

Virtex-6 ライブラリ ガイド (HDL 用)

UG623 (v12.2) 2010 年 7 月 23 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 2002–2010 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.12.2) 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。

このマニュアルについて

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。

目次

このマニュアルについて	3
デザイン エLEMENT	3
デザインの入力方法	4
1 : UniMacro について	9
BRAM_SDP_MACRO	10
BRAM_SINGLE_MACRO	19
BRAM_TDP_MACRO	29
FIFO_DUALCLOCK_MACRO	40
FIFO_SYNC_MACRO	44
ADDMACC_MACRO	48
ADDSUB_MACRO	50
COUNTER_LOAD_MACRO	52
COUNTER_TC_MACRO	54
EQ_COMPARE_MACRO	56
MACC_MACRO	58
MULT_MACRO	60
2 : ファンクション別分類	63
3 : デザイン エLEMENT	69
AND2B1L	70
BSCAN_VIRTEX6	71
BUFG	74
BUFGCE	76
BUFGCE_1	78
BUFGCTRL	80
BUFGMUX	82
BUFGMUX_1	84
BUFGMUX_CTRL	86
BUFGP	88
BUFH	89
BUFHCE	91
BUFIO	93
BUFIODQS	95
BUFR	97
CAPTURE_VIRTEX6	99

CARRY4	101
CFGLUT5.....	103
DCIRESET	106
DNA_PORT	108
DSP48E1.....	111
EFUSE_USR.....	120
FDCE.....	122
FDPE	124
FDRE	125
FDSE	126
FIFO18E1	127
FIFO36E1	131
FRAME_ECC_VIRTEX6	136
GTHE1_QUAD.....	139
GTXE1	141
IBUF	143
IBUFDS	145
IBUFDS_DIFF_OUT	147
IBUFDS_GTHE1.....	149
IBUFDS_GTXE1	150
IBUFG.....	151
IBUFGDS.....	153
IBUFGDS_DIFF_OUT.....	155
ICAP_VIRTEX6	157
IDDR.....	160
IDDR_2CLK.....	163
IDELAYCTRL.....	166
IOBUF.....	168
IOBUFDS.....	171
IODELAYE1.....	173
ISERDESE1	178
JTAG_SIM_VIRTEX6	185
KEEPER	188
LDCE.....	190
LDPE	192
LUT5	193
LUT5_D	197
LUT5_L	201

LUT6	205
LUT6_2	210
LUT6_D	215
LUT6_L	220
MMCM_ADV	225
MMCM_BASE	232
MUXF7	235
MUXF7_D	237
MUXF7_L	239
MUXF8	241
MUXF8_D	243
MUXF8_L	245
OBUF	247
OBUFDS	249
OBUF_TDS	251
ODDR	253
OR2L	256
OSERDESE1	257
PCIE_2_0	263
PULLDOWN	265
PULLUP	267
RAM128X1D	269
RAM256X1S	272
RAM32M	274
RAM32X1S	278
RAM64M	280
RAM64X1D	284
RAM64X1S	287
RAMB18E1	289
RAMB36E1	298
SIM_CONFIG_V6	311
SIM_CONFIG_V6_SERIAL	314
SRL16E	317
SRLC32E	320
STARTUP_VIRTEX6	323
SYSMON	326
TEMAC_SINGLE	333
USR_ACCESS_VIRTEX6	335

UniMacro について

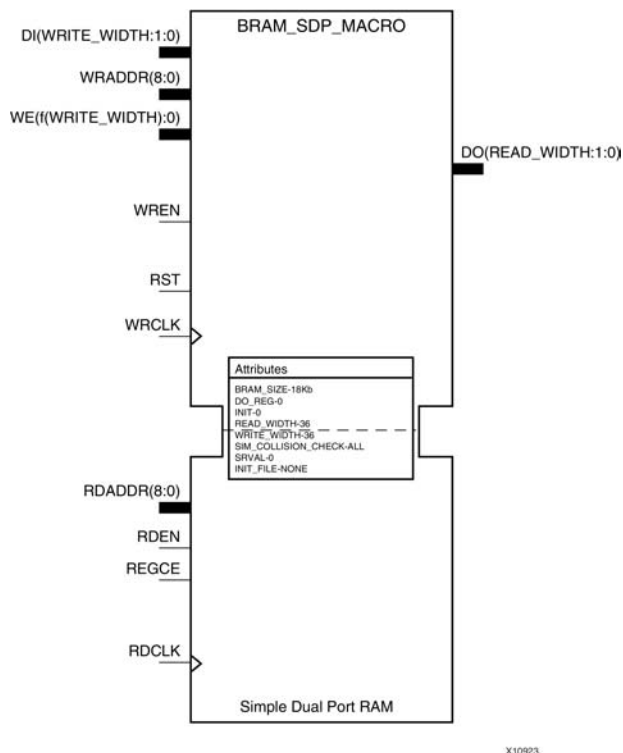
このセクションでは、このアーキテクチャで利用できる UniMacro について説明します。UniMacro は、アルファベット順に並べられています。

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
- ・ インスタンスエーション コードの例
- ・ その他のリソース

BRAM_SDP_MACRO

: Simple Dual Port RAM



概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。読み出しと書き込みは、コンポーネントに供給されるクロックに完全に同期して実行されます。ただし、READ と WRITE 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。

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	書き込み/読み出しイネーブル

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	タイプ	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。

属性	タイプ	値	デフォルト	説明
DO_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロックサイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能ですが、clock-to-out タイムが長くなります。
INIT	16 進数	72 ビット値	すべてゼロ	コンフィギュレーション後の出力の初期値を指定
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	0 ビット STRING	NONE	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にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ・ ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ GENERATE_X_ONLY に設定すると、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。 ・ NONE に設定すると、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。 <p>メモ: 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 または 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

VHDL 記述 (インスタンスーション)

```
Library UNISIM;

use UNISIM.vcomponents.all;

-- BRAM_SDP_MACRO: Simple Dual Port RAM
--               Virtex-6
-- Xilinx HDL Libraries Guide, version 12.2

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

[illegible]

[illegible]

Verilog 記述 (インスタンス化)

```
Library UNISIM;

use UNISIM.vcomponents.all;

-- BRAM_SDP_MACRO: Simple Dual Port RAM
--               Virtex-6
-- Xilinx HDL Libraries Guide, version 12.2

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

BRAM_SDP_MACRO inst : BRAM_SDP_MACRO
```

```
generic map (  
  BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"  
  DEVICE => "VIRTEX6" -- Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6"  
  WRITE_WIDTH => 0,    -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")  
  READ_WIDTH => 0,     -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")  
  DO_REG => 0, -- Optional output register (0 or 1)  
  INIT_FILE => "NONE",  
  SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",  
                                -- "GENERATE_X_ONLY" or "NONE"  
  SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST",  
                                -- see "Synthesis and Simulation Design Guide" for details  
  SRVAL => X"0000000000000000", -- Set/Reset value for port output  
  INIT => X"0000000000000000", -- Initial values on output port  
  -- The following INIT_xx declarations specify the initial contents of the RAM  
  INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",  
  INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",  
  INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",  
  INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",  
  INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",  
  INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",  
  INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",  
  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",
```


tex-6 ライブラリ ガイド (HDL 用)
623 (v12.2) 2010 年 7 月 23 日

```

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
  DI => DI,          -- Input write data port
  RDADDR => RDADDR,  -- Input read address
  RDCLK => RDCLK,    -- Input read clock
  RDEN => RDEN,      -- Input read port enable
  REGCE => REGCE,    -- Input read output register enable
  RST => RST,        -- Input reset
  WE => WE,          -- Input write enable
  WRADDR => WRADDR,  -- Input write address
  WRCLK => WRCLK,    -- Input write clock
  WREN => WREN       -- Input write port enable
);
-- End of BRAM_SDP_MACRO_inst instantiation

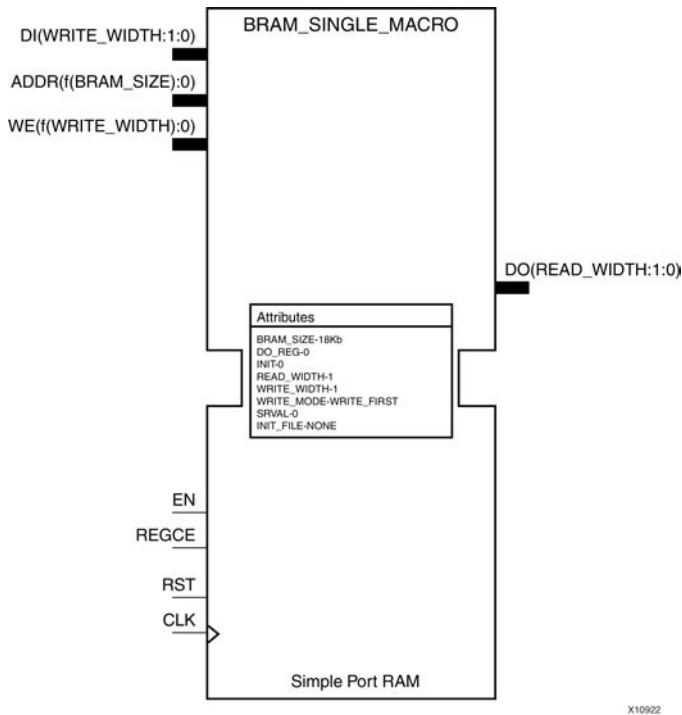
```

詳細情報

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

BRAM_SINGLE_MACRO

: Single Port RAM



概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。これらのシングル ポートのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。バイト イネーブルの書き込みが可能になり、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			
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。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	タイプ	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ンエーブルになり、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	0 ビット STRING	NONE	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 または 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 つの構文が存在しない場合はコピーし、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 12.2

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

BRAM_SINGLE_MACRO_inst : BRAM_SINGLE_MACRO
generic map (
  BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
  DEVICE => "VIRTEX6", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  DO_REG => 0, -- Optional output register (0 or 1)
  INIT_A => X"000000000", -- Initial values on output port
  INIT_FILE => "NONE",
  WRITE_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  READ_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST",
  -- see "Synthesis and Simulation Design Guide" for details
  SRVAL => X"000000000", -- Set/Reset value for port output
  WRITE_MODE => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
```

