

# Virtex-6 ライブラリ ガイド (HDL 用)

UG623 (v13.3) 2011 年 10 月 26 日



Xilinx is disclosing this user guide, manual, release note, and/or specification (the “Documentation”) to you solely for use in the development of designs to operate with Xilinx hardware devices. You may not reproduce, distribute, republish, download, display, post, or transmit the Documentation in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU “AS-IS” WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© Copyright 2002–2011 Xilinx Inc. All Rights Reserved. XILINX, the Xilinx logo, the Brand Window and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

本資料は英語版 (v.13.3) を翻訳したもので、内容に相違が生じる場合には原文を優先します。  
資料によっては英語版の更新に対応していないものがあります。  
日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

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

## 概要

---

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

このマニュアルには、次の内容が含まれます。

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

## デザイン エLEMENT

このバージョンのライブラリ ガイドには、このアーキテクチャのデザイン エLEMENTの説明とそのインスタンス化コード例が含まれます。また、インスタンス化 テンプレートは、ISE/doc/usenglish/isehelp のインストール ディレクトリにも個別 ZIP ファイルとして含まれています。

デザイン エLEMENTは、次の 3 つのカテゴリに分類されます。

- ・ **マクロ**：これらのELEMENT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ようなプリミティブをインスタンス化の際に使用します。合成ツールでは、この UniMacro が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**：ターゲットにしている FPGA デバイス用のザイリンクス コンポーネントです。プリミティブをインスタンス化して変換 (NGDBuild) プロセスを実行すると、変換後のファイルに含まれるのはまったく同じコンポーネントです。たとえば、ISERDES\_NODELAY という Virtex®-5 ELEMENTをユーザー プリミティブとしてインスタンス化し、変換 (NGDBuild) を実行すると、ISERDES\_NODELAY がそのまま残ります。一方 Virtex-5 デバイスで ISERDES を使用していると、自動的に Virtex-5 用の ISERDES\_NODELAY に変換されます。このため、「プリミティブ」の概念は、同じ分野でもユーザー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 ELEMENT (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 ソフトウェアのリリースごとに、新しいデザイン ELEMENTが組み込まれます。すべてのデザイン ELEMENT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。次にこの 4 つの使用方法を示します。

- ・ **インスタンス化**：デザインにコンポーネントが直接インスタンス化されます。これは、各ブロックをユーザーが正確に配置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ポータビリティに優れているので、複数のアーキテクチャに使用できます。推論を実行すると、パフォーマンス、エリア、電力などをユーザーが合成ツールで指定したとおりに最適化できます。
- ・ **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。推論ができない FPGA を使用して大型なブロックを構築する場合には、この方法を使用してください。このフローを使用する場合は、ターゲットにするアーキテクチャごと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ようなプリミティブをインスタンス化する際に使用します。合成ツールでは、この UniMacro が自動的に下位プリミティブに展開されます。

## UniMacro について

---

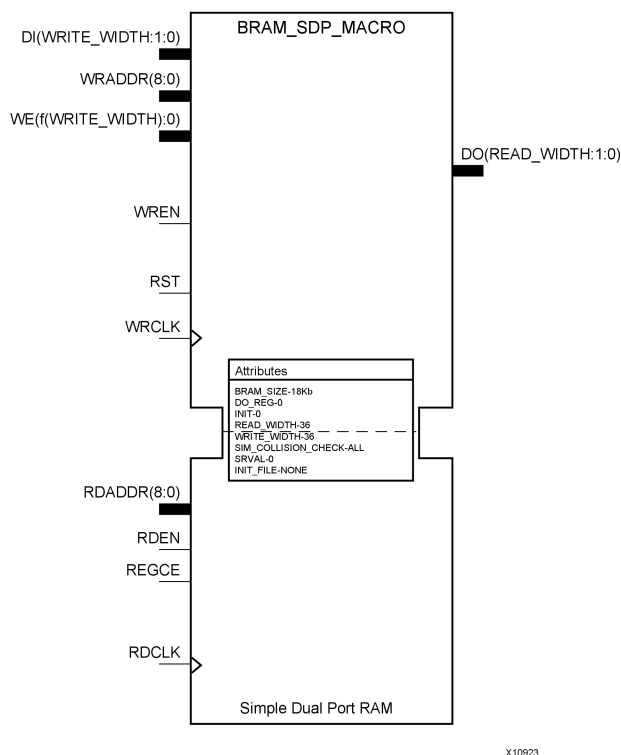
このセクションでは、このアーキテクチャで利用できる UniMacro について説明します。UniMacro は、アルファベット順に並べられています。

各 UniMacro について、次の情報が提供されています。

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

## BRAM\_SDP\_MACRO

### マクロ : Simple Dual Port RAM



X10923

## 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。

**メモ：** 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
WREN、RDEN	入力	1	ライト/リード イネーブル

ポート名	方向	幅	機能
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 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。
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。

属性	データ型	値	デフォルト	説明
INIT	16 進数	72 ビット値	すべてゼロ	コンフィギュレーション後の出力の初期値を指定します。
READ_WIDTH、 WRITE_WIDTH	整数	1 ~ 72	36	DI/DO バスの幅を指定。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
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>メモ:</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	72 ビット値	すべてゼロ	同期リセット信号 (RST) がアサートされたときの DO ポートの出力値を指定します。
INIT_00 ~ INIT_7F	16 進数	256 ビット値	すべてゼロ	16Kb または 32Kb のデータ メモリアレイの初期値を指定します。
INITP_00 ~ INITP_0F	16 進数	256 ビット値	すべてゼロ	2Kb または 4Kb のパリティ データ メモリアレイの初期値を指定します。

## VHDL 記述 (インスタンスエーション)

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



READ Width	BRAM_SIZE	READ Depth	RDADDR Width	
WRITE Width		WRITE Depth	WRADDR Width	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	14-bit	1-bit
2	"18Kb"	8192	13-bit	1-bit
1	"36Kb"	32768	15-bit	1-bit
1	"18Kb"	16384	14-bit	1-bit

[illegible]

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

```

INIT_65 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_66 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_67 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_68 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_69 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_70 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_71 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_72 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_73 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_74 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_75 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_76 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_77 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_78 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_79 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7F => X"0000000000000000000000000000000000000000000000000000000000000000",

-- The next set of INITP_xx are for the parity bits
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",

-- The next set of INIT_xx are valid when configured as 36Kb
INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DO => DO,          -- Output read data port, width defined by READ_WIDTH parameter
DI => DI,          -- Input write data port, width defined by WRITE_WIDTH parameter
RDADDR => RDADDR,  -- Input read address, width defined by read port depth
RDCLK => RDCLK,    -- 1-bit input read clock
RDEN => RDEN,      -- 1-bit input read port enable
REGCE => REGCE,    -- 1-bit input read output register enable
RST => RST,        -- 1-bit input reset
WE => WE,          -- Input write enable, width defined by write port depth
WRADDR => WRADDR,  -- Input write address, width defined by write port depth
WRCLK => WRCLK,    -- 1-bit input write clock
WREN => WREN       -- 1-bit input write port enable
);
-- End of BRAM_SDP_MACRO_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// BRAM_SDP_MACRO: Simple Dual Port RAM
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

////////////////////////////////////
// READ_WIDTH | BRAM_SIZE | READ Depth | RDADDR Width |          //
// WRITE_WIDTH |           | WRITE Depth | WRADDR Width | WE Width //

```

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

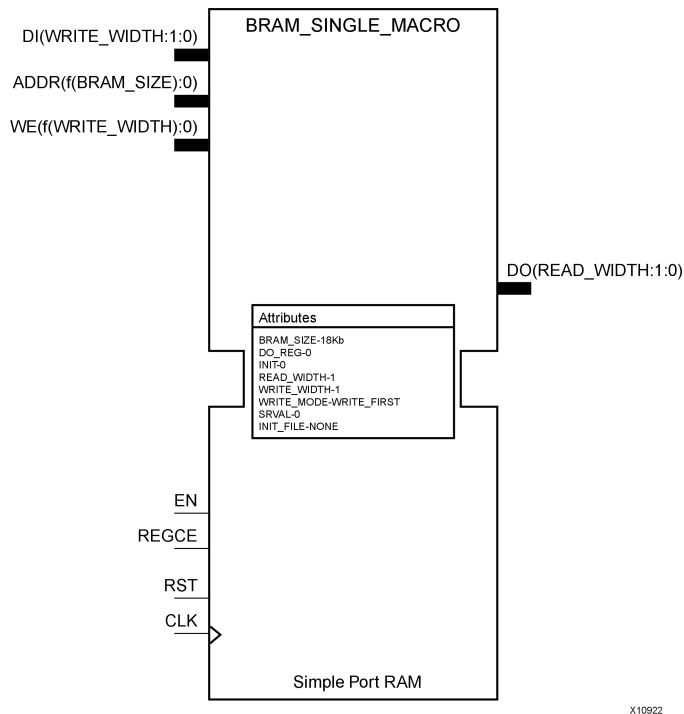
Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

## 詳細情報

14

## BRAM\_SINGLE\_MACRO

### マクロ : Single Port RAM



X10922

## 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)

ポート名	方向	幅	機能
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	



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

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

## デザインの入力方法

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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
BRAM_SIZE	文字列	"36Kb"、"18Kb"	"18Kb"	RAM 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。
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	整数	1 ~ 36	1	出力バスの幅を指定
WRITE_WIDTH	整数	1 ~ 36	1	入力バスの幅を指定
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
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
INIT	16 進数	72 ビット値	すべてゼロ	コンフィギュレーション後の出力の初期値を指定
SRVAL	16 進数	72 ビット値	すべてゼロ	同期リセット信号 (RST) がアサートされたときの DO ポートの出力値を指定します。
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	すべてゼロ	16Kb または 32Kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_0F	16 進数	256 ビット値	すべてゼロ	2Kb または 4Kb のパリティ データ メモリ アレイの初期値を指定

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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

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

```
BRAM_SINGLE_MACRO_inst : BRAM_SINGLE_MACRO
generic map (
    BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
    DEVICE => "VIRTEX6", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    DQ_REG => 0, -- Optional output register (0 or 1)
    INIT => X"00000000", -- Initial values on output port
    INIT_FILE => "NONE",
    WRITE_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
    READ_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
    SRVAL => X"00000000", -- Set/Reset value for port output
    WRITE_MODE => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
    -- The following INIT_xx declarations specify the initial contents of the RAM
    INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
    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",
```

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

[illegible]

## Verilog 記述 (インスタンス化)

```
// BRAM_SINGLE_MACRO: Single Port RAM
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3
```

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

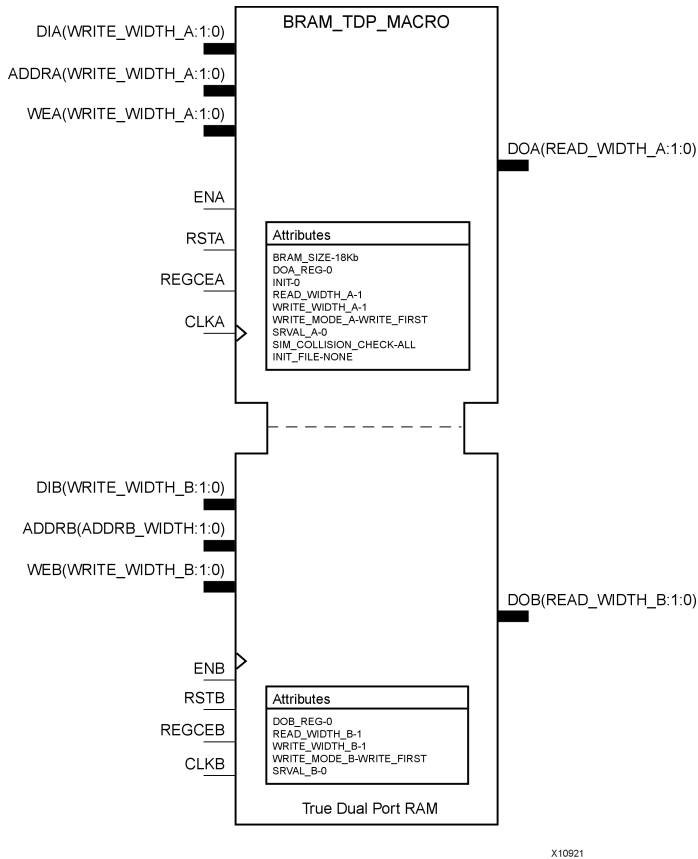
## 詳細情報

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日



## 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 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。
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	すべてゼロ	コンフィギュレーション後の出力の初期値を指定します。
INIT_FILE	文字列	ファイルの名前と場所	なし	初期値を含むファイルの名前
READ_WIDTH、WRITE_WIDTH	整数	1 ～ 72	36	DI/DO バスの幅を指定。READ_WIDTH と WRITE_WIDTH に同じ値を指定する必要があります。
SIM_COLLISION_CHECK	文字列	"ALL"、 "WARNING_ONLY"、 "GENERATE_X_ONLY"、 "NONE"	"ALL"	<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>メモ :</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>

属性	データ型	値	デフォルト	説明
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	すべてゼロ	同期リセット信号 (RST) がアサートされたときの DO ポートの出力値を指定します。
INIT_00 ～ INIT_FF	16 進数	256 ビット値	すべてゼロ	16Kb または 32Kb のデータ メモリ アレイの初期値を指定
INITP_00 ～ INITP_0F	16 進数	256 ビット値	すべてゼロ	2Kb または 4Kb のパリティ データ メモリ アレイの初期値を指定

## 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_TDP_MACRO: True Dual Port RAM
--               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

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

[illegible]

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

```

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

-- End of BRAM_TDP_MACRO_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```

// BRAM_TDP_MACRO: True Dual Port RAM
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

BRAM_TDP_MACRO #(
  .BRAM_SIZE("18Kb"), // Target BRAM: "18Kb" or "36Kb"
  .DEVICE("VIRTEX6"), // Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  .DOA_REG(0),        // Optional port A output register (0 or 1)
  .DOB_REG(0),        // Optional port B output register (0 or 1)
  .INIT_A(36'h00000000), // Initial values on port A output port
  .INIT_B(36'h00000000), // Initial values on port B output port
  .INIT_FILE("NONE"),
  .READ_WIDTH_A(0),    // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  .READ_WIDTH_B(0),    // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  .SIM_COLLISION_CHECK("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
                                // "GENERATE_X_ONLY" or "NONE"
  .SRVAL_A(36'h00000000), // Set/Reset value for port A output
  .SRVAL_B(36'h00000000), // Set/Reset value for port B output
  .WRITE_MODE_A("WRITE_FIRST"), // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
  .WRITE_MODE_B("WRITE_FIRST"), // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
  .WRITE_WIDTH_A(0), // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  .WRITE_WIDTH_B(0), // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),

```



Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

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

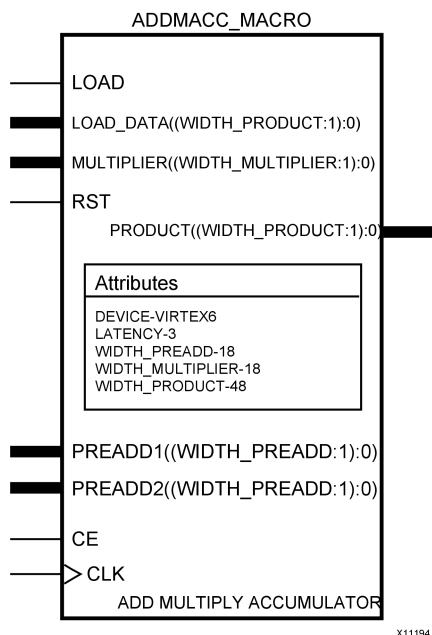
// End of BRAM_TDP_MACRO_inst instantiation
```

## 詳細情報

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

## ADDMACC\_MACRO

マクロ : Adder/Multiplier/Accumulator



### 概要

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

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
PRODUCT	出力	可変 (WIDTH_A 属性値 + WIDTH_B 属性値)	プライマリ データ出力
入力ポート			
PREADD1	入力	可変 (WIDTH_PREADD 属性を参照)	前置加算データ入力
PREADD2	入力	可変 (WIDTH_PREADD 属性を参照)	前置加算データ入力
MULTIPLIER	入力	可変 (WIDTH_MULTIPLIER 属性を参照)	乗算データ入力
CARRYIN	入力	1	キャリー入力
CLK	入力	1	クロック
CE	入力	1	クロック イネーブル
LOAD	入力	1	ロード
LOAD_DATA	入力	可変 (WIDTH_PRODUCT 属性を参照)	DSP スライスでは、LOAD がアサートされると P に $A*B+LOAD\_DATA$ が読み込まれます。

ポート名	方向	幅	機能
RST	入力	1	同期リセット

## デザインの入力方法

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

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- ADDMACC_MACRO: Add and Multiple Accumulate Function implemented in a DSP48E
--                Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

ADDMACC_MACRO_inst : ADDMACC_MACRO
generic map (
    DEVICE => "VIRTEX6", -- Target Device: "VIRTEX6", "SPARTAN6"
    LATENCY => 4,        -- Desired clock cycle latency, 1-4
```

```

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

```

## Verilog 記述 (インスタンス化)

```

// ADDMACC_MACRO: Variable width & latency - Pre-Add -> Multiplier -> Accumulate
//                               function implemented in a DSP48E
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

ADDMACC_MACRO #(
    .DEVICE("VIRTEX6"),    // Target Device: "VIRTEX6", "SPARTAN6"
    .LATENCY(4),           // Desired clock cycle latency, 0-4
    .WIDTH_PREADD(25),     // Pre-adder input width, 1-25
    .WIDTH_MULTIPLIER(18), // Multiplier input width, 1-18
    .WIDTH_PRODUCT(48)     // MACC output width, 1-48
) ADDMACC_MACRO_inst (
    .PRODUCT(PRODUCT),    // MACC result output, width defined by WIDTH_PRODUCT parameter
    .CARRYIN(CARRYIN),    // 1-bit carry-in input
    .CLK(CLK),            // 1-bit clock input
    .CE(CE),              // 1-bit clock enable input
    .LOAD(LOAD),          // 1-bit accumulator load input
    .LOAD_DATA(LOAD_DATA), // Accumulator load data input, width defined by WIDTH_PRODUCT parameter
    .MULTIPLIER(MULTIPLIER), // Multiplier data input, width defined by WIDTH_MULTIPLIER parameter
    .PREADD2(PREADD2),    // Preadder data input, width defined by WIDTH_PREADD parameter
    .PREADD1(PREADD1),    // Preadder data input, width defined by WIDTH_PREADD parameter
    .RST(RST)             // 1-bit active high synchronous reset
);

// End of ADDMACC_MACRO_inst instantiation

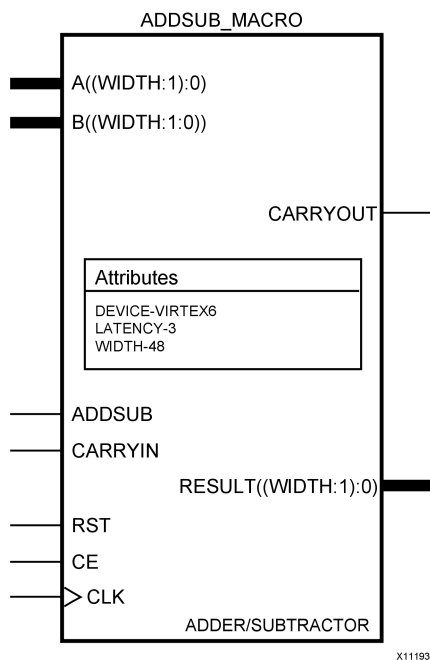
```

## 詳細情報

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

## ADDSUB\_MACRO

マクロ : Adder/Subtractor



### 概要

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

### ポートの説明

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

## デザインの入力方法

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

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- ADDSUB_MACRO: Variable width & latency - Adder / Subtractor implemented in a DSP48E
--                Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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



## Verilog 記述 (インスタンスレーション)

```
// ADDSUB_MACRO: Variable width & latency - Adder / Subtrator implemented in a DSP48E
//                      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

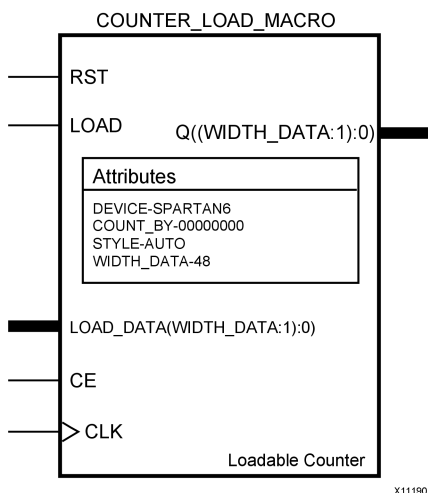
// End of ADDSUB_MACRO_inst instantiation
```

## 詳細情報

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

## COUNTER\_LOAD\_MACRO

### マクロ : Loadable Counter



### 概要

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

### ポートの説明

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

### デザインの入力方法

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

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- COUNTER_LOAD_MACRO: Loadable variable counter implemented in a DSP48E
--                               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

## Verilog 記述 (インスタンス化)

```
// COUNTER_LOAD_MACRO: Loadable variable counter implemented in a DSP48E
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

```

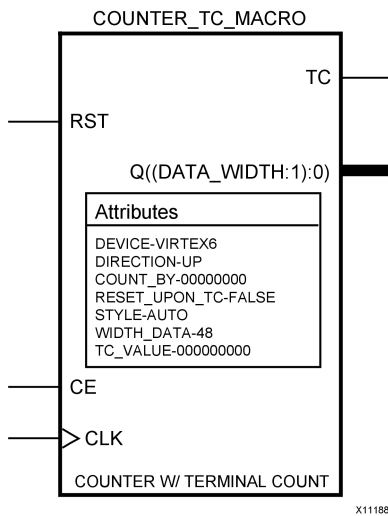
```
.RST(RST)           // 1-bit active high synchronous reset
);
// End of COUNTER_LOAD_MACRO_inst instantiation
```

## 詳細情報

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

## COUNTER\_TC\_MACRO

### マクロ : Counter with Terminal Count



### 概要

COUNTER\_TC\_MACRO を使用すると、DSP48 ブロックをターミナル カウント アップ/ダウン カウンターとして使用する場合のインスタンス化が簡単になります。出力幅、ターミナル カウント値、カウント設定、カウント方向を指定可能であり、DSP48 ブロックを HDL に統合しやすくなっています。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
TC	出力	1	TC_VALUE に達すると High になります。
Q	出力	可変 (WIDTH_DATA 属性を参照)	カウンタ出力
入力ポート			
CE	入力	1	クロック イネーブル
CLK	入力	1	クロック
RST	入力	1	同期リセット

### デザインの入力方法

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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
RESET_UPON_TC	ブール代数	TRUE、FALSE	FALSE	ターミナル カウントに達したときにカウンタをリセットするかどうかを指定します。
DEVICE	文字列	"VIRTEX6"、 "SPARTAN6"	"VIRTEX6"	ターゲットのハードウェア アーキテクチャを指定します。
DIRECTION	文字列	"UP"、"DOWN"	"UP"	カウンタの方向を指定します。
COUNT_BY	16 進数	48 ビット値	000000000001	N ごとにカウント。WIDTH_DATA より優先されます。
TC_VALUE	16 進数	48 ビット値	すべてゼロ	ターミナル カウント値を指定します。
WIDTH_DATA	整数	1 ~ 48	48	カウンタの幅を指定します。

## VHDL 記述 (インスタンスーション)

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

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

-- COUNTER_TC_MACRO: Counter with terminal count implemented in a DSP48E
--                               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

COUNTER_TC_MACRO_inst : COUNTER_TC_MACRO
generic map (
    COUNT_BY => X"000000000001", -- Count by value
    DEVICE => "VIRTEX6",         -- Target Device: "VIRTEX5", "VIRTEX6"
    DIRECTION => "UP",           -- Counter direction "UP" or "DOWN"
    RESET_UPON_TC => "FALSE",    -- Reset counter upon terminal count, TRUE or FALSE
    TC_VALUE => X"000000000000", -- Terminal count value
    WIDTH_DATA => 48)            -- Counter output bus width, 1-48
port map (
    Q => Q,                      -- Counter output, width determined by WIDTH_DATA generic
    TC => TC,                    -- 1-bit terminal count output, high = terminal count is reached
    CLK => CLK,                  -- 1-bit clock input
    CE => CE,                    -- 1-bit clock enable input
    RST => RST                   -- 1-bit active high synchronous reset
);
-- End of COUNTER_TC_MACRO_inst instantiation
```

## Verilog 記述 (インスタンスーション)

```
// COUNTER_TC_MACRO: Counter with terminal count implemented in a DSP48E
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

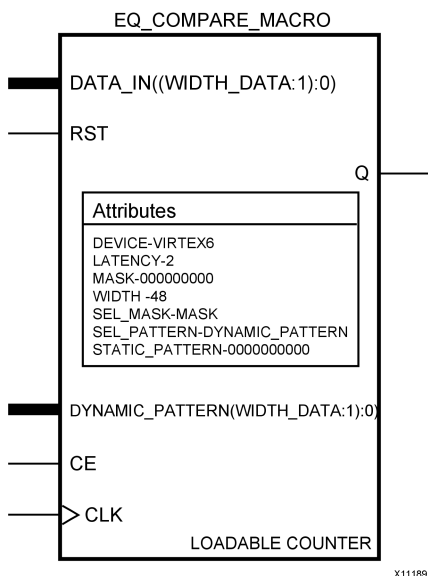
```
.RST(RST) // 1-bit active high synchronous reset
);
// End of COUNTER_TC_MACRO_inst instantiation
```

## 詳細情報

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

## EQ\_COMPARE\_MACRO

### マクロ : Equality Comparator



### 概要

EQ\_COMPARE\_MACRO を使用すると、DSP48 ブロックを等価コンパレータとして使用する場合のインスタンス化が簡単になります。入力幅、出力幅、レイテンシ、マスク、および入力ソースを指定可能であり、DSP48 ブロックを HDL に統合しやすくなっています。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
Q	出力	1	アクティブ High のパターン検出で、MASK 属性でマスクした DYNAMIC_PATTERN の値と DATA_IN の値が一致したときに High になります。結果は P と同じクロック サイクルで出力されます。
入力ポート			
DATA_IN	入力	可変 (WIDTH 属性で指定した値)	比較する入力データ
DYNAMIC_PATTERN	入力	可変 (WIDTH 属性で指定した値)	DATA_IN と比較するダイナミック データ
CLK	入力	1	クロック
CE	入力	1	クロック イネーブル
RST	入力	1	同期リセット

### デザインの入力方法

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



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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DEVICE	文字列	"VIRTEX5"、 "VIRTEX6"、 "SPARTAN6"	"VIRTEX6"	ターゲットのハードウェア アーキテクチャを指定します。
SEL_PATTERN	整数	1 ~ 24	24	PREADD1 および PREADD2 入力の幅を制御します。
MASK	16 進数	48 ビットの 16 進数	すべてゼロ	パターン検出器で使用されるマスクを指定します。
STATIC_PATTERN	16 進数	48 ビットの 16 進数	すべてゼロ	パターン検出器で使用されるパターンを指定します。
SEL_MASK	文字列	"MASK"、 "DYNAMIC_PATTERN"	"MASK"	パターン検出器のマスクに MASK を使用するか、C 入力を使用するかを指定します。
WIDTH	整数	1 ~ 48	48	DATA_IN および DYNAMIC_PATTERN の幅を指定します。
LATENCY	整数	0、1、2、3	2	パイプライン レジスタの数を指定します。  <ul style="list-style-type: none"> <li>1 : QREG == 1</li> <li>2 : AREG == BREG == CREG == QREG == 1</li> <li>3 : AREG == BREG == 2 and CREG == QREG == 1</li> </ul>

## VHDL 記述 (インスタンス化)

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;

-- EQ_COMPARE_MACRO: Equality Comparator implemented in a DSP48E
--                      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

EQ_COMPARE_MACRO_inst : EQ_COMPARE_MACRO
generic map (
  DEVICE => "VIRTEX6",           -- Target Device: "VIRTEX5", "VIRTEX6"
  LATENCY => 2,                   -- Desired clock cycle latency, 0-2
  MASK => X"000000000000",       -- Select bits to be masked, must set
                                -- SEL_MASK = "MASK"
  SEL_MASK => "MASK",            -- "MASK" = use MASK generic,
                                -- "DYNAMIC_PATTERN" = use DYNAMIC_PATTERN input bus
  SEL_PATTERN => "DYNAMIC_PATTERN", -- "DYNAMIC_PATTERN" = use DYNAMIC_PATTERN input bus
                                -- "STATIC_PATTERN" = use STATIC_PATTERN generic
)
```

```

    STATIC_PATTERN => X"000000000000", -- Specify static pattern,
                                   -- must set SEL_PATTERN = "STATIC_PATTERN"
    WIDTH => 48)                -- Comparator output bus width, 1-48
port map (
    Q => Q,                    -- 1-bit output indicating a match
    CE => CE,                  -- 1-bit active high input clock enable input
    CLK => CLK,                -- 1-bit positive edge clock input
    DATA_IN => DATA_IN, -- Input Data Bus, width determined by WIDTH generic
    DYNAMIC_PATTERN => DYNAMIC_PATTERN, -- Input Dynamic Match/Mask Bus, width determined by WIDTH generic
    RST => RST                -- 1-bit input active high reset
);
-- End of EQ_COMPARE_MACRO_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// EQ_COMPARE_MACRO: Equality Comparator implemented in a DSP48E
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

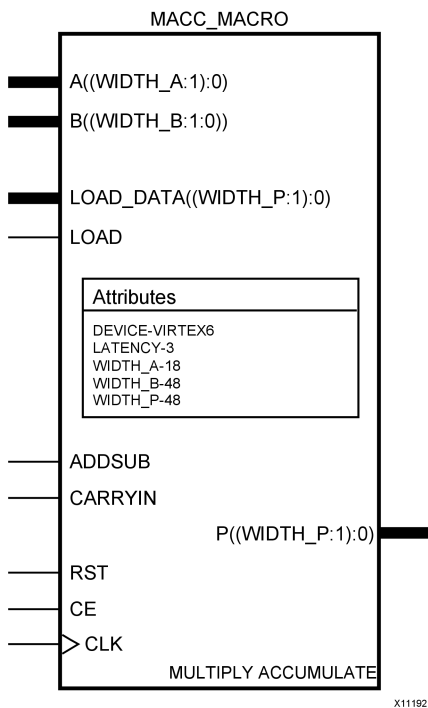
```

## 詳細情報

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

## MACC\_MACRO

マクロ : Multiplier/Accumulator



### 概要

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

### ポートの説明

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

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

## デザインの入力方法

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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
WIDTH_A	整数	1 ~ 25	25	A 入力の幅を指定します。
WIDTH_B	整数	1 ~ 18	18	B 入力の幅を指定します。
LATENCY	整数	0、1、2、3、4	3	パイプライン レジスタの数を指定します。 <ul style="list-style-type: none"> <li>1 : MREG == 1</li> <li>2 : AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1</li> <li>3 : AREG == BREG == 1 および MREG == 1 および PREG == 1</li> <li>4 : AREG == BREG == 2 および MREG == 1 および PREG == 1</li> </ul>
DEVICE	文字列	"VIRTEX5"、 "VIRTEX6"、 "SPARTAN6"	"VIRTEX6"	ターゲットのハードウェア アーキテクチャを指定します。

## VHDL 記述 (インスタンスレーション)

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;

-- MACC_MACRO: Multiple Accumulate Function implemented in a DSP48E
--             Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

```

```

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

-- End of MACC_MACRO_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// MACC_MACRO: Multiply Accumulate Function implemented in a DSP48E
//               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

MACC_MACRO #(
    .DEVICE("VIRTEX6"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    .LATENCY(3),        // Desired clock cycle latency, 1-4
    .WIDTH_A(25),       // Multiplier A-input bus width, 1-25
    .WIDTH_B(18),       // Multiplier B-input bus width, 1-18
    .WIDTH_P(48)        // Accumulator output bus width, 1-48
) MACC_MACRO_inst (
    .P(P),              // MACC output bus, width determined by WIDTH_P parameter
    .A(A),              // MACC input A bus, width determined by WIDTH_A parameter
    .ADDSUB(ADDSUB),    // 1-bit add/sub input, high selects add, low selects subtract
    .B(B),              // MACC input B bus, width determined by WIDTH_B parameter
    .CARRYIN(CARRYIN), // 1-bit carry-in input to accumulator
    .CE(CE),            // 1-bit active high input clock enable
    .CLK(CLK),          // 1-bit positive edge clock input
    .LOAD(LOAD),        // 1-bit active high input load accumulator enable
    .LOAD_DATA(LOAD_DATA), // Load accumulator input data, width determined by WIDTH_P parameter
    .RST(RST)           // 1-bit input active high reset
);

// End of MACC_MACRO_inst instantiation

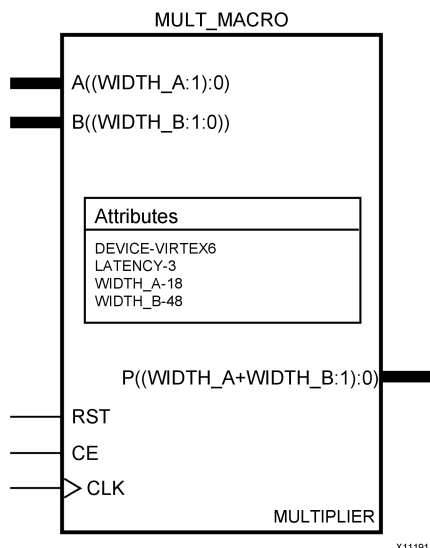
```

## 詳細情報

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

## MULT\_MACRO

### マクロ : Multiplier



### 概要

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

### ポートの説明

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

### デザインの入力方法

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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
WIDTH_A	整数	1 ~ 25	25	A 入力の幅を指定します。
WIDTH_B	整数	1 ~ 18	18	B 入力の幅を指定します。
LATENCY	整数	0、1、2、3、4	3	パイプライン レジスタの数を指定します。 <ul style="list-style-type: none"> <li>1 : MREG == 1</li> <li>2 : AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1</li> <li>3 : AREG == BREG == 1 および MREG == 1 および PREG == 1</li> <li>4 : AREG == BREG == 2 および MREG == 1 および PREG == 1</li> </ul>
DEVICE	文字列	"VIRTEX5"、 "VIRTEX6"、 "SPARTAN6"	"VIRTEX6"	ターゲットのハードウェア アーキテクチャを指定します。

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- MULT_MACRO: Multiply Function implemented in a DSP48E
--               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

```

## Verilog 記述 (インスタンスレーション)

```
// MULT_MACRO: Multiply Function implemented in a DSP48E
//               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

// End of MULT_MACRO_inst instantiation
```

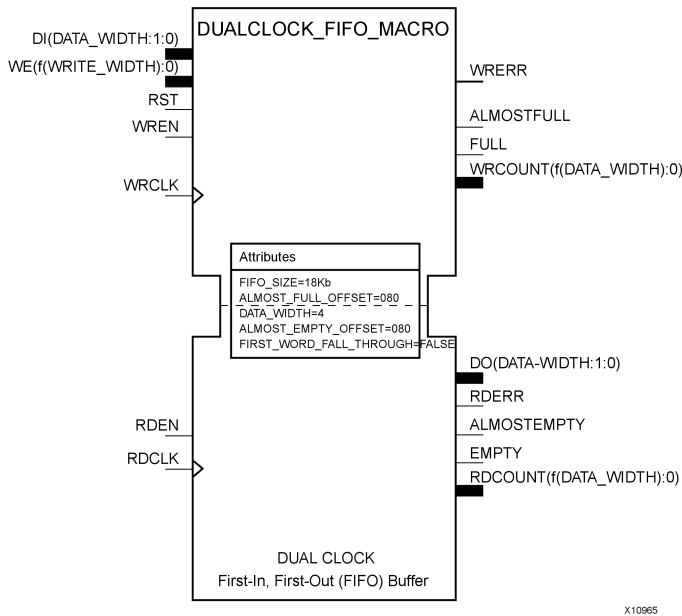
## 詳細情報

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



## FIFO\_DUALCLOCK\_MACRO

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



### 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	すべてゼロ	EMPTY と ALMOSTEMPTY の差を設定します。16 進数で指定します。
ALMOST_FULL_OFFSET	16 進数	13 ビット値	すべてゼロ	FULL と ALMOSTFULL の差を設定します。16 進数で指定します。
DATA_WIDTH	整数	1 ～ 72	4	DI/DO バスの幅を指定します。
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-6
-- Xilinx HDL Libraries Guide, version 13.3

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

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

FIFO_DUALCLOCK_MACRO_inst : FIFO_DUALCLOCK_MACRO
generic map (
    DEVICE => "VIRTEX6", -- 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
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit output almost empty
    ALMOSTFULL => ALMOSTFULL, -- 1-bit output almost full
    DO => DO, -- Output data, width defined by DATA_WIDTH parameter
    EMPTY => EMPTY, -- 1-bit output empty
    FULL => FULL, -- 1-bit output full

```

```

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

```

## Verilog 記述 (インスタンスレーション)

```

// FIFO_DUALCLOCK_MACRO: Dual Clock First-In, First-Out (FIFO) RAM Buffer
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

////////////////////////////////////
// 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("VIRTEX6"), // 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"
) FIFO_DUALCLOCK_MACRO_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit output almost empty
    .ALMOSTFULL(ALMOSTFULL), // 1-bit output almost full
    .DO(DO), // Output data, width defined by DATA_WIDTH parameter
    .EMPTY(EMPTY), // 1-bit output empty
    .FULL(FULL), // 1-bit output full
    .RDCOUNT(RDCOUNT), // Output read count, width determined by FIFO depth
    .RDERR(RDERR), // 1-bit output read error
    .WRCOUNT(WRCOUNT), // Output write count, width determined by FIFO depth
    .WRERR(WRERR), // 1-bit output write error
    .DI(DI), // Input data, width defined by DATA_WIDTH parameter
    .RDCLK(RDCLK), // 1-bit input read clock
    .RDEN(RDEN), // 1-bit input read enable
    .RST(RST), // 1-bit input reset
    .WRCLK(WRCLK), // 1-bit input write clock
    .WREN(WREN) // 1-bit input write enable
);

// End of FIFO_DUALCLOCK_MACRO_inst instantiation

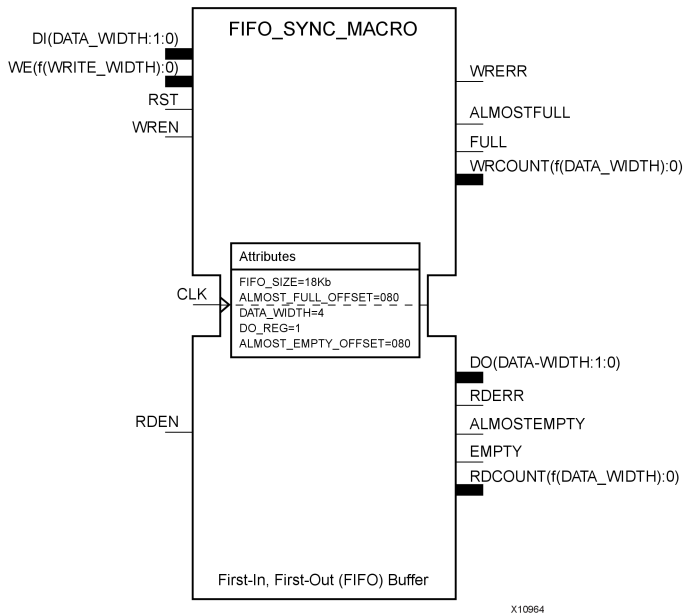
```

## 詳細情報

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

## FIFO\_SYNC\_MACRO

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



## 概要

FPGA デバイスにはブロック RAM が数個含まれ、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	すべてゼロ	EMPTY と ALMOSTEMPTY の差を設定します。16 進数で指定します。
ALMOST_FULL_OFFSET	16 進数	13 ビット値	すべてゼロ	FULL と ALMOSTFULL の差を設定します。16 進数で指定します。
DATA_WIDTH	整数	1 ~ 72	4	DI/DO バスの幅を指定します。
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-6
-- Xilinx HDL Libraries Guide, version 13.3

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

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

FIFO_SYNC_MACRO_inst : FIFO_SYNC_MACRO
generic map (
    DEVICE => "VIRTEX6", -- 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"
)
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit output almost empty
    ALMOSTFULL => ALMOSTFULL, -- 1-bit output almost full

```

```

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

```

## Verilog 記述 (インスタンスレーション)

```

// FIFO_SYNC_MACRO: Synchronous First-In, First-Out (FIFO) RAM Buffer
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

////////////////////////////////////
// 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("VIRTEX6"), // 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"
) FIFO_SYNC_MACRO_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit output almost empty
    .ALMOSTFULL(ALMOSTFULL), // 1-bit output almost full
    .DO(DO), // Output data, width defined by DATA_WIDTH parameter
    .EMPTY(EMPTY), // 1-bit output empty
    .FULL(FULL), // 1-bit output full
    .RDCOUNT(RDCOUNT), // Output read count, width determined by FIFO depth
    .RDERR(RDERR), // 1-bit output read error
    .WRCOUNT(WRCOUNT), // Output write count, width determined by FIFO depth
    .WRERR(WRERR), // 1-bit output write error
    .CLK(CLK), // 1-bit input clock
    .DI(DI), // Input data, width defined by DATA_WIDTH parameter
    .RDEN(RDEN), // 1-bit input read enable
    .RST(RST), // 1-bit input reset
    .WREN(WREN) // 1-bit input write enable
);

