブの GTS 入力ピンに接続します。コンフィギュレーションのスタートアップ シーケンスのクロックを指定するには、デザインからのクロックをこのデザイン エLEMENTの CLK ピンに接続します。CFGMCLK および CFGCLK を使用すると、内部コンフィギュレーション クロックにアクセスでき、EOS 信号はコンフィギュレーション スタートアップ シーケンスの終了を示します。

SPI PROM を使用してデバイスをコンフィギュレーションしており、コンフィギュレーション後に SPI PROM へのアクセスが必要な場合は、USRCCLKO および USRCCLKTS を使用して PROM にクロックを供給し、DINSPI ピンを使用して PROM からのデータを読み出します。PROM への SPI インターフェイスには、PROM をイネーブルにして PROM にデータを送信するため FCS\_B および MOSI ピンが必要ですが、これらは多目的ピンであり、通常の I/O としてアクセスされます。

## 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;

-- STARTUP_VIRTEX5: Startup primitive for GSR, GTS or startup sequence control,
--                  SPI PROM pins, configuration clock and start-up status
--                  Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

STARTUP_VIRTEX5_inst : STARTUP_VIRTEX5
port map (
    CFGCLK => CFGCLK, -- Config logic clock 1-bit output
    CFGMCLK => CFGMCLK, -- Config internal osc clock 1-bit output
    DINSPI => DINSPI, -- DIN SPI PROM access 1-bit output
    EOS => EOS, -- End of Startup 1-bit output
    TCKSPI => TCKSPI, -- TCK SPI PROM access 1-bit output
    CLK => CLK, -- Clock input for start-up sequence
    GSR => GSR_PORT, -- Global Set/Reset input (GSR cannot be used for the port name)
    GTS => GTS_PORT, -- Global 3-state input (GTS cannot be used for the port name)
    USRCCLKO => USRCCLKO, -- User CCLK 1-bit input
    USRCCLKTS => USRCCLKTS, -- User CCLK 3-state, 1-bit input
    USRDONEO => USRDONEO, -- User Done 1-bit input
    USRDONETS => USRDONETS -- User Done 3-state, 1-bit input
);

-- End of STARTUP_VIRTEX5_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// STARTUP_VIRTEX5: Startup primitive for accessing GSR, GTS, startup sequence
//               control, SPI PROM pins, configuration clock and start-up status
//               Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

STARTUP_VIRTEX5 STARTUP_VIRTEX5_inst (
    .CFGCLK(CFGCLK),      // Config logic clock 1-bit output
    .CFGMCLK(CFGMCLK),    // Config internal osc clock 1-bit output
    .DINSPI(DINSPI),      // DIN SPI PROM access 1-bit output
    .EOS(EOS),            // End Of Startup 1-bit output
    .TCKSPI(TCKSPI),      // TCK SPI PROM access 1-bit output
    .CLK(CLK),            // Clock input for start-up sequence
    .GSR(GSR_PORT),       // Global Set/Reset input (GSR cannot be used as a port name)
    .GTS(GTS_PORT),       // Global 3-state input (GTS cannot be used as a port name)
    .USRCCLKO(USRCCLKO),  // User CCLK 1-bit input
    .USRCCLKTS(USRCCLKTS), // User CCLK 3-state 1-bit input
    .USRDONEO(USRDONEO),  // User Done 1-bit input
    .USRDONETS(USRDONETS) // User Done 3-state 1-bit input
);

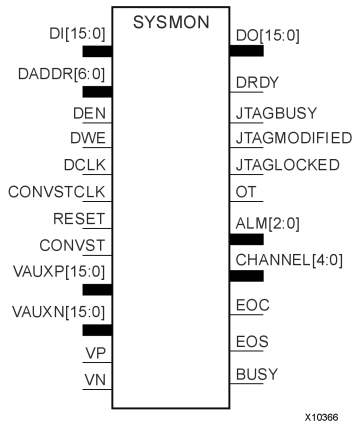
// End of STARTUP_VIRTEX5_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA コンフィギュレーション ユーザー ガイド』\(UG191\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)
- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)

## SYSMON

### プリミティブ：System Monitor



### 概要

このデザイン エLEMENTは、10 ビット、200kSPS (キロサンプル/秒) の Analog-to-Digital Converter (ADC) をベースに構築されています。ADC は、大量のオンチップ センサーと組み合わせて、オンチップ電源電圧およびチップ温度などの FPGA の物理的な動作パラメーターを計測するのに使用されます。外部電圧には、専用のアナログ入力ペア (VP/VN) と 16 のユーザーが選択可能なアナログ入力 (補助アナログ入力 (VAUXP[15:0]、VAUXN[15:0])) を介します。外部アナログ入力を使用すると、ADC でボードやエンクロージャの物理的環境を監視できます。