Virtex-6 ライブラリ ガイド (HDL 用)
UG623 (v12.2) 2010 年 7 月 23 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",

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

-- The next set of INIT_xx are valid when configured as 36Kb
INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DO => DO,      -- Output data
ADDR => ADDR,  -- Input address
CLK => CLK,    -- Input clock
DI => DI,      -- Input data port
EN => EN,      -- Input RAM enable
REGCE => REGCE, -- Input output register enable
RST => RST,    -- Input reset
WE => WE,      -- Input write enable
);

-- End of BRAM_SINGLE_MACRO_inst instantiation

```

Verilog 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

BRAM_SINGLE_MACRO_inst : BRAM_SINGLE_MACRO
generic map (
  BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
  DEVICE => "VIRTEX6", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  DO_REG => 0, -- Optional output register (0 or 1)
  INIT_A => X"000000000", -- Initial values on output port
  INIT_FILE => "NONE",
  WRITE_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  READ_WIDTH => 0, -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST",
  -- see "Synthesis and Simulation Design Guide" for details

```

Virtex-6 ライブラリ ガイド (HDL 用)
UG623 (v12.2) 2010 年 7 月 23 日

Virtex-6 ライブラリ ガイド (HDL 用)
UG623 (v12.2) 2010 年 7 月 23 日

```
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DO => DO,          -- Output data
  ADDR => ADDR,       -- Input address
  CLK => CLK,         -- Input clock
  DI => DI,           -- Input data port
  EN => EN,           -- Input RAM enable
  REGCE => REGCE,     -- Input output register enable
  RST => RST,         -- Input reset
  WE => WE            -- Input write enable
);

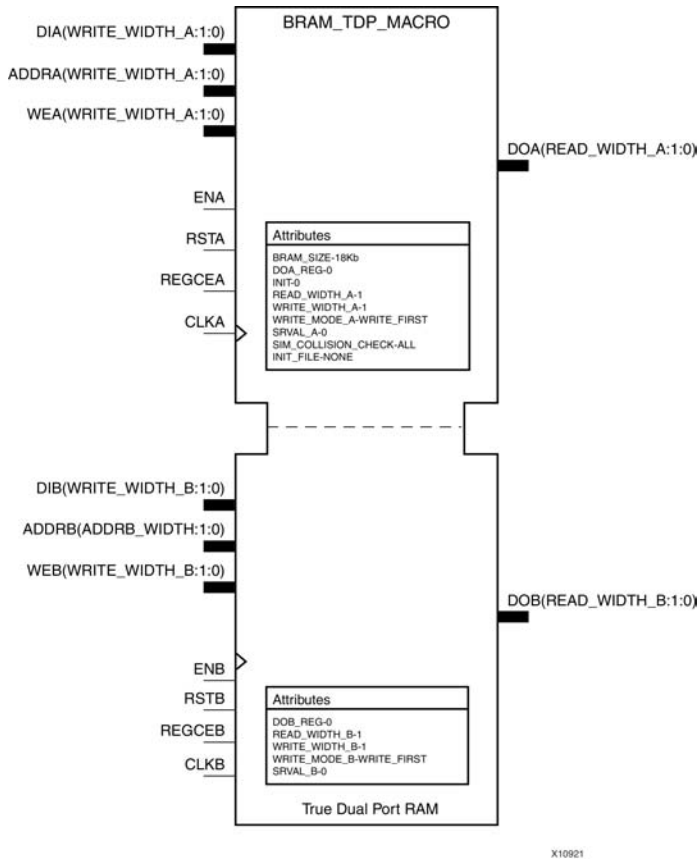
-- End of BRAM_SINGLE_MACRO_inst instantiation
```

詳細情報

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

BRAM_TDP_MACRO

: 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、READ と WRITE 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	タイプ	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
INIT_FILE	文字列	0 ビット STRING	NONE	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にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 GENERATE_X_ONLY に設定すると、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。 NONE に設定すると、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。 <p>メモ: 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	タイプ	値	デフォルト	説明
SIM_MODE	文字列	SAFE または 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 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

BRAM_TDP_MACRO_inst : BRAM_TDP_MACRO
generic map (
  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 => X"000000000", -- Initial values on A output port
  INIT_B => X"000000000", -- Initial values on B output port
  INIT_FILE => "NONE",
  READ_WIDTH_A => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  READ_WIDTH_B => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
  -- "GENERATE_X_ONLY" or "NONE"
  SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST",
  -- see "Synthesis and Simulation Design Guide" for details
  SRVAL_A => X"000000000", -- Set/Reset value for A port output
  SRVAL_B => X"000000000", -- Set/Reset value for B port output
  WRITE_MODE_A => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
  WRITE_MODE_B => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
  WRITE_WIDTH_A => 0, -- Valid values are 1, 2, 4, 9, 18 or 36 (36 only valid when BRAM_SIZE="36Kb")
  WRITE_WIDTH_B => 0, -- Valid values are 1, 2, 4, 9, 18 or 36 (36 only valid when BRAM_SIZE="36Kb")
  -- 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",

```

Virtex-6 ライブラリ ガイド (HDL 用)
UG623 (v12.2) 2010 年 7 月 23 日

Virtex-6 ライブラリ ガイド (HDL 用)
UG623 (v12.2) 2010 年 7 月 23 日

```

RSTA => RSTA,      -- Input port-A reset
RSTB => RSTB,      -- Input port-B reset
WEA  => WEA,      -- Input port-A write enable
WEB  => WEB,      -- Input port-B write enable
);

-- End of BRAM_TDP_MACRO_inst instantiation

```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

BRAM_TDP_MACRO_inst : BRAM_TDP_MACRO
generic map (
  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    => X"000000000", -- Initial values on A output port
  INIT_B    => X"000000000", -- Initial values on B output port
  INIT_FILE => "NONE",
  READ_WIDTH_A => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  READ_WIDTH_B => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
  -- "GENERATE_X_ONLY" or "NONE"
  SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST",
  -- see "Synthesis and Simulation Design Guide" for details
  SRVAL_A => X"000000000", -- Set/Reset value for A port output
  SRVAL_B => X"000000000", -- Set/Reset value for B port output
  WRITE_MODE_A => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
  WRITE_MODE_B => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
  WRITE_WIDTH_A => 0, -- Valid values are 1, 2, 4, 9, 18 or 36 (36 only valid when BRAM_SIZE="36Kb")
  WRITE_WIDTH_B => 0, -- Valid values are 1, 2, 4, 9, 18 or 36 (36 only valid when BRAM_SIZE="36Kb")
  -- The following INIT_xx declarations specify the initial contents of the RAM
  INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",

```

Virtex-6 ライブラリ ガイド (HDL 用)
UG623 (v12.2) 2010 年 7 月 23 日

```
INIT_62 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_63 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_64 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_65 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_66 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_67 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_68 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_69 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_70 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_71 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_72 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_73 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_74 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_75 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_76 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_77 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_78 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_79 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7F => X"0000000000000000000000000000000000000000000000000000000000000000",

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

-- The next set of INIT_xx are valid when configured as 36Kb
INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000")

port map (
  DOA => DOA,      -- Output port-A data
  DOB => DOB,      -- Output port-B data
  ADDRA => ADDRA,   -- Input port-A address
  ADDRb => ADDRb,   -- Input port-B address
  CLKA => CLKA,     -- Input port-A clock
  CLKB => CLKB,     -- Input port-B clock
  DIA => DIA,      -- Input port-A data
  DIB => DIB,      -- Input port-B data
  ENA => ENA,      -- Input port-A enable
  ENB => ENB,      -- Input port-B enable
  REGCEA => REGCEA, -- Input port-A output register enable
  REGCEB => REGCEB, -- Input port-B output register enable
  RSTA => RSTA,     -- Input port-A reset
  RSTB => RSTB,     -- Input port-B reset
  WEA => WEA,      -- Input port-A write enable
  WEB => WEB       -- Input port-B write enable
);

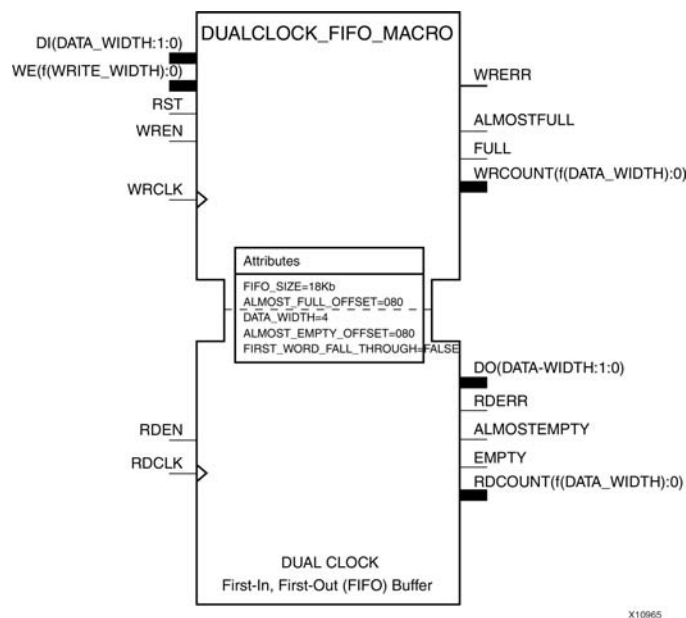
-- End of BRAM_TDP_MACRO_inst instantiation
```