// End of FIFO_SYNC_MACRO_inst instantiation

```

## 詳細情報

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



## ファンクション別分類

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

アドバンス	コンビニエンス プリミティブ	シフトレジスタ
四則演算ファンクション	I/O コンポーネント	スライス/CLB プリミティブ
クロック コンポーネント	RAM/ROM	
コンフィギュレーション/BSCAN コンポーネント	レジスタおよびラッチ	

### アドバンス

デザイン エLEMENT	説明
PCIE_2.0	プリミティブ : PCI Express version 2.0 Compliant.
SYSMON	プリミティブ : System Monitor

### 四則演算ファンクション

デザイン エLEMENT	説明
DSP48E1	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	コンビニエンス プリミティブ : Global Clock Buffer
BUFGCE	コンビニエンス プリミティブ : Global Clock Buffer with Clock Enable
BUFGCTRL	プリミティブ : Global Clock MUX Buffer
BUFGMUX_CTRL	コンビニエンス プリミティブ : 2-to-1 Global Clock MUX Buffer
BUFH	コンビニエンス プリミティブ : Clock buffer for a single clocking region
BUFHCE	プリミティブ : Clock buffer for a single clocking region with clock enable
BUFIO	プリミティブ : Local Clock Buffer for I/O
BUFIODQS	プリミティブ : Differential Clock Input for Transceiver Reference Clocks
BUFR	プリミティブ : Regional Clock Buffer for I/O and Logic Resources
IBUFDS_GTXE1	プリミティブ : Differential Clock Input for the Transceiver Reference Clocks
MMCM_ADV	プリミティブ : MMCM is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.

## 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_VIRTEX6	プリミティブ : Virtex®-6 JTAG Boundary-Scan Logic Access Circuit
CAPTURE_VIRTEX6	プリミティブ : Virtex®-6 Readback Register Capture Control
DNA_PORT	プリミティブ : Device DNA Data Access Port
EFUSE_USR	プリミティブ : 32-bit non-volatile design ID
FRAME_ECC_VIRTEX6	プリミティブ : Virtex®-6 Configuration Frame Error Detection and Correction Circuitry
ICAP_VIRTEX6	プリミティブ : Internal Configuration Access Port
JTAG_SIM_VIRTEX6	シミュレーション : JTAG TAP Controller Simulation Model
SIM_CONFIG_V6	シミュレーション : Configuration Simulation Model
SIM_CONFIG_V6_SERIAL	シミュレーション : Serial Configuration Simulation Model
STARTUP_VIRTEX6	プリミティブ : Virtex®-6 Configuration Start-Up Sequence Interface
USR_ACCESS_VIRTEX6	プリミティブ : Virtex-6 User Access Register

## コンビニエンス プリミティブ

デザイン エLEMENT	説明
BUFGCE_1	コンビニエンス プリミティブ : Global Clock Buffer with Clock Enable and Output State 1
BUFGMUX	コンビニエンス プリミティブ : Global Clock MUX Buffer
BUFGMUX_1	コンビニエンス プリミティブ : Global Clock MUX Buffer with Output State 1
BUFH	コンビニエンス プリミティブ : Clock buffer for a single clocking region
MMCM_BASE	コンビニエンス プリミティブ : Mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.

## 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)
GTHE1_QUAD	プリミティブ : Gigabit Transceiver
GTXE1	プリミティブ : Gigabit Transceiver
IBUF	プリミティブ : Input Buffer
IBUFDS	プリミティブ : Differential Signaling Input Buffer
IBUFDS_DIFF_OUT	プリミティブ : Signaling Input Buffer with Differential Output
IBUFDS_GTHE1	プリミティブ : Differential Clock Input for the GTH Transceiver Reference Clocks
IBUFG	プリミティブ : Dedicated Input Clock Buffer
IBUFGDS	プリミティブ : Differential Signaling Dedicated Input Clock Buffer and Optional Delay
IBUFGDS_DIFF_OUT	プリミティブ : Differential Signaling Input Buffer with Differential Output
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
IODELAYE1	プリミティブ : Input and Output Fixed or Variable Delay Element
ISERDESE1	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

デザイン エLEMENT	説明
OSERDESE1	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
TEMAC_SINGLE	プリミティブ : Tri-mode Ethernet Media Access Controller (MAC)

## RAM/ROM

デザイン エLEMENT	説明
FIFO18E1	プリミティブ : 18 k-bit FIFO (First In, First Out) Block RAM Memory
FIFO36E1	プリミティブ : 36 kb FIFO (First In, First Out) Block RAM Memory
RAM128X1D	プリミティブ : 128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)
RAM256X1S	プリミティブ : 256-Deep by 1-Wide Random Access Memory (Select RAM)
RAM32M	プリミティブ : 32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)
RAM32X1D	プリミティブ : 32-Deep by 1-Wide Static Dual Port Synchronous RAM
RAM32X1S	プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM
RAM32X1S_1	プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAM32X2S	プリミティブ : 32-Deep by 2-Wide Static Synchronous RAM
RAM64M	プリミティブ : 64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)
RAM64X1D	プリミティブ : 64-Deep by 1-Wide Dual Port Static Synchronous RAM
RAM64X1S	プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM
RAM64X1S_1	プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAMB18E1	プリミティブ : 18K-bit Configurable Synchronous Block RAM
RAMB36E1	プリミティブ : 36K-bit Configurable Synchronous Block RAM
ROM128X1	プリミティブ : 128-Deep by 1-Wide ROM
ROM256X1	プリミティブ : 256-Deep by 1-Wide ROM
ROM32X1	プリミティブ : 32-Deep by 1-Wide ROM
ROM64X1	プリミティブ : 64-Deep by 1-Wide ROM

## レジスタおよびラッチ

デザイン エLEMENT	説明
FDCE	プリミティブ：D Flip-Flop with Clock Enable and Asynchronous Clear
FDPE	プリミティブ：D Flip-Flop with Clock Enable and Asynchronous Preset
FDRE	プリミティブ：D Flip-Flop with Clock Enable and Synchronous Reset
FDSE	プリミティブ：D Flip-Flop with Clock Enable and Synchronous Set
IDDR	プリミティブ：Input Dual Data-Rate Register
IDDR_2CLK	プリミティブ：Input Dual Data-Rate Register with Dual Clock Inputs
LDCE	プリミティブ：Transparent Data Latch with Asynchronous Clear and Gate Enable
LDPE	プリミティブ：Transparent Data Latch with Asynchronous Preset and Gate Enable
ODDR	プリミティブ：Dedicated Dual Data Rate (DDR) Output Register

## シフト レジスタ

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

デザイン エLEMENT	説明
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

---

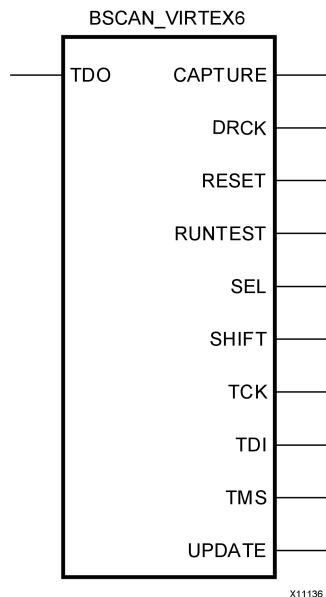
このセクションでは、このアーキテクチャで利用できるデザイン エLEMENTについて説明します。デザイン エLEMENTは、アルファベット順に並べられています。

各ライブラリ エLEMENTについて、次の情報を示します。

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

## BSCAN\_VIRTEX6

**プリミティブ**：Virtex®-6 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へアクセスできるので、内部実行デザインと FPGA の専用 JTAG ピン間の通信を可能にします。

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。USER 命令の 4 つ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。

**メモ**：各アーキテクチャのバウンダリ スキャンの詳細については、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	スキャン データレジスタ キャプチャ命令
DRCK	出力	1	スキャン クロック命令。DRCK は ゲートが付いた TCTCK で、CAPTUREDR および SHIFTDTR ステート中にトグルします。
RESET	出力	1	スキャン レジスタ リセット命令
RUNTEST	出力	1	TAP コントローラーが Run Test Idle ステートのときにアサートされます。
SEL	出力	1	スキャン モード セレクト命令
SHIFT	出力	1	スキャン チェーン シフト命令
TCK	出力	1	スキャン クロック。TAP クロック ピンへのファブリック接続。
TDI	出力	1	スキャン チェーン出力。FPGA への TDI 入力ピンのミラー
TDO	入力	1	スキャン チェーン入力
TMS	出力	1	テスト モード セレクト。TAP へのファブリック接続。
UPDATE	出力	1	スキャン レジスタ リセット命令



## 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_VIRTEX6: Boundary Scan
--               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

BSCAN_VIRTEX6_inst : BSCAN_VIRTEX6
generic map (
  DISABLE_JTAG => FALSE, -- This attribute is unsupported. Please leave it at default.
  JTAG_CHAIN => 1        -- Value for USER command. Possible values: (1,2,3 or 4).
)
port map (
  CAPTURE => CAPTURE, -- 1-bit output: CAPTURE output from TAP controller
  DRCK => DRCK,       -- 1-bit output: Data register output for USER functions
  RESET => RESET,     -- 1-bit output: Reset output for TAP controller
  RUNTEST => RUNTEST, -- 1-bit output: State output asserted when TAP controller is in Run Test Idle state.
  SEL => SEL,         -- 1-bit output: USER active output
  SHIFT => SHIFT,     -- 1-bit output: SHIFT output from TAP controller
  TCK => TCK,         -- 1-bit output: Scan Clock output. Fabric connection to TAP Clock pin.
  TDI => TDI,         -- 1-bit output: TDI output from TAP controller
  TMS => TMS,         -- 1-bit output: Test Mode Select input. Fabric connection to TAP.
  UPDATE => UPDATE,   -- 1-bit output: UPDATE output from TAP controller
  TDO => TDO          -- 1-bit input: Data input for USER function
);

-- End of BSCAN_VIRTEX6_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BSCAN_VIRTEX6: Boundary Scan
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

BSCAN_VIRTEX6 #(
    .DISABLE_JTAG("FALSE"), // This attribute is unsupported. Please leave it at default.
    .JTAG_CHAIN(1)          // Value for USER command. Possible values: (1,2,3 or 4).
)
BSCAN_VIRTEX6_inst (
    .CAPTURE(CAPTURE), // 1-bit output: CAPTURE output from TAP controller
    .DRCK(DRCK),        // 1-bit output: Data register output for USER functions
    .RESET(RESET),      // 1-bit output: Reset output for TAP controller
    .RUNTEST(RUNTEST),  // 1-bit output: State output asserted when TAP controller is in Run Test Idle state.
    .SEL(SEL),          // 1-bit output: USER active output
    .SHIFT(SHIFT),      // 1-bit output: SHIFT output from TAP controller
    .TCK(TCK),          // 1-bit output: Scan Clock output. Fabric connection to TAP Clock pin.
    .TDI(TDI),          // 1-bit output: TDI output from TAP controller
    .TMS(TMS),          // 1-bit output: Test Mode Select input. Fabric connection to TAP.
    .UPDATE(UPDATE),    // 1-bit output: UPDATE output from TAP controller
    .TDO(TDO)           // 1-bit input: Data input for USER function
);

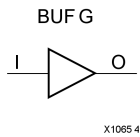
// End of BSCAN_VIRTEX6_inst instantiation
```

## 詳細情報

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

## BUFG

### コンビニエンス プリミティブ：Global Clock Buffer



### 概要

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

### ポートの説明

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

### デザインの入力方法

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

### VHDL 記述 (インスタンス化)

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

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

-- BUFG: Global Clock Buffer
-- Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of BUFG_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFG: Global Clock Buffer
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

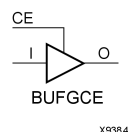
// End of BUFG_inst instantiation
```

## 詳細情報

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

## BUFGCE

### コンビニエンス プリミティブ：Global Clock Buffer with Clock Enable



### 概要

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

### 論理表

入力		出力
I	CE	O
X	0	0
I	1	I

### ポートの説明

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

### デザインの入力方法

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

## VHDL 記述 (インスタンス化)

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

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

-- BUFGCE: Global Clock Buffer with Clock Enable
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of BUFGCE_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGCE: Global Clock Buffer with Clock Enable
//          Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

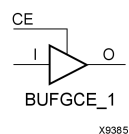
// End of BUFGCE_inst instantiation
```

## 詳細情報

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

## BUFGCE\_1

コンビニエンス 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 and Output State 1
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

BUFGCE_1_inst : BUFGCE_1
port map (
    O => O,  -- 1-bit output: Clock buffer output
    CE => CE, -- 1-bit input: Clock enable input for I0 input
    I => I   -- 1-bit input: Primary clock 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 and Output State 1
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

BUFGCE_1 BUFGCE_1_inst (
    .O(O),    // 1-bit output: Clock buffer output
    .CE(CE),  // 1-bit input: Clock enable input for I0 input
    .I(I)     // 1-bit input: Primary clock input
);

// End of BUFGCE_1_inst instantiation
```

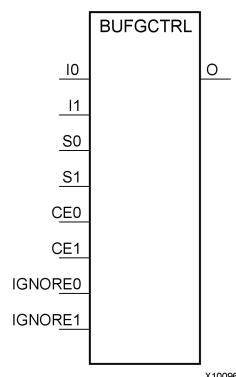
## 詳細情報

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



## BUFGCTRL

プリミティブ：Global Clock 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
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。

メモ：2 つの PRESELECT 属性を同時に TRUE にすることはできません。

## VHDL 記述（インスタンス化）

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGCTRL: Global Clock MUX Buffer
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of BUFGCTRL_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// BUFGCTRL: Global Clock MUX Buffer
//          Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

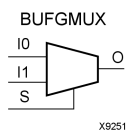
// End of BUFGCTRL_inst instantiation
```

## 詳細情報

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

## BUFGMUX

### コンビニエンス プリミティブ：Global Clock MUX Buffer



### 概要

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

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

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

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

## VHDL 記述 (インスタンスレーション)

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

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

-- BUFGMUX: Global Clock Mux Buffer
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of BUFGMUX_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFGMUX: Global Clock Mux Buffer
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

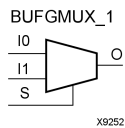
// End of BUFGMUX_inst instantiation
```

## 詳細情報

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

## BUFGMUX\_1

### コンビニエンス プリミティブ : Global Clock MUX Buffer with Output State 1



### 概要

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

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

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

## VHDL 記述 (インスタンス化)

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

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

-- BUFGMUX_1: Global Clock Mux Buffer with Output State 1
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of BUFGMUX_1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGMUX_1: Global Clock Mux Buffer with Output State 1
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

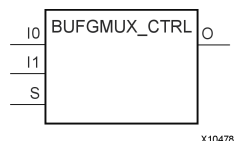
// End of BUFGMUX_1_inst instantiation
```

## 詳細情報

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

## BUFGMUX\_CTRL

### コンビニエンス プリミティブ：2-to-1 Global Clock MUX Buffer



### 概要

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

### ポートの説明

ポート名	方向	幅	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
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。

### 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: 2-to-1 Global Clock MUX Buffer
--                Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of BUFGMUX_CTRL_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFGMUX_CTRL: 2-to-1 Global Clock MUX Buffer
//                Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

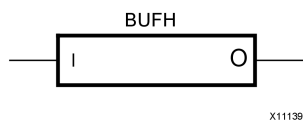
// End of BUFGMUX_CTRL_inst instantiation
```

## 詳細情報

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

## BUFH

コンビニエンス プリミティブ：Clock buffer for a single clocking region



### 概要

インスタンス化で HCLK クロック バッファ リソースを使用するための機能を提供します。このコンポーネントは手動で配置する必要があり、また特別な考慮が必要なため、アドバンス ユーザー向けです。このコンポーネントの詳細は、『[Virtex-6 FPGA クロック リソース ユーザー ガイド](#)』(UG362) を参照してください。

### ポートの説明

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

### デザインの入力方法

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

### VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFH: HROW Clock Buffer for a Single Clocking Region
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of BUFH_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// BUFH: HROW Clock Buffer for a Single Clocking Region
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

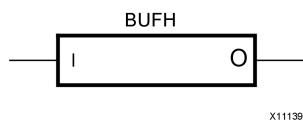
// End of BUFH_inst instantiation
```

## 詳細情報

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

## BUFH

コンビニエンス プリミティブ：Clock buffer for a single clocking region



### 概要

インスタンス化で HCLK クロック バッファ リソースを使用するための機能を提供します。このコンポーネントは手動で配置する必要があり、また特別な考慮が必要なため、アドバンス ユーザー向けです。このコンポーネントの詳細は、『[Virtex-6 FPGA クロック リソース ユーザー ガイド](#)』(UG362) を参照してください。

### ポートの説明

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

### デザインの入力方法

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

### VHDL 記述 (インスタンス化)

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

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

-- BUFH: HROW Clock Buffer for a Single Clocking Region
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of BUFH_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BUFH: HROW Clock Buffer for a Single Clocking Region
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

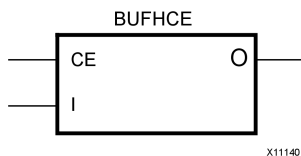
// End of BUFH_inst instantiation
```

## 詳細情報

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

## BUFHCE

**プリミティブ**：Clock buffer for a single clocking region with clock enable



### 概要

このデザイン エレメントでは、インスタンス化で HCLK クロック バッファ リソースを使用するための機能を提供します。また、CE を介してクロック イネーブル (CE) をディスエーブルにする電力消費量低減機能もあります。

### ポートの説明

ポート名	方向	幅	機能
CE	入力	1	I から O まで信号を伝播します。Low の場合出力を 0 に設定します。
I	入力	1	BUFH への入力
O	出力	1	BUFH の出力

### デザインの入力方法

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

### 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_OUT	10 進数	0、1	0	初期出力値で、停止が Low か High かを示します。

### 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;

-- BUFHCE: HROW Clock Buffer for a Single Clocking Region with Clock Enable
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

BUFHCE_inst : BUFHCE
generic map (
  INIT_OUT => 0 -- Initial output value
)
port map (
  O => O,    -- 1-bit output: Clock output
  CE => CE,  -- 1-bit input: Active high enable input
  I => I     -- 1-bit input: Clock input
```

```
);  
-- End of BUFHCE_inst instantiation
```

## Verilog 記述 (インスタンス化)

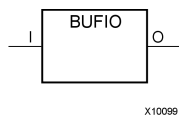
```
// BUFHCE: HROW Clock Buffer for a Single Clocking Region with Clock Enable  
//          Virtex-6  
// Xilinx HDL Libraries Guide, version 13.3  
  
BUFHCE #(  
    .INIT_OUT(0) // Initial output value  
)  
BUFHCE_inst (  
    .O(O), // 1-bit output: Clock output  
    .CE(CE), // 1-bit input: Active high enable input  
    .I(I) // 1-bit input: Clock input  
) ;  
  
// End of BUFHCE_inst instantiation
```

## 詳細情報

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

## BUFIO

### プリミティブ：Local Clock Buffer for I/O



### 概要

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 ネットワークの範囲は I/O 列までなので、CLB やブロック RAM などのロジック リソース は駆動できません。

### 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: I/O Clock Buffer
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

BUFIO_inst : BUFIO
port map (
  O => O,      -- Buffer output
  I => I        -- 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-6
// Xilinx HDL Libraries Guide, version 13.3

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

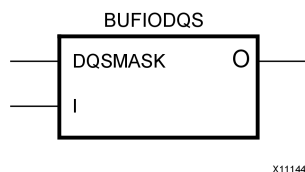
// End of BUFIO_inst instantiation
```

## 詳細情報

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

## BUFIODQS

**プリミティブ：Differential Clock Input for Transceiver Reference Clocks**



### 概要

メモリ アプリケーションの使用に理想的な追加専用回路付き BUFIO と同じクロック バッファで、オプションで余分な BUFIO 遅延を削除し、ストロブからのバースト長の後に I/O クロックをスケルチします。通常、このコンポーネントはザイリンクス MIG (Memory Interface Generator) 製品とのみ使用してください。

### ポートの説明

ポート名	方向	幅	機能
DQSMASK	入力	1	ストロブからのバースト長の後に I/O クロックをスケルチします。
I	入力	1	クロック入力ポート
O	出力	1	クロック出力ポート

### 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

### 使用可能な属性

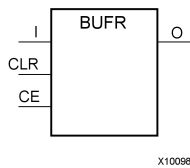
属性	データ型	値	デフォルト	説明
DQSMASK_ENABLE	ブール代数	FALSE、TRUE	FALSE	スケルチ回路を有効にします。

### 詳細情報

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

## BUFR

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



### 概要

リージョナル クロック バッファ (BUFR) は、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 6 つと、隣接するクロック領域のクロック ネット 6 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。BUFG は、クロックを使用可能なピン、ローカル インターフェイスコネク、GT、および MMCM 高性能クロックにより駆動できます。クロック入力信号を分周したクロックを出力することもできます。分周の除算値は、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。各 I/O 列では領域クロック バッファがサポートされています。内側 2 列 (中央左/右) と外側 2 列 (左、右) に最大 4 個の I/O 列があります。外側列が使用できるかどうかは、デバイスによって異なりますが、内部列は常に使用できます。Virtex®-6 アーキテクチャでは、内側列から 2 個と外側列から 2 個、領域ごとに最大 4 個の BUFR を持つことができます。Virtex-6 デバッグでは、BUFR で MMCM クロック入力および BUFG を直接駆動することもできます。

### 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 にアサートされるとポート) の出力クロックがディスエーブルにされます。High にアサートされると、分周クロック出力を生成するのに使用するカウンタがリセットされます。
CLR	入力	1	分周クロック出力用のカウンタ リセット。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	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
BUFR_DIVIDE	文字列	“BYPASS”、“1”、“2”、“3”、“4”、“5”、“6”、“7”、“8”	“BYPASS”	出力クロックに入力クロックを分周したクロックを出力するかどうかを指定します。
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;

-- BUFR: Regional (Local) Clock Buffer /w Enable, Clear and Division Capabilities
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

BUFR_inst : BUFR
generic map (
    BUFR_DIVIDE => "BYPASS",    -- "BYPASS", "1", "2", "3", "4", "5", "6", "7", "8"
    SIM_DEVICE  => "VIRTEX6")   -- 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 (インスタンス化)

```
// BUFR: Regional Clock Buffer w/ Enable, Clear and Division Capabilities
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

BUFR #(
    .BUFR_DIVIDE("BYPASS"), // "BYPASS", "1", "2", "3", "4", "5", "6", "7", "8"
    .SIM_DEVICE("VIRTEX6") // Specify target device, "VIRTEX4", "VIRTEX5", "VIRTEX6"
) BUFR_inst (
    .O(O),      // Clock buffer output
    .CE(CE),    // Active high divide counter clock enable input, when low diables output
    .CLR(CLR),  // Active high divide counter reset input
    .I(I)       // Clock buffer input driven by an IBUFG, MMCM or local interconnect
);

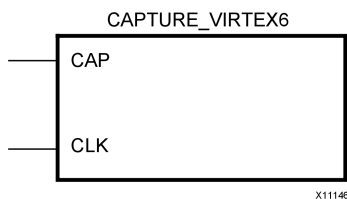
// End of BUFR_inst instantiation
```

## 詳細情報

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

## CAPTURE\_VIRTEX6

### プリミティブ：Virtex®-6 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、トリガー（CAP をアサートしているときの CLK の遷移）のたびにデータがキャプチャされます。リードバック処理を 1 回のデータ キャプチャだけに制限するには、このELEMENTに ONESHOT=TRUE 属性を追加します。

### 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

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

正しく動作するように、入力および出力をすべてデザインに接続します。

### 使用可能な属性

属性	データ型	値	デフォルト	説明
ONESHOT	ブール代数	TRUE、FALSE	TRUE	CAP トリガーごとに 1 回のリードバックを実行します。

## VHDL 記述 (インスタンスレーション)

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

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

-- CAPTURE_VIRTEX6: Register Capture
--                               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

CAPTURE_VIRTEX6_inst : CAPTURE_VIRTEX6
generic map (
    ONESHOT => TRUE -- Specifies the procedure for performing single readback per CAP trigger.
)
port map (
    CAP => CAP, -- 1-bit input: Capture Input
    CLK => CLK  -- 1-bit input: Clock Input
);

-- End of CAPTURE_VIRTEX6_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// CAPTURE_VIRTEX6: Register Capture
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

CAPTURE_VIRTEX6 #(
    .ONESHOT("TRUE") // Specifies the procedure for performing single readback per CAP trigger.
)
CAPTURE_VIRTEX6_inst (
    .CAP(CAP), // 1-bit input: Capture Input
    .CLK(CLK)  // 1-bit input: Clock Input
);

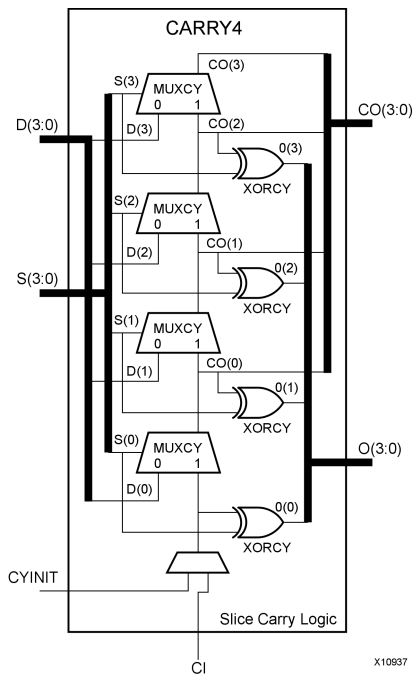
// End of CAPTURE_VIRTEX6_inst instantiation
```

## 詳細情報

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

## CARRY4

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



## 概要

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

## ポートの説明

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

## デザインの入力方法

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

## VHDL 記述 (インスタンス化)

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

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

-- CARRY4: Fast Carry Logic Component
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
```

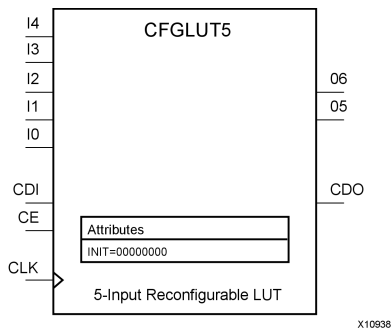
## 詳細情報

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



## CFGLUT5

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



### 概要

このデザイン エLEMENTは、ランタイムのダイナミック リコンフィギュレーションが可能な 5 入力ルックアップ テーブル (LUT) で、回路の動作中に LUT のロジック ファンクションを変更できます。CDI ピンを使用すると、クロックに同期して新しい INIT 値がシリアルにシフトされ、ロジック ファンクションが変更されます。O6 出力ピンでは、LUT に読み込まれた現在の INIT 値と現在選択されている I0 ~ I4 の入力ピン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には、1 つのスライスに含まれる 6 入力 LUT 4 個のうちの 1 つが含まれます。

このELEMENTをカスケード接続する場合は、CDO ピンを次のELEMENTの CDI ピンに接続することで、1 つのシリアルチェーンのデータ (LUT ごとに 32 ビット) で複数のELEMENTをリコンフィギュレーションできます。

### ポートの説明

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

## デザインの入力方法

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

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

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 に 32 ビットずつシフトさせることで、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 LUT に入力されるまで無視します。LUT のロジック ファンクションは、新しい INIT 値が LUT にシフトされるときに変化します。データは MSB (INIT[31]) から順に LSB (INIT[0]) までシフト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 または (I3、I2、I1、および I0)
- ・ O5 = I3、I2、I1、および 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 信号の論理値を 1 にします。INIT[31:16] が O6 出力の論理値に、INIT [15:0] の値が O5 出力の論理値に適用されます。

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

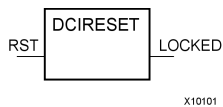
// End of CFGLUT5_inst instantiation
```

## 詳細情報

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

## DCIRESET

### プリミティブ：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;

-- DCIRESET: Digitally Controlled Impedance Reset Component
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of DCIRESET_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// DCIRESET: Digitally Controlled Impedence Reset Component
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

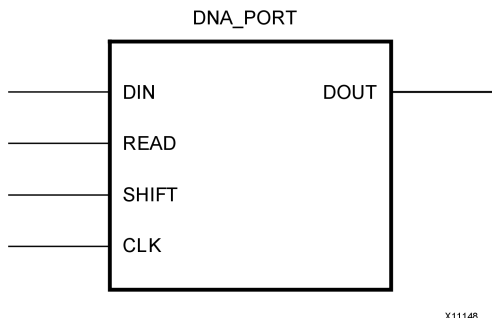
// End of DCIRESET_inst instantiation
```

## 詳細情報

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

## DNA\_PORT

### プリミティブ：Device DNA Data Access Port



### 概要

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

### ポートの説明

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

### デザインの入力方法

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

正しく動作するように、入力および出力をすべてデザインに接続します。

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- DNA_PORT: Device DNA Access Port
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of DNA_PORT_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// DNA_PORT: Device DNA Access Port
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

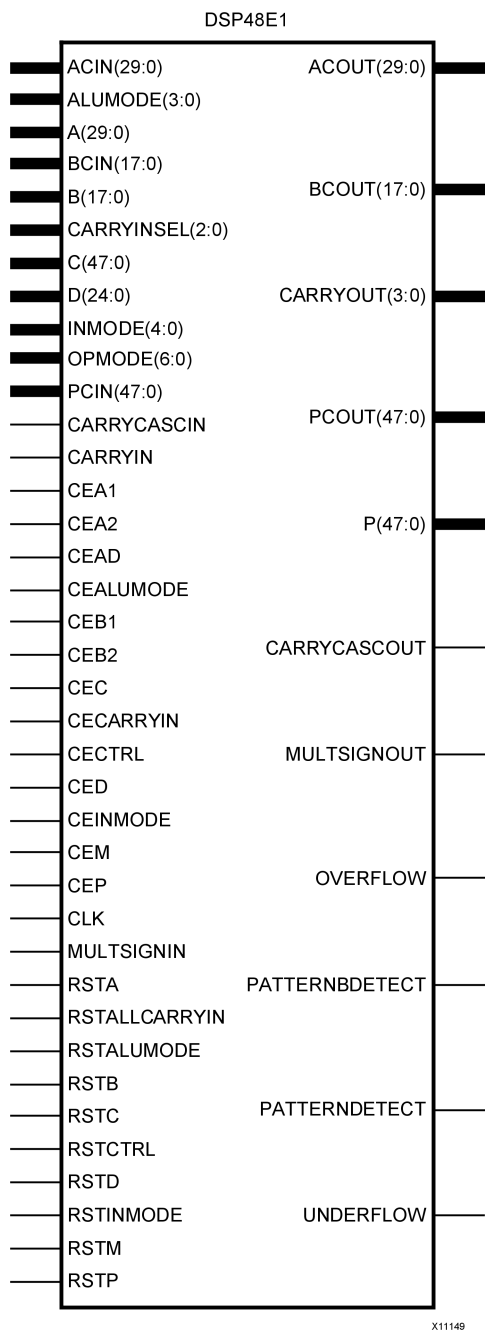
// End of DNA_PORT_inst instantiation
```

## 詳細情報

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

## DSP48E1

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



### 概要

このデザイン エレメントは、Virtex®-6 に含まれる柔軟性が高い多用途のハード IP ブロックで、多くの DSP アルゴリズムで見られる小型で高速な演算処理を作成できます。このブロックでは、乗算、加算（前置加算器を含む）、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
A[29:0]	入力	30	乗算器/前置加算器への 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
ACIN[29:0]	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 DSP スライスの ACOUT に接続します。使用しない場合は、ポートをすべてゼロにします。
ACOUT[29:0]	出力	30	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 DSP スライスの ACIN に接続します。使用しない場合は未接続にします。
ALUMODE[3:0]	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。
B[17:0]	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
BCIN[17:0]	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 DSP スライスの BCOUT に接続します。使用しない場合は、ポートをすべてゼロにします。
BCOUT[17:0]	出力	18	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 DSP スライスの BCIN に接続します。使用しない場合は未接続にします。
C[47:0]	入力	48	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
CARRYCASCIN	入力	1	上位 DSP スライスからのカスケード キャリー入力
CARRYCASCOUT	出力	1	下位 DSP スライスへのカスケード キャリー出力
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
CARRYINSEL[2:0]	入力	3	DSP スライスへのキャリー入力ソースを選択します。
CARRYOUT[3:0]	出力	4	演算処理 (加算、減算など) のキャリー出力信号。 <ul style="list-style-type: none"> <li>USE_SIMD が FOUR12 の場合、CARRYOUT は累積器/加算器/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 は累積器/加算器それぞれからの 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 は累積器/加算器/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>
CEAD	入力	1	前置加算器出力 AD パイプライン レジスタのアクティブ High のクロック イネーブルです。使用しない場合と ADREG=1 の場合は論理値を 1 に、ADREG=0 の場合は論理値を 0 にします。
CEALUMODE	入力	1	ALUMODE 入力レジスタ (ALUMODEREG=1) のアクティブ High のクロック イネーブルです。使用しない場合は論理値を 1 にします。
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、使用しない場合と AREG=2 の場合は論理値を 1 に、AREG=0 または 1 の場合は論理値を 0 にします。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。ダイナミック AB アクセスが使用されている場合、このクロック イネーブルは INMODE=1 に適用されます。
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、使用しない場合と AREG=2 の場合は論理値を 1 に、AREG=0 または 1 の場合は論理値を 0 にします。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。レジスタが使用されている場合 (AREG=1)、CEA2 がクロック イネーブルです。

ポート名	方向	幅	機能
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、使用しない場合と BREG=2 の場合は論理値を 1 に、BREG=0 または 1 の場合は論理値を 0 にします。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。ダイナミック AB アクセスが使用されている場合、このクロック イネーブルは INMODE=1 に適用されます。
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、使用しない場合と BREG=2 の場合は論理値を 1 に、BREG=0 または 1 の場合は論理値を 0 にします。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。レジスタが使用されている場合 (BREG=1)、CEB2 がクロック イネーブルです。
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値を 1 にします。
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値を 1 にします。
CECTRL	入力	1	OPMODE および CARRYINSEL レジスタのアクティブ High のクロック イネーブルで、使用しない場合は論理値を 1 にします。
CED	入力	1	D ポートレジスタ (DREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理値を 1 にします。
CEINMODE	入力	1	INMODE 入力レジスタ (INMODEREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理値を 1 にします。
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値を 1 にします。
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値を 1 にします。
CLK	入力	1	DSP スライスのクロック入力
D[24:0]	入力	25	前置加算器への 25 ビット データ入力、または乗算器への入力
INMODE[4:0]	入力	5	ALUMODE および OPMODE と共に使用する DSP スライスの演算処理を選択する制御入力です。INMODE 信号は乗算器 (前置加算器を含む) の前にある信号およびブロックの機能を制御します。
MULTSIGNIN	入力	1	カスケードされた上位 DSP スライスからの乗算器符号入力。48 ビットを超える出力が必要な場合に加算器/累積器 (MACC) の出力の符号拡張のために使用します。MULTSIGNOUT 出力ピンにのみ接続します。
MULTSIGNOUT	出力	1	カスケードされた下位 DSP スライスへの乗算器符号出力。48 ビットを超える出力が必要な場合に加算器/累積器 (MACC) の出力の符号拡張のために使用します。MULTSIGNIN 入力ピンにのみ接続します。
OPMODE[6:0]	入力	7	ALUMODE および INMODE と共に使用する DSP スライスの演算処理を選択する制御入力です。
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。
P[47:0]	出力	48	プライマリ データ出力
PATTERNBDETECT	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。
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。

ポート名	方向	幅	機能
PCIN[47:0]	入力	48	ポート P のカスケード入力で、カスケード接続されている上位の DSP スライスの PCOUT に接続します。使用しない場合は、ポートをすべてゼロにします。
PCOUT[47:0]	出力	48	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 DSP スライスの PCIN に接続します。使用しない場合は未接続にします。
RSTA	入力	1	A ポートレジスタ (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値を 0 にします。
RSTALLCARRYIN	入力	1	全キャリー入力レジスタ (CARRYINREG=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値を 0 にします。
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値を 0 にします。
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値を 0 にします。
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値を 0 にします。
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値を 0 にします。
RSTD	入力	1	D ポートレジスタ (DREG=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値を 0 にします。
RSTINMODE	入力	1	INMODE レジスタ (INMODEREG=1) のアクティブ High の同期リセットで、使用しない場合は論理値を 0 にします。
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値を 0 にします。
RSTP	入力	1	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値を 0 にします。
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。

## 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	可

## 使用可能な属性

属性	データ型	値	デフォルト	説明
A_INPUT	文字列	“DIRECT”、 “CASCADE”	“DIRECT”	A または ACIN のいずれかを選択します。
ACASCREG	整数	1、0、2	1	AREG 属性と組み合わせて使用し、A カスケード ACOUT 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。

属性	データ型	値	デフォルト	説明
ADREG	整数	1、0	1	前置加算器出力 (AD) パイプライン レジスタの使用を選択します。AD パイプライン レジスタを使用する場合は 1 に設定します。
ALUMODEREG	整数	1、0	1	ALUMODE 入力をレジスタに格納する場合は 1 に設定します。
AREG	整数	1、0、2	1	A 入力のパイプラインの段数を指定します。
AUTORESET_PATDET	文字列	“NO_RESET”、 “RESET_MATCH”、 “RESET_NOT_MATCH”	“NO_RESET”	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、DSP スライス の R 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。RESET_MATCH および RESET_NOT_MATCH 設定で、パターンが一致したとき、またはパターンが現在のサイクルで一致しないが前のサイクルでは一致していたとき、DSP スライスで 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べきかを決定します。
B_INPUT	文字列	“DIRECT”、 “CASCADE”	“DIRECT”	B または BCIN のいずれかを選択します。
BCASCREG	整数	1、0、2	1	BREG 属性と組み合わせて使用し、B カスケード BCOUT の B 入力レジスタの数を指定します。ただし、BREG の値以下にする必要があります。
BREG	整数	1、0、2	1	B 入力のパイプラインの段数を指定します。
CARRYINREG	整数	1、0	1	CARRYIN 入力をレジスタに格納する場合は 1 に設定します。
CARRYINSELREG	整数	1、0	1	CARRYINSEL 入力をレジスタに格納する場合は 1 に設定します。
CREG	整数	1、0	1	C 入力のパイプラインの段数を指定します。
DREG	整数	1、0	1	D 入力のパイプラインの段数を指定します。
INMODEREG	整数	1、0	1	INMODE 入力をレジスタに格納する場合は 1 に設定します。
MASK	16 進数	48'h000000 000000 ~ 48'hffffffff	48'h3fff ffffff	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	1、0	1	乗算器出力 (M) パイプライン レジスタの使用を選択します。使用する場合は 1 に設定します。
OPMODEREG	整数	1、0	1	OPMODE 入力をレジスタに格納する場合は 1 に設定します。
PATTERN	16 進数	48'h00000000 000000 ~ 48'hffffffff	すべてゼロ	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	1、0	1	P 出力をレジスタに格納する場合は 1 に設定します。レジスタを介す出力には、CARRYOUT、CARRYCASCOUT、MULTSIGNOUT、PATTERNB_DETECT、PATTERN_DETECT、PCOUT が含まれます。

属性	データ型	値	デフォルト	説明
SEL_MASK	文字列	“MASK”、“C”、 “ROUNDING_	“MASK”	パターン検出器で使用するマスクを指定します。C および MASK はパターン検出器で標準的に使用される値です (カウンタ、オーバーフロー検出など)。ROUNDING_MODE1 (C パー 1 つ左にシフト) および ROUNDING_MODE2 (C パー 2 つ左にシフト) は、オプションでレジスタを付けた C ポートに基づき、特別マスクを選択します。これらの丸めモードは、Virtex-6 FPGA DSP48E1 ブロック ユーザー ガイドにあるようにパターン検出器を使用して DSP スライスでの収束丸め機能をインプリメントするのに使用します。
SEL_PATTERN	文字列	“PATTERN”、“C”	“PATTERN”	パターン検出器で使用するパターンを指定します。
USE_DPORT	ブール 代数	FALSE、TRUE	FALSE	前置加算器および D ポートの使用を選択します。
USE_MULT	文字列	“MULTIPLY”、 “DYNAMIC”、 “NONE”	“MULTIPLY”	乗算器の使用方法を選択します。NONE に設定すると、加算器/ロジックユニットのみを使用するときに電力を節約できます。DYNAMIC は、ユーザーが A*B と A:B をダイナミックに切り替えていて、この 2 つのパスのワーストケース タイミングを取得することが必要であることを示します。
USE_PATTERN_DETECT	文字列	“NO_PATDET”、 “PATDET”	“NO_PATDET”	PATDET を設定するとパターン検出器がシミュレーション モデルおよびスピード ファイルで有効になります。
USE_SIMD	文字列	“ONE48”、 “FOUR12”、 “TWO24”	“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。

## 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;

-- DSP48E1: 48-bit Multi-Functional Arithmetic Block
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

```

```

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

```



```

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

-- End of DSP48E1_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// DSP48E1: 48-bit Multi-Functional Arithmetic Block
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

```

```

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

// End of DSP48E1_inst instantiation

```

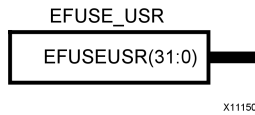
## 詳細情報

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



## EFUSE\_USR

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



### 概要

デザインに固有のビット (各デザインに関連した ID など) を格納可能な 32 個の不揮発ヒューズに JTAG を介して内部アクセスします。

### ポートの説明

ポート名	方向	幅	機能
EFUSEUSR[31:0]	出力	32	ユーザー E ヒューズ レジスタの値

### デザインの入力方法

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

### 使用可能な属性

属性	データ型	値	デフォルト	説明
SIM_EFUSE_VALUE	16 進数	32'h00000000 ~ 32'hffffff	32'h00000000	シミュレーションで使用される 32 ビットの非揮発性デザイン ID の値

### VHDL 記述 (インスタンス化)

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

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

-- EFUSE_USR: 32-bit non-volatile design ID
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of EFUSE_USR_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// EFUSE_USR: 32-bit non-volatile design ID
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

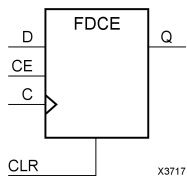
// End of EFUSE_USR_inst instantiation
```

## 詳細情報

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

## FDCE

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



## 概要

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

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

## 論理表

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

## デザインの入力方法

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

## 使用可能な属性

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

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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).
//       Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

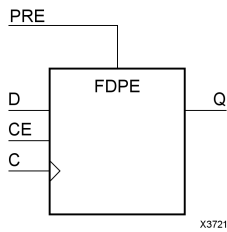
// End of FDCE_inst instantiation
```

## 詳細情報

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

## FDPE

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



## 概要

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

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

## 論理表

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

## デザインの入力方法

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

## 使用可能な属性

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

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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).
//       Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

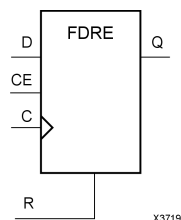
// End of FDPE_inst instantiation
```

## 詳細情報

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

## FDRE

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



### 概要

このデザイン エLEMENTは、データ (D)、クロック イネーブル (CE)、同期リセット (R) の各入力とデータ出力 (Q) がある単一の D タイプ フリップフロップです。同期リセット入力 (R) が High になると、ほかの入力は無視され、クロック (C) が Low から High に切り替わる時に出力 (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。

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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).
//       Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
```

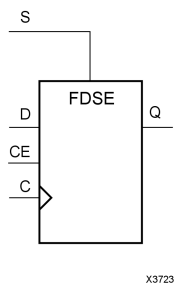
## 詳細情報

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



## FDSE

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



### 概要

FDSE は、データ (D)、クロック イネーブル (CE)、同期セット (S) の各入力とデータ出力 (Q) がある単一の D タイプ フリップフロップです。同期セット (S) 入力が高になると、クロック イネーブル (CE) 入力は無視され、クロック (C) が Low から High に切り替わるときに Q 出力が高にセットされます。S が Low、CE が High の場合、クロック (C) が Low から High に切り替わるときに 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、出力が高になります。グローバル セット/リセット (GSR) をアクティブにすると、電源投入時の状態をシミュレーションできます。GSR のデフォルトはアクティブ High ですが、STARTUP\_architecture シンボルの GSR 入力の前にインバーターを追加するとアクティブ Low にできます。

### 論理表

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

### デザインの入力方法

インスタンシエーション	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	1	コンフィギュレーション後の Q 出力の初期値を指定。

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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).
//       Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

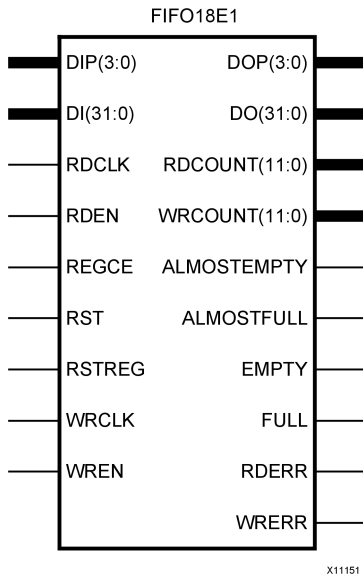
// End of FDSE_inst instantiation
```