### ポートの説明

ポート名	方向	幅	機能
ALM[2:0]	出力	3	度、Vccint、および Vccaux 用の警告出力
BUSY	出力	1	ADC ビジー信号
CHANNEL[4:0]	出力	5	チャネル選択
CONVST	入力	1	変換開始
CONVSTCLK	入力	1	変換開始クロック
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用のクロック
DEN	入力	1	ダイナミック リコンフィギュレーションのイネーブル
DI[15:0]	入力	16	ダイナミック リコンフィギュレーション用の入力データ 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用の出力データ バス
DRDY	出力	1	ダイナミック リコンフィギュレーション用のデータ 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用のライト イネーブル
EOC	出力	1	変換の終了を示す出力信号
EOS	出力	1	シーケンスの終了を示す出力信号

ポート名	方向	幅	機能
JTAGBUSY	出力	1	JTAG DRP ビジー信号
JTAGLOCKED	出力	1	DRP ポート ロック
JTAGMODIFIED	出力	1	DRP への JTAG 書き込み
OT	出力	1	温度範囲を超えたことを示す警告出力
RESET	入力	1	リセット信号 (アクティブ High)
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	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

必要な入力および出力ポートをすべて接続し、このコンポーネントに必要な動作を達成する属性を設定します。シミュレーションでは、アナログおよび温度情報をモデルに渡すためにテキストファイルを使用します。テキストファイルのフォーマットは次のとおりです。

```
// Must use valid headers on all columns
// Comments can be added to the stimulus file using '//'
TIME TEMP VCCAUX VCCINT VP VN VAUXP[0] VAUXN[0]
00000 45 2.5 1.0 0.5 0.0 0.7 0.0
05000 85 2.45 1.1 0.3 0.0 0.2 0.0
// Time stamp data is in nano seconds (ns)
// Temperature is recorded in C (degrees centigrade)
// All other channels are recorded as V (Volts)
// Valid column headers are:
// TIME, TEMP, VCCAUX, VCCINT, VP, VN,
// VAUXP[0], VAUXN[0],.....VAUXP[15], VAUXN[15]
// External analog inputs are differential so VP = 0.5 and VN = 0.0 the
// input on channel VP/VN is 0.5 - 0.0 = 0.5V
```

**注記：** このコードをコンパイルする場合は、テキストに余分なスペースを追加しないでください。コンパイル エラーが発生する場合があります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_40	16 進数	16'h0000 ~ 16'hffff	16'h0000	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	16'h0000	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	16'h0000	テスト レジスタ 0

属性	データ型	値	デフォルト	説明
INIT_44	16 進数	16'h0000 ~ 16'hffff	16'h0000	テスト レジスタ 1
INIT_45	16 進数	16'h0000 ~ 16'hffff	16'h0000	テスト レジスタ 2
INIT_46	16 進数	16'h0000 ~ 16'hffff	16'h0000	テスト レジスタ 3
INIT_47	16 進数	16'h0000 ~ 16'hffff	16'h0000	テスト レジスタ 4
INIT_48	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 0
INIT_49	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 1
INIT_4A	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 2
INIT_4B	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 3
INIT_4C	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 4
INIT_4D	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 5
INIT_4E	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 6
INIT_4F	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 7
INIT_50	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 0
INIT_51	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 1
INIT_52	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 2
INIT_53	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 3
INIT_54	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 4
INIT_55	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 5
INIT_56	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 6
INIT_57	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 7
SIM_DEVICE	文字列	"VIRTEX5"、 "VIRTEX6"	"VIRTEX5"	シミュレーション用のターゲット デバイス ファミ リを指定します。
SIM_MONITOR_FILE	文字列	ファイルの名前 と場所	design.txt	シミュレーション アナログ入力ファイルを指定 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;