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

: 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、ALMOSTEMPTY フラグと 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ます。
ALMOSTFULL	出力	1	FIFO メモリのほぼすべてのエントリがフル
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 が空です。
FULL	出力	1	FIFO メモリのすべてのエントリがフル
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

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
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	読み出しイネーブル
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	タイプ	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_EMPTY_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
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 または 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 つの構文が存在しない場合はコピーし、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 12.2

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

FIFO_DUALCLOCK_MACRO_inst : FIFO_DUALCLOCK_MACRO
generic map (
    DEVICE => "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
    SIM_MODE => "SAFE") -- Simulation "SAFE" vs "FAST",
    -- see "Synthesis and Simulation Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- Output almost empty
    ALMOSTFULL => ALMOSTFULL, -- Output almost full
    DO => DO, -- Output data
    EMPTY => EMPTY, -- Output empty
    FULL => FULL, -- Output full
    RDCOUNT => RDCOUNT, -- Output read count
    RDERR => RDERR, -- Output read error
    WRCOUNT => WRCOUNT, -- Output write count
    WRERR => WRERR, -- Output write error
    DI => DI, -- Input data
    RDCLK => RDCLK, -- Input read clock
    RDEN => RDEN, -- Input read enable
    RST => RST, -- Input reset
    WRCLK => WRCLK, -- Input write clock
    WREN => WREN -- Input write enable
);
-- End of FIFO_DUALCLOCK_MACRO_inst instantiation

```

Verilog 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

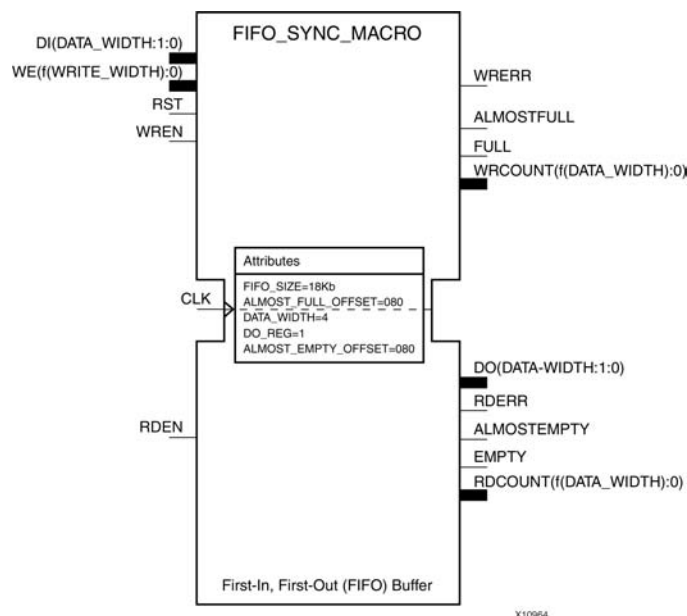
FIFO_DUALCLOCK_MACRO_inst : FIFO_DUALCLOCK_MACRO
generic map (
    DEVICE => "VIRTEX6",           -- Target Device: "VIRTEX5", "VIRTEX6"
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DATA_WIDTH => 0,              -- 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
    SIM_MODE => "SAFE") -- Simulation "SAFE" vs "FAST",
                        -- see "Synthesis and Simulation Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY,    -- Output almost empty
    ALMOSTFULL => ALMOSTFULL,      -- Output almost full
    DO => DO,                      -- Output data
    EMPTY => EMPTY,               -- Output empty
    FULL => FULL,                 -- Output full
    RDCOUNT => RDCOUNT,            -- Output read count
    RDERR => RDERR,               -- Output read error
    WRCOUNT => WRCOUNT,           -- Output write count
    WRERR => WRERR,               -- Output write error
    DI => DI,                    -- Input data
    RDCLK => RDCLK,               -- Input read clock
    RDEN => RDEN,                 -- Input read enable
    RST => RST,                   -- Input reset
    WRCLK => WRCLK,               -- Input write clock
    WREN => WREN                  -- Input write enable
);
-- End of FIFO_DUALCLOCK_MACRO_inst instantiation
```

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

: 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ます。
ALMOSTFULL	出力	1	FIFO メモリのほぼすべてのエントリがフル
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 が空です。
FULL	出力	1	FIFO メモリのすべてのエントリがフル
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ト
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ト
WRERR	出力	1	FIFO がフルのときに書き込みを行うと、アサートされます。

ポート名	方向	幅	機能
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	読み出しイネーブル
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	タイプ	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_EMPTY_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
DO_REG	2 進数	0、1	1	同期 FIFO では、標準の動作を実行するには 0 に設定します。 1 に設定した場合、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 または 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 つの構文が存在しない場合はコピーし、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 12.2

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

FIFO_SYNC_MACRO_inst : FIFO_SYNC_MACRO
generic map (
    DEVICE => "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"
    SIM_MODE => "SAFE") -- Simulation) "SAFE" vs "FAST",
                                -- see "Synthesis and Simulation Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY,    -- Output almost empty
    ALMOSTFULL => ALMOSTFULL,      -- Output almost full
    DO => DO,                      -- Output data
    EMPTY => EMPTY,               -- Output empty
    FULL => FULL,                 -- Output full
    RDCOUNT => RDCOUNT,           -- Output read count
    RDERR => RDERR,              -- Output read error
    WRCOUNT => WRCOUNT,          -- Output write count
    WRERR => WRERR,              -- Output write error
    CLK => CLK,                  -- Input clock
    DI => DI,                    -- Input data
    RDEN => RDEN,                -- Input read enable
    RST => RST,                  -- Input reset
    WREN => WREN                 -- Input write enable
);

```

```
-- End of FIFO_SYNC_MACRO_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