## 詳細情報

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

## FIFO18E1

**プリミティブ：18 k-bit FIFO (First In, First Out) Block RAM Memory**



### 概要

Virtex®-6 デバイスにはブロック RAM が数個含まれ、これらの RAM を個別に FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36Kb または 18Kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。FIFO18E1 では、FIFO 制御ロジックおよび 18Kb ブロック RAM が使用されます。このプリミティブは、4 ビット X 4K、9 ビット X 2K、18 ビット X 1K、または 36 ビット X 512 コンフィギュレーションで使用できます。また、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、同期モードまたはデュアル 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。独立したクロックでデュアル クロック モードを使用する場合、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。

**メモ：** 36 ビット X 512 ワードの FIFO には、FIFO18\_36 を使用する必要があります。ワード数が多く、データ幅の広いコンフィギュレーションには、FIFO36E1 を使用できます。ただし、エラー修正回路が必用な場合は、FIFO36\_72 の FIFO36E1 を使用する必要があります。

### ポートの説明

ポート名	方向	幅	機能
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示します。このフラグをトリガーする位置は ALMOST_EMPTY_OFFSET 属性で指定します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示します。このフラグをトリガーする位置は ALMOST_FULL_OFFSET 属性で指定します。
DI[31:0]	入力	32	FIFO データ入力バス
DIP[3:0]	入力	4	FIFO パリティ データ入力バス
DO[31:0]	出力	32	FIFO データ出力バス

ポート名	方向	幅	機能
DOP[3:0]	出力	4	FIFO パリティ データ出力バス
EMPTY	出力	1	FIFO が空であることを示します。
FULL	出力	1	FIFO がフルであることを示します。
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
REGCE	入力	1	パイプライン化された同期 FIFO の出力レジスタ クロック イネーブル
RST	入力	1	3 クロック サイクル間アクティブ High の (FIFO ロジック) の非同期リセット (デュアル レートの FIFO 向け)、同期リセット (同期 FIFO)
RSTREG	入力	1	出力レジスタの同期セット/リセット
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)
WRCOUNT、RDCOUNT	出力	12	FIFO 書き込み/読み出しポインタ
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
WRERR、RDERR	出力	1	<ul style="list-style-type: none"> <li>WRERR は FIFO がフルの間に書き込みが実行されたことを示します。</li> <li>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。</li> </ul>

## デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	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'h0000 ~ 13'h8191	13'h0080	ALMOST_EMPTY フラグをトリガーする RAM のデータ量を指定します。
ALMOST_FULL_OFFSET	16 進数	13'h0000 ~ 13'h8191	13'h0080	ALMOST_FULL フラグをトリガーする RAM のデータ量を指定します。
DATA_WIDTH	整数	4、9、18、36	4	FIFO に必要なデータ幅を指定
DO_REG	整数	1、0	1	EN_SYN のデータ バイプライン レジスタ
EN_SYN	ブール代数	FALSE、TRUE	FALSE	FIFO がデュアル クロック (独立した 2 つのクロック) または同期 (1 つのクロック) のいずれかで動作しているかを示します。デュアル クロックの場合は DO_REG=1 である必要があります。
FIFO_MODE	文字列	"FIFO18"、 "FIFO18_36"	"FIFO18"	FIFO18 または FIFO18_36 モードを選択します。

属性	データ型	値	デフォルト	説明
FIRST_WORD_FALL_THROUGH	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、RDEN がアサートされずに FIFO に最初に書き込まれた値が DO に出力されます。
INIT	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後の DO 出力の初期値を指定します。
SRVAL	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの FIFO の出力値を指定します。DO_REG=1 の場合のみ有効です。

## VHDL 記述 (インスタンスエーション)

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;

-- FIFO18E1: 18KB FIFO (First In, First Out) Block RAM Memory
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

FIFO18E1_inst : FIFO18E1
generic map (
    ALMOST_EMPTY_OFFSET => X"0080",    -- Sets the almost empty threshold
    ALMOST_FULL_OFFSET  => X"0080",    -- Sets almost full threshold
    DATA_WIDTH => 4,                  -- Sets data width to 4, 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 dual-clock (FALSE) or Synchronous (TRUE)
    FIFO_MODE => "FIFO18",            -- Sets mode to FIFO18 or FIFO18_36
    FIRST_WORD_FALL_THROUGH => FALSE,  -- Sets the FIFO FWFT to TRUE or FALSE
    INIT => X"0000000000",            -- Initial values on output port
    SRVAL => X"0000000000"            -- Set/Reset value for output port
)
port map (
    -- Read Data: 32-bit (each) output: Read output data
    DO => DO,                        -- 32-bit output: data output
    DOP => DOP,                      -- 4-bit output: parity data output
    -- Status: 1-bit (each) output: Flags and other FIFO status outputs
    ALMOSTEMPTY => ALMOSTEMPTY,      -- 1-bit output: almost empty output flag
    ALMOSTFULL => ALMOSTFULL,        -- 1-bit output: almost full output flag
    EMPTY => EMPTY,                  -- 1-bit output: empty output flag
    FULL => FULL,                    -- 1-bit output: full output flag
    RDCOUNT => RDCOUNT,              -- 12-bit output: read count output
    RDERR => RDERR,                  -- 1-bit output: read error output
    WRCOUNT => WRCOUNT,              -- 12-bit output: write count output
    WRERR => WRERR,                  -- 1-bit output: write error
    -- Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
    RDCLK => RDCLK,                  -- 1-bit input: read clock input
    RDEN => RDEN,                    -- 1-bit input: read enable input
    REGCE => REGCE,                  -- 1-bit input: clock enable input
    RST => RST,                      -- 1-bit input: reset input
    RSTREG => RSTREG,                -- 1-bit input: output register set/reset
    -- Write Control Signals: 1-bit (each) input: Write clock and enable input signals
    WRCLK => WRCLK,                  -- 1-bit input: write clock input
    WREN => WREN,                    -- 1-bit input: write enable input
    -- Write Data: 32-bit (each) input: Write input data
    DI => DI,                        -- 32-bit input: data input
    DIP => DIP                        -- 4-bit input: parity input
);

-- End of FIFO18E1_inst instantiation

```

## Verilog 記述 (インスタンス化)

```
// FIFO18E1: 18KB FIFO (First In, First Out) Block RAM Memory
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

FIFO18E1 #(
    .ALMOST_EMPTY_OFFSET(13'h0080),    // Sets the almost empty threshold
    .ALMOST_FULL_OFFSET(13'h0080),     // Sets almost full threshold
    .DATA_WIDTH(4),                    // Sets data width to 4, 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 dual-clock (FALSE) or Synchronous (TRUE)
    .FIFO_MODE("FIFO18"),              // Sets mode to FIFO18 or FIFO18_36
    .FIRST_WORD_FALL_THROUGH("FALSE"), // Sets the FIFO FWFT to TRUE or FALSE
    .INIT(36'h00000000),               // Initial values on output port
    .SRVAL(36'h00000000)               // Set/Reset value for output port
)
FIFO18E1_inst (
    // Read Data: 32-bit (each) output: Read output data
    .DO(DO),                          // 32-bit output: data output
    .DOP(DOP),                        // 4-bit output: parity data output
    // Status: 1-bit (each) output: Flags and other FIFO status outputs
    .ALMOSTEMPTY(ALMOSTEMPTY),        // 1-bit output: almost empty output flag
    .ALMOSTFULL(ALMOSTFULL),          // 1-bit output: almost full output flag
    .EMPTY(EMPTY),                    // 1-bit output: empty output flag
    .FULL(FULL),                      // 1-bit output: full output flag
    .RDCOUNT(RDCOUNT),                // 12-bit output: read count output
    .RDERR(RDERR),                    // 1-bit output: read error output
    .WRCOUNT(WRCOUNT),                // 12-bit output: write count output
    .WRERR(WRERR),                    // 1-bit output: write error
    // Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
    .RDCLK(RDCLK),                    // 1-bit input: read clock input
    .RDEN(RDEN),                      // 1-bit input: read enable input
    .REGCE(REGCE),                    // 1-bit input: clock enable input
    .RST(RST),                        // 1-bit input: reset input
    .RSTREG(RSTREG),                  // 1-bit input: output register set/reset
    // Write Control Signals: 1-bit (each) input: Write clock and enable input signals
    .WRCLK(WRCLK),                    // 1-bit input: write clock input
    .WREN(WREN),                      // 1-bit input: write enable input
    // Write Data: 32-bit (each) input: Write input data
    .DI(DI),                          // 32-bit input: data input
    .DIP(DIP)                         // 4-bit input: parity input
);

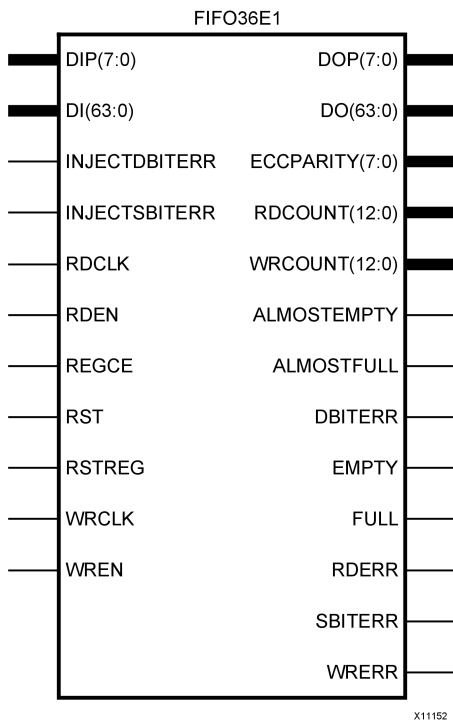
// End of FIFO18E1_inst instantiation
```

## 詳細情報

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

## FIFO36E1

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



### 概要

Virtex®-6 デバイスにはブロック RAM が数個含まれ、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36 Kb または 18 Kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。FIFO36E1 を使用すると、36Kb の FIFO のブロック RAM へアクセスできます。このコンポーネントは、関連 FIFO フラグと共に、4 ビット X 8K ワード、9 ビット X 4K ワード、18 ビット X 2K ワード、36 ビット X 1K ワード、72 ビット X 512 ワードの同期またはデュアル クロック (非同期) FIFO RAM としてコンフィギュレーションできます。

独立したクロックでデュアル クロック モードを使用する場合、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。

**メモ：** 72 ビット X 512 ワードの FIFO には、FIFO36\_72 を使用する必要があります。ワード数が少なく、データ幅の狭いコンフィギュレーションには、FIFO18E1 を使用します。ただし、エラー修正回路が必用な場合は、FIFO36\_72 を使用する必要があります。

## ポートの説明

ポート名	方向	幅	機能
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示します。このフラグをトリガーする位置は ALMOST_EMPTY_OFFSET 属性で指定します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示します。このフラグをトリガーする位置は ALMOST_FULL_OFFSET 属性で指定します。
DBITERR	出力	1	ダブル ビット エラーが検出されたことを示す ECC ファンクションからのステータス出力。使用する場合は、EN_ECC_READ を TRUE にする必要があります。
DI[63:0]	入力	64	FIFO データ入力バス
DIP[7:0]	入力	8	FIFO パリティ データ入力バス
DO[63:0]	出力	64	FIFO データ出力バス
DOP[7:0]	出力	8	FIFO パリティ データ出力バス
ECCPARITY[7:0]	出力	8	メモリ エラー検出と訂正を行う ECC デコーダーで使用される ECC エンコーダーから生成された 8 ビット データ
EMPTY	出力	1	FIFO が空であることを示します。
FULL	出力	1	FIFO がフルであることを示します。
INJECTDBITERR	入力	1	ECC 機能を使用されている場合はダブル ビット エラーが挿入されます。
INJECTSBITERR	入力	1	ECC 機能を使用されている場合はシングル ビット エラーが挿入されます。
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
REGCE	入力	1	パイプライン化された同期 FIFO の出力レジスタ クロック イネーブル
RST	入力	1	3 クロック サイクル間アクティブ High の (FIFO ロジック) の非同期リセット (デュアル レートの FIFO 向け)、同期リセット (同期 FIFO)
RSTREG	入力	1	出力レジスタの同期セット/リセット
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。
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)
WRCOUNT、 RDCOUNT	出力	13	FIFO 書き込み/読み出しポインタ
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
WRERR、 RDERR	出力	1	<ul style="list-style-type: none"> <li>WRERR は FIFO がフルの間に書き込みが実行されたことを示します。</li> <li>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。</li> </ul>



## デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	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'h0000 ~ 13'h8191	13'h0080	ALMOST_EMPTY フラグをトリガーする RAM のデータ量を指定します。
ALMOST_FULL_OFFSET	16 進数	13'h0000 ~ 13'h8191	13'h0080	ALMOST_FULL フラグをトリガーする RAM のデータ量を指定します。
DATA_WIDTH	整数	4、9、18、36、72	4	FIFO に必要なデータ幅を指定
DO_REG	整数	1、0	1	読み出しレイテンシ (パイプライン遅延 1 つ) 追加することで clock-to-out のタイミングを向上するように、FIFO の出力レジスタをイネーブルします。EN_SYN が FALSE のときは DO_REG を 1 にする必要があります。
EN_ECC_READ	ブール代数	FALSE、TRUE	FALSE	ECC デコーダー回路をイネーブルにします。
EN_ECC_WRITE	ブール代数	FALSE、TRUE	FALSE	ECC エンコーダー回路をイネーブルにします。
EN_SYN	ブール代数	FALSE、TRUE	FALSE	FALSE のときは非同期モード、TRUE のときは同期 (1 クロック) モードで FIFO が使用されることを示します。
FIFO_MODE	文字列	“FIFO36”、 “FIFO36_72”	“FIFO36”	FIFO36 または FIFO36_72 モードを選択します。
FIRST_WORD_FALL_THROUGH	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、RDEN がアサートされずに FIFO に最初書き込まれた値が DO に出力されます。
INIT	16 進数	72 ビット値	すべてゼロ	コンフィギュレーション後の DO 出力の初期値を指定します。
SRVAL	16 進数	72 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの FIFO の出力値を指定します。DO_REG=1 の場合のみ有効です。

## VHDL 記述 (インスタンス化)

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

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

-- FIFO36E1: 36KB FIFO (First In, First Out) Block RAM Memory
--      Virtex-6
--      Xilinx HDL Libraries Guide, version 13.3

FIFO36E1_inst : FIFO36E1
generic map (
    ALMOST_EMPTY_OFFSET => X"0080",    -- Sets the almost empty threshold
    ALMOST_FULL_OFFSET  => X"0080",    -- Sets almost full threshold
    DATA_WIDTH => 4,                  -- Sets data width to 4, 9, 18, 36, or 72
    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)
    FIFO_MODE => "FIFO36",             -- Sets mode to FIFO36 or FIFO36_72
    FIRST_WORD_FALL_THROUGH => FALSE,  -- Sets the FIFO FWFT to TRUE or FALSE
    INIT => X"00000000000000000000",    -- Initial values on output port
    SRVAL => X"00000000000000000000"    -- Set/Reset value for output port
)
port map (
    -- ECC Signals: 1-bit (each) output: Error Correction Circuitry ports
    DBITERR => DBITERR,                -- 1-bit output: double bit error status output
    ECCPARITY => ECCPARITY,             -- 8-bit output: generated error correction parity
    SBITERR => SBITERR,                -- 1-bit output: single bit error status output
    -- Read Data: 64-bit (each) output: Read output data
    DO => DO,                          -- 64-bit output: data output
    DOP => DOP,                        -- 8-bit output: parity data output
    -- Status: 1-bit (each) output: Flags and other FIFO status outputs
    ALMOSTEMPTY => ALMOSTEMPTY,         -- 1-bit output: almost empty output flag
    ALMOSTFULL => ALMOSTFULL,           -- 1-bit output: almost full output flag
    EMPTY => EMPTY,                    -- 1-bit output: empty output flag
    FULL => FULL,                      -- 1-bit output: full output flag
    RDCOUNT => RDCOUNT,                 -- 13-bit output: read count output
    RDERR => RDERR,                    -- 1-bit output: read error output
    WRCOUNT => WRCOUNT,                 -- 13-bit output: write count output
    WRERR => WRERR,                    -- 1-bit output: write error
    -- ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
    INJECTDBITERR => INJECTDBITERR,    -- 1-bit input: Inject a double bit error
    INJECTSBITERR => INJECTSBITERR,
    -- Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
    RDCLK => RDCLK,                    -- 1-bit input: read clock input
    RDEN => RDEN,                     -- 1-bit input: read enable input
    REGCE => REGCE,                   -- 1-bit input: clock enable input
    RST => RST,                       -- 1-bit input: reset input
    RSTREG => RSTREG,                  -- 1-bit input: output register set/reset
    -- Write Control Signals: 1-bit (each) input: Write clock and enable input signals
    WRCLK => WRCLK,                   -- 1-bit input: write clock input
    WREN => WREN,                     -- 1-bit input: write enable input
    -- Write Data: 64-bit (each) input: Write input data
    DI => DI,                         -- 64-bit input: data input
    DIP => DIP,                       -- 8-bit input: parity input
);

-- End of FIFO36E1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// FIFO36E1: 36KB FIFO (First In, First Out) Block RAM Memory
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

FIFO36E1 #(
    .ALMOST_EMPTY_OFFSET(13'h0080),    // Sets the almost empty threshold
    .ALMOST_FULL_OFFSET(13'h0080),     // Sets almost full threshold
    .DATA_WIDTH(4),                    // Sets data width to 4, 9, 18, 36, or 72
    .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)
    .FIFO_MODE("FIFO36"),              // Sets mode to FIFO36 or FIFO36_72
    .FIRST_WORD_FALL_THROUGH("FALSE"), // Sets the FIFO FWFT to TRUE or FALSE
    .INIT(72'h00000000000000000000),   // Initial values on output port
    .SRVAL(72'h00000000000000000000)   // Set/Reset value for output port
)
FIFO36E1_inst (
    // ECC Signals: 1-bit (each) output: Error Correction Circuitry ports
    .DBITERR(DBITERR),                // 1-bit output: double bit error status output
    .ECCPARITY(ECCPARITY),            // 8-bit output: generated error correction parity
    .SBITERR(SBITERR),                // 1-bit output: single bit error status output
    // Read Data: 64-bit (each) output: Read output data
    .DO(DO),                          // 64-bit output: data output
    .DOP(DOP),                        // 8-bit output: parity data output
    // Status: 1-bit (each) output: Flags and other FIFO status outputs
    .ALMOSTEMPTY(ALMOSTEMPTY),        // 1-bit output: almost empty output flag
    .ALMOSTFULL(ALMOSTFULL),          // 1-bit output: almost full output flag
    .EMPTY(EMPTY),                    // 1-bit output: empty output flag
    .FULL(FULL),                      // 1-bit output: full output flag
    .RDCOUNT(RDCOUNT),                // 13-bit output: read count output
    .RDERR(RDERR),                    // 1-bit output: read error output
    .WRCOUNT(WRCOUNT),                // 13-bit output: write count output
    .WRERR(WRERR),                    // 1-bit output: write error
    // ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
    .INJECTDBITERR(INJECTDBITERR),    // 1-bit input: Inject a double bit error
    .INJECTSBITERR(INJECTSBITERR),
    // Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
    .RDCLK(RDCLK),                    // 1-bit input: read clock input
    .RDEN(RDEN),                      // 1-bit input: read enable input
    .REGCE(REGCE),                    // 1-bit input: clock enable input
    .RST(RST),                        // 1-bit input: reset input
    .RSTREG(RSTREG),                  // 1-bit input: output register set/reset
    // Write Control Signals: 1-bit (each) input: Write clock and enable input signals
    .WRCLK(WRCLK),                    // 1-bit input: write clock input
    .WREN(WREN),                      // 1-bit input: write enable input
    // Write Data: 64-bit (each) input: Write input data
    .DI(DI),                          // 64-bit input: data input
    .DIP(DIP)                         // 8-bit input: parity input
);

// End of FIFO36E1_inst instantiation
```

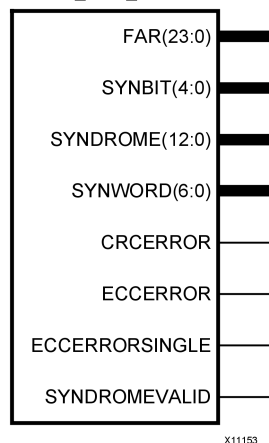
## 詳細情報

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

## FRAME\_ECC\_VIRTEX6

### プリミティブ：Virtex®-6 Configuration Frame Error Detection and Correction Circuitry

FRAME\_ECC\_VIRTEX6



### 概要

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

SEU 修正機能では、シングル ビット エラーの自動修正を行うためのハードウェア バージョンが提供されます。この修正機能で使用する追加出力には、ソフト コアで使用するハミング コード シンドロームのデコードが含まれます。

### ポートの説明

ポート名	方向	幅	機能
CRCERROR	出力	1	CRC エラーを示す出力
ECCERROR	出力	1	ECC エラーを示す出力
ECCERRORSINGLE	出力	1	シングル ビット フレーム ECC エラーが検出されたことを示します。
FAR[23:0]	出力	24	フレーム アドレス レジスタの値
SYNBIT[4:0]	出力	5	エラーのビット アドレス
SYNDROME[12:0]	出力	13	エラー ビットの出力ロケーション
SYNDROMEVALID	出力	1	SYNDROME 出力が有効であることを示す フレーム ECC 出力
SYNWORD[6:0]	出力	7	ECC エラーが検出されたフレーム内のワード

### デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
FARSRC	文字列	“EFAR”、“FAR”	“EFAR”	EFAR FAR[23:0] コンフィギュレーションレジスタの出力先を FAR または EFAR のどちらにするか決定します。コンフィギュレーション オプションレジスタ ビット CTL0[7] を設定します。
FRAME_RBT_IN_FILENAME	文字列	ファイル名およびそのロケーションを示す文字列	なし	このファイルは、ICAP_VIRTEX6 モデルにより出力され、RBT ファイルのフレーム データ情報が含まれています。FRAME_ECC モデルではこのファイルを解析して ECC が算出され、エラーがある場合は出力されます。

## VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- FRAME_ECC_VIRTEX6: Configuration Frame Error Correction
--                               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

FRAME_ECC_VIRTEX6_inst : FRAME_ECC_VIRTEX6
generic map (
    FARSRC => "EFAR",                -- Determines if the output of FAR[23:0] configuration register points
                                     -- to the FAR or EFAR. Sets configuration option register bit CTL0[7].
    FRAME_RBT_IN_FILENAME => "NONE" -- This file is output by the ICAP_VIRTEX6 model and it contains Frame
                                     -- Data information for the Raw Bitstream (RBT) file. The FRAME_ECC
                                     -- model will parse this file, calculate ECC and output any error
                                     -- conditions.
)
port map (
    CRCERROR => CRCERROR,            -- 1-bit output: Output indicating a CRC error
    ECCERROR => ECCERROR,            -- 1-bit output: Output indicating an ECC error
    ECCERRORSINGLE => ECCERRORSINGLE, -- 1-bit output: Output Indicating single-bit Frame ECC error detected.
    FAR => FAR,                      -- 24-bit output: Frame Address Register Value output
    SYNBIT => SYNBIT,                -- 5-bit output: Output bit address of error
    SYNDROME => SYNDROME,            -- 13-bit output: Output location of erroneous bit
    SYNDROMEVALID => SYNDROMEVALID, -- 1-bit output: Frame ECC output indicating the SYNDROME output is
                                     -- valid.

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

-- End of FRAME_ECC_VIRTEX6_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// FRAME_ECC_VIRTEX6: Configuration Frame Error Correction
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

FRAME_ECC_VIRTEX6 #(
    .FAR_SRC("EFAR"), // Determines if the output of FAR[23:0] configuration register points to
                      // the FAR or EFAR. Sets configuration option register bit CTL0[7].
    .FRAME_RBT_IN_FILENAME("NONE") // This file is output by the ICAP_VIRTEX6 model and it contains Frame
                                   // Data information for the Raw Bitstream (RBT) file. The FRAME_ECC model
                                   // will parse this file, calculate ECC and output any error conditions.
)
FRAME_ECC_VIRTEX6_inst (
    .CRC_ERROR(CRC_ERROR), // 1-bit output: Output indicating a CRC error
    .ECC_ERROR(ECC_ERROR), // 1-bit output: Output indicating an ECC error
    .ECC_ERROR_SINGLE(ECC_ERROR_SINGLE), // 1-bit output: Output Indicating single-bit Frame ECC error detected.
    .FAR(FAR), // 24-bit output: Frame Address Register Value output
    .SYN_BIT(SYN_BIT), // 5-bit output: Output bit address of error
    .SYNDROME(SYNDROME), // 13-bit output: Output location of erroneous bit
    .SYNDROME_VALID(SYNDROME_VALID), // 1-bit output: Frame ECC output indicating the SYNDROME output is
                                   // valid.

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

// End of FRAME_ECC_VIRTEX6_inst instantiation
```

## 詳細情報

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

## GTHE1\_QUAD

プリミティブ : Gigabit Transceiver

DADDR(15:0)	GTHE1_QUAD	DRPDO(15:0)
DI(15:0)		
MGMTPCSSELANESEL(3:0)	MGMTPCSRRDDATA(15:0)	
MGMTPCSMMDDADDR(4:0)		
MGMTPCSREGADDR(15:0)	RXCODEERR0(7:0)	
MGMTPCSRRDATA(15:0)	RXCODEERR1(7:0)	
PLLPCCLKDIV(5:0)		
PLLREFCLKSEL(2:0)	RXCODEERR2(7:0)	
RXPPOWERDOWN0(1:0)	RXCODEERR3(7:0)	
RXPPOWERDOWN1(1:0)		
RXPPOWERDOWN2(1:0)		
RXPPOWERDOWN3(1:0)		
RXRATE0(1:0)	RXCTRL0(7:0)	
RXRATE1(1:0)		
RXRATE2(1:0)	RXCTRL1(7:0)	
RXRATE3(1:0)		
SAMPLERATE0(2:0)	RXCTRL2(7:0)	
SAMPLERATE1(2:0)		
SAMPLERATE2(2:0)	RXCTRL3(7:0)	
SAMPLERATE3(2:0)		
TXCTRL0(7:0)	RXDATA0(63:0)	
TXCTRL1(7:0)		
TXCTRL2(7:0)	RXDATA1(63:0)	
TXCTRL3(7:0)		
TXDATA0(63:0)	RXDATA2(63:0)	
TXDATA1(63:0)		
TXDATA2(63:0)	RXDATA3(63:0)	
TXDATA3(63:0)		
TXDATAAMSB0(7:0)	RXDISPERR0(7:0)	
TXDATAAMSB1(7:0)		
TXDATAAMSB2(7:0)	RXDISPERR1(7:0)	
TXDATAAMSB3(7:0)		
TXMARGIN0(2:0)	RXDISPERR2(7:0)	
TXMARGIN1(2:0)		
TXMARGIN2(2:0)	RXDISPERR3(7:0)	
TXMARGIN3(2:0)		
TXPOWERDOWN0(1:0)	RXVALID0(7:0)	
TXPOWERDOWN1(1:0)		
TXPOWERDOWN2(1:0)	RXVALID1(7:0)	
TXPOWERDOWN3(1:0)		
TXRATE0(1:0)	RXVALID2(7:0)	
TXRATE1(1:0)		
TXRATE2(1:0)	RXVALID3(7:0)	
TXRATE3(1:0)		
DCLK	DRDY	
DEN		
DFETRAINCTRL0	GTINITDONE	
DFETRAINCTRL1		
DFETRAINCTRL2	MGMTPCSRRDACK	
DFETRAINCTRL3		
DISABLEDRP	RXCTRLACK0	
DWE		
GTHINIT	RXCTRLACK1	
GTHRESET		
GTHX2LANE01	RXCTRLACK2	
GTHX2LANE23		
GTHX4LANE	RXCTRLACK3	
MGMTPCSREGRD		
MGMTPCSREGWR	RXUSERCLKOUT0	
POWERDOWN0		
POWERDOWN1	RXUSERCLKOUT1	
POWERDOWN2		
POWERDOWN3	RXUSERCLKOUT2	
REFCLK		
RXBUFRESET0	RXUSERCLKOUT3	
RXBUFRESET1		
RXBUFRESET2	TSTPATH	
RXBUFRESET3		
RXENCOMMADET0	TSTREFCLKFAB	
RXENCOMMADET1		
RXENCOMMADET2	TSTREFCLKOUT	
RXENCOMMADET3		
RXN0	TXCTRLACK0	
RXN1		
RXN2	TXCTRLACK1	
RXN3		
RXP0	TXCTRLACK2	
RXP1		
RXP2	TXCTRLACK3	
RXP3		
RXPOLARITY0	TXN0	
RXPOLARITY1		
RXPOLARITY2	TXN1	
RXPOLARITY3		
RXSLIP0	TXN2	
RXSLIP1		
RXSLIP2	TXN3	
RXSLIP3		
RXUSERCLKIN0	TXP0	
RXUSERCLKIN1		
RXUSERCLKIN2	TXP1	
RXUSERCLKIN3		
TXBUFRESET0	TXP2	
TXBUFRESET1		
TXBUFRESET2	TXP3	
TXBUFRESET3		
TXDEEMPH0	TXUSERCLKOUT0	
TXDEEMPH1		
TXDEEMPH2	TXUSERCLKOUT1	
TXDEEMPH3		
TXUSERCLKIN0	TXUSERCLKOUT2	
TXUSERCLKIN1		
TXUSERCLKIN2	TXUSERCLKOUT3	
TXUSERCLKIN3		

X11000

## 概要

Virtex®-6 FPGA GTH トランシーバーのデザイン エLEMENT です。GTH は HTX ファミリの Virtex-6 FPGA の中で最高速、10G で最適化されたコンフィギュレーション可能なトランシーバーです。このELEMENTの詳細は、Virtex-6 FPGA GTH トランシーバー ユーザー ガイドを参照してください。Virtex-6 FPGA GTX Transceiver Wizard は、GTHE1\_QUAD プリミティブをインスタンスエートするラッパーの生成に使用されるツールです。このウィザードは、ザイリンクス CORE Generator™ ツールに含まれています。

## デザインの入力方法

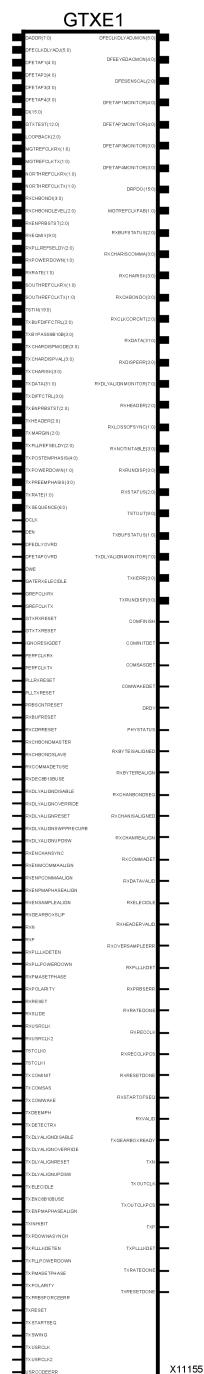
このELEMENTをインスタンスエートするには、Virtex-6 FPGA GTH Transceiver Wizard またはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンスエートしないでください。

## 詳細情報

- ・ [Virtex-6 FPGA GTH トランシーバー ユーザー ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



プリミティブ : Gigabit Transceiver



## 概要

このデザイン エLEMENTでは、効率性に優れ詳細にコンフィギュレーション可能なトランシーバーである Virtex®-6 FPGA RocketIO™ GTX トランシーバーが表現されます。このELEMENTの詳細は、Virtex-6 FPGA RocketIO GTX トランシーバー ユーザー ガイドを参照してください。Virtex-6 FPGA RocketIO GTX Transceiver Wizard は、GTXE1 プリミティブをインスタンスエートするラッパーの生成に使用されるツールです。このウィザードは、ザイリンクス CORE Generator™ ツールに含まれています。

## デザインの入力方法

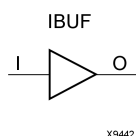
このELEMENTをインスタンスエートするには、Virtex-6 FPGA RocketIO GTX Transceiver Wizard またはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンスエートしないでください。

## 詳細情報

- ・ [Virtex-6 FPGA RocketIO GTX トランシーバー ユーザー ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUF

プリミティブ：Input Buffer



## 概要

このデザイン エLEMENTは、最上位の入力ポートまたは入出力ポートに接続されている信号に自動的に挿入されます。このバッファは通常、合成ツールで推論しますが、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 ポートをこの入力供給されるロジックに直接接続します。generic/defparam 値を設定し、バッファのビヘイビアーを適切に設定してください。

## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- IBUF: Single-ended Input Buffer
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of IBUF_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUF: Single-ended Input Buffer
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

IBUF #(
    .IBUF_LOW_PWR("TRUE"), // Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUF_inst (
    .O(O), // Buffer output
    .I(I)  // Buffer input (connect directly to top-level port)
);

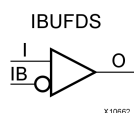
// End of IBUF_inst instantiation
```

## 詳細情報

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

## IBUFDS

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



### 概要

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

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	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 (インスタンスレーション)

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

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

-- IBUFDS: Differential Input Buffer
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of IBUFDS_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUFDS: Differential Input Buffer
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

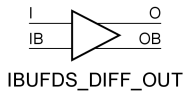
// End of IBUFDS_inst instantiation
```

## 詳細情報

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

## IBUFDS\_DIFF\_OUT

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



X10107

### 概要

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

### 論理表

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

### デザインの入力方法

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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	内部差動終端抵抗を使用するかどうかを指定
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
IBUF_LOW_POWER	ブール代数	TRUE、FALSE	FALSE	消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定

## 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_DIFF_OUT: Differential Input Buffer with Differential Output
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of IBUFDS_DIFF_OUT_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IBUFDS_DIFF_OUT: Differential Input Buffer with Differential Output
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

// End of IBUFDS_DIFF_OUT_inst instantiation
```

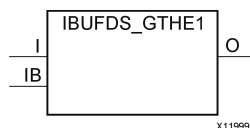
## 詳細情報

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



## IBUFDS\_GTHE1

**プリミティブ : Differential Clock Input for the GTH Transceiver Reference Clocks**



### 概要

GTH トランシーバー リファレンス クロックの専用差動クロック入力です。4 区画ある GTH の区画ごとに IBUFGDS\_GTHE1 コンポーネントが 1 つあり、GTHE1\_QUAD プリミティブの REFCLK ピンに直接接続されます。

### デザインの入力方法

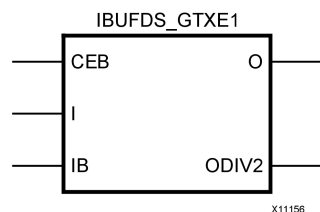
このエレメントをインスタンスエートするには、Virtex-6 FPGA GTH Transceiver Wizard またはこのエレメントを含む関連コアを使用します。このエレメントは直接インスタンスエートしないでください。

### 詳細情報

- ・ [Virtex-6 FPGA GTH トランシーバー ユーザー ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFDS\_GTXE1

**プリミティブ**：Differential Clock Input for the Transceiver Reference Clocks



### 概要

GT のリファレンス クロック入力だけでなく BUFG/MMCM などのクロック リソースを駆動することもできる、トランシーバー リファレンス クロックの差動クロック入力です。IBUFDS\_GTXE1 に関連したクワッドの 4 GTXE1にある MGTREFCLKRX/TX ピン、その上のクワッドの 4 GTXE1にある NORTHREFCLKRX/TX ピン、または下のクワッドの 4 GTXE1にある SOUTHREFCLKRX/TX ピンに接続します。

IBUFDS\_GTXE1 エレメントが接続できるデスティネーション ピンは Virtex®-6 には複数あります。GT のリファレンス クロックが接続されている場合、最適化された配線に基づき、どのピンの GT にもソフトウェアによるフル コントロールのもと配線接続ができます。複数のクロックが GT に接続されている場合、ソフトウェアにより各 IBUFDS が GT 上の指定ピンに配線されます。つまり、IBUFDS\_GTXE1 の O ピンは、GT の MGTREFCLKRX/TX ピンまたは NORTH/SOUTHREFCLKRX/TX ピンに接続されます。

**メモ**：RX および TX のマルチプレクサーは個別に選択できますが、配線はシリコン上で共有されます。

### デザインの入力方法

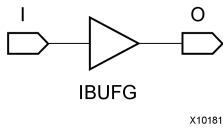
このエレメントをインスタンスエートするには、RocketIO™ Wizard またはこのエレメントを含む関連コアを使用します。このエレメントは直接インスタンスエートしないでください。

### 詳細情報

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

## IBUFG

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



### 概要

IBUFG は、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。IBUFG は最上位ポートから MMCM または 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、クロック兼用 (CC) ピンまたはグローバル クロック (GC) ピンでのみ駆動できます。

### ポートの説明

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

### デザインの入力方法

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

### 使用可能な属性

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

### VHDL 記述 (インスタンス化)

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

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

-- IBUFG: Single-ended global clock input buffer
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

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

## Verilog 記述 (インスタンス化)

```
// IBUFG: Single-ended global clock input buffer
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

IBUFG #(
    .IBUF_LOW_PWR("TRUE"), // Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFG_inst (
    .O(0), // Clock buffer output
    .I(I) // Clock buffer input (connect directly to top-level port)
);

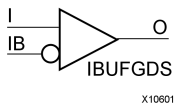
// End of IBUFG_inst instantiation
```

## 詳細情報

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

## IBUFGDS

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



### 概要

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

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- IBUFGDS: Differential Global Clock Input Buffer
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of IBUFGDS_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUFGDS: Differential Global Clock Input Buffer
//          Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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)
);

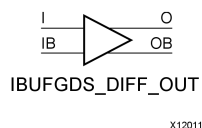
// End of IBUFGDS_inst instantiation
```

## 詳細情報

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

## IBUFGDS\_DIFF\_OUT

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



### 概要

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

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当て
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	内部差動終端抵抗を使用するかどうかを指定
IBUF_LOW_PWR	ブール代数	TRUE、FALSE	FALSE	消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定

## VHDL 記述 (インスタンスレーション)

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

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

-- IBUFGDS_DIFF_OUT: Differential Global Clock Buffer with Differential Output
--                               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

IBUFGDS_DIFF_OUT_inst : IBUFGDS_DIFF_OUT
generic map (
    DIFF_TERM => FALSE, -- Differential Termination
    IBUF_LOW_PWR => TRUE, -- Low power (TRUE) vs. performance (FALSE) setting for refernced I/O standards
    IOSTANDARD => "DEFAULT") -- Specify the input I/O standard
port map (
    O => O,      -- Buffer diff_p output
    OB => OB,    -- Buffer diff_n output
    I => I,      -- Diff_p buffer input (connect directly to top-level port)
    IB => IB     -- Diff_n buffer input (connect directly to top-level port)
);

-- End of IBUFGDS_DIFF_OUT_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// IBUFGDS_DIFF_OUT: Differential Global Clock Buffer with Differential Output
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

// End of IBUFGDS_DIFF_OUT_inst instantiation
```

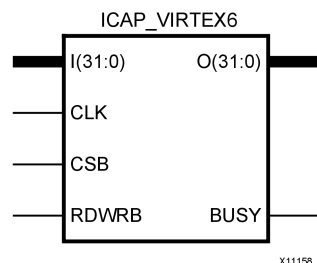
## 詳細情報

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



## ICAP\_VIRTEX6

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



### 概要

このデザイン エLEMENTからは、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。このコンポーネントを使用すると、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。このファンクションを正しく使用しないと 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限りこのELEMENTは使用しないでください。

### ポートの説明

ポート名	方向	幅	機能
BUSY	出力	1	Busy/Ready 出力
CLK	入力	1	クロック入力
CSB	入力	1	アクティブ Low の ICAP イネーブル
I[31:0]	入力	32	コンフィギュレーション データ入力バス
O[31:0]	出力	32	コンフィギュレーション データ出力バス
RDWRB	入力	1	読み出し/書き込みの選択

### デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DEVICE_ID	16 進数	32'h04244093、 32'h042CA093、 32'h042CC093、 32'h042C4093、 32'h042D0093、 32'h0423A093、 32'h0424A093、 32'h0424C093、 32'h04240093、 32'h04248093、 32'h04250093、 32'h04252093、 32'h04256093、 32'h04286093、 32'h04288093	32'h04244093	あらかじめプログラムされているデバイス ID 値を指定します。
ICAP_WIDTH	文字列	"X8"、"X16"、 "X32"	"X8"	ICAP_VIRTEX6 で使用する入力および出力データ幅を指定します。
SIM_CFG_FILE_NAME	文字列	ファイルの名前と場所	なし	シミュレーション モデルで解析するロービット ファイル (RBT) を指定します。

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- ICAP_VIRTEX6: Internal Configuration Access Port
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

ICAP_VIRTEX6_inst : ICAP_VIRTEX6
generic map (
    DEVICE_ID => X"4244093",    -- Specifies the pre-programmed Device ID value
    ICAP_WIDTH => "X8",        -- Specifies the input and output data width to be used with the
                                -- ICAP_VIRTEX6.
    SIM_CFG_FILE_NAME => "NONE" -- Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
                                -- model
)
port map (
    BUSY => BUSY,    -- 1-bit output: Busy/Ready output
    O => O,          -- 32-bit output: Configuration data output bus
    CLK => CLK,      -- 1-bit input: Clock Input
    CSB => CSB,      -- 1-bit input: Active-Low ICAP input Enable
    I => I,          -- 32-bit input: Configuration data input bus
    RDWRB => RDWRB   -- 1-bit input: Read/Write Select input
);

-- End of ICAP_VIRTEX6_inst instantiation

```

## Verilog 記述 (インスタンス化)

```
// ICAP_VIRTEX6: Internal Configuration Access Port
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

ICAP_VIRTEX6 #(
    .DEVICE_ID(0'h4244093),    // Specifies the pre-programmed Device ID value
    .ICAP_WIDTH("X8"),        // Specifies the input and output data width to be used with the
                                // ICAP_VIRTEX6.
    .SIM_CFG_FILE_NAME("NONE") // Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
                                // model
)
ICAP_VIRTEX6_inst (
    .BUSY(BUSY),    // 1-bit output: Busy/Ready output
    .O(O),          // 32-bit output: Configuration data output bus
    .CLK(CLK),      // 1-bit input: Clock Input
    .CSB(CSB),      // 1-bit input: Active-Low ICAP input Enable
    .I(I),          // 32-bit input: Configuration data input bus
    .RDWRB(RDWRB)   // 1-bit input: Read/Write Select input
);