-- SYSMON: System Monitor
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

SYSMON_inst : SYSMON
generic map (
    INIT_40 => X"0000", -- Configuration register 0
    INIT_41 => X"0000", -- Configuration register 1
    INIT_42 => X"0000", -- Configuration register 2
    INIT_43 => X"0000", -- Test register 0
    INIT_44 => X"0000", -- Test register 1
    INIT_45 => X"0000", -- Test register 2
    INIT_46 => X"0000", -- Test register 3
    INIT_47 => X"0000", -- Test register 4
    INIT_48 => X"0000", -- Sequence register 0
    INIT_49 => X"0000", -- Sequence register 1
    INIT_4A => X"0000", -- Sequence register 2
    INIT_4B => X"0000", -- Sequence register 3
    INIT_4C => X"0000", -- Sequence register 4
    INIT_4D => X"0000", -- Sequence register 5
    INIT_4E => X"0000", -- Sequence register 6
    INIT_4F => X"0000", -- Sequence register 7
    INIT_50 => X"0000", -- Alarm limit register 0
    INIT_51 => X"0000", -- Alarm limit register 1
    INIT_52 => X"0000", -- Alarm limit register 2
    INIT_53 => X"0000", -- Alarm limit register 3
    INIT_54 => X"0000", -- Alarm limit register 4
    INIT_55 => X"0000", -- Alarm limit register 5
    INIT_56 => X"0000", -- Alarm limit register 6
    INIT_57 => X"0000", -- Alarm limit register 7
    SIM_MONITOR_FILE => "design.txt") -- Simulation analog entry file
port map (
    ALM => ALM,           -- 3-bit output for temp, Vccint and Vccaux
    BUSY => BUSY,         -- 1-bit output ADC busy signal
    CHANNEL => CHANNEL,   -- 5-bit output channel selection
    DO => DO,             -- 16-bit output data bus for dynamic reconfig
    DRDY => DRDY,         -- 1-bit output data ready for dynamic reconfig
    EOC => EOC,           -- 1-bit output end of conversion
    EOS => EOS,           -- 1-bit output end of sequence
    JTAGBUSY => JTAGBUSY, -- 1-bit output JTAG DRP busy
    JTAGLOCKED => JTAGLOCKED, -- 1-bit output DRP port lock
    JTAGMODIFIED => JTAGMODIFIED, -- 1-bit output JTAG write to DRP
    OT => OT,             -- 1-bit output over temperature alarm
    CONVST => CONVST,     -- 1-bit input convert start
    CONVSTCLK => CONVSTCLK, -- 1-bit input convert start clock
    DADDR => DADDR,       -- 7-bit input address bus for dynamic reconfig
    DCLK => DCLK,         -- 1-bit input clock for dynamic reconfig
    DEN => DEN,           -- 1-bit input enable for dynamic reconfig
    DI => DI,             -- 16-bit input data bus for dynamic reconfig
    DWE => DWE,           -- 1-bit input write enable for dynamic reconfig
    RESET => RESET,       -- 1-bit input active high reset
    VAUXN => VAUXN,       -- 16-bit input N-side auxiliary analog input
    VAUXP => VAUXP,       -- 16-bit input P-side auxiliary analog input
    VN => VN,             -- 1-bit input N-side analog input
    VP => VP)             -- 1-bit input P-side analog input
);

-- End of SYSMON_inst instantiation
```



## Verilog 記述 (インスタンス化)

```
// SYSMON: System Monitor
//      Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