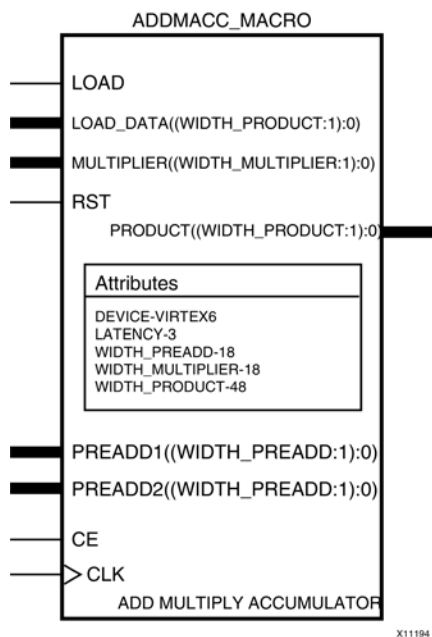
FIFO_SYNC_MACRO_inst : FIFO_SYNC_MACRO
generic map (
    DEVICE => "VIRTEX6",           -- Target Device: "VIRTEX5", "VIRTEX6"
    ALMOST_FULL_OFFSET => X"0080",  -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DATA_WIDTH => 0,              -- Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
    FIFO_SIZE => "18Kb",           -- Target BRAM, "18Kb" or "36Kb"
    SIM_MODE => "SAFE") -- Simulation) "SAFE" vs "FAST",
                        -- see "Synthesis and Simulation Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY,    -- Output almost empty
    ALMOSTFULL => ALMOSTFULL,      -- Output almost full
    DO => DO,                      -- Output data
    EMPTY => EMPTY,               -- Output empty
    FULL => FULL,                 -- Output full
    RDCOUNT => RDCOUNT,           -- Output read count
    RDERR => RDERR,              -- Output read error
    WRCOUNT => WRCOUNT,          -- Output write count
    WRERR => WRERR,              -- Output write error
    CLK => CLK,                  -- Input clock
    DI => DI,                    -- Input data
    RDEN => RDEN,                -- Input read enable
    RST => RST,                  -- Input reset
    WREN => WREN                 -- Input write enable
);
-- End of FIFO_SYNC_MACRO_inst instantiation
```

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

: Adder/Multiplier/Accumulator



X11194

概要

前置加算器、乗算/累積ファンクションとして使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な入力幅および出力幅、レイテンシがあり、DSP48 ブロックを HDL に統合しやすくします。

ポートの説明

ポート名	方向	幅	機能
出力ポート			
PRODUCT	出力	可変幅。WIDTH_A 属性値 + WIDTH_B 属性値に等しい。	プライマリ データ出力
入力ポート			
PREADD1	入力	可変値。WIDTH_PREADD 属性を参照。	前置加算データ入力
PREADD2	入力	可変値。WIDTH_PREADD 属性を参照。	前置加算データ入力
MULTIPLIER	入力	可変値。WIDTH_MULTIPLIER 属性を参照。	乗算器データ入力
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	可変値。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 は A*B+LOAD_DATA でロードされます。
RST	入力	1	同期リセット

デザインの入力方法

この UniMacro はインスタンス化のみが可能で、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

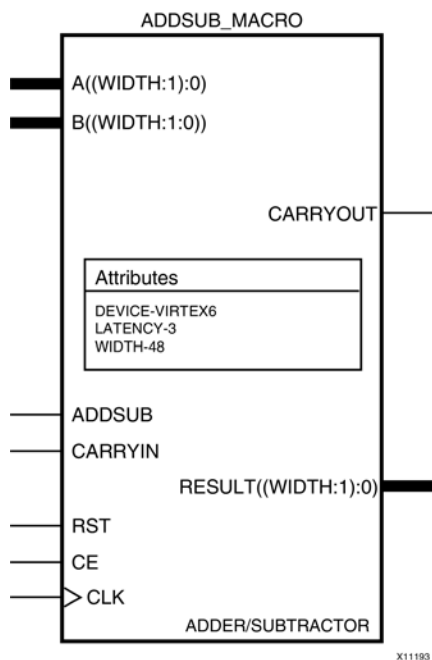
属性	タイプ	値	デフォルト	説明
WIDTH_PREADD	整数	1 ~ 24	24	PREADD1 および PREADD2 入力の幅を制御します。
WIDTH_MULTIPLIER	整数	1 ~ 18	18	MULTIPLIER 入力の幅を制御します。
WIDTH_PRODUCT	整数	1 ~ 48	48	MULTIPLIER 出力の幅を制御します。
LATENCY	整数	0、1、2、3、4	3	パイプライン レジスタの数 <ul style="list-style-type: none"> ・ 1 - MREG == 1 ・ 2 - AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1 ・ 3 - AREG == BREG == 1 および MREG == 1 および PREG == 1 ・ 4 - AREG == BREG == 2 および MREG == 1 および PREG == 1
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

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

: Adder/Subtractor



概要

単純な加算器/減算器として使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な入力幅および出力幅、レイテンシがあり、DSP48 ブロックを HDL に統合しやすくします。

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	可変値。WIDTH 属性を参照。	RDADDR で指定されたデータ出力バス
入力ポート			
ADDSUB	入力	1	High の場合、RESULT は加算器です。Low の場合、RESULT は減算器です。
A	入力	可変値。WIDTH 属性を参照。	加算器/減算器へのデータ入力
B	入力	可変値。WIDTH 属性を参照。	加算器/減算器へのデータ入力
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

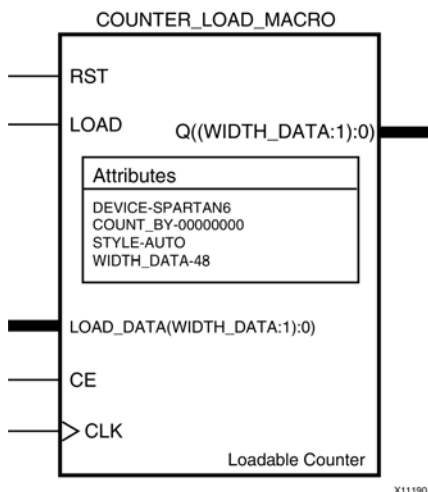
属性	タイプ	値	デフォルト	説明
DEVICE	文字列	VIRTEX6、 SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ
LATENCY	整数	0、1、2	2	パイプライン レジスタの数 <ul style="list-style-type: none"> 1 - PREG == 1 2 - AREG == BREG == CREG == PREG
WIDTH	整数	1 ～ 48	48	A、B、RESULT ポート幅。B および RESULT ポート幅をほかのパラメータ使 用し上書きできます。
WIDTH_RESULT	整数	1 ～ 48	48	RESULT ポート幅上書き

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

: Loadable Counter



概要

ダイナミック ロード アップ/ダウン カウンタとして使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な出力幅およびカウント値があり、DSP48 ブロックを HDL に統合しやすくします。

ポートの説明

ポート名	方向	幅	機能
出力ポート			
Q	出力	可変値。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	可変値。WIDTH_DATA 属性を参照。	アサートされると、LOAD_DATA からカウンタをロードします (2 クロックのレイテンシ)。
LOAD_DATA	入力	可変値。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	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

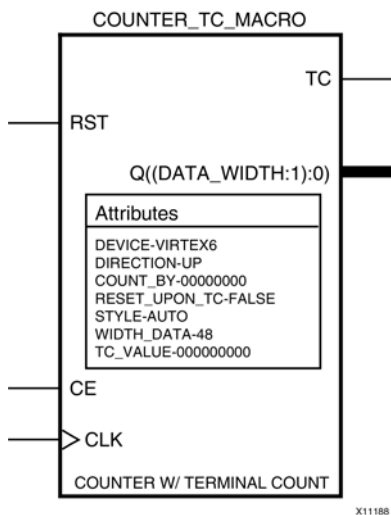
属性	タイプ	値	デフォルト	説明
DEVICE	文字列	VIRTEX6、 SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ
COUNT_BY	16 進数	48 ビット値	000000000001	N ごとにカウント。WIDTH_DATA より優先されます。
WIDTH_DATA	整数	1 ～ 48	48	カウンタ幅を指定します。

詳細情報

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

COUNTER_TC_MACRO

: Counter with Terminal Count



概要

ターミナル アップ/ダウン カウンタとして使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な出力幅、ターミナル カウンタ値、カウント設定、カウント方向があり、DSP48 ブロックを HDL に統合しやすくします。

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 が得られると、ターミナル カウントが High になります。
Q	出力	可変値。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

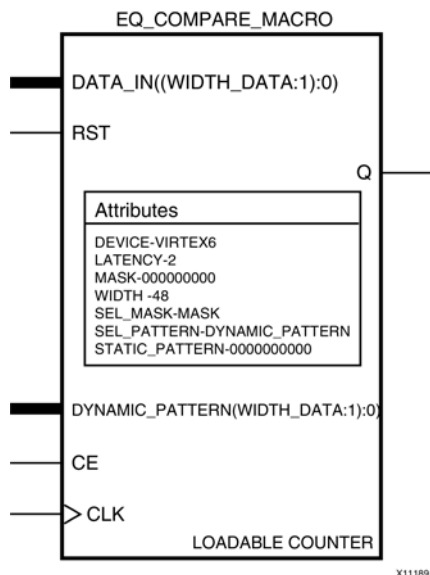
属性	タイプ	値	デフォルト	説明
RESET_UPON_TC	ブール代数	TRUE、FALSE	FALSE	ターミナル カウント達成時にカウンタをリセットするかどうかを指定します。
DEVICE	文字列	VIRTEX6、SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ
DIRECTION	文字列	UP、DOWN	UP	カウンタ方向を指定します。
COUNT_BY	16 進数	48 ビット値	000000000001	N ごとにカウント。WIDTH_DATA より優先されます。
TC_VALUE	16 進数	48 ビット値	すべてゼロ	ターミナル カウント値
WIDTH_DATA	整数	1 ~ 48	48	カウンタ幅を指定します。

詳細情報

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

EQ_COMPARE_MACRO

: Equality Comparator



概要

等価コンパレータとして使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な入力幅および出力幅、レイテンシ、マスク、入力ソースがあり、DSP48 ブロックを HDL に統合しやすくします。

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 属性で指定した 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	可変幅。WIDTH 属性値に等しい。	比較する入力データ
DYNAMIC_PATTERN	入力	可変幅。WIDTH 属性値に等しい。	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

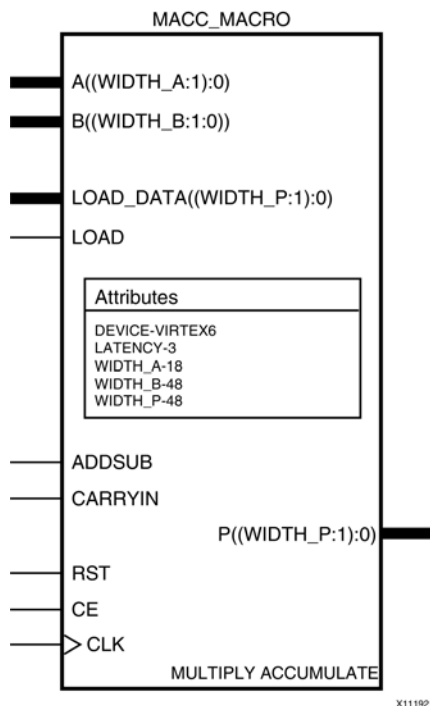
属性	タイプ	値	デフォルト	説明
DEVICE	文字列	VIRTEX5、 VIRTEX6、 SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ
SEL_PATTERN	整数	1 ～ 24	24	PREADD1 および PREADD2 入力の幅を制御します。
MASK	16 進数	48 ビットの 16 進数	すべてゼロ	パターン検出器で使用するマスクを指定します。
STATIC_PATTERN	16 進数	48 ビットの 16 進数	すべてゼロ	パターン検出器で使用するパターンを指定します。
SEL_MASK	文字列	MASK、 DYNAMIC_PATTERN	MASK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

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

: Multiplier/Accumulator



概要

単純な符号付乗算器/アキュムレータ モードで使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な入力幅および出力幅、レイテンシがあり、DSP48 ブロックを HDL に統合しやすくします。

ポートの説明

ポート名	方向	幅	機能
出力ポート			
P	出力	可変幅。WIDTH_A 属性値 + WIDTH_B 属性値に等しい。	プライマリ データ出力
入力ポート			
A	入力	可変値。WIDTH_A 属性を参照。	乗算器データ入力
B	入力	可変値。WIDTH_B 属性を参照。	乗算器データ入力
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	可変幅。WIDTH_A 属性値 + WIDTH_B 属性値に等しい。	DSP スライスでは、LOAD がアサートされると P は $A*B + \text{LOAD_DATA}$ でロードされます。

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乗算モードに、Low の場合は減算モードに設定します。

デザインの入力方法

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

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

使用可能な属性

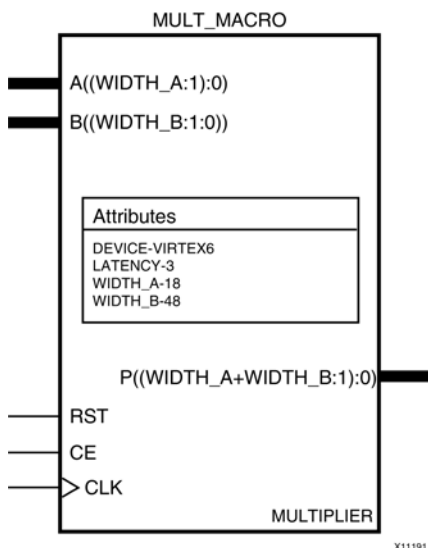
属性	タイプ	値	デフォルト	説明
WIDTH_A	整数	1 ~ 25	25	A 入力の幅を制御します。
WIDTH_B	整数	1 ~ 18	18	B 入力の幅を制御します。
LATENCY	整数	0、1、2、3、4	3	パイプライン レジスタの数 <ul style="list-style-type: none"> 1 - MREG == 1 2 - AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1 3 - AREG == BREG == 1 および MREG == 1 および PREG == 1 4 - AREG == BREG == 2 および MREG == 1 および PREG == 1
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

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

: Multiplier



概要

単純な乗算器として使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な入力幅および出力幅、レイテンシがあり、DSP48 ブロックを HDL に統合しやすくします。

ポートの説明

ポート名	方向	幅	機能
出力ポート			
P	出力	可変幅。WIDTH_A 属性値 + WIDTH_B 属性値に等しい。	プライマリ データ出力
入力ポート			
A	入力	可変値。WIDTH_A 属性を参照。	乗算器データ入力
B	入力	可変値。WIDTH_B 属性を参照。	乗算器データ入力
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	タイプ	値	デフォルト	説明
WIDTH_A	整数	1 ~ 25	25	A 入力の幅を制御します。
WIDTH_B	整数	1 ~ 18	18	B 入力の幅を制御します。
LATENCY	整数	0、1、2、3、4	3	パイプライン レジスタの数 <ul style="list-style-type: none">・ 1 - MREG == 1・ 2 - AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1・ 3 - AREG == BREG == 1 および MREG == 1 および PREG == 1・ 4 - AREG == BREG == 2 および MREG == 1 および PREG == 1
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

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	I/O コンポーネント	スライス/CLB プリミティブ
四則演算ファンクション	ロジック	
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	
コンビニエンス プリミティブ	シフト レジスタ	

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	プリミティブ : Differential Signaling Input Buffer
IBUFDS_GTHE1	プリミティブ : Differential Clock Input for the GTH Transceiver Reference Clocks
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.
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.

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	コンビニエンス プリミティブ：Global Clock Buffer with Clock Enable
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
BUFGMUX_CTRL	コンビニエンス プリミティブ：2-to-1 Global Clock MUX Buffer
BUFGP	コンビニエンス プリミティブ：Primary Global Buffer for Driving Clocks
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
OBUFDS	プリミティブ : 3-State Output Buffer with Differential Signaling, Active-Low Output Enable
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)

ロジック

デザイン エLEMENT	説明
AND2B1L	プリミティブ : Two input AND gate implemented in place of a Slice Latch
OR2L	プリミティブ : Two input OR gate implemented in place of a Slice Latch

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)
RAM32X1S	プリミティブ : 32-Deep by 1-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
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

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)
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

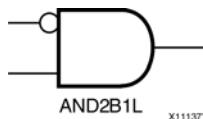
このセクションでは、このアーキテクチャで利用できるデザイン エLEMENTについて説明します。デザイン エLEMENTは、アルファベット順に並べられています。

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)
- ・ インスタンスエーション コードの例
- ・ その他のリソース

AND2B1L

: Two input AND gate implemented in place of a Slice Latch



概要

このデザイン エレメントでは、コンフィギュレーション可能なスライス ラッチで 1 入力反転される 2 入力 AND ゲートのファンクションが使用されます。このエレメントを使用すると、ロジックのレジスタ/ラッチ リソース数をトレードオフにすることで、ロジック レベルを削減して、デバイスのロジック集積度を高めることができます。このエレメントはレジスタのパックおよび集積度に影響を与えるので注意してください。AND2B1L または OR2L エレメントをスライスに指定すると、残りのレジスタおよびラッチが使用できなくなります。

論理表

入力		出力
DI	SRI	O
0	0	0
0	1	0
1	0	1
1	1	0

ポートの説明

ポート名	タイプ	幅	機能
O	出力	1	AND ゲートの出力
DI	入力	1	同じスライスにあるソース LUT に通常接続されるアクティブ High の入力
SRI	入力	1	通常スライス外からソースされるアクティブ Low の入力 メモ : 複数の AND2B1L または OR2B1L を 1 つのスライスにパックするには、この入力に共通の信号を接続する必要があります。

デザインの入力方法

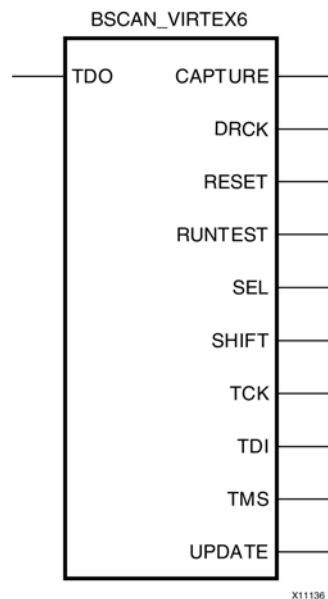
インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

詳細情報

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

BSCAN_VIRTEX6

: Virtex®-6 JTAG Boundary-Scan Logic Access Circuit



概要

このデザイン エLEMENTを使用すると、JTAG バウンダリ スキャン ロジック コントローラ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LEMENTの各インスタンスでは、JTAG_CHAIN 属性の設定に従い、JTAG USER 命令 1 つ (USER1 から USER4 まで) が処理されます。USER 命令の 4 つすべてを処理するには、ELEMENTを 4 つインスタンス化し JTAG_CHAIN 属性を設定します。

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	タイプ	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 コントローラが Run Test Idle ステートのときにアサートされます。Virtex®-6 製品で BSCAN プリミティブと同じ名前が使用されていることを確認してください。
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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.2

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

-- End of BSCAN_VIRTEX6_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

-- End of BSCAN_VIRTEX6_inst instantiation
```