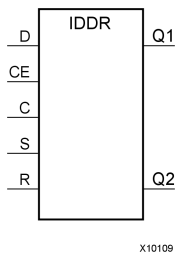
// End of ICAP_VIRTEX6_inst instantiation
```

## 詳細情報

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

## IDDR

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



### 概要

このデザイン エレメントは、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ており、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ます。これにより、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。

**メモ：** 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なときに使用します。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 出力です。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	クロック入力ピンです。
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、また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

## 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 Q2 ピンの初期値を指定します。
SRTYPE	文字列	"SYNC"、"ASYNC"	"SYNC"	セット/リセットのタイプを選択します。"SYNC" に設定するとリセット (R) およびセット (S) ピンの動作が C クロック ピンの立ち上がりエッジに同期し、"ASYNC" に設定すると非同期動作になります。

## VHDL 記述 (インスタンス化)

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

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

-- IDDR: Double Data Rate Input Register with Set, Reset
--       and Clock Enable.
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of IDDR_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IDDR: Input Double Data Rate Input Register with Set, Reset
//      and Clock Enable.
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

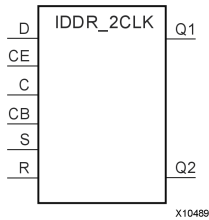
// End of IDDR_inst instantiation
```

## 詳細情報

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

## IDDR\_2CLK

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



### 概要

このデザイン エLEMENTは、ザイリンクス FPGA 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 の各立ち下がりエッジの後に変化します。
- SAME\_EDGE モード：**データは各クロックの立ち上がりエッジで受信されますが、CB クロック データレジスタの前にレジスタが 1 つ追加されます。このレジスタはクロック信号 C の立ち上がりエッジで動作するので、DDR データは同じクロック エッジで FPGA に送信されます。ただし、データ ペアは分離されているように見えます。Q1 と Q2 にはペア 1 および 2 が同時に送信されず、最初のペアがペア 1 とドントケアとなり、次のクロックサイクルでペア 2 と 3 が送信されます。
- SAME\_EDGE\_PIPELINED モード：**SAME\_EDGE モードと同様にデータが処理されますが、SAME\_EDGE モードでのデータ ペアの分離を回避するため、C のクロック データレジスタの前にもレジスタが追加されます。これにより、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 出力です。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、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-6
-- Xilinx HDL Libraries Guide, version 13.3

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)

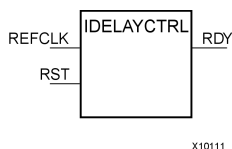
```
// IDDR_2CLK: Dual-Clock, Input Double Data Rate Input Register with
//           Set, Reset and Clock Enable.
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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 primay 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エート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エートされており、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、一定の周波数リファレンス クロック REFCLK を使用する場合に、プロセス、電圧、および温度の変化にかかわらず、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	リファレンス 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assertされます。
REFCLK	入力	1	プロセス、電圧、温度の変化にかかわらず、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 (リファレンス クロック)：**プロセス、電圧、温度の変化にかかわらず、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)：**リファレンス 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assertされます。

### 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

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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: IDELAY Tap Delay Value Control
//             Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

(* IDELAY_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
);

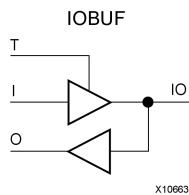
// End of IDELAYCTRL_inst instantiation
```

## 詳細情報

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

## IOBUF

プリミティブ：Bi-Directional Buffer



## 概要

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

## 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
I/O	入出力	1	バッファの入出力
I	入力	1	バッファの入力
T	入力	1	トリステート イネーブル入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	I/O 規格として LVTTTL、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;

-- IOBUF: Single-ended Bi-directional Buffer
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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 (インスタンス化)

```
// IOBUF: Single-ended Bi-directional Buffer
//      All devices
// Xilinx HDL Libraries Guide, version 13.3

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
);

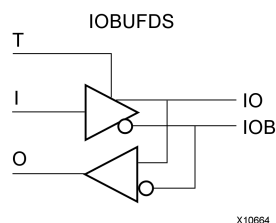
// End of IOBUF_inst instantiation
```

## 詳細情報

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

## IOBUFDS

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



## 概要

このデザイン エレメントは、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
I/O	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-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

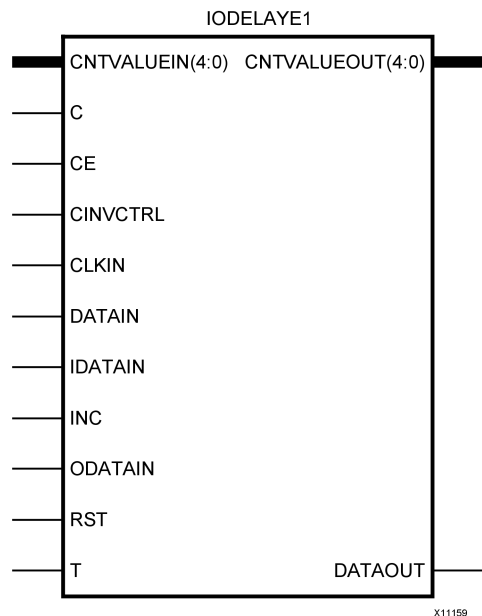
// End of IOBUFDS_inst instantiation
```

## 詳細情報

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

## IODELAYE1

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



### 概要

各 I/O ブロックには、IODELAYE1 と呼ばれるプログラム可能な絶対遅延エレメントが含まれています。この遅延エレメントは、入力レジスタ/ISERDESE1 または出力レジスタ/OSERDESE1 ブロック、あるいはその両方に接続できます。IODELAYE1 は、キャリブレーションされたタップ精度を使用する 31 タップのラップアラウンド遅延エレメントです。遅延値は、Virtex-6 FPGA データシートを参照してください。組み合わせ入力パス、レジスタ付き入力パス、組み合わせ出力パス、またはレジスタ付き出力パスに使用できます。FPGA ロジックで直接アクセスすることも可能です。IODELAYE1 を使用すると、入力信号を個別に遅延できます。タップ遅延精度は、IDELAYCTRL リファレンス クロックを Virtex-6 FPGA データシートで指定された範囲から選択することによって変更できます。IODELAYE1 リソースは、入力、出力、または双方向遅延として使用できます。

### ポートの説明

ポート名	方向	幅	機能
C	入力	1	VARIABLE または VAR_LOADABLE モードで使用するクロック入力
CE	入力	1	インクリメント/デクリメントを有効にするアクティブ High のイネーブル
CINVCTRL	入力	1	クロック (C) の極性を動的に反転します。
CLKIN	入力	1	IODELAY へのクロック入力 (IO CLKMUX から)
CNTVALUEIN[4:0]	入力	5	動的に読み込まれるタップ値用の FPGA ロジックからのタップ カウンター値
CNTVALUEOUT[4:0]	出力	5	タップ値モニター用に FPGA ロジックに送信されるタップ カウンター値
DATAIN	入力	1	FPGA ロジックで直接駆動され、ロジックでアクセス可能な遅延ラインとなります。データは、DATAOUT ポートを介して IDELAY_VALUE で設定された遅延で FPGA ロジックにフィードバックされます。DATAIN はローカルに反転可能です。データを IOB に駆動することはできません。



ポート名	方向	幅	機能
DATAOUT	出力	1	3 つのデータ入力ポートからの遅延データです。IDELAY モードでは FPGA ロジックに、ODELAY モードでは IOB に、双方向遅延モードでは両方に接続されます。双方向遅延モードでは、T ポートは IDATAIN パスと ODATAIN パスを動的に切り替え、OLOGIC ブロックからのトライステート信号 T で示される方向に基づき入力遅延と出力遅延を切り替えます。
IDATAIN	入力	1	関連付けられている IOB で駆動されます。IDELAY モードでは、データは ILOGIC/ISERDES ブロックに入力するか、FPGA ロジックに直接入力するか、または DATAOUT ポートを介して IDELAY_VALUE で設定された遅延で両方に入力できます。
INC	入力	1	タップ遅延のインクリメント/デクリメント数
ODATAIN	入力	1	OLOGIC/OSERDES で駆動されます。ODELAY モードでは、ODATAIN は IOB に接続されている DATAOUT ポートを ODELAY_VALUE で設定された遅延で駆動します。
RST	入力	1	VARIABLE モードでは、遅延エレメントを IDELAY_VALUE または ODELAY_VALUE 属性で設定された値にリセットします。これらの属性が設定されていない場合は、0 にリセットされます。RST はアクティブ High のリセットで、入力クロック入力 (C) に同期しています。VAR_LOADABLE モードでは、遅延エレメントを CNTVALUEIN で設定された値にリセットします。CNTVALUEIN[4:0] の値が新しいタップ値になります。この場合、IDELAY_VALUE および ODELAY_VALUE 属性は無視されます。
T	入力	1	トライステート入力制御ポート。双方向動作では、OBUFT の T ピンも制御します。入力のみまたは内部遅延の場合は High に、出力のみの場合は Low にします。

## デザインの入力方法

インスタンス化	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
CINVCTRL_SEL	ブール代数	FALSE、TRUE	FALSE	クロック (C) の極性を動的に反転します。
DELAY_SRC	文字列	"CLKIN"、"DATAIN"、 "I"、"IO"、"O"	"I"	IODELAY コンポーネントのソースを指定します。 <ul style="list-style-type: none"> <li>"CLKIN"：CLKIN が IODELAYE1 入力となります。</li> <li>"DATAIN"：どのポートにも接続しません (内部モード)。</li> <li>"I"：入力ポートまたは IBUF に直接接続します (入力モード)。</li> <li>"IO"：ポートに接続します。</li> </ul>

属性	データ型	値	デフォルト	説明
				・ “O”：出力ポートまたは OBUF に接続します (出力モード)。
HIGH_PERFORMANCE_MODE	ブール代数	TRUE、FALSE	TRUE	TRUE の場合、出力ジッターが減少します。FALSE の場合、消費電力量が減少します。消費電力量の差異は、XPower Estimator (XPE) ツールで確認できます。
IDELAY_TYPE	文字列	“DEFAULT”、 “FIXED”、 “VARIABLE”、 “VAR_LOADABLE”	“DEFAULT”	タップ遅延ラインのタイプを設定します。“DEFAULT” に設定すると、ゼロ ホールド タイムになります。“FIXED” に設定すると、スタティック遅延値に設定されます。“VAR_LOADABLE” に設定すると、タップ値が動的に読み込まれます。“VARIABLE” に設定すると、遅延値が動的に調整されます。
IDELAY_VALUE	整数	0、1、2、3、4、5、6、7、 8、9、10、11、12、13、 14、15、16、17、18、 19、20、21、22、23、 24、25、26、27、28、 29、30、31	0	“FIXED” モードでは遅延タップ数、“VARIABLE” モードでは遅延タップ数の初期値を指定します (入力パス)。IDELAY_TYPE が “VAR_LOADABLE” に設定されている場合、この属性は無視されます。
ODELAY_TYPE	文字列	“FIXED”、 “VARIABLE”、 “VAR_LOADABLE”	“FIXED”	出力遅延タイプを “FIXED” (固定)、“VARIABLE” (変動)、または “VAR_LOADABLE” (タップ値を動的に読み込み) のいずれかに設定します。
ODELAY_VALUE	整数	0、1、2、3、4、5、6、7、 8、9、10、11、12、13、 14、15、16、17、18、 19、20、21、22、23、 24、25、26、27、28、 29、30、31	0	“FIXED” モードでは遅延タップ数、“VARIABLE” モードでは遅延タップ数の初期値を指定します (出力パス)。IDELAY_TYPE が “VAR_LOADABLE” に設定されている場合、この属性は無視されます。
REFCLK_FREQUENCY	1 上位ビット FLOAT	190.0 ～ 210.1 および 290.0 ～ 310.0	200.0	Timing Analyzer でスタティック タイミング解析、論理シミュレーション、タイミングシミュレーションに使用するタップ値 (MHz) を設定します。適切なタップ遅延値およびパフォーマンスを得るためには、REFCLK の周波数をデータシートに記載された範囲内にする必要があります。
SIGNAL_PATTERN	文字列	“DATA”、“CLOCK”	“DATA”	Timing Analyzer でデータパスまたはクロックパスに対して適切な遅延チェーンジッター量が使用されるようにします。

## VHDL 記述 (インスタンス化)

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;

-- IDELAYE1: Input and Output Fixed or Variable Delay Element
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

IDELAYE1_inst : IDELAYE1
generic map (
    CINVERTSEL => FALSE,          -- Enable dynamic clock inversion ("TRUE"/"FALSE")
    DELAY_SRC => "I",             -- Delay input ("I", "CLKIN", "DATAIN", "IO", "O")
    HIGH_PERFORMANCE_MODE => TRUE, -- Reduced jitter ("TRUE"), Reduced power ("FALSE")
    IDELAY_TYPE => "DEFAULT",     -- "DEFAULT", "FIXED", "VARIABLE", or "VAR_LOADABLE"
    IDELAY_VALUE => 0,            -- Input delay tap setting (0-32)

```

```

ODELAY_TYPE => "FIXED",          -- "FIXED", "VARIABLE", or "VAR_LOADABLE"
ODELAY_VALUE => 0,               -- Output delay tap setting (0-32)
REFCLK_FREQUENCY => 200.0,       -- IDELAYCTRL clock input frequency in MHz
SIGNAL_PATTERN => "DATA"        -- "DATA" or "CLOCK" input signal
)
port map (
  CNTVALUEOUT => CNTVALUEOUT, -- 5-bit output - Counter value for monitoring purpose
  DATAOUT => DATAOUT,       -- 1-bit output - Delayed data output
  C => C,                     -- 1-bit input - Clock input
  CE => CE,                   -- 1-bit input - Active high enable increment/decrement function
  CINVCTRL => CINVCTRL,       -- 1-bit input - Dynamically inverts the Clock (C) polarity
  CLKIN => CLKIN,             -- 1-bit input - Clock Access into the IDELAY
  CNTVALUEIN => CNTVALUEIN,   -- 5-bit input - Counter value for loadable counter application
  DATAIN => DATAIN,         -- 1-bit input - Internal delay data
  IDATAIN => IDATAIN,         -- 1-bit input - Delay data input
  INC => INC,                  -- 1-bit input - Increment / Decrement tap delay
  ODATAIN => ODATAIN,         -- 1-bit input - Data input for the output datapath from the device
  RST => RST,                  -- 1-bit input - Active high, synchronous reset, resets delay chain to IDELAY_VALUE/
                                -- ODELAY_VALUE tap. If no value is specified, the default is 0.
  T => T                       -- 1-bit input - 3-state input control. Tie high for input-only or internal delay or
                                -- tie low for output only.
);

-- End of IDELAYE1_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```

// IDELAYE1: Input / Output Fixed or Variable Delay Element
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

(* IDELAY_GROUP = "<iodelay_group_name>" *) // Specifies group name for associated IDELAYs and IDELAYCTRL
IDELAYE1 #(
  .CINVCTRL_SEL("FALSE"),          // Enable dynamic clock inversion ("TRUE"/"FALSE")
  .DELAY_SRC("I"),                 // Delay input ("I", "CLKIN", "DATAIN", "IO", "O")
  .HIGH_PERFORMANCE_MODE("FALSE"), // Reduced jitter ("TRUE"), Reduced power ("FALSE")
  .IDELAY_TYPE("DEFAULT"),         // "DEFAULT", "FIXED", "VARIABLE", or "VAR_LOADABLE"
  .IDELAY_VALUE(0),                // Input delay tap setting (0-32)
  .ODELAY_TYPE("FIXED"),           // "FIXED", "VARIABLE", or "VAR_LOADABLE"
  .ODELAY_VALUE(0),                // Output delay tap setting (0-32)
  .REFCLK_FREQUENCY(200.0),        // IDELAYCTRL clock input frequency in MHz
  .SIGNAL_PATTERN("DATA")         // "DATA" or "CLOCK" input signal
)
IDELAYE1_inst (
  .CNTVALUEOUT(CNTVALUEOUT), // 5-bit output - Counter value for monitoring purpose
  .DATAOUT(DATAOUT),         // 1-bit output - Delayed data output
  .C(C),                     // 1-bit input - Clock input
  .CE(CE),                   // 1-bit input - Active high enable increment/decrement function
  .CINVCTRL(CINVCTRL),       // 1-bit input - Dynamically inverts the Clock (C) polarity
  .CLKIN(CLKIN),             // 1-bit input - Clock Access into the IDELAY
  .CNTVALUEIN(CNTVALUEIN),   // 5-bit input - Counter value for loadable counter application
  .DATAIN(DATAIN),           // 1-bit input - Internal delay data
  .IDATAIN(IDATAIN),         // 1-bit input - Delay data input
  .INC(INC),                  // 1-bit input - Increment / Decrement tap delay
  .ODATAIN(ODATAIN),         // 1-bit input - Data input for the output datapath from the device
  .RST(RST),                  // 1-bit input - Active high, synchronous reset, resets delay chain to IDELAY_VALUE/
                                // ODELAY_VALUE tap. If no value is specified, the default is 0.
  .T(T)                       // 1-bit input - 3-state input control. Tie high for input-only or internal delay or
                                // tie low for output only.
);

// End of IDELAYE1_inst instantiation

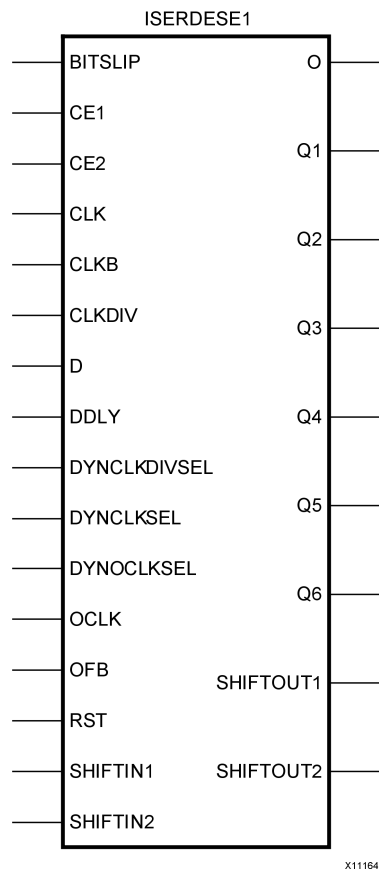
```

## 詳細情報

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

## ISERDESE1

### プリミティブ：Input SERial/DESerializer



### 概要

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

### ポートの説明

ポート名	方向	幅	機能
BITSLLIP	入力	1	BITSLLIP ピンがアサートされると (アクティブ High)、CLKDIV に同期してビットスリップ処理が実行されます。ビットスリップ処理が実行されるごとに、バレル シフターと同様、Q1 ~ Q6 出力ポートのデータが 1 ずつシフトします (DDR と SDR では動作が異なる)。
CE1	入力	1	データレジスタ クロック イネーブル
CE2	入力	1	データレジスタ 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	入力シリアル データ ストリームの入力に使用される高速セカンダリ クロック入力。MEMORY_QDR 以外のモードでは、CLKB を CLK を反転したクロックに接続します。MEMORY_QDR モードでは、CLKB を固有の位相シフトされたクロックに接続する必要があります。

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
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
DDLY	入力	1	IODELAY からのシリアル入力
DYNCLKDIVSEL	入力	1	CLKDIV の反転を動的に選択
DYNCLKSEL	入力	1	CLK および CLKB の反転を動的に選択
O	出力	1	組み合わせ出力
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
OCLKB	入力	1	非同期オーバーサンプリングに使用されます。
OFB	入力	1	出力フィードバック ポート。OSERDESE1 の高速シリアル データ出力ポートまたは CLKPERF のバイパスされたバージョンです。ODELAYUSED 属性が 0 に設定されている場合、ISERDESE1 にシリアル データを送信するために OFB ポートを使用できません。ODELAYUSED 属性が 1 に設定されており、OSERDESE1 が MEMORY_DDR3 モードの場合、高パフォーマンス クロック入力 (CLKPERF) を IODELAYE1 にリンクするために OFB ポートを使用できます。
Q1 ~ Q6	出力	1	ISERDESE1 モジュールのレジスタ付き出力。1 つの ISERDESE1 ブロックで最大 6 ビット (1:6 デシリアル化) までサポートできます。6 より大きいビット幅 (10 ビットまで) もサポート可能です。
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
SHIFTIN1/ SHIFTIN2	入力	1	ISERDES_MODE が SLAVE の場合は、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。
SHIFTOUT1/ SHIFTOUT2	出力	1	ISERDES_MODE が MASTER で、2 つの ISERDES_NODELAY をカスケード接続する場合は、スレーブの SHIFTIN1 と SHIFTIN2 入力に接続します。

## デザインの入力方法

インスタンス化	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
DATA_RATE	文字列	"DDR"、"SDR"	"DDR"	入力されるデータ ストリーム レートを SDR または DDR のいずれかに指定します。
DATA_WIDTH	整数	4、2、3、5、6、7、8、10	4	シリアル/パラレル コンバーターの幅を指定します。有効な値は、DATA_RATE 属性 ("SDR" または "DDR") によって異なります。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> <li>DATA_RATE = "DDR" の場合の有効値は 4、6、8、または 10 です。</li> <li>DATA_RATE = "SDR" の場合の有効値は 2、3、4、5、6、7、8 です。</li> </ul>
DYN_CLKDIV_INV_EN	ブール代数	FALSE、TRUE	FALSE	TRUE の場合、DYNCLKDIVINSEL の反転がイネーブルになり、CLKDIV ピンの HDL 反転がディスエーブルになります。
DYN_CLK_INV_EN	ブール代数	FALSE、TRUE	FALSE	TRUE の場合、DYNCLKINSEL の反転がイネーブルになり、CLK および CLKB ピンの HDL 反転がディスエーブルになります。
INIT_Q1 ~ INIT_Q4	2 進数	1'b0 ~ 1'b1	1'b0	Q 出力の初期値を指定します。
INTERFACE_TYPE	文字列	"MEMORY"、 "MEMORY_DDR3"、 "MEMORY_QDR"、 "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。
IOBDELAY	文字列	"NONE"、"BOTH"、 "IBUF"、"IFD"	"NONE"	ISERDES モジュールの入力ソースを指定します。
NUM_CE	整数	2、1	2	クロック イネーブルの数を指定します。
OFB_USED	ブール代数	FALSE、TRUE	FALSE	<p>ISERDESE1 および OSERDESE1 の OFB ポートは、OSERDESE1 に送信されたデータを ISERDESE1 にフィードバックするために使用できます。この機能をイネーブルにするには、OFB_USED 属性を TRUE に設定します。正しいデータがフィードバックされるようにするには、OSERDESE1 と ISERDESE1 の DATA_RATE および DATA_WIDTH を同じ設定にする必要があります。</p> <p>ISERDESE1 および OSERDESE1 をデータ幅拡張モードで使用する場合は、マスター OSERDESE1 をマスター ISERDESE1 に接続します。ISERDESE1 をフィードバックポートとして使用する場合、外部データの入力として使用することはできません。</p> <p><b>メモ：</b> OFB を OSERDES 出力を遅延するためにのみ使用する場合は、OFB_USED を FALSE に設定する必要があります。</p>
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 モードかスレーブ モードに設定するかどうかを指定します。
SRVAL_Q1 ~ SRVAL_Q4	2 進数	1'b0 ~ 1'b1	1'b0	SR をアサートした場合の Q 出力の値を指定します。

## 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;

-- ISERDESE1: Input SERIAL/DESerializer
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

ISERDESE1_inst : ISERDESE1
generic map (
    DATA_RATE => "DDR",           -- "SDR" or "DDR"
    DATA_WIDTH => 4,              -- Parallel data width (2-8, 10)
    DYN_CLKDIV_INV_EN => FALSE,    -- Enable DYNCLKDIVINSEL inversion (TRUE/FALSE)
    DYN_CLK_INV_EN => FALSE,       -- Enable DYNCLKINSEL inversion (TRUE/FALSE)
    -- INIT_Q1 - INIT_Q4: Initial value on the Q outputs (0/1)
    INIT_Q1 => '0',
    INIT_Q2 => '0',
    INIT_Q3 => '0',
    INIT_Q4 => '0',
    INTERFACE_TYPE => "MEMORY",   -- "MEMORY", "MEMORY_DDR3", "MEMORY_QDR", "NETWORKING", or "OVERSAMPLE"
    IOBDELAY => "NONE",           -- "NONE", "IBUF", "IFD", "BOTH"
    NUM_CE => 2,                  -- Number of clock enables (1 or 2)
    OFB_USED => FALSE,            -- Select OFB path (TRUE/FALSE)
    SERDES_MODE => "MASTER",      -- "MASTER" or "SLAVE"
    -- SRVAL_Q1 - SRVAL_Q4: Q output values when SR is used (0/1)
    SRVAL_Q1 => '0',
    SRVAL_Q2 => '0',
    SRVAL_Q3 => '0',
    SRVAL_Q4 => '0'
)
port map (
    O => O,                        -- 1-bit output: Combinatorial output
    -- Q1 - Q6: 1-bit (each) output: Registered data outputs
    Q1 => Q1,
    Q2 => Q2,
    Q3 => Q3,
    Q4 => Q4,
    Q5 => Q5,
    Q6 => Q6,
    -- SHIFTOUT1-SHIFTOUT2: 1-bit (each) output: Data width expansion output ports
    SHIFTOUT1 => SHIFTOUT1,
    SHIFTOUT2 => SHIFTOUT2,
    BITSLIP => BITSLIP,           -- 1-bit input: Bitflip enable input
    -- CE1, CE2: 1-bit (each) input: Data register clock enable inputs
    CE1 => CE1,
    CE2 => CE2,
    -- Clocks: 1-bit (each) input: ISERDESE1 clock input ports
    CLK => CLK,                   -- 1-bit input: High-speed clock input
    CLKB => CLKB,                 -- 1-bit input: High-speed secondary clock input
    CLKDIV => CLKDIV,             -- 1-bit input: Divided clock input
    OCLK => OCLK,                 -- 1-bit input: High speed output clock input used when
    -- INTERFACE_TYPE="MEMORY"

    -- Dynamic Clock Inversions: 1-bit (each) input: Dynamic clock inversion pins to switch clock polarity
    DYNCLKDIVSEL => DYNCLKDIVSEL, -- 1-bit input: Dynamic CLKDIV inversion input
    DYNCLKSEL => DYNCLKSEL,       -- 1-bit input: Dynamic CLK/CLKB inversion input
    -- Input Data: 1-bit (each) input: ISERDESE1 data input ports
    D => D,                       -- 1-bit input: Data input
    DDLY => DDLY,                 -- 1-bit input: Serial input data from IODELAYE1
    OFB => OFB,                   -- 1-bit input: Data feedback input from OSERDESE1
    RST => RST,                   -- 1-bit input: Active high asynchronous reset input
    -- SHIFTIN1-SHIFTIN2: 1-bit (each) input: Data width expansion input ports
    SHIFTIN1 => SHIFTIN1,
    SHIFTIN2 => SHIFTIN2
);

-- End of ISERDESE1_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// ISERDESE1: Input SERIAL/DESerializer
//          Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

ISERDESE1 #(
    .DATA_RATE("DDR"),           // "SDR" or "DDR"
    .DATA_WIDTH(4),              // Parallel data width (2-8, 10)
    .DYN_CLKDIV_INV_EN("FALSE"), // Enable DYNCLKDIVINVSEL inversion (TRUE/FALSE)
    .DYN_CLK_INV_EN("FALSE"),    // Enable DYNCLKINVSEL inversion (TRUE/FALSE)
    // INIT_Q1 - INIT_Q4: Initial value on the Q outputs (0/1)
    .INIT_Q1(1'b0),
    .INIT_Q2(1'b0),
    .INIT_Q3(1'b0),
    .INIT_Q4(1'b0),
    .INTERFACE_TYPE("MEMORY"),   // "MEMORY", "MEMORY_DDR3", "MEMORY_QDR", "NETWORKING", or "OVERSAMPLE"
    .IOBDelay("NONE"),           // "NONE", "IBUF", "IFD", "BOTH"
    .NUM_CE(2),                  // Number of clock enables (1 or 2)
    .OFB_USED("FALSE"),          // Select OFB path (TRUE/FALSE)
    .SERDES_MODE("MASTER"),      // "MASTER" or "SLAVE"
    // SRVAL_Q1 - SRVAL_Q4: Q output values when SR is used (0/1)
    .SRVAL_Q1(1'b0),
    .SRVAL_Q2(1'b0),
    .SRVAL_Q3(1'b0),
    .SRVAL_Q4(1'b0)
)
ISERDESE1_inst (
    .O(0),                       // 1-bit output: Combinatorial output
    // Q1 - Q6: 1-bit (each) output: Registered data outputs
    .Q1(Q1),
    .Q2(Q2),
    .Q3(Q3),
    .Q4(Q4),
    .Q5(Q5),
    .Q6(Q6),
    // SHIFTOUT1-SHIFTOUT2: 1-bit (each) output: Data width expansion output ports
    .SHIFTOUT1(SHIFTOUT1),
    .SHIFTOUT2(SHIFTOUT2),
    .BITSLIP(BITSLIP),           // 1-bit input: Bit-slip enable input
    // CE1, CE2: 1-bit (each) input: Data register clock enable inputs
    .CE1(CE1),
    .CE2(CE2),
    // Clocks: 1-bit (each) input: ISERDESE1 clock input ports
    .CLK(CLK),                   // 1-bit input: High-speed clock input
    .CLKB(CLKB),                 // 1-bit input: High-speed secondary clock input
    .CLKDIV(CLKDIV),             // 1-bit input: Divided clock input
    .OCLK(OCLK),                 // 1-bit input: High speed output clock input used when
    // INTERFACE_TYPE="MEMORY"

    // Dynamic Clock Inversions: 1-bit (each) input: Dynamic clock inversion pins to switch clock polarity
    .DYNCLKDIVSEL(DYNCLKDIVSEL), // 1-bit input: Dynamic CLKDIV inversion input
    .DYNCLKSEL(DYNCLKSEL),       // 1-bit input: Dynamic CLK/CLKB inversion input
    // Input Data: 1-bit (each) input: ISERDESE1 data input ports
    .D(D),                       // 1-bit input: Data input
    .DDL(DDL),                   // 1-bit input: Serial input data from IODELAYE1
    .OFB(OFB),                   // 1-bit input: Data feedback input from OSERDESE1
    .RST(RST),                   // 1-bit input: Active high asynchronous reset input
    // SHIFTTIN1-SHIFTTIN2: 1-bit (each) input: Data width expansion input ports
    .SHIFTTIN1(SHIFTTIN1),
    .SHIFTTIN2(SHIFTTIN2)
);

// End of ISERDESE1_inst instantiation
```

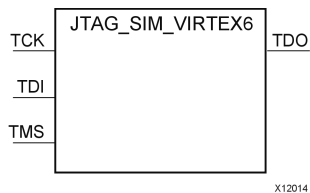
## 詳細情報

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



## JTAG\_SIM\_VIRTEX6

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



### 概要

このシミュレーション コンポーネントを使用すると、JTAG TAP コントローラー インターフェイス、ファンクション、およびコマンドの論理シミュレーションを実行でき、JTAG およびバウンダリ スキャン動作、USER コマンドおよび BSCAN\_VIRTEX6 コンポーネントに関連する動作をボード レベルで理解し、デバッグするのに役立ちます。このモデルは、FPGA ソフトウェアの特定のプリミティブにはマップされず、デザインに直接インスタンス化することはできませんが、テストベンチなどのシミュレーションのみのファイルで指定するなど、合成で除外してデザイン ネットリストに含まれないようにすれば、ソース デザインと共に使用できます。このモデルは、論理 (RTL) シミュレーションおよびタイミング シミュレーションで使用できます。

### ポートの説明

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

### デザインの入力方法

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

テストベンチ ファイルにインスタンス化し、インプリメンテーション ファイルまたはデザインの合成に使用されるファイルには含めないことをお勧めします。コンフィギュレーションの読み込みとデバイスのスタートアップの関係およびスタートアップ シーケンスを決定するために使用できます。

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
PART_NAME	文字列	"CX75T"、"LX75T"、 "CX130T"、 "LX130T"、 "CX195T"、 "LX195T"、 "CX240T"、 "LX240T"、 "HX250T"、 "SX315T"、 "LX365T"、 "HX380T"、 "SX475T"、 "LX550T"、 "HX565T"、"LX760"	"LX75T"	IDCODE およびその他のデバイス特定の属性を正しく設定するため、ターゲット デバイスを指定します。

## VHDL 記述 (インスタンス化)

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

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

-- JTAG_SIM_VIRTEX6: JTAG Interface Simulation Model
--                      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

JTAG_SIM_VIRTEX6_inst : JTAG_SIM_VIRTEX6
generic map (
  PART_NAME => "LX75T") -- Specify target V6 device. Possible values are:
                        -- "CX130T", "CX195T", "CX240T", "CX75T", "HX250T",
                        -- "HX255T", "HX380T", "HX45T", "HX565T",
                        -- "LX115T", "LX130T", "LX130TL", "LX195T",
                        -- "LX195TL", "LX240T", "LX240TL", "LX365T",
                        -- "LX365TL", "LX40T", "LX550T", "LX550TL",
                        -- "LX75T", "LX760", "SX315T", "SX475T"

port map (
  TDO => TDO,          -- JTAG data output (1-bit)
  TCK => TCK,          -- Clock input (1-bit)
  TDI => TDI,          -- JTAG data input (1-bit)
  TMS => TMS           -- JTAG command input (1-bit)
);

-- End of JTAG_SIM_VIRTEX6_inst instantiation
```

## Verilog 記述 (インスタンスエーション)

```
// JTAG_SIM_VIRTEX6: JTAG Interface Simulation Model
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

JTAG_SIM_VIRTEX6 #(
    .PART_NAME("LX75T") // Specify target V6 device. Possible values are:
                        // "CX130T", "CX195T", "CX240T", "CX75T", "HX255T", "HX380T", "HX45T", "HX565T",
                        // "LX115T", "LX130T", "LX130TL", "LX195T", "LX195TL", "LX240T", "LX240TL", "LX365T", "LX365TL",
                        // "LX40T", "LX550T", "LX550TL", "LX75T", "LX760", "SX315T", "SX475T"
) JTAG_SIM_VIRTEX6_inst (
    .TDO(TDO), // 1-bit JTAG data output
    .TCK(TCK), // 1-bit Clock input
    .TDI(TDI), // 1-bit JTAG data input
    .TMS(TMS) // 1-bit JTAG command input
);

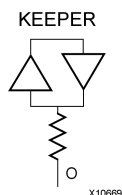
// End of JTAG_SIM_VIRTEX6_inst instantiation
```

## 詳細情報

- ・ [合成/シミュレーション デザイン ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## KEEPER

### プリミティブ：KEEPER Symbol



### 概要

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

### ポートの説明

ポート名	方向	幅	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-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of KEEPER_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// KEEPER: I/O Buffer Weak Keeper
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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

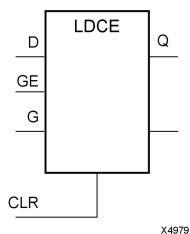
// End of KEEPER_inst instantiation
```

## 詳細情報

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

## LDCE

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



## 概要

このデザイン エレメントは、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

## 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-6
--       Xilinx HDL Libraries Guide, version 13.3

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.
//       Virtex-6
//       Xilinx HDL Libraries Guide, version 13.3

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
);

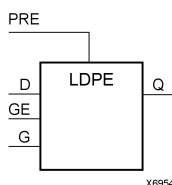
// End of LDCE_inst instantiation
```

## 詳細情報

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

## LDPE

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



## 概要

このデザイン エレメントは、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 Q ポートに対する GSR のアサート時の初期値を指定



## 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-6
--       Xilinx HDL Libraries Guide, version 13.3

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.
//       Virtex-6
//       Xilinx HDL Libraries Guide, version 13.3

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

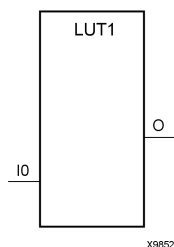
// End of LDPE_inst instantiation
```

## 詳細情報

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

## LUT1

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



## 概要

このデザイン エレメントは一般出力 (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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

## 論理表

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT1: 1-input Look-Up Table with general output
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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

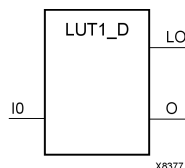
// End of LUT1_inst instantiation
```

## 詳細情報

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

## LUT1\_D

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



### 概要

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

出力 O は一般的なインターコネクトです。LO 出力は同じ CLB スライス内の別の出力、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 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

### デザインの入力方法

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

### 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT1_D: 1-input Look-Up Table with general and local outputs
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

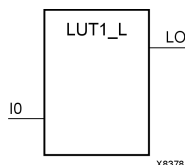
// End of LUT1_D_inst instantiation
```

## 詳細情報

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

## LUT1\_L

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



### 概要

このデザイン エレメント 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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

### デザインの入力方法

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

### 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT1_L: 1-input Look-Up Table with local output
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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

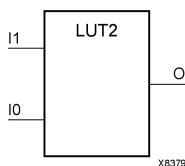
// End of LUT1_L_inst instantiation
```

## 詳細情報

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

## LUT2

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



### 概要

このデザイン エLEMENTは一般出力 (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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

### デザインの入力方法

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



## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT2: 2-input Look-Up Table with general output
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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

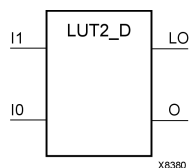
// End of LUT2_inst instantiation
```

## 詳細情報

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

## LUT2\_D

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



### 概要

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

出力 O は一般的なインターコネクトです。LO 出力は同じ CLB スライス内の別の出力、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 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法 :** LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

### デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT2_D: 2-input Look-Up Table with general and local outputs
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

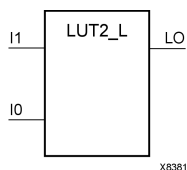
// End of LUT2_D_inst instantiation
```

## 詳細情報

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

## LUT2\_L

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



### 概要

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

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

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

**論理表を使用する方法 :** LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

### デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT2_L: 2-input Look-Up Table with local output
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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

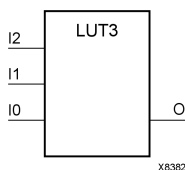
// End of LUT2_L_inst instantiation
```

## 詳細情報

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

## LUT3

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



## 概要

このデザイン エLEMENTは一般出力 (O) を持つ 3 ビットのルックアップ テーブル (LUT) です。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。このELEMENTは、バッファまたはインバーターの機能を果たします。これらのELEMENTは基本ブロックで、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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

## 論理表

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- LUT3: 3-input Look-Up Table with general output
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

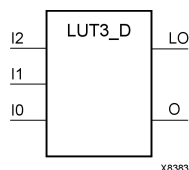
// End of LUT3_inst instantiation
```

## 詳細情報

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

## LUT3\_D

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



### 概要

このデザイン エレメントは 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 は一般的なインターコネクトです。LO 出力は同じ CLB スライス内の別の出力、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 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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



## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

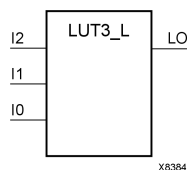
// End of LUT3_D_inst instantiation
```

## 詳細情報

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

## LUT3\_L

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



### 概要

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

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

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

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- LUT3_L: 3-input Look-Up Table with local output
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

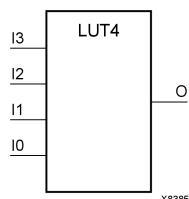
// End of LUT3_L_inst instantiation
```

## 詳細情報

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

## LUT4

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



### 概要

このデザイン エLEMENTは一般出力 (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。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4: 4-input Look-Up Table with general output
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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

-- End of LUT4_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// LUT4: 4-input Look-Up Table with general output
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

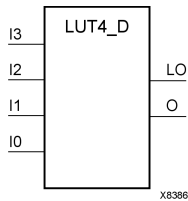
// End of LUT4_inst instantiation
```

## 詳細情報

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

## LUT4\_D

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



### 概要

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

出力 O は一般的なインターコネクトです。LO 出力は同じ CLB スライス内の別の出力、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 値を設定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_D: 4-input Look-Up Table with general and local outputs
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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

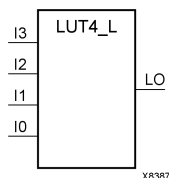
// End of LUT4_D_inst instantiation
```

## 詳細情報

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

## LUT4\_L

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



### 概要

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

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

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

**論理表を使用する方法 :** LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_L: 4-input Look-Up Table with local output
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

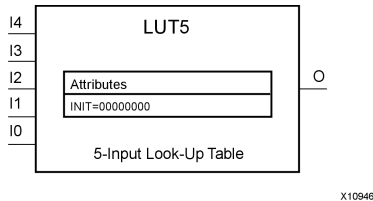
// End of LUT4_L_inst instantiation
```

## 詳細情報

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

## LUT5

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



### 概要

このデザイン エLEMENTは、入力 5 個、出力 1 個のルックアップ 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) として動作するか、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で、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

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

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

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

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

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

## ポートの説明

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

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- LUT5: 5-input Look-Up Table with general output
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

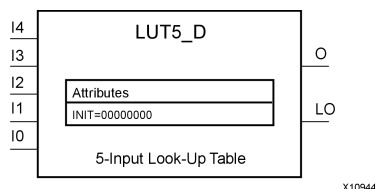
// End of LUT5_inst instantiation
```

## 詳細情報

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

## LUT5\_D

### プリミティブ：5-Input Lookup Table with General and Local Outputs



### 概要

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

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

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

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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



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

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

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	5 入力 LUT 出力
LO	出力	1	内部 CLB 接続用の 5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- LUT5_D: 5-input Look-Up Table with general and local outputs
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

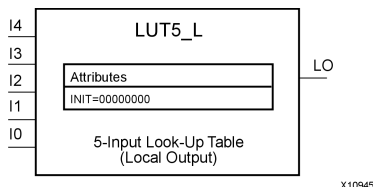
// End of LUT5_D_inst instantiation
```

## 詳細情報

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

## LUT5\_L

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



### 概要

このデザイン エレメントは、入力 5 個、出力 1 個のルックアップ 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) として動作するか、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。入力に適used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、入力すべてがゼロの場合以外は出力は 1 になります (5 入力 OR ゲート)。

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

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。バイナリの真理値表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

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

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

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

## ポートの説明

ポート名	方向	幅	機能
L0	出力	1	内部 CLB 接続用の 6/5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力

## デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- LUT5_L: 5-input Look-Up Table with local output
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

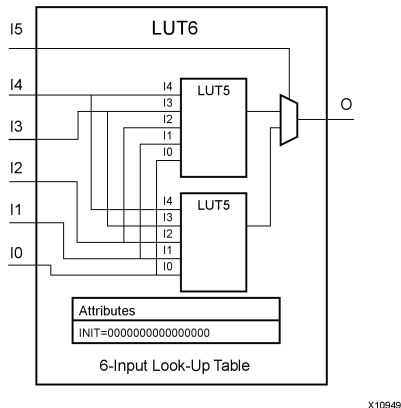
// End of LUT5_L_inst instantiation
```

## 詳細情報

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

## LUT6

### プリミティブ：6-Input Lookup Table with General Output



### 概要

このデザイン エLEMENTは、入力 6 個、出力 1 個のルックアップ 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) として動作するか、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のので、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数値を設定する必要があります。入力 が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を `64'h8000000000000000` (VHDL では `X"8000000000000000"`) に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値を `64'hffffffffffffff` (VHDL では `X"FFFFFFFFFFFFFFFF"`) に設定すると、入力すべてがゼロの場合以外は出力は 1 になります (6 入力 OR ゲート)。