SYSMON #(
    .INIT_40(16'h0), // Configuration register 0
    .INIT_41(16'h0), // Configuration register 1
    .INIT_42(16'h0), // Configuration register 2
    .INIT_43(16'h0), // Test register 0
    .INIT_44(16'h0), // Test register 1
    .INIT_45(16'h0), // Test register 2
    .INIT_46(16'h0), // Test register 3
    .INIT_47(16'h0), // Test register 4
    .INIT_48(16'h0), // Sequence register 0
    .INIT_49(16'h0), // Sequence register 1
    .INIT_4A(16'h0), // Sequence register 2
    .INIT_4B(16'h0), // Sequence register 3
    .INIT_4C(16'h0), // Sequence register 4
    .INIT_4D(16'h0), // Sequence register 5
    .INIT_4E(16'h0), // Sequence register 6
    .INIT_4F(16'h0), // Sequence register 7
    .INIT_50(16'h0), // Alarm limit register 0
    .INIT_51(16'h0), // Alarm limit register 1
    .INIT_52(16'h0), // Alarm limit register 2
    .INIT_53(16'h0), // Alarm limit register 3
    .INIT_54(16'h0), // Alarm limit register 4
    .INIT_55(16'h0), // Alarm limit register 5
    .INIT_56(16'h0), // Alarm limit register 6
    .INIT_57(16'h0), // Alarm limit register 7
    .SIM_MONITOR_FILE("design.txt") // Simulation analog entry file
) SYSMON_inst (
    .ALM(ALM), // 3-bit output for temp, Vccint and Vccaux
    .BUSY(BUSY), // 1-bit output ADC busy signal
    .CHANNEL(CHANNEL), // 5-bit output channel selection
    .DO(DO), // 16-bit output data bus for dynamic reconfig
    .DRDY(DRDY), // 1-bit output data ready for dynamic reconfig
    .EOC(EOC), // 1-bit output end of conversion
    .EOS(EOS), // 1-bit output end of sequence
    .JTAGBUSY(JTAGBUSY), // 1-bit output JTAG DRP busy
    .JTAGLOCKED(JTAGLOCKED), // 1-bit output DRP port lock
    .JTAGMODIFIED(JTAGMODIFIED), // 1-bit output JTAG write to DRP
    .OT(OT), // 1-bit output over temperature alarm
    .CONVST(CONVST), // 1-bit input convert start
    .CONVSTCLK(CONVSTCLK), // 1-bit input convert start clock
    .DADDR(DADDR), // 7-bit input address bus for dynamic reconfig
    .DCLK(DCLK), // 1-bit input clock for dynamic reconfig
    .DEN(DEN), // 1-bit input enable for dynamic reconfig
    .DI(DI), // 16-bit input data bus for dynamic reconfig
    .DWE(DWE), // 1-bit input write enable for dynamic reconfig
    .RESET(RESET), // 1-bit input active high reset
    .VAUXN(VAUXN), // 16-bit input N-side auxiliary analog input
    .VAUXP(VAUXP), // 16-bit input P-side auxiliary analog input
    .VN(VN), // 1-bit input N-side analog input
    .VP(VP) // 1-bit input P-side analog input
);

// End of SYSMON_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)

## TEMAC

プリミティブ：Tri-mode Ethernet Media Access Controller (MAC)

### 概要

このデザイン エLEMENTには、イーサネット システムの一般的な接続要件を満たすため、別々にコンフィギュレーション可能なエンベデッド イーサネット MAC が 1 ペア含まれます。

### デザインの入力方法

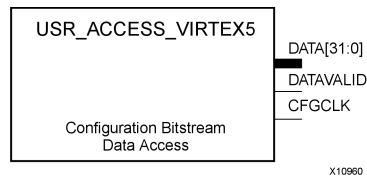
インスタンス化	不可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

### 詳細情報

- ・ [『Virtex-5 エンベデッドトライモード イーサネット MAC ユーザー ガイド』\(UG194\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)
- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)

## USR\_ACCESS\_VIRTEX5

### プリミティブ：Virtex-5 User Access Register



### 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、ビットストリームからのデータを読み出すことができます。たとえば、コンフィギュレーション後に FPGA デザインからビットストリーム格納ソースに保存されたデータにアクセスすることができます。

### ポートの説明

ポート名	方向	幅	機能
DATA	出力	32	コンフィギュレーション出力データ
DATAVALID	出力	1	DATA ポートに有効なデータが含まれているかどうかを示すアクティブ High の信号
CFGCLK	出力	1	コンフィギュレーション クロック

### 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;

-- USR_ACCESS_VIRTEX5: Configuration Data Memory Access Port
--                      Virtex-5
-- Xilinx HDL Libraries Guide, version 14.5

USR_ACCESS_VIRTEX5_inst : USR_ACCESS_VIRTEX5
port map (
    CFGCLK => CFGCLK, -- 1-bit configuration clock output
    DATA => DATA,    -- 32-bit config data output
    DATAVALID => DATAVALID -- 1-bit data valid output
);

-- End of USR_ACCESS_VIRTEX5_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// USR_ACCESS_VIRTEX5: Configuration Data Memory Access Port
//                               Virtex-5
// Xilinx HDL Libraries Guide, version 14.5

USR_ACCESS_VIRTEX5 USR_ACCESS_VIRTEX5_inst (
    .CFGCLK(CFGCLK),           // 1-bit configuration clock output
    .DATA(DATA),               // 32-bit config data output
    .DATAVALID(DATAVALID)     // 1-bit data valid output
);

// End of USR_ACCESS_VIRTEX5_inst instantiation
```

## 詳細情報

- ・ [『Virtex-5 FPGA ユーザー ガイド』\(UG190\)](#)
- ・ [『Virtex-5 FPGA データシート：DC 特性およびスイッチ特性』\(DS202\)](#)