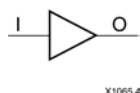
詳細情報

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

BUFG

: Global Clock Buffer

BUFG



概要

このデザイン エレメントは、ファンアウトが大きいバッファで、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	タイプ	幅	機能
I	入力	1	クロック バッファ出力
O	出力	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

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

-- End of BUFG_inst instantiation
```

Verilog 記述 (インスタンス化)

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

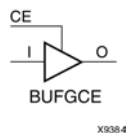
```
Library UNISIM;  
use UNISIM.vcomponents.all;  
  
-- BUFG: Global Clock Buffer  
--      Virtex-6  
-- Xilinx HDL Libraries Guide, version 12.2  
  
BUFG_inst : BUFG  
port map (  
    O => O, -- 1-bit Clock buffer output  
    I => I  -- 1-bit Clock buffer input  
);  
  
-- End of BUFG_inst instantiation
```

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

: Global Clock Buffer with Clock Enable



概要

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

論理表

入力		出力
I	CE	O
X	0	0
I	1	I

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 つの構文が存在しない場合はコピーし、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 12.2

BUFGCE_inst : BUFGCE
port map (
  O => O,      -- 1-bit Clock Output pin
  -- CE0-CE1: 1-bit (each) Clock Enable Input
  CE => CE,
  -- I0-I1: 1-bit (each) Clock Input Pin
  I => I
);

-- End of BUFGCE_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

BUFGCE_inst : BUFGCE
port map (
  O => O,    -- 1-bit Clock Output pin
  -- CE0-CE1: 1-bit (each) Clock Enable Input
  CE => CE,
  -- I0-I1: 1-bit (each) Clock Input Pin
  I => I
);

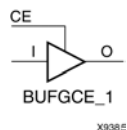
-- End of BUFGCE_inst instantiation
```

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

: Global Clock Buffer with Clock Enable and Output State 1



概要

このデザイン エレメントは、クロック イネーブル付きグローバル クロック バッファです。O 出力は、クロック イネーブル (CE) が Low (非アクティブ) のときに 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

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 つの構文が存在しない場合はコピーし、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 12.2

BUFGCE_1_inst : BUFGCE_1
port map (
    O => O,    -- 1-bit Clock Output pin
    -- CE0-CE1: 1-bit (each) Clock Enable Input
    CE => CE,
    -- I0-I1: 1-bit (each) Clock Input Pin
    I => I
);

-- End of BUFGCE_1_inst instantiation
```

Verilog 記述 (インスタンス化)

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

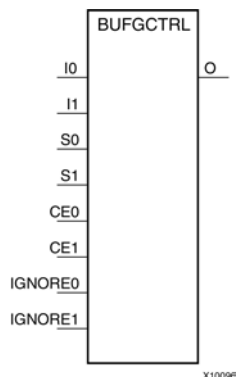
```
Library UNISIM;  
use UNISIM.vcomponents.all;  
  
-- BUFGCE_1: Global Clock Buffer with Clock Enable and Output State 1  
--           Virtex-6  
-- Xilinx HDL Libraries Guide, version 12.2  
  