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

**論理表を使用する方法：** LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

入力						出力
I5	I4	I3	I2	I1	I0	O
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]
0	0	0	0	1	0	INIT[2]

入力						出力
I5	I4	I3	I2	I1	I0	O
0	0	0	0	1	1	INIT[3]
0	0	0	1	0	0	INIT[4]
0	0	0	1	0	1	INIT[5]
0	0	0	1	1	0	INIT[6]
0	0	0	1	1	1	INIT[7]
0	0	1	0	0	0	INIT[8]
0	0	1	0	0	1	INIT[9]
0	0	1	0	1	0	INIT[10]
0	0	1	0	1	1	INIT[11]
0	0	1	1	0	0	INIT[12]
0	0	1	1	0	1	INIT[13]
0	0	1	1	1	0	INIT[14]
0	0	1	1	1	1	INIT[15]
0	1	0	0	0	0	INIT[16]
0	1	0	0	0	1	INIT[17]
0	1	0	0	1	0	INIT[18]
0	1	0	0	1	1	INIT[19]
0	1	0	1	0	0	INIT[20]
0	1	0	1	0	1	INIT[21]
0	1	0	1	1	0	INIT[22]
0	1	0	1	1	1	INIT[23]
0	1	1	0	0	0	INIT[24]
0	1	1	0	0	1	INIT[25]
0	1	1	0	1	0	INIT[26]
0	1	1	0	1	1	INIT[27]
0	1	1	1	0	0	INIT[28]
0	1	1	1	0	1	INIT[29]
0	1	1	1	1	0	INIT[30]
0	1	1	1	1	1	INIT[31]
1	0	0	0	0	0	INIT[32]
1	0	0	0	0	1	INIT[33]
1	0	0	0	1	0	INIT[34]
1	0	0	0	1	1	INIT[35]
1	0	0	1	0	0	INIT[36]
1	0	0	1	0	1	INIT[37]



入力						出力
I5	I4	I3	I2	I1	I0	O
1	0	0	1	1	0	INIT[38]
1	0	0	1	1	1	INIT[39]
1	0	1	0	0	0	INIT[40]
1	0	1	0	0	1	INIT[41]
1	0	1	0	1	0	INIT[42]
1	0	1	0	1	1	INIT[43]
1	0	1	1	0	0	INIT[44]
1	0	1	1	0	1	INIT[45]
1	0	1	1	1	0	INIT[46]
1	0	1	1	1	1	INIT[47]
1	1	0	0	0	0	INIT[48]
1	1	0	0	0	1	INIT[49]
1	1	0	0	1	0	INIT[50]
1	1	0	0	1	1	INIT[51]
1	1	0	1	0	0	INIT[52]
1	1	0	1	0	1	INIT[53]
1	1	0	1	1	0	INIT[54]
1	1	0	1	1	1	INIT[55]
1	1	1	0	0	0	INIT[56]
1	1	1	0	0	1	INIT[57]
1	1	1	0	1	0	INIT[58]
1	1	1	0	1	1	INIT[59]
1	1	1	1	0	0	INIT[60]
1	1	1	1	0	1	INIT[61]
1	1	1	1	1	0	INIT[62]
1	1	1	1	1	1	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値						

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	6/5 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- LUT6: 6-input Look-Up Table with general output
-- Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

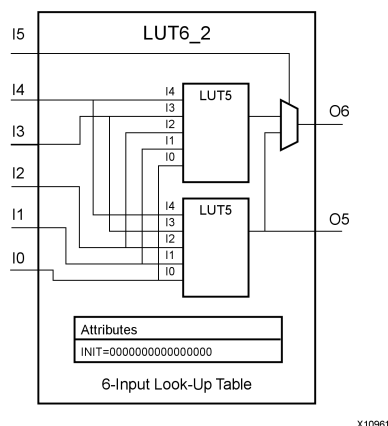
// End of LUT6_inst instantiation
```

## 詳細情報

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

## LUT6\_2

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



### 概要

このデザイン エLEMENTは、入力 6 個、出力 2 個のルックアップ 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) として動作するか、入力を共有する 5 入力のロジック ファンクション 2 つ、または入力とロジック 値を共有する 6 入力および 5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブ ロックで、デザインに含まれる多くのロジック ファンクションのインプリメントに使用されます。LUT6\_2 は、スライスに含ま れる 4 個のルックアップ テーブル (LUT) のいずれかにマップされます。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数を設定する必要があります。入力 が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば、Verilog で INIT 値を 64'hXXXXXXXXXXXXXXXX に設定すると、入力がすべてゼロの場合以外は O6 出力は 1 になり、[4:0] がすべてゼロの場合以外は O5 出力は 1 になります (5 または 6 入力の OR ゲート)。INIT 値の下位半 分 (ビット 31:0) は O5 出力のロジック ファンクションに適用されます。

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

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出 力のロジック値を指定し、これらの出力値から、初期値を作成します。

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

### 論理表

入力						出力	
I5	I4	I3	I2	I1	I0	O5	O6
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	0	1	1	INIT[3]	INIT[3]
0	0	0	1	0	0	INIT[4]	INIT[4]

入力						出力	
0	0	0	1	0	1	INIT[5]	INIT[5]
0	0	0	1	1	0	INIT[6]	INIT[6]
0	0	0	1	1	1	INIT[7]	INIT[7]
0	0	1	0	0	0	INIT[8]	INIT[8]
0	0	1	0	0	1	INIT[9]	INIT[9]
0	0	1	0	1	0	INIT[10]	INIT[10]
0	0	1	0	1	1	INIT[11]	INIT[11]
0	0	1	1	0	0	INIT[12]	INIT[12]
0	0	1	1	0	1	INIT[13]	INIT[13]
0	0	1	1	1	0	INIT[14]	INIT[14]
0	0	1	1	1	1	INIT[15]	INIT[15]
0	1	0	0	0	0	INIT[16]	INIT[16]
0	1	0	0	0	1	INIT[17]	INIT[17]
0	1	0	0	1	0	INIT[18]	INIT[18]
0	1	0	0	1	1	INIT[19]	INIT[19]
0	1	0	1	0	0	INIT[20]	INIT[20]
0	1	0	1	0	1	INIT[21]	INIT[21]
0	1	0	1	1	0	INIT[22]	INIT[22]
0	1	0	1	1	1	INIT[23]	INIT[23]
0	1	1	0	0	0	INIT[24]	INIT[24]
0	1	1	0	0	1	INIT[25]	INIT[25]
0	1	1	0	1	0	INIT[26]	INIT[26]
0	1	1	0	1	1	INIT[27]	INIT[27]
0	1	1	1	0	0	INIT[28]	INIT[28]
0	1	1	1	0	1	INIT[29]	INIT[29]
0	1	1	1	1	0	INIT[30]	INIT[30]
0	1	1	1	1	1	INIT[31]	INIT[31]
1	0	0	0	0	0	INIT[0]	INIT[32]
1	0	0	0	0	1	INIT[1]	INIT[33]
1	0	0	0	1	0	INIT[2]	INIT[34]
1	0	0	0	1	1	INIT[3]	INIT[35]
1	0	0	1	0	0	INIT[4]	INIT[36]
1	0	0	1	0	1	INIT[5]	INIT[37]
1	0	0	1	1	0	INIT[6]	INIT[38]
1	0	0	1	1	1	INIT[7]	INIT[39]
1	0	1	0	0	0	INIT[8]	INIT[40]

入力						出力	
1	0	1	0	0	1	INIT[9]	INIT[41]
1	0	1	0	1	0	INIT[10]	INIT[42]
1	0	1	0	1	1	INIT[11]	INIT[43]
1	0	1	1	0	0	INIT[12]	INIT[44]
1	0	1	1	0	1	INIT[13]	INIT[45]
1	0	1	1	1	0	INIT[14]	INIT[46]
1	0	1	1	1	1	INIT[15]	INIT[47]
1	1	0	0	0	0	INIT[16]	INIT[48]
1	1	0	0	0	1	INIT[17]	INIT[49]
1	1	0	0	1	0	INIT[18]	INIT[50]
1	1	0	0	1	1	INIT[19]	INIT[51]
1	1	0	1	0	0	INIT[20]	INIT[52]
1	1	0	1	0	1	INIT[21]	INIT[53]
1	1	0	1	1	0	INIT[22]	INIT[54]
1	1	0	1	1	1	INIT[23]	INIT[55]
1	1	1	0	0	0	INIT[24]	INIT[56]
1	1	1	0	0	1	INIT[25]	INIT[57]
1	1	1	0	1	0	INIT[26]	INIT[58]
1	1	1	0	1	1	INIT[27]	INIT[59]
1	1	1	1	0	0	INIT[28]	INIT[60]
1	1	1	1	0	1	INIT[29]	INIT[61]
1	1	1	1	1	0	INIT[30]	INIT[62]
1	1	1	1	1	1	INIT[31]	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値							

## ポートの説明

ポート名	方向	幅	機能
O6	出力	1	6/5 LUT 出力
O5	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	LUT5/6 の出力ファンクションを指定します。

## VHDL 記述 (インスタンスレーション)

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

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

-- LUT6_2: 6-input 2 output Look-Up Table
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

LUT6_2_inst : LUT6_2
generic map (
  INIT => X"0000000000000000") -- Specify LUT Contents
port map (
  O6 => O6, -- 6/5-LUT output (1-bit)
  O5 => O5, -- 5-LUT output (1-bit)
  I0 => I0, -- LUT input (1-bit)
  I1 => I1, -- LUT input (1-bit)
  I2 => I2, -- LUT input (1-bit)
  I3 => I3, -- LUT input (1-bit)
  I4 => I4, -- LUT input (1-bit)
  I5 => I5  -- LUT input (1-bit)
);

-- End of LUT6_2_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// LUT6_2: 6-input, 2 output Look-Up Table
//          Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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)
);

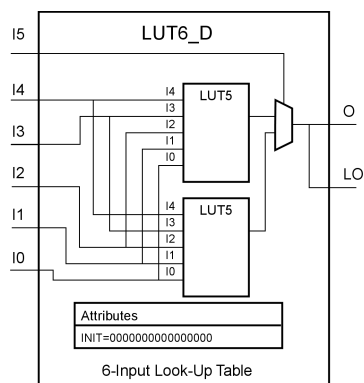
// End of LUT6_2_inst instantiation
```

## 詳細情報

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

## LUT6\_D

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



X10947

## 概要

このデザイン エLEMENTは、入力 6 個、出力 1 個のルックアップ 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) として動作するか、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のので、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数値を設定する必要があります。入力 が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を 64'h8000000000000000 (VHDL では X"8000000000000000") に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値を 64'hffffffff (VHDL では X"FFFFFFFFFFFFFFFF") に設定すると、入力すべてがゼロの場合以外は出力は 1 になります (6 入力 OR ゲート)。

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

**論理表を使用する方法：**LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

**論理式を使用する方法：**リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、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	すべてゼロ	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- LUT6_D: 6-input Look-Up Table with general and local outputs
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

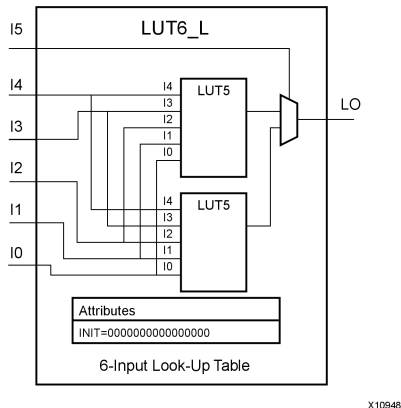
// End of LUT6_D_inst instantiation
```

## 詳細情報

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

## LUT6\_L

**プリミティブ：6-Input Lookup Table with Local Output**



### 概要

このデザイン エLEMENTは、入力 6 個、出力 1 個のルックアップ 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) として動作するか、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のので、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がある場合以外は、常に使用する必要があります。

LUT のロジック ファンクションを指定するため、INIT 属性に 64 ビットの 16 進数値を設定する必要があります。入力 が適用されたときに出力される値を、その入力に対応する INIT ビットに設定します。たとえば Verilog で INIT 値を `64'h8000000000000000` (VHDL では `X"8000000000000000"`) に設定すると、入力すべてが 1 の場合以外は出力は 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値を `64'hffffffffffffff` (VHDL では `X"FFFFFFFFFFFFFFFF"`) に設定すると、入力すべてがゼロの場合以外は出力は 1 になります (6 入力 OR ゲート)。

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

**論理表を使用する方法：** LUT の INIT 値を決定する一般的な方法。バイナリの真理値表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

**論理式を使用する方法：** リストされた論理表の値に対応する LUT の各入力にパラメーターを定義し、パラメーターを基にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単ですが、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	すべてゼロ	ルックアップ テーブルの論理値を指定します。

## VHDL 記述 (インスタンス化)

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

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

-- LUT6_L: 6-input Look-Up Table with local output
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
```

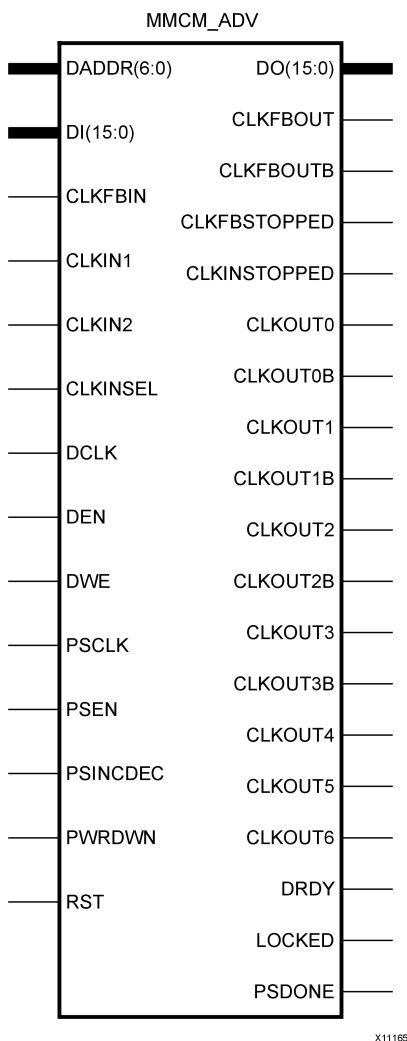


## 詳細情報

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

## MMCM\_ADV

**プリミティブ：** MMCM is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.



### 概要

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

### ポートの説明

ポート名	方向	幅	機能
CLKFBIN	入力	1	クロック フィードバック入力
CLKFBOUT	出力	1	専用 MMCM フィードバック出力
CLKFBOUTB	出力	1	CLKFBOUT を反転したクロック出力
CLKFBSTOPPED	出力	1	フィードバック クロックが停止したことを示すステータス ビン

ポート名	方向	幅	機能
CLKINSEL	入力	1	入力マルチプレクサーのステートを制御する信号で、High の場合は CLKIN1、Low の場合は CLKIN2 です。
CLKINSTOPPED	出力	1	入力クロックが停止したことを示すステータス ビン
CLKIN1	入力	1	汎用クロック入力
CLKIN2	入力	1	MMCM リファレンス クロックのセカンダリ クロック入力
CLKOUT[0:6]	出力	7、1 ビット	コンフィギュレーション可能なクロック出力 (0 ~ 6) で、VCO 位相出力 (ユーザー制御可能) を 1 (バイパス) から 128 までの値で分周したものに設定できます。出力クロックは、位相シフトされていない場合はお互いに位相が揃っており、適切なフィードバック コンフィギュレーションにより入力クロックに揃えられます。
CLKOUT[0:3]B	出力	4、1 ビット	CLKOUT[0:3] を反転したものです。
DADDR[6:0]	入力	7	ダイナミック リコンフィギュレーション用のリコンフィギュレーション アドレスを供給する入力バス。このバスを使用しない場合は、すべてのビットを 0 にする必要があります。
DCLK	入力	1	ダイナミック リコンフィギュレーション ポートのリファレンス クロック
DEN	入力	1	ダイナミック リコンフィギュレーション機能にアクセスするためのイネーブル制御信号。ダイナミック リコンフィギュレーションを使用しない場合は Low に接続する必要があります。
DI[15:0]	入力	16	リコンフィギュレーション データを供給するデータ入力バス。このバスを使用しない場合は、すべてのビットを 0 にする必要があります。
DO[15:0]	出力	16	ダイナミック リコンフィギュレーションを使用する場合の MMCM データ出力バス
DRDY	出力	1	MMCM のダイナミック リコンフィギュレーション機能の DEN 信号への応答を供給する READY 出力
DWE	入力	1	DADDR アドレスへの DI データの書き込みを制御するライト イネーブル信号。使用しない場合は、Low に接続する必要があります。
LOCKED	出力	1	位相アライメントが定義されている時間内で完了し、周波数が定義されている PPM 範囲内で一致したことを示します。MMCM は電源投入時に自動的にロック状態になるので、リセットは不要です。入力クロックが停止した場合、または位相アライメントに違反が発生した場合 (入力クロックの位相シフトなど) は、LOCKED がディアサートされます。LOCKED がディアサートされると、自動的にロックが達成されます。
PSCLK	入力	1	位相シフト クロック
PSDONE	出力	1	位相シフト終了
PSEN	入力	1	位相シフト イネーブル
PSINCDEC	入力	1	位相シフト インクリメント/デクリメント制御
PWRDWN	入力	1	インスタンス化されているが未使用の MMCM をパワー ダウンします。
RST	入力	1	非同期リセット信号。この信号が解放されると、MMCM はクロックに同期して再びイネーブルになります。入力クロックの条件 (周波数など) を変更する場合、リセットは不要です。

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
BANDWIDTH	文字列	"OPTIMIZED"、 "HIGH"、 "LOW"	"OPTIMIZED"	ジッター、位相マージンなどの MMCM 特性に影響する MMCM プログラム アルゴリズムを指定します。
CLKFBOUT_MULT_F	3 上位ビット 浮動小数点	5.0 ~ 64.0	5.0	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。この値は実数で指定する必要がありますが、整数値のみがサポートされます。たとえば、6.0 はサポートされますが 6.5 はサポートされません。
CLKFBOUT_PHASE	3 上位ビット 浮動小数点	-360,000 ~ 360,000	0.000	クロック フィードバック出力の位相オフセットを度数で指定します。フィードバック クロックをシフトすると、MMCM の出力クロックがすべて負の値に位相シフトします。
CLKIN_PERIOD	フロート (nS)	1.000 ~ 100.000	0.000	CLKIN1 の入力周期を指定します。精度は ps です。この値は必ず設定する必要があります。
CLKOUT0_DIVIDE_F	3 上位ビット 浮動小数点	1.000 ~ 128.000	1.000	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。
CLKOUT[0:6]_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。
CLKOUT[0:6]_DUTY_CYCLE	3 上位ビット 浮動小数点	0.001 ~ 0.999	0.500	CLKOUT クロック出力のデューティ サイクルをパーセントで指定します。0.50 の場合、デューティ サイクルは 50% になります。
CLKOUT[0:6]_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	クロック フィードバック出力の位相オフセットを度数で指定します。フィードバック クロックをシフトすると、MMCM の出力クロックがすべて負の値に位相シフトします。
CLKOUT4_CASCADE	ブール代数	FALSE、TRUE	FALSE	出力クロック分周が 128 よりも大きい場合、出力分周 (カウンタ) を CLKOUT4 分周の入力にカスケードします。
CLOCK_HOLD	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、VCO 周波数が CLKIN を損失する前の周波数に保持されます。

属性	データ型	値	デフォルト	説明
COMPENSATION	文字列	"ZHOLD"、 "BUF_IN"、 "CASCADE"、 "EXTERNAL"、 "INTERNAL"	"ZHOLD"	<p>クロック入力の補正。"ZHOLD" に設定する必要があります。MMCM フィードバックのコンフィギュレーション方法を定義します。</p> <ul style="list-style-type: none"> <li>"ZHOLD"：I/O レジスタでのホールド タイムが負になるように MMCM をコンフィギュレーションします。</li> <li>"INTERNAL"：MMCM の内部フィードバックパスが使用され、遅延は調整されません。</li> <li>"EXTERNAL"：FPGA の外部ネットワークが調整されます。</li> <li>"CASCADE"：2 つの MMCM がカスケード接続されます。</li> <li>"BUF_IN"：その他の補正モードには一致せず、遅延は補正されません。クロック入力が BUFG/BUFH/BUFR/GT で駆動される場合です。</li> </ul>
DIVCLK_DIVIDE	整数	1 ～ 128	1	すべての出力クロックの入力クロックに対する分周比を指定し、PFD に入力される CLKIN を分周します。
REF_JITTER1	3 上位ビット 浮動小数点	0.000 ～ 0.999	0.010	MMCM のパフォーマンスを最適化するため、CLKIN1 に予測されるジッター値を指定します。BANDWIDTH が "OPTIMIZED" に設定されている場合、値が既知でない場合は入力クロックに最適なパラメーターが選択されます。値が既知である場合は、値を入力クロックに予測されるジッターの UI パーセント (最大ピークトゥピーク値) で指定する必要があります。
REF_JITTER2	3 上位ビット 浮動小数点	0.000 ～ 0.999	0.010	MMCM のパフォーマンスを最適化するため、CLKIN2 に予測されるジッター値を指定します。BANDWIDTH が "OPTIMIZED" に設定されている場合、値が既知でない場合は入力クロックに最適なパラメーターが選択されます。値が既知である場合は、値を入力クロックに予測されるジッターの UI パーセント (最大ピークトゥピーク値) で指定する必要があります。
STARTUP_WAIT	ブール代数	FALSE	FALSE	この属性はサポートされていません。
CLKFBOUT_USE_FINE_PS	ブール代数	FALSE、TRUE	FALSE	CLKFBOUT カウンター ファイン可変位相シフト イネーブル
CLKOUT[0:6]_USE_FINE_PS	ブール代数	FALSE、TRUE	FALSE	CLKOUT[1:6] の可変ファイン位相シフト イネーブル

## VHDL 記述 (インスタンス化)

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;
```

```

-- MMCM_ADV: Advanced Mixed Mode Clock Manager
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

MMCM_ADV_inst : MMCM_ADV
generic map (
    BANDWIDTH => "OPTIMIZED",          -- Jitter programming ("HIGH","LOW","OPTIMIZED")
    CLKFBOUT_MULT_F => 5.0,            -- Multiply value for all CLKOUT (5.0-64.0).
    CLKFBOUT_PHASE => 0.0,             -- Phase offset in degrees of CLKFB (0.00-360.00).
    -- CLKIN_PERIOD: Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    CLKIN1_PERIOD => 0.0,
    CLKIN2_PERIOD => 0.0,
    CLKOUT0_DIVIDE_F => 1.0,           -- Divide amount for CLKOUT0 (1.000-128.000).
    -- CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for CLKOUT outputs (0.01-0.99).
    CLKOUT0_DUTY_CYCLE => 0.5,
    CLKOUT1_DUTY_CYCLE => 0.5,
    CLKOUT2_DUTY_CYCLE => 0.5,
    CLKOUT3_DUTY_CYCLE => 0.5,
    CLKOUT4_DUTY_CYCLE => 0.5,
    CLKOUT5_DUTY_CYCLE => 0.5,
    CLKOUT6_DUTY_CYCLE => 0.5,
    -- CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for CLKOUT outputs (-360.000-360.000).
    CLKOUT0_PHASE => 0.0,
    CLKOUT1_PHASE => 0.0,
    CLKOUT2_PHASE => 0.0,
    CLKOUT3_PHASE => 0.0,
    CLKOUT4_PHASE => 0.0,
    CLKOUT5_PHASE => 0.0,
    CLKOUT6_PHASE => 0.0,
    -- CLKOUT1_DIVIDE - CLKOUT6_DIVIDE: Divide amount for CLKOUT (1-128)
    CLKOUT1_DIVIDE => 1,
    CLKOUT2_DIVIDE => 1,
    CLKOUT3_DIVIDE => 1,
    CLKOUT4_DIVIDE => 1,
    CLKOUT5_DIVIDE => 1,
    CLKOUT6_DIVIDE => 1,
    CLKOUT4_CASCADE => FALSE,          -- Cascade CLKOUT4 counter with CLKOUT6 (TRUE/FALSE)
    CLOCK_HOLD => FALSE,              -- Hold VCO Frequency (TRUE/FALSE)
    COMPENSATION => "ZHOLD",           -- "ZHOLD", "INTERNAL", "EXTERNAL", "CASCADE" or "BUF_IN"
    DIVCLK_DIVIDE => 1,               -- Master division value (1-80)
    -- REF_JITTER: Reference input jitter in UI (0.000-0.999).
    REF_JITTER1 => 0.0,
    REF_JITTER2 => 0.0,
    STARTUP_WAIT => FALSE,            -- Not supported. Must be set to FALSE.
    -- USE_FINE_PS: Fine phase shift enable (TRUE/FALSE)
    CLKFBOUT_USE_FINE_PS => FALSE,
    CLKOUT0_USE_FINE_PS => FALSE,
    CLKOUT1_USE_FINE_PS => FALSE,
    CLKOUT2_USE_FINE_PS => FALSE,
    CLKOUT3_USE_FINE_PS => FALSE,
    CLKOUT4_USE_FINE_PS => FALSE,
    CLKOUT5_USE_FINE_PS => FALSE,
    CLKOUT6_USE_FINE_PS => FALSE
)
port map (
    -- Clock Outputs: 1-bit (each) output: User configurable clock outputs
    CLKOUT0 => CLKOUT0,               -- 1-bit output: CLKOUT0 output
    CLKOUT0B => CLKOUT0B,             -- 1-bit output: Inverted CLKOUT0 output
    CLKOUT1 => CLKOUT1,               -- 1-bit output: CLKOUT1 output
    CLKOUT1B => CLKOUT1B,             -- 1-bit output: Inverted CLKOUT1 output
    CLKOUT2 => CLKOUT2,               -- 1-bit output: CLKOUT2 output
    CLKOUT2B => CLKOUT2B,             -- 1-bit output: Inverted CLKOUT2 output
    CLKOUT3 => CLKOUT3,               -- 1-bit output: CLKOUT3 output
    CLKOUT3B => CLKOUT3B,             -- 1-bit output: Inverted CLKOUT3 output
    CLKOUT4 => CLKOUT4,               -- 1-bit output: CLKOUT4 output
    CLKOUT5 => CLKOUT5,               -- 1-bit output: CLKOUT5 output
    CLKOUT6 => CLKOUT6,               -- 1-bit output: CLKOUT6 output
    -- DRP Ports: 16-bit (each) output: Dynamic reconfiguration ports
    DO => DO,                         -- 16-bit output: DRP data output
    DRDY => DRDY,                     -- 1-bit output: DRP ready output
    -- Dynamic Phase Shift Ports: 1-bit (each) output: Ports used for dynamic phase shifting of the outputs
    PSDONE => PSDONE,                 -- 1-bit output: Phase shift done output

```

```

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

-- End of MMCM_ADV_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```

// MMCM_ADV: Advanced Mixed Mode Clock Manager
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

MMCM_ADV #(
    .BANDWIDTH("OPTIMIZED"),      // Jitter programming ("HIGH","LOW","OPTIMIZED")
    .CLKFBOUT_MULT_F(5.0),        // Multiply value for all CLKOUT (5.0-64.0).
    .CLKFBOUT_PHASE(0.0),         // Phase offset in degrees of CLKFB (0.00-360.00).
    // CLKIN_PERIOD: Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    .CLKIN1_PERIOD(0.0),
    .CLKIN2_PERIOD(0.0),
    .CLKOUT0_DIVIDE_F(1.0),        // Divide amount for CLKOUT0 (1.000-128.000).
    // CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for CLKOUT outputs (0.01-0.99).
    .CLKOUT0_DUTY_CYCLE(0.5),
    .CLKOUT1_DUTY_CYCLE(0.5),
    .CLKOUT2_DUTY_CYCLE(0.5),
    .CLKOUT3_DUTY_CYCLE(0.5),
    .CLKOUT4_DUTY_CYCLE(0.5),
    .CLKOUT5_DUTY_CYCLE(0.5),
    .CLKOUT6_DUTY_CYCLE(0.5),
    // CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for CLKOUT outputs (-360.000-360.000).
    .CLKOUT0_PHASE(0.0),
    .CLKOUT1_PHASE(0.0),
    .CLKOUT2_PHASE(0.0),
    .CLKOUT3_PHASE(0.0),
    .CLKOUT4_PHASE(0.0),
    .CLKOUT5_PHASE(0.0),
    .CLKOUT6_PHASE(0.0),
    // CLKOUT1_DIVIDE - CLKOUT6_DIVIDE: Divide amount for CLKOUT (1-128)
    .CLKOUT1_DIVIDE(1),
    .CLKOUT2_DIVIDE(1),
    .CLKOUT3_DIVIDE(1),
    .CLKOUT4_DIVIDE(1),
    .CLKOUT5_DIVIDE(1),
    .CLKOUT6_DIVIDE(1),
    .CLKOUT4_CASCADE("FALSE"),    // Cascade CLKOUT4 counter with CLKOUT6 (TRUE/FALSE)
    .CLOCK_HOLD("FALSE"),        // Hold VCO Frequency (TRUE/FALSE)
    .COMPENSATION("ZHOLD"),       // "ZHOLD", "INTERNAL", "EXTERNAL", "CASCADE" or "BUF_IN"
    .DIVCLK_DIVIDE(1),           // Master division value (1-80)

```

```

// REF_JITTER: Reference input jitter in UI (0.000-0.999).
.REF_JITTER1(0.0),
.REF_JITTER2(0.0),
.STARTUP_WAIT("FALSE"),           // Not supported. Must be set to FALSE.
// USE_FINE_PS: Fine phase shift enable (TRUE/FALSE)
.CLKFBOUT_USE_FINE_PS("FALSE"),
.CLKOUT0_USE_FINE_PS("FALSE"),
.CLKOUT1_USE_FINE_PS("FALSE"),
.CLKOUT2_USE_FINE_PS("FALSE"),
.CLKOUT3_USE_FINE_PS("FALSE"),
.CLKOUT4_USE_FINE_PS("FALSE"),
.CLKOUT5_USE_FINE_PS("FALSE"),
.CLKOUT6_USE_FINE_PS("FALSE")
)
MMCM_ADV_inst (
// Clock Outputs: 1-bit (each) output: User configurable clock outputs
.CLKOUT0(CLKOUT0),                // 1-bit output: CLKOUT0 output
.CLKOUT0B(CLKOUT0B),              // 1-bit output: Inverted CLKOUT0 output
.CLKOUT1(CLKOUT1),                // 1-bit output: CLKOUT1 output
.CLKOUT1B(CLKOUT1B),              // 1-bit output: Inverted CLKOUT1 output
.CLKOUT2(CLKOUT2),                // 1-bit output: CLKOUT2 output
.CLKOUT2B(CLKOUT2B),              // 1-bit output: Inverted CLKOUT2 output
.CLKOUT3(CLKOUT3),                // 1-bit output: CLKOUT3 output
.CLKOUT3B(CLKOUT3B),              // 1-bit output: Inverted CLKOUT3 output
.CLKOUT4(CLKOUT4),                // 1-bit output: CLKOUT4 output
.CLKOUT5(CLKOUT5),                // 1-bit output: CLKOUT5 output
.CLKOUT6(CLKOUT6),                // 1-bit output: CLKOUT6 output
// DRP Ports: 16-bit (each) output: Dynamic reconfiguration ports
.DO(DO),                          // 16-bit output: DRP data output
.DRDY(DRDY),                      // 1-bit output: DRP ready output
// Dynamic Phase Shift Ports: 1-bit (each) output: Ports used for dynamic phase shifting of the outputs
.PSDONE(PSDONE),                  // 1-bit output: Phase shift done output
// Feedback Clocks: 1-bit (each) output: Clock feedback ports
.CLKFBOUT(CLKFBOUT),              // 1-bit output: Feedback clock output
.CLKFBOUTB(CLKFBOUTB),            // 1-bit output: Inverted CLKFBOUT
// Status Ports: 1-bit (each) output: MMCM status ports
.CLKFBSTOPPED(CLKFBSTOPPED),      // 1-bit output: Feedback clock stopped output
.CLKINSTOPPED(CLKINSTOPPED),      // 1-bit output: Input clock stopped output
.LOCKED(LOCKED),                  // 1-bit output: LOCK output
// Clock Inputs: 1-bit (each) input: Clock inputs
.CLKIN1(CLKIN1),                  // 1-bit input: Primary clock input
.CLKIN2(CLKIN2),                  // 1-bit input: Secondary clock input
// Control Ports: 1-bit (each) input: MMCM control ports
.CLKINSEL(CLKINSEL),              // 1-bit input: Clock select input
.PWRDWN(PWRDWN),                  // 1-bit input: Power-down input
.RST(RST),                        // 1-bit input: Reset input
// DRP Ports: 7-bit (each) input: Dynamic reconfiguration ports
.DADDR(DADDR),                    // 7-bit input: DRP address input
.DCLK(DCLK),                      // 1-bit input: DRP clock input
.DEN(DEN),                        // 1-bit input: DRP enable input
.DI(DI),                          // 16-bit input: DRP data input
.DWE(DWE),                        // 1-bit input: DRP write enable input
// Dynamic Phase Shift Ports: 1-bit (each) input: Ports used for dynamic phase shifting of the outputs
.PSCLK(PCLK),                     // 1-bit input: Phase shift clock input
.PSEN(PSEN),                      // 1-bit input: Phase shift enable input
.PSINCDEC(PSINCDEC),              // 1-bit input: Phase shift increment/decrement input
// Feedback Clocks: 1-bit (each) input: Clock feedback ports
.CLKFBIN(CLKFBIN)                 // 1-bit input: Feedback clock input
);

// End of MMCM_ADV_inst instantiation

```

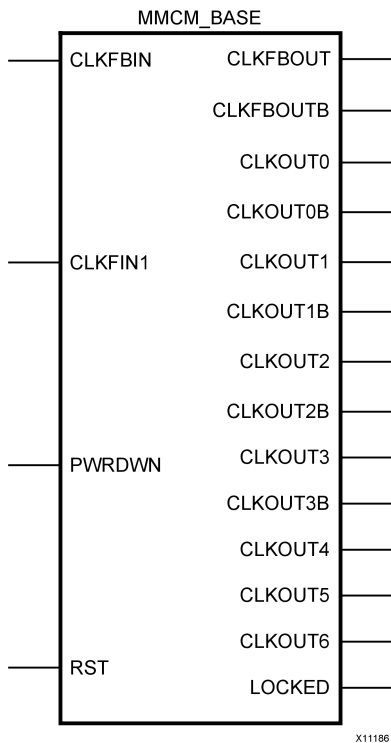
## 詳細情報

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



## MMCM\_BASE

**コンビニエンス プリミティブ**：Mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.



## 概要

このコンポーネントは、クロック ネットワークのスキュー調整、周波数合成、ジッター低減をサポートするための混合信号ブロックです。O0 を 2 で分周、O1 を 3 で分周するようプログラムするなど、7 つの O カウンターを個別にプログラムできます。ただし、1 つの VCO ですべてのカウンターを駆動するため、VCO 動作周波数をすべての出力カウンターに対して同じにする必要があります。CLKFBOUT および CLKFBOUTB はロジックを駆動するのに使用できますが、CLKIN 周波数と同じにする必要があります。

## ポートの説明

ポート名	方向	幅	機能
CLKFBIN	入力	1	クロック フィードバック入力
CLKFBOUT	出力	1	専用 MMCM フィードバック出力
CLKFBOUTB	出力	1	反転した MMCM フィードバック クロック出力
CLKIN1	入力	1	汎用クロック入力
CLKOUT[0:6]	出力	7、1 ビット	コンフィギュレーション可能なクロック出力 (0 ~ 6) で、VCO 位相出力 (ユーザー制御可能) を 1 (バイパス) から 128 までの値で分周したものに設定できます。出力クロックは、位相シフトされていない場合はお互いに位相が揃っており、適切なフィードバック コンフィギュレーションにより入力クロックに揃えられます。
CLKOUT[0:3]B	出力	4、1 ビット	CLKOUT[0:3] を反転したものです。

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

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
BANDWIDTH	文字列	"OPTIMIZED"、 "HIGH"、 "LOW"	"OPTIMIZED"	ジッター、位相マージンなどの MMCM 特性に影響する MMCM プログラム アルゴリズムを指定します。
CLKFBOUT_MULT_F	3 上位ビット 浮動小数点	5.0 ~ 64.0	5.0	すべての CLKOUT クロック出力を逡倍する値を指定します。この値と、CLKOUT#_DIVIDE 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKFBOUT_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	クロック フィードバック出力の位相オフセットを度数で指定します。フィードバック クロックをシフトすると、MMCM の出力クロックがすべて負の値に位相シフトします。
CLKIN1_PERIOD	フロート (nS)	1.000 ~ 1000.000	0.000	CLKIN1 の入力周期を指定します。精度は ps です。この値は必ず設定する必要があります。
CLKOUT0_DIVIDE_F	3 上位ビット 浮動小数点	1.000 ~ 128.000	1.000	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。
CLKOUT[0:6]_DUTY_CYCLE	3 上位ビット 浮動小数点	0.001 ~ 0.999	0.500	CLKOUT クロック出力のデューティ サイクルをパーセントで指定します。0.50 の場合、デューティ サイクルは 50% になります。
CLKOUT[0:6]_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	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。
CLOCK_HOLD	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、VCO 周波数が CLKIN を損失する前の周波数に保持されます。

属性	データ型	値	デフォルト	説明
DIVCLK_DIVIDE	整数	1 ～ 128	1	すべての出力クロックの入力クロックに対する分周比を指定し、PFD に入力される CLKIN を分周します。
REF_JITTER1	3 上位ビット 浮動小数点	0.000 ～ 0.999	0.010	MMCM パフォーマンスを最適化するため、リファレンス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、値が既知でない場合は入力クロックに最適なパラメーターが選択されます。値が既知である場合は、値を入力クロックに予測されるジッターの UI パーセント (最大ピークトゥピーク値) で指定する必要があります。
STARTUP_WAIT	ブール代数	FALSE	FALSE	この属性はサポートされていません。

## 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;

-- MMCM_BASE: Base Mixed Mode Clock Manager
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

MMCM_BASE_inst : MMCM_BASE
generic map (
    BANDWIDTH => "OPTIMIZED", -- Jitter programming ("HIGH","LOW","OPTIMIZED")
    CLKFBOUT_MULT_F => 5.0,    -- Multiply value for all CLKOUT (5.0-64.0).
    CLKFBOUT_PHASE => 0.0,     -- Phase offset in degrees of CLKFB (0.00-360.00).
    CLKIN1_PERIOD => 0.0,      -- Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    CLKOUT0_DIVIDE_F => 1.0,   -- Divide amount for CLKOUT0 (1.000-128.000).
    -- CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for each CLKOUT (0.01-0.99).
    CLKOUT0_DUTY_CYCLE => 0.5,
    CLKOUT1_DUTY_CYCLE => 0.5,
    CLKOUT2_DUTY_CYCLE => 0.5,
    CLKOUT3_DUTY_CYCLE => 0.5,
    CLKOUT4_DUTY_CYCLE => 0.5,
    CLKOUT5_DUTY_CYCLE => 0.5,
    CLKOUT6_DUTY_CYCLE => 0.5,
    -- CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for each CLKOUT (-360.000-360.000).
    CLKOUT0_PHASE => 0.0,
    CLKOUT1_PHASE => 0.0,
    CLKOUT2_PHASE => 0.0,
    CLKOUT3_PHASE => 0.0,
    CLKOUT4_PHASE => 0.0,
    CLKOUT5_PHASE => 0.0,
    CLKOUT6_PHASE => 0.0,
    -- CLKOUT1_DIVIDE - CLKOUT6_DIVIDE: Divide amount for each CLKOUT (1-128)
    CLKOUT1_DIVIDE => 1,
    CLKOUT2_DIVIDE => 1,
    CLKOUT3_DIVIDE => 1,
    CLKOUT4_DIVIDE => 1,
    CLKOUT5_DIVIDE => 1,
    CLKOUT6_DIVIDE => 1,
    CLKOUT4_CASCADE => FALSE, -- Cascade CLKOUT4 counter with CLKOUT6 (TRUE/FALSE)
    CLOCK_HOLD => FALSE,     -- Hold VCO Frequency (TRUE/FALSE)
    DIVCLK_DIVIDE => 1,      -- Master division value (1-80)
    REF_JITTER1 => 0.0,      -- Reference input jitter in UI (0.000-0.999).
    STARTUP_WAIT => FALSE    -- Not supported. Must be set to FALSE.
)
port map (
    -- Clock Outputs: 1-bit (each) output: User configurable clock outputs
    CLKOUT0 => CLKOUT0,      -- 1-bit output: CLKOUT0 output

```

```
CLKOUT0B => CLKOUT0B,    -- 1-bit output: Inverted CLKOUT0 output
CLKOUT1 => CLKOUT1,      -- 1-bit output: CLKOUT1 output
CLKOUT1B => CLKOUT1B,    -- 1-bit output: Inverted CLKOUT1 output
CLKOUT2 => CLKOUT2,      -- 1-bit output: CLKOUT2 output
CLKOUT2B => CLKOUT2B,    -- 1-bit output: Inverted CLKOUT2 output
CLKOUT3 => CLKOUT3,      -- 1-bit output: CLKOUT3 output
CLKOUT3B => CLKOUT3B,    -- 1-bit output: Inverted CLKOUT3 output
CLKOUT4 => CLKOUT4,      -- 1-bit output: CLKOUT4 output
CLKOUT5 => CLKOUT5,      -- 1-bit output: CLKOUT5 output
CLKOUT6 => CLKOUT6,      -- 1-bit output: CLKOUT6 output
-- Feedback Clocks: 1-bit (each) output: Clock feedback ports
CLKFBOUT => CLKFBOUT,    -- 1-bit output: Feedback clock output
CLKFBOUTB => CLKFBOUTB,  -- 1-bit output: Inverted CLKFBOUT output
-- Status Port: 1-bit (each) output: MMCM status ports
LOCKED => LOCKED,        -- 1-bit output: LOCK output
-- Clock Input: 1-bit (each) input: Clock input
CLKIN1 => CLKIN1,
-- Control Ports: 1-bit (each) input: MMCM control ports
PWRDWN => PWRDWN,        -- 1-bit input: Power-down input
RST => RST,               -- 1-bit input: Reset input
-- Feedback Clocks: 1-bit (each) input: Clock feedback ports
CLKFBIN => CLKFBIN       -- 1-bit input: Feedback clock input
);

-- End of MMCM_BASE_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// MMCM_BASE: Base Mixed Mode Clock Manager
//          Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

MMCM_BASE #(
    .BANDWIDTH("OPTIMIZED"), // Jitter programming ("HIGH","LOW","OPTIMIZED")
    .CLKFBOUT_MULT_F(5.0),   // Multiply value for all CLKOUT (5.0-64.0).
    .CLKFBOUT_PHASE(0.0),    // Phase offset in degrees of CLKFB (0.00-360.00).
    .CLKIN1_PERIOD(0.0),     // Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    .CLKOUT0_DIVIDE_F(1.0),   // Divide amount for CLKOUT0 (1.000-128.000).
    // CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for each CLKOUT (0.01-0.99).
    .CLKOUT0_DUTY_CYCLE(0.5),
    .CLKOUT1_DUTY_CYCLE(0.5),
    .CLKOUT2_DUTY_CYCLE(0.5),
    .CLKOUT3_DUTY_CYCLE(0.5),
    .CLKOUT4_DUTY_CYCLE(0.5),
    .CLKOUT5_DUTY_CYCLE(0.5),
    .CLKOUT6_DUTY_CYCLE(0.5),
    // CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for each CLKOUT (-360.000-360.000).
    .CLKOUT0_PHASE(0.0),
    .CLKOUT1_PHASE(0.0),
    .CLKOUT2_PHASE(0.0),
    .CLKOUT3_PHASE(0.0),
    .CLKOUT4_PHASE(0.0),
    .CLKOUT5_PHASE(0.0),
    .CLKOUT6_PHASE(0.0),
    // CLKOUT1_DIVIDE - CLKOUT6_DIVIDE: Divide amount for each CLKOUT (1-128)
    .CLKOUT1_DIVIDE(1),
    .CLKOUT2_DIVIDE(1),
    .CLKOUT3_DIVIDE(1),
    .CLKOUT4_DIVIDE(1),
    .CLKOUT5_DIVIDE(1),
    .CLKOUT6_DIVIDE(1),
    .CLKOUT4_CASCADE("FALSE"), // Cascade CLKOUT4 counter with CLKOUT6 (TRUE/FALSE)
    .CLOCK_HOLD("FALSE"),     // Hold VCO Frequency (TRUE/FALSE)
    .DIVCLK_DIVIDE(1),        // Master division value (1-80)
    .REF_JITTER1(0.0),        // Reference input jitter in UI (0.000-0.999).
    .STARTUP_WAIT("FALSE")    // Not supported. Must be set to FALSE.
)
MMCM_BASE_inst (
    // Clock Outputs: 1-bit (each) output: User configurable clock outputs
    .CLKOUT0(CLKOUT0),        // 1-bit output: CLKOUT0 output
    .CLKOUT0B(CLKOUT0B),      // 1-bit output: Inverted CLKOUT0 output
    .CLKOUT1(CLKOUT1),        // 1-bit output: CLKOUT1 output
    .CLKOUT1B(CLKOUT1B),      // 1-bit output: Inverted CLKOUT1 output
    .CLKOUT2(CLKOUT2),        // 1-bit output: CLKOUT2 output
    .CLKOUT2B(CLKOUT2B),      // 1-bit output: Inverted CLKOUT2 output
    .CLKOUT3(CLKOUT3),        // 1-bit output: CLKOUT3 output
    .CLKOUT3B(CLKOUT3B),      // 1-bit output: Inverted CLKOUT3 output
    .CLKOUT4(CLKOUT4),        // 1-bit output: CLKOUT4 output
    .CLKOUT5(CLKOUT5),        // 1-bit output: CLKOUT5 output
    .CLKOUT6(CLKOUT6),        // 1-bit output: CLKOUT6 output
    // Feedback Clocks: 1-bit (each) output: Clock feedback ports
    .CLKFBOUT(CLKFBOUT),      // 1-bit output: Feedback clock output
    .CLKFBOUTB(CLKFBOUTB),    // 1-bit output: Inverted CLKFBOUT output
    // Status Port: 1-bit (each) output: MMCM status ports
    .LOCKED(LOCKED),          // 1-bit output: LOCK output
    // Clock Input: 1-bit (each) input: Clock input
    .CLKIN1(CLKIN1),
    // Control Ports: 1-bit (each) input: MMCM control ports
    .PWRDWN(PWRDWN),          // 1-bit input: Power-down input
    .RST(RST),                // 1-bit input: Reset input
    // Feedback Clocks: 1-bit (each) input: Clock feedback ports
    .CLKFBIN(CLKFBIN)        // 1-bit input: Feedback clock input
);

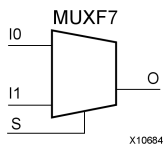
// End of MMCM_BASE_inst instantiation
```

## 詳細情報

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

## MUXF7

### プリミティブ：2-to-1 Look-Up Table Multiplexer with General Output



## 概要

このデザイン エLEMENTは、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 8:1 マルチプレクサーを作成するためのマルチプレクサー ファンクションをインプリメントします。I0 および I1 入力には、MUXF6 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。

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 (インスタンス化)

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

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

-- MUXF7: CLB MUX to tie two MUXF6's together with general output
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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 (インスタンス化)

```
// MUXF7: CLB MUX to tie two LUT6's or MUXF6's together with general output
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

MUXF7 MUXF7_inst (
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie to LUT6 O6 pin)
    .I1(I1),    // Input (tie to LUT6 O6 pin)
    .S(S)       // Input select to MUX
);

// End of MUXF7_inst instantiation
```

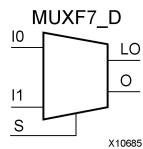
## 詳細情報

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



## MUXF7\_D

プリミティブ：2-to-1 Look-Up Table Multiplexer with Dual Output



### 概要

このデザイン エLEMENTは、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マルチプレクサー ファンクションをインプリメントします。I0 および I1 入力には、MUXF6 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 は一般的な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 (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_D: CLB MUX to tie two MUXF6's together with general and local outputs
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

MUXF7_D_inst : MUXF7_D
port map (
    LO => LO,  -- Ouput 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 (インスタンス化)

```
// MUXF7_D: CLB MUX to tie two LUT6's or MUXF6's together with general and local outputs
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

MUXF7_D MUXF7_D_inst (
    .LO(LO),  // Ouput of MUX to local routing
    .O(O),    // Output of MUX to general routing
    .IO(IO),  // Input (tie to LUT6 O6 pin)
    .I1(I1),  // Input (tie to LUT6 O6 pin)
    .S(S)     // Input select to MUX
);

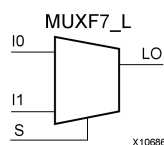
// End of MUXF7_D_inst instantiation
```

## 詳細情報

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

## MUXF7\_L

プリミティブ：2-to-1 look-up table Multiplexer with Local Output



### 概要

このデザイン エLEMENTは、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マルチプレクサー ファンクションをインプリメントします。I0 および I1 入力には、MUXF6 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
I1	入力	1	入力
S	入力	1	MUX への入力セレクト

### デザインの入力方法

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

## VHDL 記述 (インスタンスレーション)

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

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

-- MUXF7_L: CLB MUX to tie two MUXF6's together with local output
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

MUXF7_L_inst : MUXF7_L
port map (
    LO => LO,  -- Output of MUX to local routing
    IO => IO,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S     -- Input select to MUX
);

-- End of MUXF7_L_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MUXF7_L: CLB MUX to tie two LUT6's or MUXF6's together with local output
//          Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

MUXF7_L MUXF7_L_inst (
    .LO(IO),  // Output of MUX to local routing
    .IO(IO),  // Input (tie to LUT6 O6 pin)
    .I1(I1),  // Input (tie to LUT6 O6 pin)
    .S(S)     // Input select to MUX
);

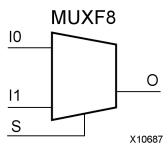
// End of MUXF7_L_inst instantiation
```

## 詳細情報

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

## MUXF8

プリミティブ：2-to-1 Look-Up Table Multiplexer with General Output



### 概要

このデザイン エLEMENTは、ルックアップ テーブルと MUXF5、MUXF6、および MUXF7 を組み合わせて、8 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マルチプレクサー ファンクションを、スライス 8 個にインプリメントします。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-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

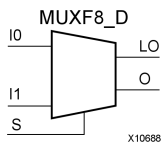
// End of MUXF8_inst instantiation
```

## 詳細情報

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

## MUXF8\_D

プリミティブ：2-to-1 Look-Up Table Multiplexer with Dual Output



### 概要

このデザイン エLEMENTは、4 つのルックアップ テーブルと 2 つの MUXF7 を組み合わせて、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 ファンクションを、スライス 8 個にインプリメントします。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 は一般的な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-6
-- Xilinx HDL Libraries Guide, version 13.3

MUXF8_D_inst : MUXF8_D
port map (
    LO => LO,  -- Ouput 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-6
// Xilinx HDL Libraries Guide, version 13.3

MUXF8_D MUXF8_D_inst (
    .LO(LO),  // Ouput 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
```

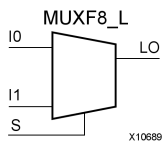
## 詳細情報

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



## MUXF8\_L

プリミティブ：2-to-1 Look-Up Table Multiplexer with Local Output



### 概要

このデザイン エLEMENTは、4 つのルックアップ テーブルと 2 つの MUXF7 を組み合わせて、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 ファンクションを、CLB 2 個 (スライス 8 個) にインプリメントします。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-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

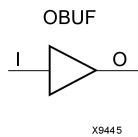
// End of MUXF8_L_inst instantiation
```

## 詳細情報

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

## OBUF

### プリミティブ：Output Buffer



### 概要

このデザイン エLEMENTは単純な出力バッファで、トライステートにならない (常に駆動される) FPGA デバイス ピンへの出力信号を駆動するために使用します。デザインのすべての出力ポートに OBUF、OBUFT、OBUFDS、OBUFTDS のいずれかを接続する必要があります。

このELEMENTは内部回路を外部から分離し、チップから出力する信号の駆動電流を供給します。I/O ブロック (IOB) 内にあります。出力 (O) は、OPAD または IOPAD に接続されます。このELEMENTでは、LVTTL 規格が使用され、DRIVE 制約と SLOW または FAST 制約を使用して駆動電流とスルー レートを選択できます。デフォルトでは、DRIVE は 12mA、スルー レートは SLOW に設定されています。

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	最上位出力ポートに直接接続される OBUF の出力
I	入力	1	OBUF の入力。出力ポートを駆動するロジックに接続

### デザインの入力方法

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

### 使用可能な属性

属性	データ型	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	出力の駆動電流を指定します。許容範囲で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"	ELEMENTに I/O 規格を割り当てます。
SLEW	文字列	"SLOW"、"FAST"	"SLOW"	出力ドライバのスルー レートを指定します。この属性の最適な設定方法は、データシートを参照してください。

## VHDL 記述 (インスタンスレーション)

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

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

-- OBUF: Single-ended Output Buffer
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

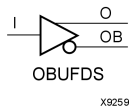
// End of OBUF_inst instantiation
```

## 詳細情報

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

## OBUFDS

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



### 概要

このデザイン エLEMENTは、低電圧の差動信号 (1.8V CMOS) をサポートする単一の出力バッファです。内部回路を外部から分離し、チップから出力する信号の駆動電流を供給します。出力には 2 つの異なるポート (O および OB) があり、これらのポートをそれぞれ「マスター」および「スレーブ」と呼びます。マスターとスレーブは MYNET と MYNETB のように、同じ論理信号の反対の状態を示します。

### 論理表

入力	出力	
I	O	OB
0	0	1
1	1	0

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	Diff_p 出力 (最上位ポートに直接接続)
OB	出力	1	Diff_n 出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力

### デザインの入力方法

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

### 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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

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

-- OBUFDS: Differential Output Buffer
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

OBUFDS_inst : OBUFDS
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Diff_p output (connect directly to top-level port)
    OB => OB,    -- Diff_n output (connect directly to top-level port)
    I => I       -- Buffer input
);

-- End of OBUFDS_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// OBUFDS: Differential Output Buffer
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

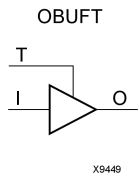
// End of OBUFDS_inst instantiation
```

## 詳細情報

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

## OBUFT

### プリミティブ：3-State Output Buffer with Active Low Output Enable



### 概要

このデザイン エレメントは、入力 (I)、出力 (O)、アクティブ Low 出力イネーブル (T) を持つ単一のトリステート出力バッファです。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=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、トリ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	トリステート イネーブル入力

### デザインの入力方法

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

### 使用可能な属性

属性	データ型	値	デフォルト	説明
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;

-- OBUFT: Single-ended 3-state Output Buffer
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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 (インスタンスレーション)

```
// OBUFT: Single-ended 3-state Output Buffer
//      All devices
// Xilinx HDL Libraries Guide, version 13.3

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
```

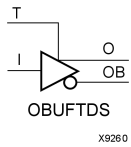
## 詳細情報

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



## OBUFTDS

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



### 概要

このデザイン エ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;

-- OBUFTDS: Differential 3-state Output Buffer
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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 (インスタンス化)

```
// OBUFTDS: Differential 3-state Output Buffer
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

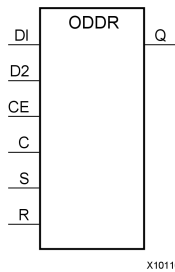
// End of OBUFTDS_inst instantiation
```

## 詳細情報

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

## ODDR

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



### 概要

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

#### 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"	DDR のデータ送信モードを選択
INIT	2 進数	0, 1	1	Q の初期値
SRTYPE	文字列	"SYNC"、"ASYNC"	"SYNC"	セット/リセットのタイプを選択

## 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;

-- ODDR: Output Double Data Rate Output Register with Set, Reset
--       and Clock Enable.
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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 ("ASYNC" 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 (インスタンス化)

```
// ODDR: Output Double Data Rate Output Register with Set, Reset
//       and Clock Enable.
//       Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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 "ASYNC"
) 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
);

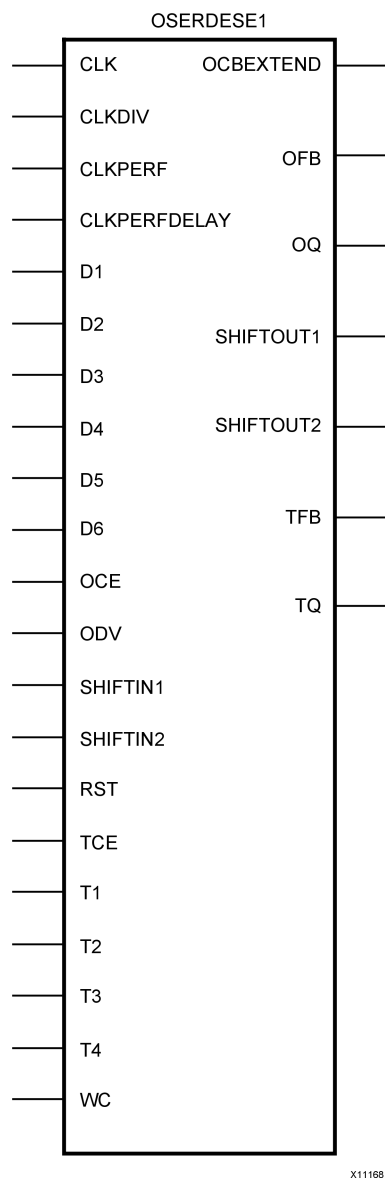
// End of ODDR_inst instantiation
```

## 詳細情報

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

## OSERDESE1

プリミティブ : Dedicated IOB Output Serializer



## 概要

このエレメントは、高速ソース同期インターフェイスのインプリメンテーションに特化したクロックおよびロジックリソースを持つ、専用パラレル/シリアルコンバーターです。各 OSERDES モジュールに、データおよびトライステート制御用の専用シリアライザーが含まれています。データシリアライザーおよびトライステートシリアライザーは、両方とも SDR および DDR モードにコンフィギュレーションできます。データでは 6:1 まで（データ幅拡張を使用する場合は 10:1 まで）のシリアル化、トライステート制御では 4:1 までのシリアル化が可能です。高速メモリアプリケーションをサポートするための専用 DDR3 モードもあります。

## ポートの説明

ポート名	方向	幅	機能
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。
CLKDIV	入力	1	高速分周クロック入力。パラレル/シリアル コンバーターのパラレル側に駆動します。CLK ポートに接続されたクロックを分周したクロックです。
CLKPERF	入力	1	MMCM から的高パフォーマンス クロックを OSERDESE1 に供給するための専用パスの一部です。DDR3 アプリケーションの MEMORY_DDR3 モードでのみ使用可能です。
CLKPERFDELAY	入力	1	MMCM から IODELAYE1 を介して遅延された高パフォーマンス クロックを OSERDESE1 に供給するための専用パスの一部です。DDR3 アプリケーションの MEMORY_DDR3 モードでのみ使用可能です。IODELAYE1 を使用して CLKPERF を遅延していない場合は、CLKPERFDELAY を CLKPERF と同じソースに接続してください。
D1 ~ D6	入力	1	パラレル データ入力。パラレル データは、D1 ~ D6 から OSERDES モジュールに入力されます。これらのポートは FPGA に接続され、2 ~ 6 ビット (6:1 シリアル化) にコンフィギュレーションできます。2 つ目の OSERDES を SLAVE モードで使用するにより、10 ビットまでの幅をサポートできます。
OCBEXTEND	出力	1	DDR3 モードで使われ、CLK を CLKPERF または CLKPERFDELAY に一致させるために出力循環バッファによりレイテンシを増加したことを示します。
OCE	入力	1	データ パス用のアクティブ High のクロック イネーブル
ODV	入力	1	MEMORY_DDR3 モード用の専用ロジックの一部です。IODELAYE1 を介した CLKPERFDELAY 遅延が半周期よりも大きい場合に、ユーザーが High にアサートします。DDR3 アプリケーションの MEMORY_DDR3 モードでのみ使用可能です。MEMORY_DDR3 モードを使用しない場合は、このポートを GND に接続してください。
OFB	出力	1	出力フィードバック ポート。OSERDESE1 の高速シリアル データ 出力ポートまたは CLKPERF のバイパスされたバージョンです。ODELAYUSED 属性が 0 に設定されている場合、OSERDESE1 にシリアル データを送信するために OFB ポートを使用できます。ODELAYUSED 属性が 1 に設定されており、OSERDESE1 が MEMORY_DDR3 モードの場合、高パフォーマンス クロック入力 (CLKPERF) を IODELAYE1 にリンクするために OFB ポートを使用できます。
OQ	出力	1	OSERDES モジュールのデータ出力ポート。入力ポート D1 のデータが OQ に一番最初に出されます。このポートは、データ パラレル/シリアル コンバーターの出力と IOB のデータ入力を接続します。このポートで IODELAYE1 を駆動することはできません。OFB ピンを使用する必要があります。
RST	入力	1	CLK および CLKDIV ドメインのすべてのデータフリップフロップの出力を非同期で Low に駆動します。OSERDES 回路が CLK ドメインで動作しており、タイミングがクリティカルな場合は、内部専用回路を使用して RST 入力のタイミングを調整し、CLK ドメインに同期したリセット信号を生成してください。同様に、CLKDIV ドメインに同期したリセット信号を生成する RST 入力のタイミングを調整する専用回路があります。RST 入力のタイミングを調整する OSERDES 回路があるので、CLKDIV 周波数ドメイン (CLKDIV に同期) でタイミングを満たす RST 入力にリセット パルスを提供するだけですみます。RST は、CLKDIV の 1 サイクル以上 High に駆動する必要があります。複数の OSERDES ポートを含むインターフェイスを構築する場合は、すべての OSERDES ポートを同期化する必要があります。RST 入力の内部タイミングは、同じリセット パルスを受信するすべての OSERDES ブロックがお互いに同期してリセット状態から戻るように調整されます。

ポート名	方向	幅	機能
SHIFTIN1/ SHIFTIN2	入力	1	データ入力を拡張するためのカスケード入力。スレーブの SHIFTOUT1、SHIFTOUT2 に接続します。
SHIFTOUT1/ SHIFTOUT2	出力	1	データ入力を拡張するためのカスケード出力。マスターの SHIFTIN1、SHIFTIN2 に接続します。
TCE	入力	1	トリステスト制御パス用のアクティブ High のクロック イネーブル
TFB	出力	1	IODELAY に送信される OSERDES モジュールのトリステスト制御出力。トリステスト パラレル/シリアル コンバーターの出力を IODELAY の制御/トリステスト入力に接続します。
TQ	出力	1	OSERDES モジュールのトリステスト制御出力。トリステスト パラレル/シリアル コンバーターの出力を IOB の制御/トリステスト入力に接続します。
T1 ~ T4	入力	1	パラレルトリステスト入力。パラレルトリステスト信号は、T1 ~ T4 から OSERDES モジュールに入力されます。このポートは FPGA に接続され、1、2、または 4 ビットにコンフィギュレーションできます。
WC	入力	1	MEMORY_DDR3 モード用の専用ロジックの一部です。データの書き込みからデータの読み出しに切り替わるときに書き込みコマンドが発行されます。DDR3 アプリケーションの MEMORY_DDR3 モードでのみ使用可能です。MEMORY_DDR3 モードを使用しない場合は、このポートを GND に接続してください。

## 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
DATA_RATE_OQ	文字列	"DDR"、"SDR"	"DDR"	データ (OQ) 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	"DDR"、"BUF"、"SDR"	"DDR"	トリステスト (TQ) をクロック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、またはバッファのコンフィギュレーションに設定するかを指定します。
DATA_WIDTH	整数	4、2、3、5、6、7、8、10	4	パラレル/シリアル データ コンバーターの幅を指定します。設定可能な値は、DATA_RATE_OQ = "DDR" の場合は 4、6、8、10、DATA_RATE_OQ = "SDR" の場合は 2、3、4、5、6、7、8 です。
DDR3_DATA	整数	0、1	1	DDR3 では、I/O が DQ または DQS ピンの場合は 1 に、制御、アドレス、クロックなどの場合は 0 に設定されます。
INIT_OQ	2 進数	1'b0 ~ 1'b1	1'b0	OQ 出力の初期値を指定します。
INIT_TQ	2 進数	1'b0 ~ 1'b1	1'b0	TQ 出力の初期値を指定します。
INTERFACE_TYPE	文字列	"DEFAULT"、"MEMORY_DDR3"	"DEFAULT"	OSERDESE1 の使用モデルを選択します。



属性	データ型	値	デフォルト	説明
ODELAY_USED	整数	0、1	0	DDR3 モードでのみ使用されます。ODELAY を使用している場合に出力循環バッファを正しいモードに設定するのに役立ちます。その他のモードでは、デザインで ODELAY を使用している場合でも、0 に設定してください。
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	1'b0 ~ 1'b1	1'b0	SR をアサートした場合の OQ 出力の値を指定します。
SRVAL_TQ	2 進数	1'b0 ~ 1'b1	1'b0	SR をアサートした場合の TQ 出力の値を指定します。
TRISTATE_WIDTH	整数	4、1	4	パラレル/シリアルトライステートコンバーターの幅を指定します。設定可能な値は、DATA_RATE_TQ = "DDR"、DATA_WIDTH = 4、および DATA_RATE_OQ = "DDR" の場合は 1 または 4、DATA_RATE_TQ、DATA_WIDTH、および DATA_RATE_OQ がそれ以外の値に設定されている場合は 1 です。

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- OSERDESE1: Output Serial/DESerializer
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

OSERDESE1_inst : OSERDESE1
generic map (
    DATA_RATE_OQ => "DDR",      -- "SDR" or "DDR"
    DATA_RATE_TQ => "DDR",      -- "BUF", "SDR" or "DDR"
    DATA_WIDTH => 4,            -- Parallel data width (1-8,10)
    DDR3_DATA => 1,             -- Must leave at 1 (MIG-only parameter)
    INIT_OQ => '0',             -- Initial value of OQ output (0/1)
    INIT_TQ => '0',             -- Initial value of TQ output (0/1)
    INTERFACE_TYPE => "DEFAULT", -- Must leave at "DEFAULT" (MIG-only parameter)
    ODELAY_USED => 0,           -- Must leave at 0 (MIG-only parameter)
    SERDES_MODE => "MASTER",    -- "MASTER" or "SLAVE"
    SRVAL_OQ => '0',            -- OQ output value when SR is used (0/1)
    SRVAL_TQ => '0',            -- TQ output value when SR is used (0/1)
    TRISTATE_WIDTH => 4         -- Parallel to serial 3-state converter width (1 or 4)
)
port map (
    -- MIG-only Signals: 1-bit (each) output: Do not use unless generated by MIG
    OCBEXTEND => OCBEXTEND,      -- 1-bit output: Leave unconnected (MIG-only connected signal)
    -- Outputs: 1-bit (each) output: Serial output ports
    OFB => OFB,                  -- 1-bit output: Data feedback output to ISERDESE1
    OQ => OQ,                    -- 1-bit output: Data output (connect to I/O port)
    TFB => TFB,                  -- 1-bit output: 3-state control output
    TQ => TQ,                    -- 1-bit output: 3-state path output
    -- SHIFTOUT1-SHIFTOUT2: 1-bit (each) output: Data width expansion output ports
    SHIFTOUT1 => SHIFTOUT1,      -- 1-bit output: Connect to SHIFTIN1 of slave or unconnected
    SHIFTOUT2 => SHIFTOUT2,      -- 1-bit output: Connect to SHIFTIN2 of slave or unconnected
    -- Clocks: 1-bit (each) input: OSERDESE1 clock input ports
    CLK => CLK,                  -- 1-bit input: High-speed clock input
    CLKDIV => CLKDIV,            -- 1-bit input: Divided clock input
    -- Control Signals: 1-bit (each) input: Clock enable and reset input ports

```

```
OCE => OCE,                -- 1-bit input: Active high clock data path enable input
RST => RST,                -- 1-bit input: Active high reset input
TCE => TCE,                -- 1-bit input: Active high clock enable input for 3-state
-- D1 - D6: 1-bit (each) input: Parallel data inputs
D1 => D1,
D2 => D2,
D3 => D3,
D4 => D4,
D5 => D5,
D6 => D6,
-- MIG-only Signals: 1-bit (each) input: Do not use unless generated by MIG
CLKPERF => CLKPERF,        -- 1-bit input: Ground input (MIG-only connected signal)
CLKPERFDELAY => CLKPERFDELAY, -- 1-bit input: Ground input (MIG-only connected signal)
ODV => ODV,                -- 1-bit input: Ground input (MIG-only connected signal)
WC => WC,                  -- 1-bit input: Ground input (MIG-only connected signal)
-- SHIFTIN1-SHIFTIN2: 1-bit (each) input: Data width expansion input ports
SHIFTIN1 => SHIFTIN1,      -- 1-bit input: Connect to SHIFTOUT1 of master or GND
SHIFTIN2 => SHIFTIN2,      -- 1-bit input: Connect to SHIFTOUT2 of master or GND
-- T1 - T4: 1-bit (each) input: Parallel 3-state inputs
T1 => T1,
T2 => T2,
T3 => T3,
T4 => T4
);

-- End of OSERDESE1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// OSERDESE1: Output SERIAL/DESerializer
//          Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

OSERDESE1 #(
    .DATA_RATE_OQ("DDR"),           // "SDR" or "DDR"
    .DATA_RATE_TQ("DDR"),           // "BUF", "SDR" or "DDR"
    .DATA_WIDTH(4),                 // Parallel data width (1-8,10)
    .DDR3_DATA(1),                  // Must leave at 1 (MIG-only parameter)
    .INIT_OQ(1'b0),                 // Initial value of OQ output (0/1)
    .INIT_TQ(1'b0),                 // Initial value of TQ output (0/1)
    .INTERFACE_TYPE("DEFAULT"),     // Must leave at "DEFAULT" (MIG-only parameter)
    .ODELAY_USED(0),                 // Must leave at 0 (MIG-only parameter)
    .SERDES_MODE("MASTER"),         // "MASTER" or "SLAVE"
    .SRVAL_OQ(1'b0),                 // OQ output value when SR is used (0/1)
    .SRVAL_TQ(1'b0),                 // TQ output value when SR is used (0/1)
    .TRISTATE_WIDTH(4)               // Parallel to serial 3-state converter width (1 or 4)
)
OSERDESE1_inst (
    // MIG-only Signals: 1-bit (each) output: Do not use unless generated by MIG
    .OCBEXTEND(OCBEXTEND),          // 1-bit output: Leave unconnected (MIG-only connected signal)
    // Outputs: 1-bit (each) output: Serial output ports
    .OFB(OFB),                      // 1-bit output: Data feedback output to ISERDESE1
    .OQ(OQ),                        // 1-bit output: Data output (connect to I/O port)
    .TFB(TFB),                      // 1-bit output: 3-state control output
    .TQ(TQ),                        // 1-bit output: 3-state path output
    // SHIFTOUT1-SHIFTOUT2: 1-bit (each) output: Data width expansion output ports
    .SHIFTOUT1(SHIFTOUT1),          // 1-bit output: Connect to SHIFTOUT1 of slave or unconnected
    .SHIFTOUT2(SHIFTOUT2),          // 1-bit output: Connect to SHIFTOUT2 of slave or unconnected
    // Clocks: 1-bit (each) input: OSERDESE1 clock input ports
    .CLK(CLK),                      // 1-bit input: High-speed clock input
    .CLKDIV(CLKDIV),                // 1-bit input: Divided clock input
    // Control Signals: 1-bit (each) input: Clock enable and reset input ports
    .OCE(OCE),                      // 1-bit input: Active high clock data path enable input
    .RST(RST),                      // 1-bit input: Active high reset input
    .TCE(TCE),                      // 1-bit input: Active high clock enable input for 3-state
    // D1 - D6: 1-bit (each) input: Parallel data inputs
    .D1(D1),
    .D2(D2),
    .D3(D3),
    .D4(D4),
    .D5(D5),
    .D6(D6),
    // MIG-only Signals: 1-bit (each) input: Do not use unless generated by MIG
    .CLKPERF(CLKPERF),              // 1-bit input: Ground input (MIG-only connected signal)
    .CLKPERFDELAY(CLKPERFDELAY),    // 1-bit input: Ground input (MIG-only connected signal)
    .ODV(ODV),                      // 1-bit input: Ground input (MIG-only connected signal)
    .WC(WC),                        // 1-bit input: Ground input (MIG-only connected signal)
    // SHIFTOUT1-SHIFTOUT2: 1-bit (each) input: Data width expansion input ports
    .SHIFTOUT1(SHIFTOUT1),          // 1-bit input: Connect to SHIFTOUT1 of master or GND
    .SHIFTOUT2(SHIFTOUT2),          // 1-bit input: Connect to SHIFTOUT2 of master or GND
    // T1 - T4: 1-bit (each) input: Parallel 3-state inputs
    .T1(T1),
    .T2(T2),
    .T3(T3),
    .T4(T4)
);

// End of OSERDESE1_inst instantiation
```

## 詳細情報

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

## PCIE\_2\_0

プリミティブ : PCI Express version 2.0 compliant port



## 概要

このエレメントは、RocketIO™ トランシーバー、ブロック RAM、さまざまなクロック リソースなど、FPGA のほかのリソースと併用します。エンドポイント、ルート ポート、またはカスタム PCI EXPRESS® デザインを PCIe\_2.0 を使用してインプリメントするには、必ず CORE Generator™ (ISE® Design Suite に含まれる) を使用して PCI EXPRESS デザイン用の LogiCORE™ IP コアを作成してください。LogiCORE は、PCI\_E\_2\_0 ソフトウェア プリミティブをインスタンス化し、インターフェイスを FPGA リソースに接続し、すべての属性を設定して、シンプルでユーザーにとって使いやすいインターフェイスを提供します。

## デザインの入力方法

このエレメントをインスタンス化するには、PCI EXPRESS コアまたはこのエレメントを含む関連コアを使用します。このエレメントは直接インスタンス化しないでください。

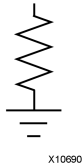
## 詳細情報

- ・ [Virtex-6 FPGA RocketIO GTX トランシーバー ユーザー ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## PULLDOWN

**プリミティブ：Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs**

PULLDOWN



X10690

### 概要

この抵抗エレメントは、入力、出力、双方向のパッドに接続し、フロートする可能性のあるノードのロジックレベルを Low にします。

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	プルダウン出力 (最上位ポートに直接接続)

### デザインの入力方法

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

### VHDL 記述 (インスタンス化)

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

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

-- PULLDOWN: I/O Buffer Weak Pull-down
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

PULLDOWN_inst : PULLDOWN
port map (
  O => O      -- Pulldown output (connect directly to top-level port)
);

-- End of PULLDOWN_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// PULLDOWN: I/O Buffer Weak Pull-down
//          Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

PULLDOWN PULLDOWN_inst (
    .O(0)      // Pulldown output (connect directly to top-level port)
);

// End of PULLDOWN_inst instantiation
```

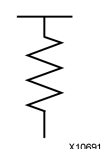
## 詳細情報

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

## PULLUP

**プリミティブ : Resistor to VCC for Input PADs, Open-Drain, and 3-State Outputs**

PULLUP



### 概要

このデザイン エレメントは、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 エレメント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-6
-- Xilinx HDL Libraries Guide, version 13.3

PULLUP_inst : PULLUP
port map (
  O => O      -- Pullup output (connect directly to top-level port)
);

-- End of PULLUP_inst instantiation
```



## Verilog 記述 (インスタンスレーション)

```
// PULLUP: I/O Buffer Weak Pull-up
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

PULLUP PULLUP_inst (
    .O(0)      // Pullup output (connect directly to top-level port)
);

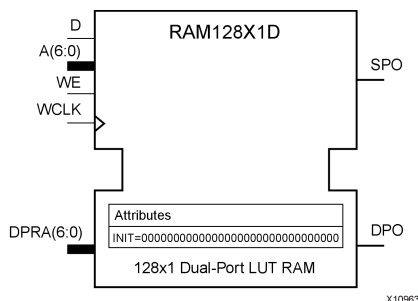
// End of PULLUP_inst instantiation
```

## 詳細情報

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

## RAM128X1D

### プリミティブ：128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)



### 概要

このデザイン エLEMENTは 128 ワード X 1 ビットの RAM で読み出し/書き込みポートがあり、ライト イネーブル (WE) が High のときにアドレス バス A で指定されたロケーションに D 入力データピンの値が書き込まれます。この書き込みは WCLK の立ち上がりエッジの直後に実行され、同じ値が SPO に出力されます。WE が Low のときは非同期読み出しが実行され、アドレス バス A で指定されたメモリ ロケーションの値が SPO に非同期で出力されます。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	すべてゼロ	RAM の初期値を指定

## VHDL 記述 (インスタンス化)

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

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

-- RAM128X1D: 128-deep by 1-wide positive edge write, asynchronous read
--             dual-port distributed LUT RAM
--             Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

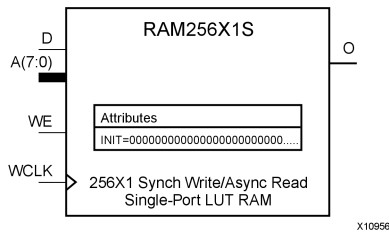
// End of RAM128X1D_inst instantiation
```

## 詳細情報

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

## RAM256X1S

### プリミティブ：256-Deep by 1-Wide Random Access Memory (Select RAM)



## 概要

このデザイン エLEMENTは、256 ワード X 1 ビットの RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM と呼ばれる) を使用してインプリメントされるため、ブロック RAM リソースは使用しません。同期読み出しを行う場合は、出力にレジスタを付けて同じスライスに配置できます。この場合、RAM とレジスタで同じクロックを使用する必要があります。アクティブ High のライト イネーブル (WE) が High になると、WCLK ピンの立ち上がりエッジで D 入力データピンの値がメモリ アレイに書き込まれます。出力 O は、WE の値にかかわらず、アドレス バス A で指定されたメモリ ロケーションの値を出力します。書き込みが実行されると、その直後に出力の値が新しい値に更新されます。

## ポートの説明

ポート名	方向	幅	機能
O	出力	1	アドレス バス A で指定される読み出し/書き込みポートのデータ出力
D	入力	1	アドレス バス A で指定される書き込みデータ入力
A	入力	8	読み出し/書き込みポートのアドレス バス
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

## デザインの入力方法

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

インスタンス化する場合、このコンポーネントを次のように接続します。

- WCLK 入力をクロック ソースに、D 入力を格納するデータ ソースに、O 出力を FDCE の D 入力などの適切なデスティネーションに接続します。
- クロック イネーブル ピン (WE) は、適切なライト イネーブル ソースに接続します。
- 8 ビット バス A は、読み出し/書き込みのソースに接続します。
- 256 ビットの 16 進数で構成される INIT 属性で、RAM の初期値を指定できます。

指定しない場合は、初期値はすべて 0 になります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	256 ビット値	すべてゼロ	RAM の初期値を指定

## 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;

-- RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
--             single-port distributed LUT RAM
--             Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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 (インスタンス化)

```
// RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
//             single-port distributed LUT RAM
//             Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

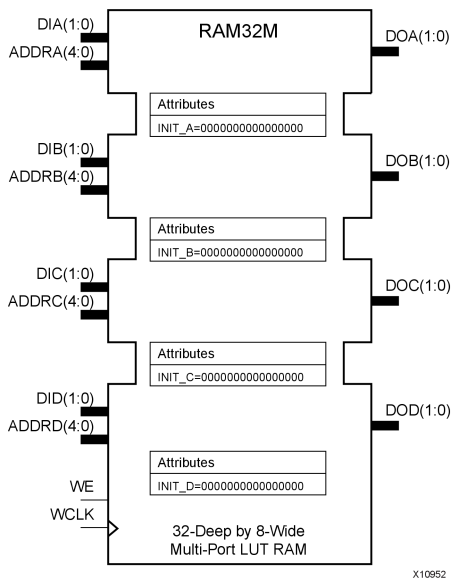
// End of RAM256X1S_inst instantiation
```

## 詳細情報

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

## RAM32M

**プリミティブ：32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)**



## 概要

このデザイン エLEMENTは、32 ワード X 8 ビットのマルチポート RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM™) を使用してインプリメントされるため、デバイスのブロック RAM リソースを使用しません。RAM32M コンポーネントは 1 つのスライスにインプリメントされ、8 ビット書き込み、2 ビット読み出しのポート 1 つと、同じメモリからの 2 ビット読み出しポート 3 つから構成されます。これにより、RAM のバイト幅の書き込みと独立した 2 ビットの読み出しが可能です。DIA、DIB、DIC、および DID 入力をすべて同じデータ入力に接続すると、読み出し/書き込みポート 1 つ、独立した読み出しポート 3 つの 32x2 クワッドポートメモリになります。DID をグラウンドに接続した場合、DOD は使用されません。ADDRa、ADDRb、ADDRc を同じアドレスに接続すると、32x6 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、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ELEMENT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。同期読み出しを行う場合は、RAM32M の出力を FDRSE 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、通常の RAM の操作では不要です。

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、 $\text{ADDRy}[z] = \text{INIT}_y[2*z+1:2*z]$  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	すべてゼロ	A ポートの RAM の初期値を指定
INIT_B	16 進数	64 ビット値	すべてゼロ	B ポートの RAM の初期値を指定
INIT_C	16 進数	64 ビット値	すべてゼロ	C ポートの RAM の初期値を指定
INIT_D	16 進数	64 ビット値	すべてゼロ	D ポートの RAM の初期値を指定



## 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;

-- RAM32M: 32-deep by 8-wide Multi Port LUT RAM
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

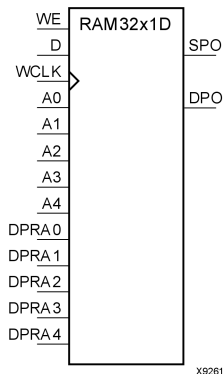
// End of RAM32M_inst instantiation
```

## 詳細情報

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

## RAM32X1D

### プリミティブ：32-Deep by 1-Wide Static Dual Port Synchronous RAM



### 概要

このデザイン エ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	すべてゼロ	RAM の初期値を指定

## 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;

-- RAM32X1D: 32 x 1 positive edge write, asynchronous read
--           dual-port distributed RAM
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

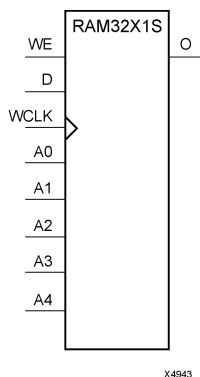
// End of RAM32X1D_inst instantiation
```

## 詳細情報

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

## RAM32X1S

プリミティブ：32-Deep by 1-Wide Static Synchronous RAM



### 概要

このデザイン エLEMENTは 32 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わる時に、データ入力 (D) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が 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	すべてゼロ	RAM の初期値を指定

## VHDL 記述 (インスタンス化)

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

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

-- RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

RAM32X1S_inst : RAM32X1S
generic map (
  INIT => X"00000000")
port map (
  O => O,      -- RAM output
  A0 => A0,     -- RAM address[0] input
  A1 => A1,     -- RAM address[1] input
  A2 => A2,     -- RAM address[2] input
  A3 => A3,     -- RAM address[3] input
  A4 => A4,     -- RAM address[4] input
  D => D,       -- RAM data input
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);

-- End of RAM32X1S_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

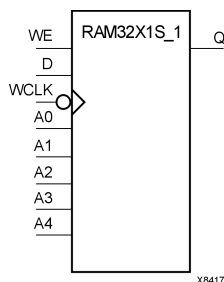
// End of RAM32X1S_inst instantiation
```

## 詳細情報

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

## RAM32X1S\_1

プリミティブ：32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



### 概要

このデザイン エレメントは 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 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 の初期値を指定

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

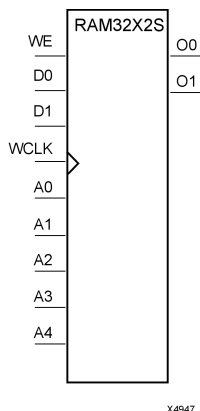
// End of RAM32X1S_1_inst instantiation
```

## 詳細情報

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

## RAM32X2S

プリミティブ：32-Deep by 2-Wide Static Synchronous RAM



### 概要

このデザイン エLEMENTは 32 ワード X 2 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、入力 (D1 ~ D0) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。出力ピン (O1 ~ O0) に出力される値は、アドレスピンで指定された RAM 内の位置に格納されている値です。

INIT\_00 および INIT\_01 属性を使用して RAM32X2S の初期値を指定できます。

### 論理表

入力			出力
WE (モード)	WCLK	D	O0 ~ O1
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D1 ~ D0	D1 ~ D0
1 (読み出し)	↓	X	データ
データ = A4 ~ A0 で指定されたワード			

### デザインの入力方法

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

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT_00	16 進数	32 ビット値	すべてゼロ	RAM のビット 0 の初期値を指定
INIT_01	16 進数	32 ビット値	すべてゼロ	RAM のビット 1 の初期値を指定