BUFGCE_1_inst : BUFGCE_1  
port map (  
    O => O,    -- 1-bit Clock Output pin  
    -- CE0-CE1: 1-bit (each) Clock Enable Input  
    CE => CE,  
    -- I0-I1: 1-bit (each) Clock Input Pin  
    I => I  
);  
  
-- 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

: 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	タイプ	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	タイプ	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。

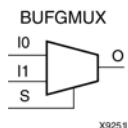
メモ：2 つの PRESELECT 属性を同時に TRUE にすることはできません。

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

: Global Clock MUX Buffer



概要

BUFGMUX はマルチプレクサ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	タイプ	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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

BUFGMUX_inst : BUFGMUX
port map (
  O => O,    -- 1-bit Clock Output pin
  -- CE0-CE1: 1-bit (each) Clock Enable Input
  S => S,
  -- I0-I1: 1-bit (each) Clock Input Pin
  I0 => I0,
  I1 => I1
);

-- End of BUFGMUX_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

BUFGMUX_inst : BUFGMUX
port map (
  O => O,    -- 1-bit Clock Output pin
  -- CE0-CE1: 1-bit (each) Clock Enable Input
  S => S,
  -- I0-I1: 1-bit (each) Clock Input Pin
  I0 => I0,
  I1 => I1
);

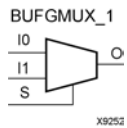
-- End of BUFGMUX_inst instantiation
```

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

: 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。

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

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 つの構文が存在しない場合はコピーし、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 12.2

BUFGMUX_1_inst : BUFGMUX_1
port map (
  O => O,    -- 1-bit Clock Output pin
  -- CE0-CE1: 1-bit (each) Clock Enable Input
  S => S,
  -- I0-I1: 1-bit (each) Clock Input Pin
  I0 => I0,
  I1 => I1
);

-- End of BUFGMUX_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

BUFGMUX_1_inst : BUFGMUX_1
port map (
  O => O,    -- 1-bit Clock Output pin
  -- CE0-CE1: 1-bit (each) Clock Enable Input
  S => S,
  -- I0-I1: 1-bit (each) Clock Input Pin
  I0 => I0,
  I1 => I1
);

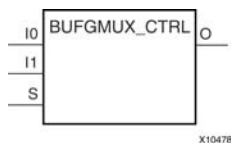
-- End of BUFGMUX_1_inst instantiation
```

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

: 2-to-1 Global Clock MUX Buffer



概要

このデザイン エレメントは、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。このエレメントは、S ピンを 2:1 マルチプレクサのセレクト 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 ビット	クロック出力
I0	入力	1 ビット	2 つのクロック入力の 1 つ
I1	入力	1 ビット	2 つのクロック入力の 1 つ
S	入力	1 ビット	I0 (S=0) または I1 (S=1) クロック出力の選択

デザインの入力方法

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

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

BUFGMUX_CTRL_inst : BUFGMUX_CTRL
port map (
  O => O,    -- 1-bit Clock Output pin
  -- I0-I1: 1-bit (each) Clock Input Pin
  I0 => I0,
  I1 => I1,
  -- S0-S1: 1-bit (each) Clock Select Input
  S => S
);

-- End of BUFGMUX_CTRL_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

BUFGMUX_CTRL_inst : BUFGMUX_CTRL
port map (
  O => O,    -- 1-bit Clock Output pin
  -- I0-I1: 1-bit (each) Clock Input Pin
  I0 => I0,
  I1 => I1,
  -- S0-S1: 1-bit (each) Clock Select Input
  S => S
);

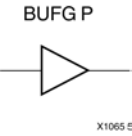
-- End of BUFGMUX_CTRL_inst instantiation
```

詳細情報

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

BUFGP

: Primary Global Buffer for Driving Clocks



概要

このデザイン エLEMENTはプライマリ グローバル バッファであり、FPGA デバイス内でファンアウトの高いクロックまたは制御信号を分配するために使用されます。これは BUFG を駆動する IBUFG と同等です。

このELEMENTを使用すると、コンフィギャブル ロジック ブロック (CLB) および I/O ブロック (IOB) のクロック ピンに直接アクセスでき、ほかの CLB 入力には制限付きでアクセスできます。BUFGP への入力は、専用 IOB からのみ供給されます。構造上、このELEMENTは常にクロック ピンに直接アクセスできるようになっています。ただし、BUFGP がデバイスのどのエッジに配置されているかによって、F3、G1、C3、C1 ピンのいずれか 1 つにしかアクセスできません。必要なピンに垂直ラインから直接アクセスできない場合、PAR で別の CLB を通して信号が送られ、一般配線を使用してそのロード ピンにアクセスします。

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

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

: 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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.1

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

-- End of BUFH_inst instantiation
```

Verilog 記述 (インスタンス化)

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

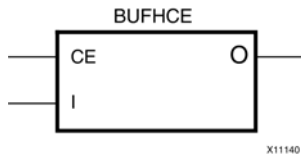
```
Library UNISIM;  
use UNISIM.vcomponents.all;  
  
-- BUFH: HROW Clock Buffer for a Single Clocking Region  
--      Virtex-6  
-- Xilinx HDL Libraries Guide, version 12.1  
  
BUFH_inst : BUFH  
port map (  
    O => O, -- 1-bit Clock Output  
    I => I  -- 1-bit Clock Input  
);  
  
-- End of 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

: Clock buffer for a single clocking region with clock enable



概要

このデザイン エLEMENT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	タイプ	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	タイプ	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 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 with Clock Enable
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 12.2

BUFHCE_inst : BUFHCE
generic map (
  INIT_OUT => 0  -- Initial output value
)
port map (
  O => O,  -- 1-bit Data output
  CE => CE, -- 1-bit Buffer enable input
  I => I   -- 1-bit Data input
);

-- End of BUFHCE_inst instantiation
```

Verilog 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 with Clock Enable
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 12.2

BUFHCE_inst : BUFHCE
generic map (
  INIT_OUT => 0  -- Initial output value
)
port map (
  O => O,  -- 1-bit Data output
  CE => CE, -- 1-bit Buffer enable input
  I => I   -- 1-bit Data input
);

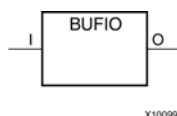
-- End of BUFHCE_inst instantiation
```

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

: 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信号を処理できる 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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.2

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

-- End of BUFIO_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

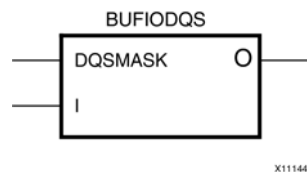
-- End of BUFIO_inst instantiation
```

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

: 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。

ポートの説明

ポート名	タイプ	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	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	タイプ	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。

VHDL 記述 (インスタンス化)

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

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

-- BUFIODQS: Differential Clock Input for Transceiver Reference Clocks
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 12.2

BUFIODQS_inst : BUFIODQS
generic map (
    DQSMASK_ENABLE => FALSE -- Enables the squelch circuitry
)
port map (
    O => O, -- 1-bit Clock output port
    DQSMASK => DQSMASK, -- 1-bit "squelch" the I/O clock after a given burst length from strobe
    I => I -- 1-bit Clock input port
);

-- End of BUFIODQS_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

-- BUFIODQS: Differential Clock Input for Transceiver Reference Clocks
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 12.2

BUFIODQS_inst : BUFIODQS
generic map (
    DQSMASK_ENABLE => FALSE -- Enables the squelch circuitry
)
port map (
    O => O, -- 1-bit Clock output port
    DQSMASK => DQSMASK, -- 1-bit "squelch" the I/O clock after a given burst length from strobe
    I => I -- 1-bit Clock input port
);

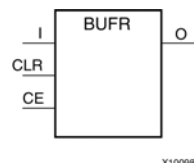
-- End of BUFIODQS_inst instantiation
```

詳細情報

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

BUFR

: Regional Clock Buffer for I/O and Logic Resources



概要

領域クロック 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	タイプ	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	タイプ	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 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

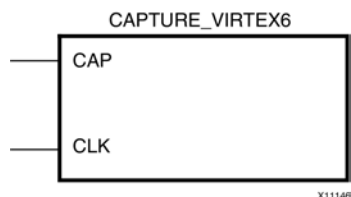
-- End of BUFR_inst instantiation
```

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

: 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。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 に切り替わる時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 (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
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	タイプ	値	デフォルト	説明
ONESHOT	ブール代数	TRUE、FALSE	TRUE	CAPトリガごとに 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 つの構文が存在しない場合はコピーし、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 12.2

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 Capture Input
    CLK => CLK  -- 1-bit Clock Input
);

-- End of CAPTURE_VIRTEX6_inst instantiation
```

Verilog 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 Capture Input
    CLK => CLK  -- 1-bit Clock Input
);

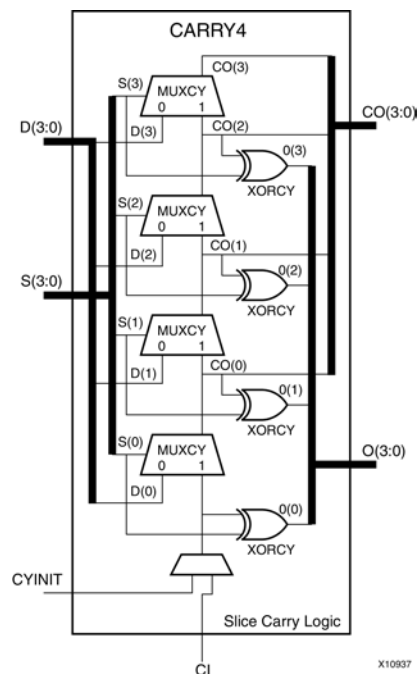
-- End of CAPTURE_VIRTEX6_inst instantiation
```