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

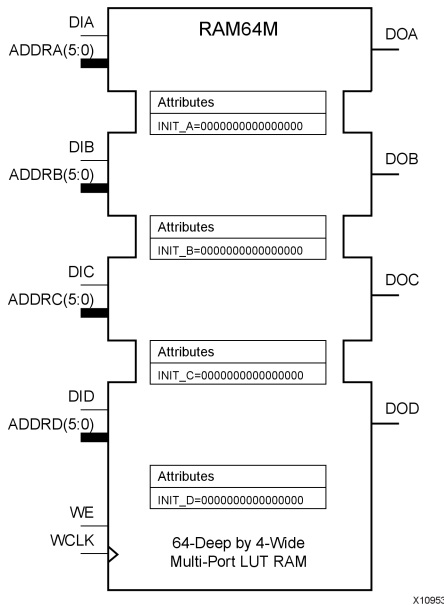
// End of RAM32X2S_inst instantiation
```

## 詳細情報

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

## RAM64M

**プリミティブ：64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)**



### 概要

このデザイン エLEMENTは、64 ワード X 4 ビットのマルチポート RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM™ と呼ばれる) を使用してインプリメントされるため、ブロック RAM リソースを使用しません。RAM64M コンポーネントは 1 つのスライスにインプリメントされます。4 ビット書き込みポート 1 つ、1 ビット読み出しのポート 1 つ、および同じメモリからの 1 ビット読み出しポート 3 つから構成されており、RAM の 4 ビット書き込みおよび個別ビット読み出しが可能です。DIA、DIB、DIC、および DID 入力をすべて同じデータ入力に接続すると、読み出し/書き込みポート 1 つ、独立した読み出しポート 3 つの 64x1 クワッド ポートメモリになります。DID をグラウンドに接続した場合、DOD は使用されません。ADDRA、ADDRB、ADDRC を同じアドレスに接続すると、64x3 のシングル デュアル ポート RAM になります。ADDRD を ADDRA、ADDRB、ADDRC に接続すると、64x4 のシングル ポート RAM になります。この RAM には、ほかにも可能なコンフィギュレーションがあります。

### ポートの説明

ポート名	方向	幅	機能
DOA	出力	1	アドレス バス ADDRA で指定される読み出しポートのデータ出力
DOB	出力	1	アドレス バス ADDRb で指定される読み出しポートのデータ出力
DOC	出力	1	アドレス バス ADDRc で指定される読み出しポートのデータ出力
DOD	出力	1	アドレス バス ADDRd で指定される読み出し/書き込みポートのデータ出力
DIA	入力	1	ADDRd で指定される書き込みデータ入力 (読み出し出力は ADDRA で指定)
DIB	入力	1	ADDRd で指定される書き込みデータ入力 (読み出し出力は ADDRb で指定)

ポート名	方向	幅	機能
DIC	入力	1	ADDRD で指定される書き込みデータ入力 (読み出し出力は ADDRRC で指定)
DID	入力	1	アドレス バス ADDRD で指定される書き込みデータ入力
ADDRA	入力	6	読み出しアドレス バス A
ADDRB	入力	6	読み出しアドレス バス B
ADDRC	入力	6	読み出しアドレス バス C
ADDRD	入力	6	4 ビットのデータ書き込みポート、1 ビットのデータ読み出しポートのアドレス バス D
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

## デザインの入力方法

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

このエレメントは、同期書き込みと非同期読み出し機能を備えた RAM を記述することにより、一部の合成ツールで推論できます。RAM の推論およびコード例の詳細は、合成ツールのマニュアルを参照してください。RAM64M のインスタンス化は、RAM ファンクションを暗示的に指定する必要がある場合、コンポーネントを手動でまたは相対的に配置する必要がある場合に実行することをお勧めします。同期読み出しを行う場合は、RAM64M の出力を FDRSE 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、通常の RAM の操作では不要です。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、 $\text{ADDRy}[z] = \text{INITy}[z]$  で計算されます。

たとえば、RAM の ADDRRC ポートが 00001 の場合、INIT\_C[1] 値がそのアドレスで最初の書き込みが行われる前の DOC ポートの初期値になります。指定しない場合は、初期値はすべて 0 になります。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_A	16 進数	64 ビット値	すべてゼロ	A ポートの RAM の初期値を指定
INIT_B	16 進数	64 ビット値	すべてゼロ	B ポートの RAM の初期値を指定
INIT_C	16 進数	64 ビット値	すべてゼロ	C ポートの RAM の初期値を指定
INIT_D	16 進数	64 ビット値	すべてゼロ	D ポートの RAM の初期値を指定

## 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;

-- RAM64M: 64-deep by 4-wide Multi Port LUT RAM
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

RAM64M_inst : RAM64M
generic map (
  INIT_A => X"0000000000000000", -- Initial contents of A port
  INIT_B => X"0000000000000000", -- Initial contents of B port
  INIT_C => X"0000000000000000", -- Initial contents of C port
  INIT_D => X"0000000000000000"  -- Initial contents of D port
) port map (
  DOA => DOA, -- Read port A 1-bit output
  DOB => DOB, -- Read port B 1-bit output
  DOC => DOC, -- Read port C 1-bit output
  DOD => DOD, -- Read/Write port D 1-bit output
  ADDRA => ADDRA, -- Read port A 6-bit address input
  ADDR_B => ADDR_B, -- Read port B 6-bit address input
  ADDR_C => ADDR_C, -- Read port C 6-bit address input
  ADDR_D => ADDR_D, -- Read/Write port D 6-bit address input
  DIA => DIA, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDRA
  DIB => DIB, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_B
  DIC => DIC, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_C
  DID => DID, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_D
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);
-- End of RAM64M_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// RAM64M: 64-deep by 4-wide Multi Port LUT RAM
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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), // Readw/rite port D 1-bit output
```

```
.DIA(DIA),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRA  
.DIB(DIB),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRb  
.DIC(DIC),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRc  
.DID(DID),      // RAM 1-bit data write input addressed by ADDRd,  
                // read addressed by ADDRd  
.ADDRA(ADDRA),  // Read port A 6-bit address input  
.ADDRB(ADDRB),  // Read port B 6-bit address input  
.ADDRC(ADDRC),  // Read port C 6-bit address input  
.ADDRD(ADDRD),  // Readw/rite port D 6-bit address input  
.WE(WE),        // Write enable input  
.WCLK(WCLK)     // Write clock input  
);  
  
// End of RAM64M_inst instantiation
```

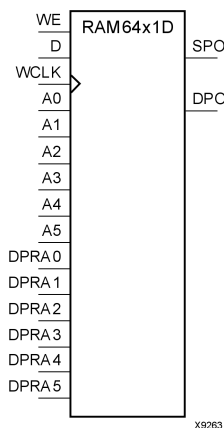
## 詳細情報

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



## RAM64X1D

**プリミティブ：64-Deep by 1-Wide Dual Port Static Synchronous RAM**



### 概要

このデザイン エLEMENTは 64 ワード X 1 ビットのデュアル ポート SRAM で、同期書き込み機能を備えています。デバイスには、読み出しアドレス (DPRA5 ～ DPRA0) 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。

**メモ：** 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

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	RAM の初期値を指定

## 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;

-- RAM64X1D: 64 x 1 negative edge write, asynchronous read
--           dual-port distributed RAM
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

RAM64X1D_1_inst : RAM64X1D_1
generic map (
  INIT => X"0000000000000000") -- Initial contents of RAM
port map (
  DPO => DPO,      -- Read-only 1-bit data output
  SPO => SPO,      -- R/W 1-bit data output
  A0 => A0,        -- R/W address[0] input bit
  A1 => A1,        -- R/W address[1] input bit
  A2 => A2,        -- R/W address[2] input bit
  A3 => A3,        -- R/W address[3] input bit
  A4 => A4,        -- R/W address[4] input bit
  A5 => A5,        -- R/W address[5] input bit
  D => D,          -- Write 1-bit data input
  DPRA0 => DPRA0,  -- Read-only address[0] input bit
  DPRA1 => DPRA1,  -- Read-only address[1] input bit
  DPRA2 => DPRA2,  -- Read-only address[2] input bit
  DPRA3 => DPRA3,  -- Read-only address[3] input bit
  DPRA4 => DPRA4,  -- Read-only address[4] input bit
  DPRA5 => DPRA5,  -- Read-only address[5] input bit
  WCLK => WCLK,    -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM64X1D_1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// RAM64X1D: 64 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

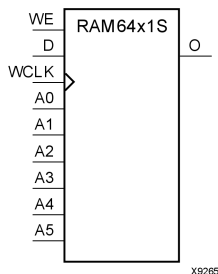
// End of RAM64X1D_inst instantiation
```

## 詳細情報

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

## RAM64X1S

**プリミティブ：64-Deep by 1-Wide Static Synchronous RAM**



### 概要

このデザイン エレメントは 64 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 6 ビットのアドレス (A5 ～ A0) で選択されたワードに書き込まれます。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。

INIT 属性を使用すると、コンフィギュレーション中にこのエレメントを初期化できます。

### 論理表

モード選択を次の論理表に示します。

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D	D
1 (読み出し)	↓	X	データ
データ = A5 ～ A0 で指定されたワード			

### デザインの入力方法

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

### 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	RAM の初期値を指定

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

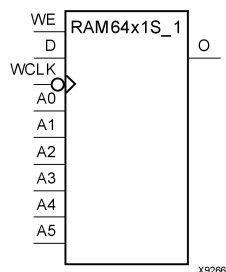
// End of RAM64X1S_inst instantiation
```

## 詳細情報

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

## RAM64X1S\_1

プリミティブ：64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



### 概要

このデザイン エLEMENTは 64 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が High から Low に切り替わるときに、データ入力 (D) の値が 6 ビットのアドレス (A5 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が High から Low に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ Low ですが、インバーターを使用してアクティブ High にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。

INIT 属性を使用すると、コンフィギュレーション中にこのELEMENTを初期化できます。

### 論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↓	D	D
1 (読み出し)	↑	X	データ
データ = A5 ~ A0 で指定されたワード			

### デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	RAM の初期値を指定

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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
//           Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

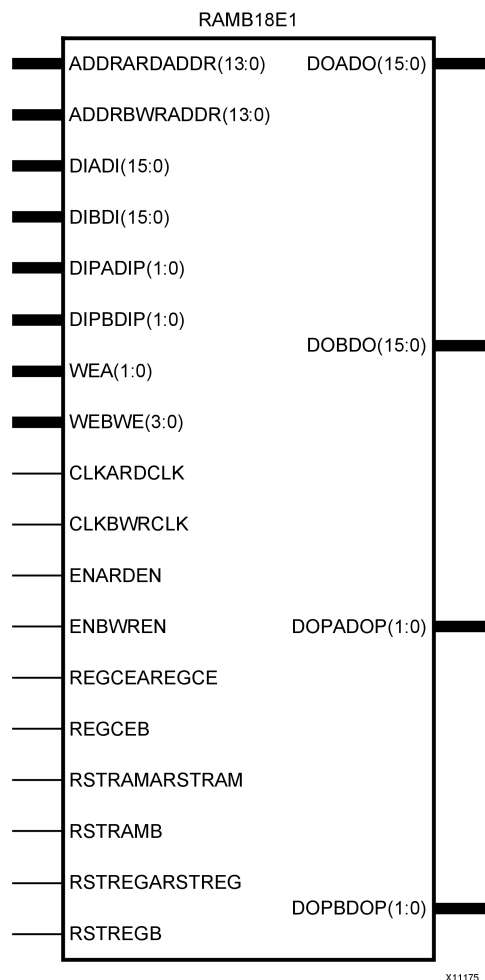
// End of RAM64X1S_1_inst instantiation
```

## 詳細情報

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

## RAMB18E1

**プリミティブ：18K-bit Configurable Synchronous Block RAM**



### 概要

Virtex®-6 デバイスにはブロック RAM が数個含まれ、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36 kb または 18 kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。このデザイン エLEMENT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ELEMENTは、1 ビット X 16K ワード～ 18 ビット X 1029 ワードの完全なデュアル ポート RAM としてコンフィギュレーションできます。また、36 ビット幅 X 512 ワードのシンプル デュアル ポート RAM にコンフィギュレーションすることもできます。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、読み出しと書き込みが同時に実行されます。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。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
ADDRARDADDR[13:0]	入力	14	ポート A アドレス入力バス/読み出しアドレス入力バス
ADDRBRWADDR[13:0]	入力	14	ポート B アドレス入力バス/書き込みアドレス入力バス



ポート名	方向	幅	機能
CLKARDCLK	入力	1	ポート A クロック入力/読み出しクロック入力
CLKBWRCLK	入力	1	ポート B クロック入力/書き込みクロック入力
DIADI[15:0]	入力	16	ポート A データ入力バス/WRADDR でアドレス指定されるデータ入力バス。RAM_MODE=SDP の場合、DIADI の論理値は DI[15:0] です。
DIBDI[15:0]	入力	16	ポート B データ入力バス/WRADDR でアドレス指定されるデータ入力バス。RAM_MODE=SDP の場合、DIBDI の論理値は DI[31:16] です。
DIPADIP[1:0]	入力	2	ポート A パリティ データ入力バス/WRADDR でアドレス指定されるデータ パリティ入力バス。RAM_MODE=SDP の場合、DIPADIP の論理値は DIP[1:0] です。
DIPBDIP[1:0]	入力	2	ポート B パリティ データ入力バス/WRADDR でアドレス指定されるデータ パリティ入力バス。RAM_MODE=SDP の場合、DIPBDIP の論理値は DIP[3:2] です。
DOADO[15:0]	出力	16	ポート A データ出力バス/RDADDR でアドレス指定されるデータ出力バス。RAM_MODE=SDP の場合、DOADO の論理値は DO[15:0] です。
DOBDO[15:0]	出力	16	ポート B データ出力バス/RDADDR でアドレス指定されるデータ出力バス。RAM_MODE=SDP の場合、DOBDO の論理値は DO[31:16] です。
DOPADOP[1:0]	出力	2	ポート A パリティ データ出力バス/RDADDR でアドレス指定されるパリティ データ出力バス。RAM_MODE=SDP の場合、DOPADOP の論理値は DOP[1:0] です。
DOPBDOP[1:0]	出力	2	ポート B パリティ データ出力バス/RDADDR でアドレス指定されるパリティ データ出力バス。RAM_MODE=SDP の場合、DOPBDOP の論理値は DOP[3:2] です。
ENARDEN	入力	1	ポート A RAM イネーブル/リード イネーブル
ENBWREN	入力	1	ポート B RAM イネーブル/ライト イネーブル
REGCEAREGCE	入力	1	ポート A 出力レジスタ クロック イネーブル入力/出力レジスタ クロック イネーブル入力 (DO_REG=1 の場合のみ有効)
REGCEB	入力	1	ポート B 出力レジスタ クロック イネーブル (DO_REG=1 および RAM_MODE=TDP の場合のみ有効)
RSTRAMARSTRAM	入力	1	同期データ ラッチを SRVAL_A で指定された値にセット/リセットします。RSTRAMARSTRAM は、DO_REG=0 または 1 のとき BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合は、RSTRAMARSTRAM でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。この信号は、RAM_MODE=TDP の場合はポート A の RSTRAMA、RAM_MODE=SDP の場合は RSTRAM です。
RSTRAMB	入力	1	同期データ ラッチを SRVAL_B で指定された値にセット/リセットします。RSTRAMB は、DO_REG=0 または 1 のとき BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合は、RSTRAMB でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。RAM_MODE=SDP の場合は使用されません。
RSTREGARSTREG	入力	1	同期出力レジスタを SRVAL_A で指定された値にセット/リセットします。RSTREGARSTREG は、DO_REG=1 のとき出力レジスタをセット/リセットします。RSTREG_PRIORITY_A は、この信号の優先度が REGCEAREGCE よりも高いかどうかを決定します。この信号は、RAM_MODE=TDP の場合はポート A の RSTREGA、RAM_MODE=SDP の場合は RSTREG です。

ポート名	方向	幅	機能
RSTREGB	入力	1	同期出力レジスタを SRVAL_B で指定された値にセット/リセットします。RSTREGB は、DO_REG=1 のとき出力レジスタをセット/リセットします。RSTREG_PRIORITY_B は、この信号の優先度が REGCEB よりも高いかどうかを決定します。RAM_MODE=SDP の場合は使用されません。
WEA[1:0]	入力	2	ポート A のバイト幅ライト イネーブル。RAM_MODE=SDP の場合は使用されません。異なるポート幅の WEA マッピングについてはユーザー ガイドを参照してください。
WEBWE[3:0]	入力	4	ポート B のバイト幅ライト イネーブル/ライト イネーブル。異なるポート幅の WEBWE マッピングについてはユーザー ガイドを参照してください。

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
COLLISION CHECK	文字列	"ALL"、"GENERATE_X_ONLY"、"NONE"、"WARNING_ONLY"	"ALL"	<p>メモリの競合が発生した場合のシミュレーションの動作を指定します。</p> <ul style="list-style-type: none"> <li>・ "ALL"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。</li> <li>・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。</li> </ul> <p><b>メモ</b>： "ALL" 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。</p>
DOA_REG	整数	0、1	0	<p>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。</p>

属性	データ型	値	デフォルト	説明
				が、clock-to-out タイムが長くなります。TDP モードでポート A に、SDP では下位 18 ビット (パリティビットを含む) に適用されます。
DOB_REG	整数	0、1	0	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。TDP モードではポート B に、SDP モードでは上位ビット (パリティビットを含む) に適用されます。
INIT_A	16 進数	18 ビット値	すべてゼロ	コンフィギュレーション後のポート A の出力の初期値を指定します。TDP モードでポート A に、SDP では下位 18 ビット (パリティビットを含む) に適用されます。
INIT_B	16 進数	18 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。TDP でポート B に、SDP では高いほうのビット (パリティビットを含む) に適用されます。
INIT_FILE	文字列	ファイルの名前と場所	なし	RAM の初期内容を記述するファイルの名前を指定します。
INIT_00 ~ INIT_3F	16 進数	すべてゼロ ~ すべて 1	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_07	16 進数	すべてゼロ ~ すべて 1	すべてゼロ	2kb のパリティ メモリ アレイの初期値を指定
RAM_MODE	文字列	"TDP"、"SDP"	"TDP"	シンプル デュアル ポート ("SDP") または完全なデュアル ポート ("TDP") を選択します。
RDADDR_COLLISION_HWCONFIG	文字列	"DELAYED_WRITE"、 "PERFORMANCE"	"DELAYED_WRITE"	<ul style="list-style-type: none"> <li>・ "PERFORMANCE" に設定すると、READ_FIRST モードでのクロック パフォーマンス (周波数) が向上します。</li> <li>・ RAM の両方のポートで同じクロックを使用してる場合に "PERFORMANCE" に設定すると、アドレスが重なった場合の競合規則が適用されます。</li> <li>・ "DELAYED_WRITE" モードでは、競合を発生させずに RAM を使用できます。</li> <li>・ ES シリコンではサポートされていないので、ES デバイスをターゲットとする場合は "DELAYED_WRITE" に設定する必要があります。</li> </ul>

属性	データ型	値	デフォルト	説明
READ_WIDTH_A	整数	0、1、2、4、9、18、36、72	0	ポート A の読み出しのデータ幅を指定します (パリティビットを含む)。ポート A を使用しない場合は、ポート幅を 0 にする必要があります。ポートを使用する場合は、必要なポート幅に設定してください。SDP の場合は、パリティビットを含む読み出し幅です。
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)。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。SDP では使用されません。
RSTREG_PRIORITY_A	文字列	"RSTREG"、 "REGCE"	"RSTREG"	RSTREG または REGCE のレジスタ優先順位を選択します。TDP モードではポート A に、SDP モードでは下位 18 ビット (パリティビットを含む) に適用されます。
RSTREG_PRIORITY_B	文字列	"RSTREG"、 "REGCE"	"RSTREG"	RSTREG または REGCE のレジスタ優先順位を選択します。TDP モードではポート B に、SDP モードでは上位ビット (パリティビットを含む) に適用されます。
SRVAL_A	16 進数	18 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの RAM の出力値を指定します。TDP モードでポート A に、SDP では下位 18 ビット (パリティビットを含む) に適用されます。
SRVAL_B	16 進数	18 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの RAM の出力値を指定します。TDP でポート B に、SDP では高いほうのビット (パリティビットを含む) に適用されます。
WRITEMODE	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	書き込みが実行されるときポートの動作を指定します。 <ul style="list-style-type: none"> <li>・ "WRITE_FIRST" : 書き込まれた値が出力ポートに出力されます。</li> <li>・ "READ_FIRST" : 書き込み前にそのメモリ ロケーションに格納されていた値が出力ポートに出力されます。</li> <li>・ "NO_CHANGE" : 出力ポートの以前の値が保持されます。</li> </ul>
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)。ポートを使用しない場合は、0 に設定する必要があります。それ以外の場合は、適切なデータ幅に設定してください。SDP モードでは使用されません。
WRITE_WIDTH_B	整数	0、1、2、4、9、18、36、72	0	ポート B への書き込みのデータ幅を指定します (パリティビットを含む)。ポートを使用しない場合は、0 に設定する必要があります。それ以外の場合は、適切なデータ幅に設定してください。SDP モードでは、パリティビットを含む書き込み幅です。

## VHDL 記述 (インスタンス化)

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

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

-- RAMB18E1: 18K-bit Configurable Synchronous Block RAM
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

RAMB18E1_inst : RAMB18E1
generic map (
  -- Collision check: Values ("ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE")
  SIM_COLLISION_CHECK => "ALL",
  -- DOA_REG, DOB_REG: Optional output register (0 or 1)
  DOA_REG => 0,
  DOB_REG => 0,
  -- INITP_00 to INITP_07: Initial contents of parity memory array
  INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  -- INIT_00 to INIT_3F: Initial contents of data memory array
  INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```

INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- INIT_A, INIT_B: Initial values on output ports
INIT_A => X"00000",
INIT_B => X"00000",
INIT_FILE => "NONE",
RAM_MODE => "TDP",
RDADDR_COLLISION_HWCONFIG => "DELAYED_WRITE",

-- READ_WIDTH A/B, WRITE_WIDTH A/B: Read/write width per port
READ_WIDTH_A => 0,
READ_WIDTH_B => 0,
WRITE_WIDTH_A => 0,
WRITE_WIDTH_B => 0,
-- RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG" or "REGCE")
RSTREG_PRIORITY_A => "RSTREG",
RSTREG_PRIORITY_B => "RSTREG",
-- SRVAL_A, SRVAL_B: Set/reset value for output
SRVAL_A => X"00000",
SRVAL_B => X"00000",
-- WriteMode: Value on output upon a write ("WRITE_FIRST", "READ_FIRST", or "NO_CHANGE")
WRITE_MODE_A => "WRITE_FIRST",
WRITE_MODE_B => "WRITE_FIRST"
)
port map (
-- Port A Data: 16-bit (each) output: Port A data
DOADO => DOADO,
DOPADOP => DOPADOP,
-- Port B Data: 16-bit (each) output: Port B data
DOBDO => DOBDO,
DOPBDOP => DOPBDOP,
-- Port A Address/Control Signals: 14-bit (each) input: Port A address and control signals (read port
-- when RAM_MODE="SDP")
ADDRARDADDR => ADDRARDADDR,
CLKARDCLK => CLKARDCLK,
ENARDEN => ENARDEN,
REGCEAREGCE => REGCEAREGCE,
RSTRAMARSTRAM => RSTRAMARSTRAM,
RSTREGARSTREG => RSTREGARSTREG,
WEA => WEA,
-- Port A Data: 16-bit (each) input: Port A data
DIADI => DIADI,
DIPADIP => DIPADIP,
-- Port B Address/Control Signals: 14-bit (each) input: Port B address and control signals (write port
-- when RAM_MODE="SDP")
ADDRBWRADDR => ADDRBWRADDR,
CLKBWRCLK => CLKBWRCLK,
ENBWREN => ENBWREN,
REGCEB => REGCEB,
RSTRAMB => RSTRAMB,
RSTREGB => RSTREGB,
WEBWE => WEBWE,
-- Port B Data: 16-bit (each) input: Port B data
DIBDI => DIBDI,

```

## Verilog 記述 (インスタンス化)

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

336



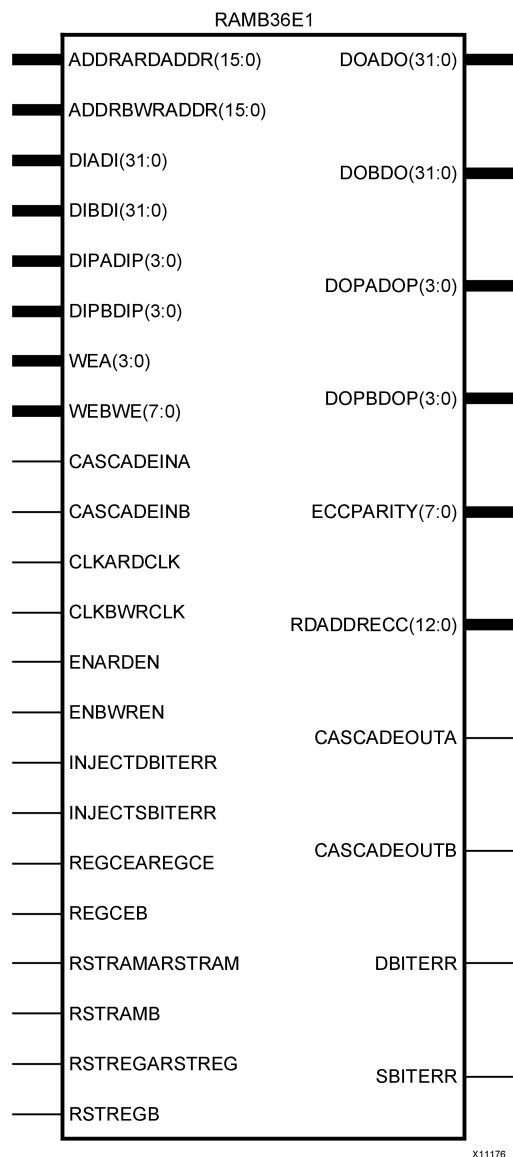
```
);  
// End of RAMB18E1_inst instantiation
```

## 詳細情報

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

## RAMB36E1

### プリミティブ : 36K-bit Configurable Synchronous Block RAM



## 概要

Virtex®-6 デバイスにはブロック RAM が数個含まれ、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36 Kb または 18 Kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。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ELEMENTをカスケード 接続すると、大型の RAM を作成できます。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。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、読み出しと書き込みが同時に実行されます。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。広いデータ幅でコンフィギュレーションすると、バイト イネーブルの書き込みが可能になり、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。エラー検出と訂正回路をイネーブルにすると、メモリ破損を検出し、修正することもできます。

## ポートの説明

ポート名	方向	幅	機能
ADDRARDADDR[15:0]	入力	16	ポート A アドレス入力バス/読み出しアドレス入力バス
ADDRBWRADDR[15:0]	入力	16	ポート B アドレス入力バス/書き込みアドレス入力バス
CASCADEINA	入力	1	ポート A カスケード入力。RAM_MODE=SDP の場合は使用されません。
CASCADEINB	入力	1	ポート B カスケード入力。RAM_MODE=SDP の場合は使用されません。
CASCADEOUTA	出力	1	ポート A カスケード出力。RAM_MODE=SDP の場合は使用されません。
CASCADEOUTB	出力	1	ポート B カスケード出力。RAM_MODE=SDP の場合は使用されません。
CLKARDCLK	入力	1	ポート A クロック入力/読み出しクロック入力
CLKBWRCLK	入力	1	ポート B クロック入力/書き込みクロック入力
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 フังก์ションからのステータス出力。この機能を使用するには EN_ECC_READ を TRUE に設定します。RAM_MODE=TDP の場合は使用されません。
DIADI[31:0]	入力	32	ポート A データ入力バス/WRADDR でアドレス指定されるデータ入力バス。RAM_MODE=SDP の場合、DIADI の論理値は DI[31:0] です。
DIBDI[31:0]	入力	32	ポート B データ入力バス/WRADDR でアドレス指定されるデータ入力バス。RAM_MODE=SDP の場合、DIBDI の論理値は DI[63:32] です。
DIPADIP[3:0]	入力	4	ポート A パリティ データ入力バス/WRADDR でアドレス指定されるデータ パリティ入力バス。RAM_MODE=SDP の場合、DIPADIP の論理値は DIP[3:0] です。
DIPBDIP[3:0]	入力	4	ポート B パリティ データ入力バス/WRADDR でアドレス指定されるデータ パリティ入力バス。RAM_MODE=SDP の場合、DIPBDIP の論理値は DIP[7:4] です。
DOADO[31:0]	出力	32	ポート A データ出力バス/RDADDR でアドレス指定されるデータ出力バス。RAM_MODE=SDP の場合、DOADO の論理値は DO[31:0] です。
DOBDO[31:0]	出力	32	ポート B データ出力バス/RDADDR でアドレス指定されるデータ出力バス。RAM_MODE=SDP の場合、DOBDO の論理値は DO[63:32] です。
DOPADOP[3:0]	出力	4	ポート A パリティ データ出力バス/RDADDR でアドレス指定されるパリティ データ出力バス。RAM_MODE=SDP の場合、DOPADOP の論理値は DOP[3:0] です。
DOPBDOP[3:0]	出力	4	ポート B パリティ データ出力バス/RDADDR でアドレス指定されるパリティ データ出力バス。RAM_MODE=SDP の場合、DOPBDOP の論理値は DOP[7:4] です。
ECCPARITY[7:0]	出力	8	ECC デコーダーでメモリ エラーを検出および訂正するために使用される、ECC エンコーダーで生成された 8 ビット データ。RAM_MODE=TDP の場合は使用されません。
ENARDEN	入力	1	ポート A RAM イネーブル/リード イネーブル
ENBWREN	入力	1	ポート B RAM イネーブル/ライト イネーブル
INJECTDBITERR	入力	1	ECC 機能が使用されている場合はダブル ビット エラーが挿入されます。

ポート名	方向	幅	機能
INJECTSBITERR	入力	1	ECC 機能が使用されている場合はシングル ビット エラーが挿入されます。
RDADDRECC[8:0]	出力	9	9 ビット ECC 読み出しアドレス。RAM_MODE=TDP の場合は使用されません。
REGCEAREGCE	入力	1	ポート A 出力レジスタ クロック イネーブル入力/出力レジスタ クロック イネーブル入力 (DO_REG=1 の場合のみ有効)
REGCEB	入力	1	ポート B 出力レジスタ クロック イネーブル (DO_REG=1 および RAM_MODE=TDP の場合のみ有効)
RSTRAMARSTRAM	入力	1	同期データ ラッチを SRVAL_A で指定された値にセット/リセットします。RSTRAMARSTRAM は、DO_REG=0 または 1 のとき BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合は、RSTRAMARSTRAM でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。この信号は、RAM_MODE=TDP の場合はポート A の RSTRAMA、RAM_MODE=SDP の場合は RSTRAM です。
RSTRAMB	入力	1	同期データ ラッチを SRVAL_B で指定された値にセット/リセットします。RSTRAMB は、DO_REG=0 または 1 のとき BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合は、RSTRAMB でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。RAM_MODE=SDP の場合は使用されません。
RSTREGARSTREG	入力	1	同期出力レジスタを SRVAL_A で指定された値にセット/リセットします。RSTREGARSTREG は、DO_REG=1 のとき出力レジスタをセット/リセットします。RSTREG_PRIORITY_A は、この信号の優先度が REGCEAREGCE よりも高いかどうかを決定します。この信号は、RAM_MODE=TDP の場合はポート A の RSTREGA、RAM_MODE=SDP の場合は RSTREG です。
RSTREGB	入力	1	同期出力レジスタを SRVAL_B で指定された値にセット/リセットします。RSTREGB は、DO_REG=1 のとき出力レジスタをセット/リセットします。RSTREG_PRIORITY_B は、この信号の優先度が REGCEB よりも高いかどうかを決定します。RAM_MODE=SDP の場合は使用されません。
SBITERR	出力	1	シングル ビット エラーが検出されたことを示す ECC 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 に設定する必要があります。RAM_MODE=TDP の場合は使用されません。
WEA[3:0]	入力	4	ポート A のバイト幅ライト イネーブル。RAM_MODE=SDP の場合は使用されません。異なるポート幅の WEA マッピングについてはユーザー ガイドを参照してください。
WEBWE[7:0]	入力	8	ポート B のバイト幅ライト イネーブル/ライト イネーブル。異なるポート幅の WEBWE マッピングについては『Virtex®-6ユーザー ガイド』を参照してください。

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
COLLISION_CHECK	文字列	"ALL"、 "GENERATE_X_ONLY"、 "NONE"、 "WARNING_ONLY"	"ALL"	<p>メモリの競合が発生した場合のシミュレーションの動作を指定します。</p> <ul style="list-style-type: none"> <li>・ "ALL"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。</li> <li>・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。</li> <li>・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。</li> </ul> <p><b>メモ：</b> "ALL" 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。</p>
DOA_REG	整数	0、1	0	<p>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。TDP モードでポート A に、SDP では下位 36 ビット (パリティビットを含む) に適用されます。</p>
DOB_REG	整数	0、1	0	<p>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。TDP でポート B に、SDP では高いほうのビット (パリティビットを含む) に適用されます。</p>
EN_ECC_READ	ブール代数	FALSE、TRUE	FALSE	ECC デコーダー回路をイネーブルにします。
EN_ECC_WRITE	ブール代数	FALSE、TRUE	FALSE	ECC エンコーダー回路をイネーブルにします。
INIT_A	16 進数	36 ビット値	すべてゼロ	<p>コンフィギュレーション後のポート A の出力の初期値を指定します。"TDP" モードでポート A に、"SDP" では下位 36 ビット (パリティビットを含む) に適用されます。</p>

属性	データ型	値	デフォルト	説明
INIT_B	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。“TDP” モードではポート B に、“SDP” モードでは上位ビット (パリティビットを含む) に適用されます。
INIT_FILE	文字列	ファイルの名前と場所	なし	RAM の初期内容を記述するファイルの名前を指定します。
INIT_00 ~ INIT_7F	16 進数	すべてゼロ ~ すべて 1	すべてゼロ	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	すべてゼロ ~ すべて 1	すべてゼロ	4kb のパリティ メモリ アレイの初期値を指定します。
RAM_EXTENSION_A	文字列	“LOWER”、 “NONE”、 “UPPER”	“NONE”	ポート A カスケード モードを選択します。2 つのブロック RAM をカスケード接続して 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。RAM_MODE=SDP の場合は使用されません。
RAM_EXTENSION_B	文字列	“LOWER”、 “NONE”、 “UPPER”	“NONE”	ポート B カスケード モードを選択します。2 つのブロック RAM をカスケード接続して 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。RAM_MODE=SDP の場合は使用されません。
RAM_MODE	文字列	“TDP”、“SDP”	“TDP”	シングル デュアル ポート (“SDP”) または完全なデュアル ポート (“TDP”) を選択します。
RDADDR_COLLISION_HWCONFIG	文字列	“DELAYED_WRITE”、 “PERFORMANCE”	“DELAYED_WRITE”	<ul style="list-style-type: none"> <li>“PERFORMANCE” に設定すると、READ_FIRST モードでのクロック パフォーマンス (周波数) が向上します。</li> <li>RAM の両方のポートで同じクロックを使用してる場合に “PERFORMANCE” に設定すると、アドレスが重なった場合の競合規則が適用されます。</li> <li>“DELAYED_WRITE” モードでは、競合を発生させずに RAM を使用できます。</li> <li>ES シリコンではサポートされていないので、ES デバイスをターゲットとする場合は “DELAYED_WRITE” に設定する必要があります。</li> </ul>

属性	データ型	値	デフォルト	説明
READ_WIDTH_A	整数	0、1、2、4、9、18、36、72	0	ポート A の読み出しのデータ幅を指定します (パリティビットを含む)。ポートを使用しない場合は、0 に設定する必要があります。ポートを使用する場合は、必要なポート幅に設定してください。
READ_WIDTH_B	整数	0、1、2、4、9、18、36、72	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)。ポートを使用しない場合は、0 に設定する必要があります。ポートを使用する場合は、必要なポート幅に設定してください。
RSTREG_PRIORITY_A	文字列	"RSTREG"、 "REGCE"	"RSTREG"	RSTREG または REGCE のレジスタ優先順位を選択します。TDP モードでポート A に、SDP では下位 36 ビット (パリティビットを含む) に適用されます。
RSTREG_PRIORITY_B	文字列	"RSTREG"、 "REGCE"	"RSTREG"	RSTREG または REGCE のレジスタ優先順位を選択します。TDP でポート B に、SDP では高いほうのビット (パリティビットを含む) に適用されます。
SRVAL_A	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの RAM の出力値を指定します。
SRVAL_B	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの RAM の出力値を指定します。
WRITEMODE	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	書き込みが実行されるときポートの動作を指定します。 <ul style="list-style-type: none"> <li>・ "WRITE_FIRST": 書き込まれた値が出力ポートに出力されます。</li> <li>・ "READ_FIRST": 書き込み前にそのメモリ ロケーションに格納されていた値が出力ポートに出力されます。</li> <li>・ "NO_CHANGE": 出力ポートの以前の値が保持されます。</li> </ul>
WRITE_WIDTH_A	整数	0、1、2、4、9、18、36	0	ポート A 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。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。
WRITE_WIDTH_B	整数	0、1、2、4、9、18、36、72	0	ポート B 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。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。

## 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;

-- RAMB36E1: 36K-bit Configurable Synchronous Block RAM
```

```
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

RAMB36E1_inst : RAMB36E1
generic map (
  -- Collision check: Values ("ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE")
  SIM_COLLISION_CHECK => "ALL",
  -- DOA_REG, DOB_REG: Optional output register (0 or 1)
  DOA_REG => 0,
  DOB_REG => 0,
  -- Error Correction Circuitry (ECC): Encoder/decoder enable (TRUE/FALSE)
  EN_ECC_READ => FALSE,
  EN_ECC_WRITE => FALSE,
  -- INITP_00 to INITP_0F: Initial contents of the parity memory array
  INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  -- INIT_00 to INIT_7F: Initial contents of the data memory array
  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",
```



Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

```

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",
-- INIT_A, INIT_B: Initial values on output ports
INIT_A => X"0000000000",
INIT_B => X"0000000000",
INIT_FILE => "NONE", -- RAM initialization
-- file

-- RAM_EXTENSION_A, RAM_EXTENSION_B: Selects cascade mode ("UPPER", "LOWER", or "NONE")
RAM_EXTENSION_A => "NONE",
RAM_EXTENSION_B => "NONE",
RAM_MODE => "TDP", -- "SDP" or "TDP"
RDADDR_COLLISION_HWCONFIG => "DELAYED_WRITE", -- "PERFORMANCE" or "DELAYED_WRITE"

-- READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per port
READ_WIDTH_A => 0, -- 0, 1, 2, 4, 9, 18,
-- 36, or 72
READ_WIDTH_B => 0, -- 0, 1, 2, 4, 9, 18, or
-- 36
WRITE_WIDTH_A => 0, -- 0, 1, 2, 4, 9, 18, or
-- 36
WRITE_WIDTH_B => 0, -- 0, 1, 2, 4, 9, 18,
-- 36, or 72

-- RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG" or "REGCE")
RSTREG_PRIORITY_A => "RSTREG",
RSTREG_PRIORITY_B => "RSTREG",
-- SRVAL_A, SRVAL_B: Set/reset value for output
SRVAL_A => X"0000000000",
SRVAL_B => X"0000000000",
-- WriteMode: Value on output upon a write ("WRITE_FIRST", "READ_FIRST", or "NO_CHANGE")
WRITE_MODE_A => "WRITE_FIRST",
WRITE_MODE_B => "WRITE_FIRST"
)
port map (
-- Cascade Signals: 1-bit (each) output: BRAM cascade ports (to create 64kx1)
CASCADEOUTA => CASCADEOUTA, -- 1-bit output: A port cascade output
CASCADEOUTB => CASCADEOUTB, -- 1-bit output: B port cascade output
-- ECC Signals: 1-bit (each) output: Error Correction Circuitry ports
DBITERR => DBITERR, -- 1-bit output: double bit error status output
ECCPARITY => ECCPARITY, -- 8-bit output: generated error correction parity
RDADDRECC => RDADDRECC, -- 9-bit output: ECC read address
SBITERR => SBITERR, -- 1-bit output: Single bit error status output
-- Port A Data: 32-bit (each) output: Port A data
DOADO => DOADO, -- 32-bit output: A port data/LSB data output
DOPADOP => DOPADOP, -- 4-bit output: A port parity/LSB parity output
-- Port B Data: 32-bit (each) output: Port B data
DOBDO => DOBDO, -- 32-bit output: B port data/MSB data output
DOPBDOP => DOPBDOP, -- 4-bit output: B port parity/MSB parity output
-- Cascade Signals: 1-bit (each) input: BRAM cascade ports (to create 64kx1)
CASCADEINA => CASCADEINA, -- 1-bit input: A port cascade input
CASCADEINB => CASCADEINB, -- 1-bit input: B port cascade input
-- ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
INJECTDBITERR => INJECTDBITERR, -- 1-bit input: Inject a double bit error
INJECTSBITERR => INJECTSBITERR, -- 1-bit input: Inject a single bit error
-- Port A Address/Control Signals: 16-bit (each) input: Port A address and control signals (read port
-- when RAM_MODE="SDP")
ADDRARDADDR => ADDRARDADDR, -- 16-bit input: A port address/Read address input
CLKARDCLK => CLKARDCLK, -- 1-bit input: A port clock/Read clock input
ENARDEN => ENARDEN, -- 1-bit input: A port enable/Read enable input
REGCEAREGCE => REGCEAREGCE, -- 1-bit input: A port register enable/Register enable input
RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit input: A port set/reset input
RSTREGARSTREG => RSTREGARSTREG, -- 1-bit input: A port register set/reset input
WEA => WEA, -- 4-bit input: A port write enable input

```



Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v13.3) 2011 年 10 月 26 日

```

.RDADDRECC(RDADDRECC),          // 9-bit output: ECC read address
.SBITERR(SBITERR),              // 1-bit output: Single bit error status output
// Port A Data: 32-bit (each) output: Port A data
.DOADO(DOADO),                  // 32-bit output: A port data/LSB data output
.DOPADOP(DOPADOP),              // 4-bit output: A port parity/LSB parity output
// Port B Data: 32-bit (each) output: Port B data
.DOBDO(DOBDO),                  // 32-bit output: B port data/MSB data output
.DOPBDOP(DOPBDOP),              // 4-bit output: B port parity/MSB parity output
// Cascade Signals: 1-bit (each) input: BRAM cascade ports (to create 64kx1)
.CASCADEINA(CASCADEINA),        // 1-bit input: A port cascade input
.CASCADEINB(CASCADEINB),        // 1-bit input: B port cascade input
// ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
.INJECTDBITERR(INJECTDBITERR), // 1-bit input: Inject a double bit error
.INJECTSBITERR(INJECTSBITERR), // 1-bit input: Inject a single bit error
// Port A Address/Control Signals: 16-bit (each) input: Port A address and control signals (read port
// when RAM_MODE="SDP")
.ADDRARDADDR(ADDRARDADDR),      // 16-bit input: A port address/Read address input
.CLKARDCLK(CLKARDCLK),          // 1-bit input: A port clock/Read clock input
.ENARDEN(ENARDEN),              // 1-bit input: A port enable/Read enable input
.REGCEAREGCE(REGCEAREGCE),      // 1-bit input: A port register enable/Register enable input
.RSTRAMARSTRAM(RSTRAMARSTRAM), // 1-bit input: A port set/reset input
.RSTREGARSTREG(RSTREGARSTREG), // 1-bit input: A port register set/reset input
.WEA(WEA),                      // 4-bit input: A port write enable input
// Port A Data: 32-bit (each) input: Port A data
.DIADI(DIADI),                  // 32-bit input: A port data/LSB data input
.DIPADIP(DIPADIP),              // 4-bit input: A port parity/LSB parity input
// Port B Address/Control Signals: 16-bit (each) input: Port B address and control signals (write port
// when RAM_MODE="SDP")
.ADDRBWRADDR(ADDRBWRADDR),      // 16-bit input: B port address/Write address input
.CLKBWRCLK(CLKBWRCLK),          // 1-bit input: B port clock/Write clock input
.ENBWREN(ENBWREN),              // 1-bit input: B port enable/Write enable input
.REGCEB(REGCEB),                // 1-bit input: B port register enable input
.RSTRAMB(RSTRAMB),              // 1-bit input: B port set/reset input
.RSTREGB(RSTREGB),              // 1-bit input: B port register set/reset input
.WEBWE(WEBWE),                  // 8-bit input: B port write enable/Write enable input
// Port B Data: 32-bit (each) input: Port B data
.DIBDI(DIBDI),                  // 32-bit input: B port data/MSB data input
.DIPBDIP(DIPBDIP),              // 4-bit input: B port parity/MSB parity input
);

// End of RAMB36E1_inst instantiation

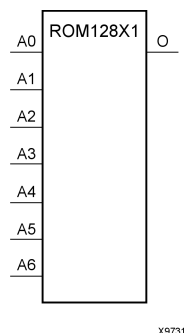
```

## 詳細情報

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

## ROM128X1

プリミティブ：128-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 128 ワード X 1 ビットの ROM です。データ出力 (O) には、7 ビットのアドレス (A6 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 32 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	128 ビット値	すべてゼロ	ROM の値を指定

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
```

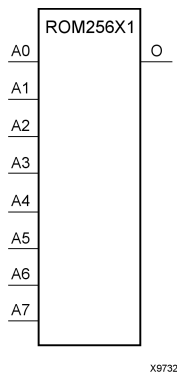


## 詳細情報

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

## ROM256X1

プリミティブ：256-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 256 ワード X 1 ビットの ROM です。データ出力 (O) には、8 ビットのアドレス (A7 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 64 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。

INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	256 ビット値	すべてゼロ	ROM の値を指定

## VHDL 記述 (インスタンス化)

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

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

-- ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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]
);

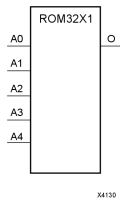
// End of ROM256X1_inst instantiation
```

## 詳細情報

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

## ROM32X1

プリミティブ：32-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 32 ワード X 1 ビットの ROM です。データ出力 (O) には、5 ビットのアドレス (A4 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 8 桁の 16 進数で、ROM には最上位ビット A=1FH から最下位ビット A=00H の順に書き込まれます。

たとえば INIT=10A78F39 と指定すると、「0001 0000 1010 0111 1000 1111 0011」というデータストリームが生成されます。INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## デザインの入力方法

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

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべてゼロ	ROM の値を指定

## VHDL 記述 (インスタンス化)

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

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

-- ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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
//          Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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]
);

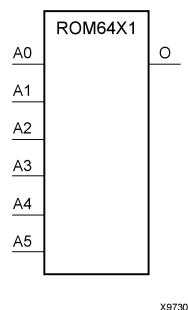
// End of ROM32X1_inst instantiation
```

## 詳細情報

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

## ROM64X1

プリミティブ：64-Deep by 1-Wide ROM



### 概要

このデザイン エLEMENTは 64 ワード X 1 ビットの ROM です。データ出力 (O) には、6 ビットのアドレス (A5 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 16 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。INIT を指定しないと、エラーが発生します。

### 論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## デザインの入力方法

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	ROM の値を指定

## VHDL 記述 (インスタンス化)

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

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

-- ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
```

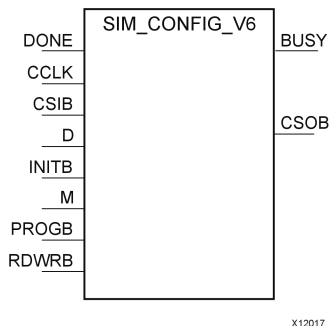


## 詳細情報

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

## SIM\_CONFIG\_V6

## シミュレーション：Configuration Simulation Model



## 概要

このシミュレーション コンポーネントを使用すると、多数の一般的なコンフィギュレーション インターフェイス、ファンクション、およびコマンドの論理シミュレーションを実行でき、コンフィギュレーション動作をボード レベルで理解し、デバッグするのに役立ちます。また、デザインのグローバル セット/リセット (GSR) やグローバル トライステート (GTS) などの一部のスタートアップ動作もシミュレーションできます。このモデルは、FPGA ソフトウェアの特定のプリミティブにはマップされず、デザインに直接インスタンス化することはできませんが、テストベンチなどのシミュレーションのみのファイルで指定するなど、合成で除外してデザイン ネットリストに含まれないようにすれば、ソース デザインと共に使用できます。このモデルは、論理 (RTL) シミュレーションおよびタイミング シミュレーションで使用できます。また、ICAP\_VIRTEX6 をインスタンス化し、そのコンフィギュレーション アクセスをシミュレーションする際にも間接的に使用されます。

## ポートの説明

ポート名	方向	幅	機能
BUSY	出力	1	リードバック中に使用されます。
CSOB	出力	1	アクティブ Low のパラレル デイジー チェーン チップ セレクト出力。FPGA が 1 つのみのアプリケーションでは使用されません。
DONE	入出力	1	コンフィギュレーションの完了を示すアクティブ High の信号 <ul style="list-style-type: none"> <li>0 = FPGA のコンフィギュレーションは完了していない</li> <li>1 = FPGA のコンフィギュレーション完了</li> </ul>
CCLK	入力	1	JTAG を除くすべてのコンフィギュレーション モードのコンフィギュレーション クロック ソース
CSIB	入力	1	SelectMAP データ バスをイネーブルにするアクティブ Low のチップ セレクト <ul style="list-style-type: none"> <li>0 = SelectMAP データ バスをイネーブル</li> <li>1 = SelectMAP データ バスをディスエーブル</li> </ul>
D	入力	32	コンフィギュレーションおよびリードバック データ バス。CCLK の立ち上がりエッジで供給されます。
INITB	入力	1	モード ピンが読み込まれる前は、Low に保持することでコンフィギュレーションを遅延できます。モード ピンが読み込まれた後は、オープン ドレインのアクティブ Low 出力となり、コンフィギュレーション中の CRC エラーの有無を示します。 <ul style="list-style-type: none"> <li>0 = CRC エラー</li> </ul>

ポート名	方向	幅	機能
			<ul style="list-style-type: none"> <li>1 = CRC エラーなし</li> </ul> SEU 検出ファンクションが有効の場合、リードバック CRC エラーが検出されると Low に駆動されます (オプション)。
M	入力	2	モード ピン。コンフィギュレーション モードを指定します。
PROGB	入力	1	アクティブ Low の非同期フルチップ リセット
RDWRB	入力	1	D[x:0] データ バスの方向を指定します。 <ul style="list-style-type: none"> <li>0 = 入力</li> <li>1 = 出力</li> </ul> RDWRB 入力は、CSLB がディアサートの場合にのみ変更可能です。CSLB がディアサートされていない場合は、ABORT が発生します。

## デザインの入力方法

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

テストベンチ ファイルにインスタンス化し、インプリメンテーション ファイルまたはデザインの合成に使用されるファイルには含めないことをお勧めします。コンフィギュレーションの読み込みとデバイスのスタートアップの関係およびスタートアップ シーケンスを決定するために使用できます。通常このモデルは、コンフィギュレーションの動作を観察するためコンフィギュレーション ビットストリーム ファイルで使います。

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DEVICE_ID	32 ビットの 16 進数	有効なデバイス ID コード	32'h00000000	ターゲット デバイスのデバイス ID コードを指定します。ビットストリームの処理およびデバイスの識別読み出しで使用されます。

## VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- SIM_CONFIG_V6: Behavioral Simulation-only Model of FPGA SelectMap Configuration
--               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

SIM_CONFIG_V6_inst : SIM_CONFIG_V6
generic map (
  ICAP_SUPPORT => FALSE,      -- Using ICAP, TRUE or FALSE
  ICAP_WIDTH => "X8",        -- ICAP width, "X8", "X16", "X32"
)

```

```

-- Do not need to change/specify if
-- ICAP_SUPPORT=FALSE
DEVICE_ID => X"00000000") -- Specifies the Pre-programmed Device ID value
port map (
  BUSY => BUSY,    -- 1-bit output Busy pin
  CSOB => CSOB,    -- 1-bit output chip select pin
  DONE => DONE,    -- 1-bit bi-directional Done pin
  CCLK => CCLK,    -- 1-bit input configuration clock
  D => D,          -- 8-bit bi-directional configuration data
  INITB => INITB,  -- 1-bit bi-directional INIT status pin
  M => M,          -- 3-bit input Mode pins
  PROGB => PROGB,  -- 1-bit input Program pin
  RDWRB => RDWRB  -- 1-bit input Read/Write pin
);

-- End of SIM_CONFIG_V6_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// SIM_CONFIG_V6: Behavioral Simulation-only Model of FPGA SelectMap Configuration
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

SIM_CONFIG_V6 #(
  .DEVICE_ID(32'h00000000), // Specify DEVICE_ID
  .ICAP_SUPPORT("FALSE"),  // Using ICAP, "TRUE" or "FALSE"
  .ICAP_WIDTH("X8")        // ICAP width, "X8", "X16", "X32"
                          // Do not need to change/specify if ICAP_SUPPORT="FALSE"
) SIM_CONFIG_V6_inst (
  .BUSY(BUSY), // 1-bit output Busy pin
  .CSOB(CSOB), // 1-bit output chip select pin
  .DONE(DONE), // 1-bit bi-directional Done pin
  .CCLK(CCLK), // 1-bit input configuration clock
  .CSB(CSB),   // 1-bit input chip select
  .D(D),       // 32-bit bi-directional configuration data
  .INITB(INITB), // 1-bit bi-directional INIT status pin
  .M(M),       // 3-bit input Mode pins
  .PROGB(PROGB), // 1-bit input Program pin
  .RDWRB(RDWRB) // 1-bit input Read/write pin
);

// End of SIM_CONFIG_V6_inst instantiation

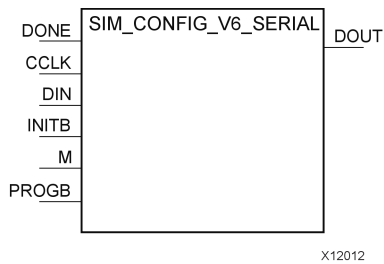
```

## 詳細情報

- ・ [合成/シミュレーション デザイン ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## SIM\_CONFIG\_V6\_SERIAL

### シミュレーション：Serial Configuration Simulation Model



### 概要

このシミュレーション コンポーネントを使用すると、多数の一般的なシリアル コンフィギュレーション インターフェイス、ファンクション、およびコマンドの論理シミュレーションを実行でき、コンフィギュレーション動作をボード レベルで理解し、デバッグするのに役立ちます。また、デザインのグローバル セット/リセット (GSR) やグローバル トライステート (GTS) などの一部のスタートアップ動作もシミュレーションできます。このモデルは、FPGA ソフトウェアの特定のプリミティブにはマップされず、デザインに直接インスタンスエートすることはできませんが、テストベンチなどのシミュレーションのみのファイルで指定するなど、合成で除外してデザイン ネットリストに含まれないようにすれば、ソース デザインと共に使用できます。このモデルは、論理 (RTL) シミュレーションおよびタイミング シミュレーションで使用できます。

### ポートの説明

ポート名	方向	幅	機能
DONE	入出力	1	コンフィギュレーションの完了を示すアクティブ High の信号 <ul style="list-style-type: none"> <li>0 = FPGA のコンフィギュレーションは完了していない</li> <li>1 = FPGA のコンフィギュレーション完了</li> </ul>
DOUT	出力	1	デイジー チェーンのダウストリーム デバイス用のシリアル データ出力。データは CCLK の立ち下がりエッジで供給されます。
CCLK	入力	1	JTAG を除くすべてのコンフィギュレーション モードのコンフィギュレーション クロック ソース
DIN	入力	1	シリアル コンフィギュレーション データ入力 (CCLK の立ち上がりエッジに同期)
INITB	入力	1	モード ピンが読み込まれる前は、Low に保持することでコンフィギュレーションを遅延できます。モード ピンが読み込まれた後は、オープンドレインのアクティブ Low 出力となり、コンフィギュレーション中の CRC エラーの有無を示します。 <ul style="list-style-type: none"> <li>0 = CRC エラー</li> <li>1 = CRC エラーなし</li> </ul> SEU 検出ファンクションが有効の場合、リードバック CRC エラーが検出されると Low に駆動されます (オプション)。
M	入力	2	モード ピン。コンフィギュレーション モードを指定します。
PROGB	入力	1	アクティブ Low の非同期フルチップ リセット

## デザインの入力方法

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

テストベンチ ファイルにインスタンス化し、インプリメンテーション ファイルまたはデザインの合成に使用されるファイルには含めないことをお勧めします。コンフィギュレーションの読み込みとデバイスのスタートアップの関係およびスタートアップ シーケンスを決定するために使用できます。通常このモデルは、コンフィギュレーションの動作を観察するためコンフィギュレーション ビットストリーム ファイルで使われます。

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

## 使用可能な属性

属性	データ型	値	デフォルト	説明
DEVICE_ID	32 ビットの 16 進数	有効なデバイス ID コード	32'h00000000	ターゲット デバイスのデバイス ID コードを指定します。ビットストリームの処理およびデバイスの識別読み出しで使用されます。

## VHDL 記述 (インスタンス化)

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

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

-- SIM_CONFIG_V6_SERIAL: Behavioral Simulation-only Model of FPGA Serial Configuration
--                               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

SIM_CONFIG_V6_SERIAL_inst : SIM_CONFIG_V6_SERIAL
generic map (
  DEVICE_ID => X"00000000") -- Specifies the Pre-programmed Device ID value
port map (
  DONE => DONE,    -- 1-bit bi-directional Done pin
  CCLK => CCLK,    -- 1-bit input configuration clock
  DIN => DIN,      -- 1-bit input configuration data
  INITB => INITB,  -- 1-bit bi-directional INIT status pin
  M => M,          -- 3-bit input Mode pins
  PROGB => PROGB  -- 1-bit input Program pin
);

-- End of SIM_CONFIG_V6_SERIAL_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// SIM_CONFIG_V6_SERIAL: Behavioral Simulation-only Model of FPGA Serial Configuration
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

SIM_CONFIG_V6_SERIAL #(
    .DEVICE_ID(32'h00000000) // Specify DEVICE_ID
) SIM_CONFIG_V6_SERIAL_inst (
    .DONE(DONE),           // 1-bit bi-directional Done pin
    .DOUT(DOUT),           // 1-bit data output pin
    .CCLK(CCLK),           // 1-bit input configuration clock
    .DIN(DIN),             // 1-bit input configuration data
    .INITB(INITB),         // 1-bit bi-directional INIT status pin
    .M(M),                 // 3-bit input Mode pins
    .PROGB(PROGB)         // 1-bit input Program pin
);

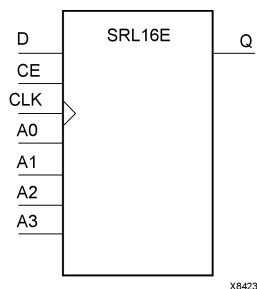
// End of SIM_CONFIG_V6_SERIAL_inst instantiation
```

## 詳細情報

- ・ [合成/シミュレーション デザイン ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## SRL16E

## プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Clock Enable



## 概要

このデザイン エ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 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには：** 入力 A3 ～ A0 の値を変化させます。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ゼロ (0000) にクリアされます。

CE が High の場合、クロック (CLK) が Low から High に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。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	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

## 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-6
-- Xilinx HDL Libraries Guide, version 13.3

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 (インスタンス化)

```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

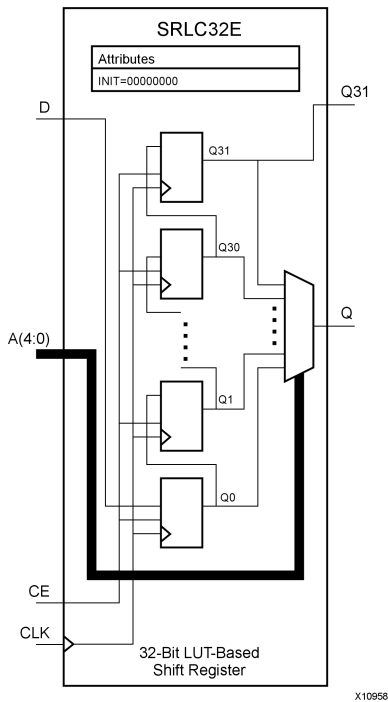
// End of SRL16E_inst instantiation
```

## 詳細情報

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

## SRLC32E

プリミティブ：32 Clock Cycle, Variable Length Shift Register Look-Up Table (LUT) with Clock Enable



### 概要

このデザイン エLEMENTは、1 つのルックアップ テーブル (LUT) にインプリメントされる、可変長で 1 ～ 32 クロック サイクルのシフトレジスタです。シフトレジスタの長さは、固定することも、変動させることもできます。このELEMENTは、アクティブ High のクロック イネーブルおよびカスケード機能も備えているため、複数の SRLC32E をカスケード接続でき、より大型のシフトレジスタを作成できます。

### ポートの説明

ポート名	方向	幅	機能
Q	出力	1	シフトレジスタ データ出力
Q31	出力	1	シフトレジスタ カスケード出力 (後続 SRLC32E の D 入力に接続)
D	入力	1	シフトレジスタ データ入力
CLK	入力	1	クロック
CE	入力	1	アクティブ High のクロック イネーブル
A	入力	5	SRL の長さのダイナミック選択 A=00000 ==> 1 ビット A=11111 ==> 32 ビット

## デザインの入力方法

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

インスタンス化する場合、このコンポーネントを次のように接続します。

- ・ CLK 入力を適切なクロック ソースに、D 入力をシフト/格納するデータ ソースに、Q 出力を FDCPE 入力または FDRSE 入力などの適切なデスティネーションに接続します。
- ・ クロック イネーブル ピン (CE) はクロック イネーブル信号に接続するか、使用しない場合は論理 1 にします。
- ・ 5 ビット バス A は、一定の値 (0 ~ 31) にしてシフトレジスタの長さを 1 ~ 32 ビットに固定するか、または適切な論理値にしてシフトレジスタの長さを 1 ~ 32 ビットの範囲で変更できます。
- ・ シフトレジスタの長さを 32 ビットより長くする場合は、Q31 出力ピンを後続の SRLC32E の D 入力に接続します。
- ・ Q31 出力は、別の SRLC32E 以外には接続できません。
- ・ Q 出力は、カスケード モードでも使用できます。
- ・ 32 ビットの 16 進数の INIT 属性で、シフトレジスタの初期シフト パターンを指定できます。
- ・ シフトアウトされる最初の値は INIT[0] です。

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべてゼロ	SRLC32E の初期シフト パターンを指定

## VHDL 記述 (インスタンス化)

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

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

-- SRLC32E: 32-bit variable length shift register LUT
--           with clock enable
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

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-6
// Xilinx HDL Libraries Guide, version 13.3

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
);

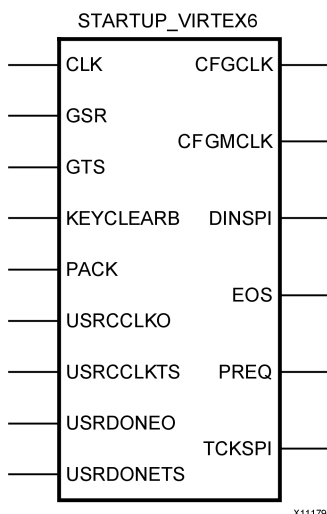
// End of SRLC32E_inst instantiation
```

## 詳細情報

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

## STARTUP\_VIRTEX6

### プリミティブ：Virtex®-6 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終わりにスタートアップ シーケンスで別のクロックを使用するのを指定したり、コンフィギュレーション クロックを内部ロジックにアクセスさせるのにも使用されます。

### ポートの説明

ポート名	方向	幅	機能
CFGCLK	出力	1	コンフィギュレーションのメイン クロック出力。FPGA ファブリックへの出力です。BitGen オプションで指定された周波数のクロック信号を出力します。ソースは内部リング オシレーターです。
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。FPGA ファブリックへの出力です。周波数が 50MHz のクロック信号を出力します。ソースは内部リング オシレーターです。
CLK	入力	1	ユーザー スタートアップ クロック。FPGA ファブリックからの入力です。デバイス スタートアップ シーケンスのスタートアップ クロックを駆動します。基本的にはユーザー定義 CCLK です。
DINSPI	出力	1	DIN SPI PROM アクセス出力。FPGA ファブリックへの出力です。このピン上のデータが、シリアル データとして FPGA に接続されている SPI PROM から読み出されます。検証を実行するために SPI PROM の内容をリードバックする際に有益です。
EOS	出力	1	コンフィギュレーションの終了を示すアクティブ High の信号。FPGA ファブリックへの出力です。スタートアップの終了を示すフラグと同じ信号を FPGA ファブリックに送信します。このピンは、リセット信号として使用できます。
GSR	入力	1	グローバル セット/リセット (GSR) 入力 (ポート名に GSR は使用不可)。FPGA ファブリックからの入力です。グローバル セット/リセット ピンのステートを手動で駆動します。ほとんどのアプリケーションでは、このピンは Low に接続する必要があります。

ポート名	方向	幅	機能
GTS	入力	1	グローバルトライステート (GTS) 入力 (ポート名に GTS は使用不可)。FPGA ファブリックからの入力です。グローバルトライステートのステートを手動で駆動します。ほとんどのアプリケーションでは、このピンは Low に接続する必要があります。
KEYCLEARB	入力	1	バッテリー充電 RAM (BBRAM) からのクリア AES デクリプタ。FPGA ファブリックからの入力です。200ns 間 Low に保持すると、BBRAM からの暗号キーの内容が消去されます。このピンは、PROG_USR 属性を設定した場合にのみイネーブルになります。セーフ操作では High に接続できます。
PACK	入力	1	PROGRAM 確認入力。FPGA ファブリックからの入力です。PROG_B 信号のアサートを確認し、PROG_B ステートマシンによる FPGA のリセットを続行させます。このピンは、PROG_USR 属性を設定した場合にのみイネーブルになります。セーフ操作では Low に接続できます。
PREQ	出力	1	デバイス出力への PROGRAM 要求。FPGA ファブリックへの出力です。PROG_B ステートマシンからのデバイスリセット要求です。これにより、デザインがリセットを完了できるステートになるまで PROG_B のアサート要求が制御されます。このピンは、PROG_USR 属性を設定した場合にのみイネーブルになります。セーフ操作では未接続のままにできます。
TCKSPI	出力	1	TCK コンフィギュレーション ピン アクセス出力。FPGA ファブリックへの出力です。FPGA のコンフィギュレーション インターフェイスに駆動される CCLK を直接コピーしたものです。内部ステートマシンを CCLK に同期する場合に有益です。
USRCCLKO	入力	1	ユーザー CCLK 入力。FPGA ファブリックからの入力です。ファブリックで生成されたカスタム クロック周波数を FPGA ピンの CCLK に駆動します。コンフィギュレーション後に外部 PROM (特に SPI PROM) にアクセスする場合に有益です。
USRCCLKTS	入力	1	内部ユーザー CCLK のトライステート イネーブル。FPGA ファブリックからの入力です。FPGA の CCLK ピンのトライステート機能をイネーブルにします。通常はこのピンを Low に接続し、CCLK ピンがトライステートにならないようにします。
USRDONEO	入力	1	内部ユーザー DONE ピンの出力制御。FPGA ファブリックからの入力です。FPGA DONE ピンを直接駆動します。
USRDONETS	入力	1	ユーザー DONE ピンのトライステート イネーブル。FPGA ファブリックからの入力です。FPGA の DONE ピンのトライステート機能をイネーブルにします。通常は、このピンは Low に接続する必要があります。High に接続すると、DONE がアサートされなくなります。

## 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

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伝えます。

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、このコンポーネントの USRCCLKO ピンと DINSPI ピンを使用して、専用コンフィギュレーション入力ピンにアクセスできるようにします。

## 使用可能な属性

属性	データ型	値	デフォルト	説明
PROG_USR	ブール代数	FALSE、TRUE	FALSE	プログラム イベント セキュリティ機能を有効にします。暗号化ビットストリームを使用している場合にのみ使用可能です。

## 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;

-- STARTUP_VIRTEX6: STARTUP Block
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

STARTUP_VIRTEX6_inst : STARTUP_VIRTEX6
generic map (
  PROG_USR => FALSE -- Activate program event security feature. Requires encrypted bitstreams.
)
port map (
  CFGCLK => CFGCLK,      -- 1-bit output: Configuration main clock output
  CFGMCLK => CFGMCLK,    -- 1-bit output: Configuration internal oscillator clock output
  DINSPI => DINSPI,      -- 1-bit output: DIN SPI PROM access output
  EOS => EOS,            -- 1-bit output: Active high output signal indicating the End Of Configuration.
  PREQ => PREQ,          -- 1-bit output: PROGRAM request to fabric output
  TCKSPI => TCKSPI,      -- 1-bit output: TCK configuration pin access output
  CLK => CLK,            -- 1-bit input: User start-up clock input
  GSR => GSR,            -- 1-bit input: Global Set/Reset input (GSR cannot be used for the port name)
  GTS => GTS,            -- 1-bit input: Global 3-state input (GTS cannot be used for the port name)
  KEYCLEARB => KEYCLEARB, -- 1-bit input: Clear AES Decrypter Key input from Battery-Backed RAM (BBRAM)
  PACK => PACK,          -- 1-bit input: PROGRAM acknowledge input
  USRCCLKO => USRCCLKO,  -- 1-bit input: User CCLK input
  USRCCLKTS => USRCCLKTS, -- 1-bit input: User CCLK 3-state enable input
  USRDONEO => USRDONEO,  -- 1-bit input: User DONE pin output control
  USRDONETS => USRDONETS -- 1-bit input: User DONE 3-state enable output
);

-- End of STARTUP_VIRTEX6_inst instantiation
```



## Verilog 記述 (インスタンスエーション)

```
// STARTUP_VIRTEX6: STARTUP Block
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

STARTUP_VIRTEX6 #(
    .PROG_USR("FALSE") // Activate program event security feature. Requires encrypted bitstreams.
)
STARTUP_VIRTEX6_inst (
    .CFGCLK(CFGCLK), // 1-bit output: Configuration main clock output
    .CFGMCLK(CFGMCLK), // 1-bit output: Configuration internal oscillator clock output
    .DINSPI(DINSPI), // 1-bit output: DIN SPI PROM access output
    .EOS(EOS), // 1-bit output: Active high output signal indicating the End Of Configuration.
    .PREQ(PREQ), // 1-bit output: PROGRAM request to fabric output
    .TCKSPI(TCKSPI), // 1-bit output: TCK configuration pin access output
    .CLK(CLK), // 1-bit input: User start-up clock input
    .GSR(GSR), // 1-bit input: Global Set/Reset input (GSR cannot be used for the port name)
    .GTS(GTS), // 1-bit input: Global 3-state input (GTS cannot be used for the port name)
    .KEYCLEARB(KEYCLEARB), // 1-bit input: Clear AES Decrypter Key input from Battery-Backed RAM (BBRAM)
    .PACK(PACK), // 1-bit input: PROGRAM acknowledge input
    .USRCCLKO(USRCCLKO), // 1-bit input: User CCLK input
    .USRCCLKTS(USRCCLKTS), // 1-bit input: User CCLK 3-state enable input
    .USRDONEO(USRDONEO), // 1-bit input: User DONE pin output control
    .USRDONETS(USRDONETS) // 1-bit input: User DONE 3-state enable output
);

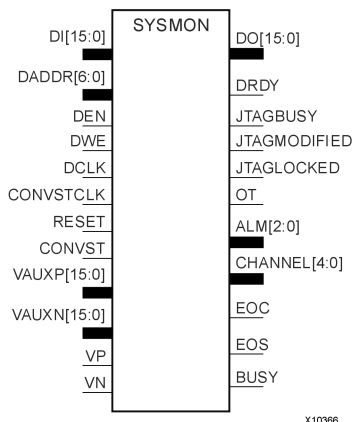
// End of STARTUP_VIRTEX6_inst instantiation
```

## 詳細情報

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

## 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	温度、Vccint および Vccaux の 3 ビット出力アラーム
BUSY	出力	1	1 ビット出力 ADC ビジー信号
CHANNEL[4:0]	出力	5	5 ビット出力チャネル選択
CONVST	入力	1	1 ビット入力変換開始
CONVSTCLK	入力	1	1 ビット入力変換開始クロック
DADDR[6:0]	入力	7	ダイナミック リコンフィギュレーションの 7 ビット入力アドレス バス
DCLK	入力	1	ダイナミック リコンフィギュレーションの 1 ビット入力クロック
DEN	入力	1	ダイナミック リコンフィギュレーションの 1 ビット入力イネーブル
DI[15:0]	入力	16	ダイナミック リコンフィギュレーションの 16 ビット入力データ 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の 16 ビット出力データ バス
DRDY	出力	1	ダイナミック リコンフィギュレーションの 1 ビット出力データレディ
DWE	入力	1	ダイナミック リコンフィギュレーションの 1 ビット入力ライト イネーブル

ポート名	方向	幅	機能
EOC	出力	1	変換の 1 ビット出力エンド
EOS	出力	1	シーケンスの 1 ビット出力エンド
JTAGBUSY	出力	1	1 ビット出力 JTAG DRP ビジー
JTAGLOCKED	出力	1	1 ビット出力 DRP ポート ロック
JTAGMODIFIED	出力	1	DRP への 1 ビット出力 JTAG 書き込み
OT	出力	1	温度アラームの 1 ビット出力
RESET	入力	1	1 ビット入力アクティブ High リセット
VAUXN[15:0]	入力	16	16 ビット入力 N 側補助アナログ入力
VAUXP[15:0]	入力	16	16 ビット入力 P 側補助アナログ入力
VN	入力	1	1 ビット入力 N 側アナログ入力
VP	入力	1	1 ビット入力 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

該当する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の該当するビヘイビアの属性を設定します。シミュレーションには、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
```

**メモ：** 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

属性	データ型	値	デフォルト	説明
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
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

属性	データ型	値	デフォルト	説明
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
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

## 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;

-- SYSMON: System Monitor
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

SYSMON_inst : SYSMON
generic map (
  -- INIT_40 - INIT_42: System Monitor configuration registers
  INIT_40 => X"0000",
  INIT_41 => X"0000",
  INIT_42 => X"0800",
  -- INIT_43 - INIT_47: System Monitor Test registers (do not edit)
  INIT_43 => X"0000",
  INIT_44 => X"0000",
  INIT_45 => X"0000",
  INIT_46 => X"0000",
  INIT_47 => X"0000",
  -- INIT_48 - INIT_4F: Sequence registers for the Channel Sequencer
  INIT_48 => X"0000",
  INIT_49 => X"0000",
  INIT_4A => X"0000",
  INIT_4B => X"0000",
  INIT_4C => X"0000",
  INIT_4D => X"0000",
  INIT_4E => X"0000",
  INIT_4F => X"0000",
  -- INIT_50 - INIT_57: Alarm threshold registers
  INIT_50 => X"0000",
  INIT_51 => X"0000",
  INIT_52 => X"0000",
  INIT_53 => X"0000",
  INIT_54 => X"0000",
  INIT_55 => X"0000",
  INIT_56 => X"0000",
  INIT_57 => X"0000",
  -- Simulation attributes: Set for proepr simulation behavior
  SIM_DEVICE => "VIRTEX5",           -- Must be set to VIRTEX6
  SIM_MONITOR_FILE => "design.txt" -- Analog simulation data file name
)
port map (
  -- Alarm Ports: 3-bit (each) output: ALM, OT
  ALM => ALM,           -- 3-bit output: output alarm for temp, Vccint and Vccaux
  OT => OT,             -- 1-bit output: Over-Temperature alarm output
  -- DRP Ports: 16-bit (each) output: Dynamic Reconfiguration Ports
  DO => DO,             -- 16-bit output: DRP output data bus
  DRDY => DRDY,         -- 1-bit output: DRP data ready output signal
  -- Status Ports: 1-bit (each) output: SYSMON status ports
  BUSY => BUSY,         -- 1-bit output: ADC busy output
  CHANNEL => CHANNEL,   -- 5-bit output: Channel selection outputs
  EOC => EOC,           -- 1-bit output: End of Conversion output
  EOS => EOS,           -- 1-bit output: End of Sequence output
  JTAGBUSY => JTAGBUSY,  -- 1-bit output: JTAG DRP transaction in progress output
  JTAGLOCKED => JTAGLOCKED, -- 1-bit output: JTAG requested DRP port lock output
  JTAGMODIFIED => JTAGMODIFIED, -- 1-bit output: JTAG Write to the DRP has occurred output
  -- Auxilliary Analog-Input Pairs: 16-bit (each) input: VAUXP[15:0], VAUXN[15:0]

```

```

VAUXN => VAUXN,          -- 16-bit input: N-side auxiliary analog input
VAUXP => VAUXP,          -- 16-bit input: P-side auxiliary analog input
-- Control and Clock Ports: 1-bit (each) input: Reset and Conversion Start
CONVST => CONVST,        -- 1-bit input: Convert start input
CONVSTCLK => CONVSTCLK,  -- 1-bit input: Convert start input
RESET => RESET,          -- 1-bit input: Active-high reset input
-- DRP Ports: 7-bit (each) input: Dynamic Reconfiguration Ports
DADDR => DADDR,          -- 7-bit input: DRP input address bus
DCLK => DCLK,            -- 1-bit input: DRP clock input
DEN => DEN,              -- 1-bit input: DRP input enable signal
DI => DI,                -- 16-bit input: DRP input data bus
DWE => DWE,              -- 1-bit input: DRP write enable input
-- Dedicated Analog Input Pair: 1-bit (each) input: VP/VN
VN => VN,                -- 1-bit input: N-side analog input
VP => VP,                -- 1-bit input: P-side analog input
);

-- End of SYSMON_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// SYSMON: System Monitor
//      Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

SYSMON #(
    // INIT_40 - INIT_42: System Monitor configuration registers
    .INIT_40(16'h0000),
    .INIT_41(16'h0000),
    .INIT_42(16'h0800),
    // INIT_43 - INIT_47: System Monitor Test registers (do not edit)
    .INIT_43(16'h0000),
    .INIT_44(16'h0000),
    .INIT_45(16'h0000),
    .INIT_46(16'h0000),
    .INIT_47(16'h0000),
    // INIT_48 - INIT_4F: Sequence registers for the Channel Sequencer
    .INIT_48(16'h0000),
    .INIT_49(16'h0000),
    .INIT_4A(16'h0000),
    .INIT_4B(16'h0000),
    .INIT_4C(16'h0000),
    .INIT_4D(16'h0000),
    .INIT_4E(16'h0000),
    .INIT_4F(16'h0000),
    // INIT_50 - INIT_57: Alarm threshold registers
    .INIT_50(16'h0000),
    .INIT_51(16'h0000),
    .INIT_52(16'h0000),
    .INIT_53(16'h0000),
    .INIT_54(16'h0000),
    .INIT_55(16'h0000),
    .INIT_56(16'h0000),
    .INIT_57(16'h0000),
    // Simulation attributes: Set for proepr simulation behavior
    .SIM_DEVICE("VIRTEX5"),          // Must be set to VIRTEX6
    .SIM_MONITOR_FILE("design.txt")   // Analog simulation data file name
)
SYSMON_inst (
    // Alarm Ports: 3-bit (each) output: ALM, OT
    .ALM(ALM),          // 3-bit output: output alarm for temp, Vccint and Vccaux
    .OT(OT),            // 1-bit output: Over-Temperature alarm output
    // DRP Ports: 16-bit (each) output: Dynamic Reconfiguration Ports
    .DO(DO),            // 16-bit output: DRP output data bus
    .DRDY(DRDY),        // 1-bit output: DRP data ready output signal
    // Status Ports: 1-bit (each) output: SYSMON status ports
    .BUSY(BUSY),        // 1-bit output: ADC busy output
    .CHANNEL(CHANNEL),  // 5-bit output: Channel selection outputs
    .EOC(EOC),          // 1-bit output: End of Conversion output
    .EOS(EOS),          // 1-bit output: End of Sequence output
    .JTAGBUSY(JTAGBUSY), // 1-bit output: JTAG DRP transaction in progress output

```

```
.JTAGLOCKED(JTAGLOCKED), // 1-bit output: JTAG requested DRP port lock output
.JTAGMODIFIED(JTAGMODIFIED), // 1-bit output: JTAG Write to the DRP has occurred output
// Auxiliary Analog-Input Pairs: 16-bit (each) input: VAUXP[15:0], VAUXN[15:0]
.VAUXN(VAUXN), // 16-bit input: N-side auxiliary analog input
.VAUXP(VAUXP), // 16-bit input: P-side auxiliary analog input
// Control and Clock Ports: 1-bit (each) input: Reset and Conversion Start
.CONVST(CONVST), // 1-bit input: Convert start input
.CONVSTCLK(CONVSTCLK), // 1-bit input: Convert start input
.RESET(RESET), // 1-bit input: Active-high reset input
// DRP Ports: 7-bit (each) input: Dynamic Reconfiguration Ports
.DADDR(DADDR), // 7-bit input: DRP input address bus
.DCLK(DCLK), // 1-bit input: DRP clock input
.DEN(DEN), // 1-bit input: DRP input enable signal
.DI(DI), // 16-bit input: DRP input data bus
.DWE(DWE), // 1-bit input: DRP write enable input
// Dedicated Analog Input Pair: 1-bit (each) input: VP/VN
.VN(VN), // 1-bit input: N-side analog input
.VP(VP) // 1-bit input: P-side analog input
);

// End of SYSMON_inst instantiation
```

## 詳細情報

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

## TEMAC\_SINGLE

プリミティブ : Tri-mode Ethernet Media Access Controller (MAC)

TEMAC_SINGLE	
CLIENTEMACPAUSEVAL(15:0)	EMACCLIENTRXD(15:0)
CLIENTEMACTXD(15:0)	EMACCLIENTRXSTATS(6:0)
CLIENTEMACTXIFGDELAY(7:0)	EMACDCRDBUS(0:31)
DCREMACABUS(0:9)	EMACPHYTXD(7:0)
DCREMACDBUS(0:31)	HOSTRDATA(31:0)
HOSTADDR(9:0)	DCRHOSTDONEIR
HOSTOPCODE(1:0)	EMACCLIENTANINTERRUPT
HOSTWRDATA(31:0)	EMACCLIENTRXBADFRAME
PHYEMACPHYAD(4:0)	EMACCLIENTRXCLIENTCLKOUT
PHYEMACRXBUFSTATUS(1:0)	EMACCLIENTRXDVL
PHYEMACRXCLKCORCNT(2:0)	EMACCLIENTRXDVLMSW
PHYEMACRXD(7:0)	EMACCLIENTRXFRAMEDROP
CLIENTEMACDCMLCKED	EMACCLIENTRXGOODFRAME
CLIENTEMACPAUSEREQ	EMACCLIENTRXSTATSBYTEVLD
CLIENTEMACRXCLIENTCLKIN	EMACCLIENTRXSTATSVLD
CLIENTEMACTXCLIENTCLKIN	EMACCLIENTTXACK
CLIENTEMACTXDVL	EMACCLIENTTXCLIENTCLKOUT
CLIENTEMACTXDVLMSW	EMACCLIENTTXCOLLISION
CLIENTEMACTXFIRSTBYTE	EMACCLIENTTXRETRANSMIT
CLIENTEMACTXUNDERRUN	EMACCLIENTTXSTATS
DCREMACCLK	EMACCLIENTTXSTATSBYTEVLD
DCREMACENABLE	EMACCLIENTTXSTATSVLD
DCREMACREAD	EMACDCRACK
DCREMACWRITE	EMACPHYENCOMMAALIGN
HOSTCLK	EMACPHYLOOPBACKMSB
HOSTMIIMSEL	EMACPHYMCLKOUT
HOSTREQ	EMACPHYMDOUT
PHYEMACCOL	EMACPHYMDTRI
PHYEMACCRS	EMACPHYMGTRXRESET
PHYEMACGTCLK	EMACPHYMGTTXRESET
PHYEMACMCLKIN	EMACPHYPOWERDOWN
PHYEMACMDIN	EMACPHYSYNCACQSTATUS
PHYEMACMIITXCLK	EMACPHYTXCHARDISPMODE
PHYEMACRXCHARISCOMMA	EMACPHYTXCHARDISPVLD
PHYEMACRXCHARISK	EMACPHYTXCHARISK
PHYEMACRXCLK	EMACPHYTXCLK
PHYEMACRXDISPERR	EMACPHYTXEN
PHYEMACRXDV	EMACPHYTXER
PHYEMACRXER	EMACPHYTXGMIIICLKOUT
PHYEMACRXNOTINTABLE	EMACSPEDIS10100
PHYEMACRXRUNDISP	HOSTMIIMRDY
PHYEMACSIGNLDET	
PHYEMACTXBUFERR	
PHYEMACTXGMIIICLKIN	
RESET	

X11181

## 概要

Virtex®-6 FPGA エンベデッドトライモード イーサネット MAC をインスタンス化するために必要なポートおよび属性を提供します。SecureIP 暗号化 HDL を含むため、論理およびタイミング シミュレーションにも使用されます。このプリミティブは、イーサネット MAC ラッパーを作成するために、CORE Generator™ を使用して特定ニーズに合わせて変更することができます。



## デザインの入力方法

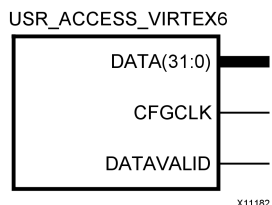
このエレメントをインスタンスエートするには、エンベデッド開発キット (EDK) またはこのエレメントを含む関連コアを使用します。このエレメントは直接インスタンスエートしないでください。

## 詳細情報

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

## USR\_ACCESS\_VIRTEX6

### プリミティブ：Virtex-6 User Access Register



### 概要

このデザイン エレメントを使用すると、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
CFGCLK	出力	1	コンフィギュレーション クロック出力
DATA[31:0]	出力	32	コンフィギュレーション データ出力
DATAVALID	出力	1	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

### デザインの入力方法

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

### VHDL 記述（インスタンス化）

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- USR_ACCESS_VIRTEX6: Configuration Data Access
--                               Virtex-6
-- Xilinx HDL Libraries Guide, version 13.3

USR_ACCESS_VIRTEX6_inst : USR_ACCESS_VIRTEX6
port map (
  CFGCLK => CFGCLK,      -- 1-bit output: Configuration Clock output
  DATA  => DATA,        -- 32-bit output: Configuration Data output
  DATAVALID => DATAVALID -- 1-bit output: Active high data valid output
);

-- End of USR_ACCESS_VIRTEX6_inst instantiation

```

## Verilog 記述 (インスタンス化)

```
// USR_ACCESS_VIRTEX6: Configuration Data Access
//                               Virtex-6
// Xilinx HDL Libraries Guide, version 13.3

USR_ACCESS_VIRTEX6 USR_ACCESS_VIRTEX6_inst (
    .CFGCLK(CFGCLK),           // 1-bit output: Configuration Clock output
    .DATA(DATA),               // 32-bit output: Configuration Data output
    .DATAVALID(DATAVALID)     // 1-bit output: Active high data valid output
);

// End of USR_ACCESS_VIRTEX6_inst instantiation
```

## 詳細情報

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