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

: 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 つの構文が存在しない場合はコピーし、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 12.2

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）

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

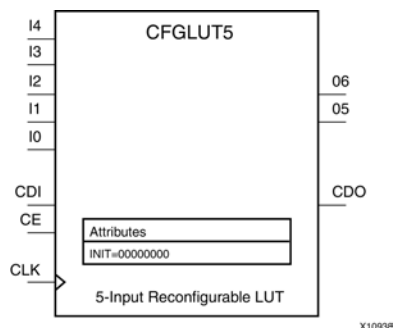
-- End of CARRY4_inst instantiation
```

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

： 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 $(I3, I2, I1, \text{および } I0)$
- ・ $O5 = I3, I2, I1, \text{および } I0$

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	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	このエレメントの初期値を指定します。

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

-- End of CFGLUT5_inst instantiation
```

Verilog 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
```

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

: DCI State Machine Reset (After Configuration Has Been Completed)



概要

このデザイン エレメントは、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	タイプ	幅	機能
LOCKED	出力	1	DCIRESET LOCK ステータス出力
RST	入力	1	DCIRESET 非同期リセット入力

デザインの入力方法

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

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 Impedence Reset Component
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 12.2

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

-- End of DCIRESET_inst instantiation
```

Verilog 記述 (インスタンス化)

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

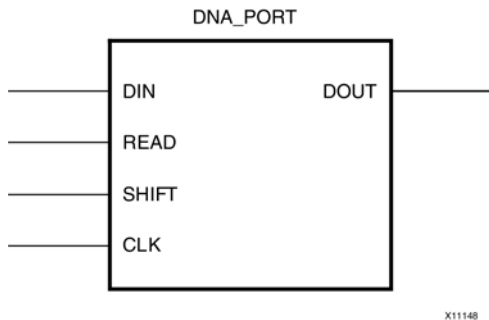
```
Library UNISIM;  
use UNISIM.vcomponents.all;  
  
-- DCIRESET: Digitally Controlled Impedence Reset Component  
--           Virtex-6  
-- Xilinx HDL Libraries Guide, version 12.2  
  
DCIRESET_inst : DCIRESET  
port map (  
    LOCKED => LOCKED, -- 1-bit DCIRESET LOCK status output  
    RST => RST         -- 1-bit DCIRESET 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

: 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 (重複しない 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 (シフトアウト後の DNA データの反復) することもできます。このコンポーネントは主に FPGA ビットストリームの不正なコピー防止用に、ほかの回路と合わせて使用されます。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ます。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読み込む必要があります。シフトレジスタが読み込まれると、アクティブ High の SHIFT 入力がいネーブルになり、DOUT 出力ポートからデータが取り込まれ、データが同期シフトアウトします。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終わりに追加できます。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	タイプ	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。

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読み込む必要があります。シフトレジスタが読み込まれると、アクティブ High の SHIFT 入力がいネーブルになり、DOUT 出力ポートからデータが取り込まれ、データが同期シフトアウトします。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終わりに追加できます。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。

使用可能な属性

属性	タイプ	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 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 DNA output data
    CLK => CLK,   -- 1-bit Clock input
    DIN => DIN,   -- 1-bit User data input pin
    READ => READ, -- 1-bit Active high load DNA, active low read input
    SHIFT => SHIFT -- 1-bit Active high shift enable input
);

-- End of DNA_PORT_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 DNA output data
    CLK => CLK,      -- 1-bit Clock input
    DIN => DIN,      -- 1-bit User data input pin
    READ => READ,    -- 1-bit Active high load DNA, active low read input
    SHIFT => SHIFT   -- 1-bit Active high shift enable input
);

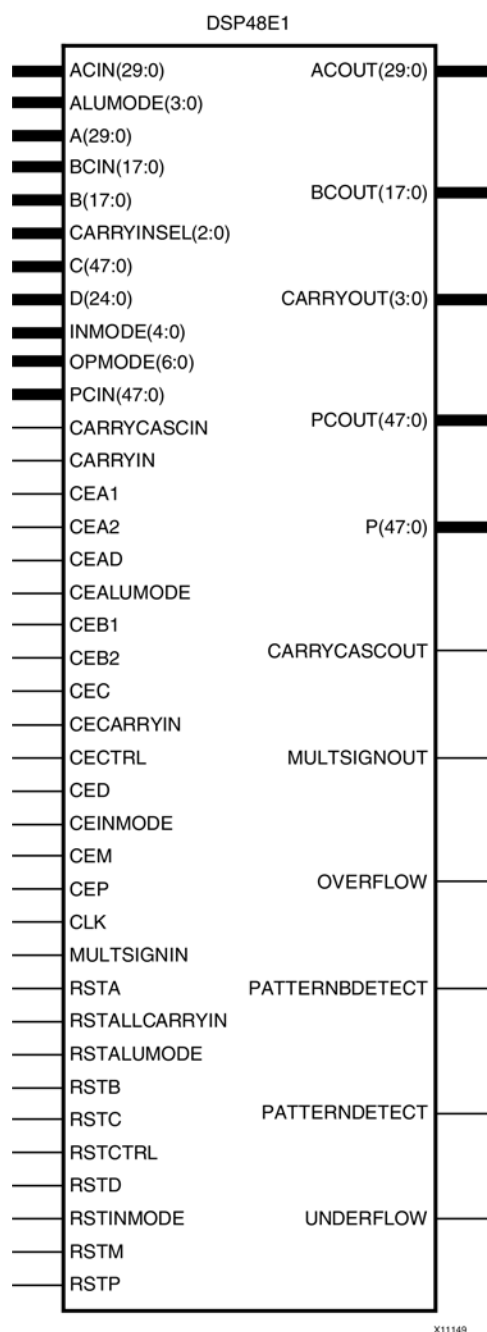
-- End of DNA_PORT_inst instantiation
```

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

: 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LEMENT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	タイプ	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"> 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。 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。 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。
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	タイプ	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	タイプ	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	タイプ	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	タイプ	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	タイプ	値	デフォルト	説明
SEL_MASK	文字列	MASK、C、 ROUNDING_ MODE1、 ROUNDING_ MODE2	MASK	パターン検出器で使用され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 バー 1 つ左にシフト) および ROUNDING_MODE2 (C バー 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され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 つの構文が存在しない場合はコピーし、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 12.2

DSP48E1_inst : DSP48E1
generic map (
    ACASCREG => 1,
    ADREG => 1,
    ALUMODEREG => 1,
    AREG => 1,
    AUTORESET_PATDET => "NO_RESET",
    A_INPUT => "DIRECT",
    BCASCREG => 1,
    -- Number of pipeline registers between A/ACIN input and ACOUT output,
    -- 0, 1, or 2
    -- Number of pipeline registers on pre-adder output, 0 or 1
    -- Number of pipeline registers on ALUMODE input, 0 or 1
    -- Number of pipeline registers on the A input, 0, 1 or 2
    -- NO_RESET, RESET_MATCH, RESET_NOT_MATCH
    -- Selects A input used, "DIRECT" (A port) or "CASCADE" (ACIN port)
    -- Number of pipeline registers between B/BCIN input and BCOUT output,

```

```

-- 0, 1, or 2
BREG => 1, -- Number of pipeline registers on the B input, 0, 1 or 2
B_INPUT => "DIRECT", -- Selects B input used, "DIRECT" (B port) or "CASCADE" (BCIN port)
CARRYINREG => 1, -- Number of pipeline registers for the CARRYIN input, 0 or 1
CARRYINSELREG => 1, -- Number of pipeline registers for the CARRYINSEL input, 0 or 1
CREG => 1, -- Number of pipeline registers on the C input, 0 or 1
DREG => 1, -- Number of pipeline registers on the D input, 0 or 1
INMODEREG => 1, -- Number of pipeline registers on INMODE input, 0 or 1
MASK => X"3fffffff", -- 48-bit Mask value for pattern detect
MREG => 1, -- Number of multiplier pipeline registers, 0 or 1
OPMODEREG => 1, -- Number of pipeline registers on OPMODE input, 0 or 1
PATTERN => X"000000000000", -- 48-bit Pattern match for pattern detect
PREG => 1, -- Number of pipeline registers on the P output, 0 or 1
SEL_MASK => "MASK", -- "C", "MASK", "ROUNDING_MODE1", "ROUNDING_MODE2"
SEL_PATTERN => "PATTERN", -- Select pattern value between the "PATTERN" value or the value on
-- the "C" port
USE_DPORT => FALSE, -- Select D port usage, TRUE or FALSE
USE_MULT => "MULTIPLY", -- Select multiplier usage, "MULTIPLY", "DYNAMIC", or "NONE" (no
-- multiplier)
USE_PATTERN_DETECT => "NO_PATDET", -- Enable pattern detect, "PATDET", "NO_PATDET"
USE_SIMD => "ONE48", -- SIMD selection, "ONE48", "TWO24", "FOUR12"
)
port map (
-- Cascade: 30-bit (each) Cascade
ACOUT => ACOUT, -- 30-bit A port cascade output
BCOUT => BCOUT, -- 18-bit B port cascade output
CARRYCASCOUT => CARRYCASCOUT, -- 1-bit cascade carry output
MULTSIGNOUT => MULTSIGNOUT, -- 1-bit multiplier sign cascade output
PCOUT => PCOUT, -- 48-bit cascade output
-- Control: 1-bit (each) Control
OVERFLOW => OVERFLOW, -- 1-bit overflow in add/acc output
PATTERNBDETECT => PATTERNBDETECT, -- 1-bit active high pattern bar detect output
PATTERNDETECT => PATTERNDETECT, -- 1-bit active high pattern detect output
UNDERFLOW => UNDERFLOW, -- 1-bit active high underflow in add/acc output
-- Data: 4-bit (each) Data
CARRYOUT => CARRYOUT, -- 4-bit carry output
P => P, -- 48-bit output
-- Cascade: 30-bit (each) Cascade
ACIN => ACIN, -- 30-bit A cascade data input
BCIN => BCIN, -- 18-bit B cascade input
CARRYCASCIN => CARRYCASCIN, -- 1-bit cascade carry input
MULTSIGNIN => MULTSIGNIN, -- 1-bit multiplier sign input
PCIN => PCIN, -- 48-bit P cascade input
-- Control: 4-bit (each) Control
ALUMODE => ALUMODE, -- 4-bit ALU control input
CARRYINSEL => CARRYINSEL, -- 3-bit carry select input
CEINMODE => CEINMODE, -- 1-bit active high clock enable input for INMODE registers
CLK => CLK, -- 1-bit Clock input
INMODE => INMODE, -- 5-bit INMODE control input
OPMODE => OPMODE, -- 7-bit operation mode input
RSTINMODE => RSTINMODE, -- 1-bit reset input for INMODE pipeline registers
-- Data: 30-bit (each) Data
A => A, -- 30-bit A data input
B => B, -- 18-bit B data input
C => C, -- 48-bit C data input
CARRYIN => CARRYIN, -- 1-bit carry input signal
D => D, -- 25-bit D data input
-- Reset/Clock Enable: 1-bit (each) Reset/Clock Enable
CEA1 => CEA1, -- 1-bit active high clock enable input for 1st stage A registers
CEA2 => CEA2, -- 1-bit active high clock enable input for 2nd stage A registers
CEAD => CEAD, -- 1-bit active high clock enable input for pre-adder output registers
CEALUMODE => CEALUMODE, -- 1-bit active high clock enable input for ALUMODE registers
CEB1 => CEB1, -- 1-bit active high clock enable input for 1st stage B registers
CEB2 => CEB2, -- 1-bit active high clock enable input for 2nd stage B registers
CEC => CEC, -- 1-bit active high clock enable input for C registers
CECARRYIN => CECARRYIN, -- 1-bit active high clock enable input for CARRYIN register
CECTRL => CECTRL, -- 1-bit active high clock enable input for OPMODE and carry registers
CED => CED, -- 1-bit active high clock enable input for D registers
CEM => CEM, -- 1-bit active high clock enable input for multiplier registers
CEP => CEP, -- 1-bit active high clock enable input for P registers
RSTA => RSTA, -- 1-bit reset input for A pipeline registers
RSTALLCARRYIN => RSTALLCARRYIN, -- 1-bit reset input for carry pipeline registers
RSTALUMODE => RSTALUMODE, -- 1-bit reset input for ALUMODE pipeline registers

```

```

RSTB => RSTB,          -- 1-bit reset input for B pipeline registers
RSTC => RSTC,          -- 1-bit reset input for C pipeline registers
RSTCTRL => RSTCTRL,    -- 1-bit reset input for OPMODE pipeline registers
RSTD => RSTD,          -- 1-bit reset input for D pipeline registers
RSTM => RSTM,          -- 1-bit reset input for multiplier registers
RSTP => RSTP,          -- 1-bit reset input for P pipeline registers
);

-- End of DSP48E1_inst instantiation

```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

DSP48E1_inst : DSP48E1
generic map (
    ACASCREG => 1,          -- Number of pipeline registers between A/ACIN input and ACOUT output,
                           -- 0, 1, or 2
    ADREG => 1,             -- Number of pipeline registers on pre-adder output, 0 or 1
    ALUMODEREG => 1,        -- Number of pipeline registers on ALUMODE input, 0 or 1
    AREG => 1,              -- Number of pipeline registers on the A input, 0, 1 or 2
    AUTORESET_PATDET => "NO_RESET", -- NO_RESET, RESET_MATCH, RESET_NOT_MATCH
    A_INPUT => "DIRECT",    -- Selects A input used, "DIRECT" (A port) or "CASCADE" (ACIN port)
    BCASCREG => 1,          -- Number of pipeline registers between B/BCIN input and BCOUT output,
                           -- 0, 1, or 2
    BREG => 1,             -- Number of pipeline registers on the B input, 0, 1 or 2
    B_INPUT => "DIRECT",    -- Selects B input used, "DIRECT" (B port) or "CASCADE" (BCIN port)
    CARRYINREG => 1,        -- Number of pipeline registers for the CARRYIN input, 0 or 1
    CARRYINSELREG => 1,     -- Number of pipeline registers for the CARRYINSEL input, 0 or 1
    CREG => 1,             -- Number of pipeline registers on the C input, 0 or 1
    DREG => 1,             -- Number of pipeline registers on the D input, 0 or 1
    INMODEREG => 1,        -- Number of pipeline registers on INMODE input, 0 or 1
    MASK => X"3fffffff",   -- 48-bit Mask value for pattern detect
    MREG => 1,             -- Number of multiplier pipeline registers, 0 or 1
    OPMODEREG => 1,        -- Number of pipeline registers on OPMODE input, 0 or 1
    PATTERN => X"000000000000", -- 48-bit Pattern match for pattern detect
    PREG => 1,             -- Number of pipeline registers on the P output, 0 or 1
    SEL_MASK => "MASK",     -- "C", "MASK", "ROUNDING_MODE1", "ROUNDING_MODE2"
    SEL_PATTERN => "PATTERN", -- Select pattern value between the "PATTERN" value or the value on
                           -- the "C" port
    USE_DPORT => FALSE,     -- Select D port usage, TRUE or FALSE
    USE_MULT => "MULTIPLY", -- Select multiplier usage, "MULTIPLY", "DYNAMIC", or "NONE" (no
                           -- multiplier)
    USE_PATTERN_DETECT => "NO_PATDET", -- Enable pattern detect, "PATDET", "NO_PATDET"
    USE_SIMD => "ONE48",    -- SIMD selection, "ONE48", "TWO24", "FOUR12"
)
port map (
    -- Cascade: 30-bit (each) Cascade
    ACOUT => ACOUT,          -- 30-bit A port cascade output
    BCOUT => BCOUT,          -- 18-bit B port cascade output
    CARRYCASCOUT => CARRYCASCOUT, -- 1-bit cascade carry output
    MULTSIGNOUT => MULTSIGNOUT, -- 1-bit multiplier sign cascade output
    PCOUT => PCOUT,          -- 48-bit cascade output
    -- Control: 1-bit (each) Control
    OVERFLOW => OVERFLOW,    -- 1-bit overflow in add/acc output
    PATTERNBDETECT => PATTERNBDETECT, -- 1-bit active high pattern bar detect output
    PATTERNDETECT => PATTERNDETECT, -- 1-bit active high pattern detect output
    UNDERFLOW => UNDERFLOW, -- 1-bit active high underflow in add/acc output
    -- Data: 4-bit (each) Data
    CARRYOUT => CARRYOUT,    -- 4-bit carry output
    P => P,                  -- 48-bit output
    -- Cascade: 30-bit (each) Cascade
    ACIN => ACIN,            -- 30-bit A cascade data input
    BCIN => BCIN,            -- 18-bit B cascade input

```



```

CARRYCASCIN => CARRYCASCIN,      -- 1-bit cascade carry input
MULTSIGNIN  => MULTSIGNIN,        -- 1-bit multiplier sign input
PCIN        => PCIN,              -- 48-bit P cascade input
-- Control: 4-bit (each) Control
ALUMODE     => ALUMODE,           -- 4-bit ALU control input
CARRYINSEL  => CARRYINSEL,        -- 3-bit carry select input
CEINMODE    => CEINMODE,          -- 1-bit active high clock enable input for INMODE registers
CLK         => CLK,               -- 1-bit Clock input
INMODE      => INMODE,           -- 5-bit INMODE control input
OPMODE      => OPMODE,           -- 7-bit operation mode input
RSTINMODE   => RSTINMODE,         -- 1-bit reset input for INMODE pipeline registers
-- Data: 30-bit (each) Data
A           => A,                 -- 30-bit A data input
B           => B,                 -- 18-bit B data input
C           => C,                 -- 48-bit C data input
CARRYIN     => CARRYIN,          -- 1-bit carry input signal
D           => D,                 -- 25-bit D data input
-- Reset/Clock Enable: 1-bit (each) Reset/Clock Enable
CEA1        => CEA1,             -- 1-bit active high clock enable input for 1st stage A registers
CEA2        => CEA2,             -- 1-bit active high clock enable input for 2nd stage A registers
CEAD        => CEAD,             -- 1-bit active high clock enable input for pre-adder output registers
CEALUMODE   => CEALUMODE,        -- 1-bit active high clock enable input for ALUMODE registers
CEB1        => CEB1,             -- 1-bit active high clock enable input for 1st stage B registers
CEB2        => CEB2,             -- 1-bit active high clock enable input for 2nd stage B registers
CEC         => CEC,              -- 1-bit active high clock enable input for C registers
CECARRYIN   => CECARRYIN,        -- 1-bit active high clock enable input for CARRYIN register
CECTRL      => CECTRL,           -- 1-bit active high clock enable input for OPMODE and carry registers
CED         => CED,              -- 1-bit active high clock enable input for D registers
CEM         => CEM,              -- 1-bit active high clock enable input for multiplier registers
CEP         => CEP,              -- 1-bit active high clock enable input for P registers
RSTA        => RSTA,             -- 1-bit reset input for A pipeline registers
RSTALLCARRYIN => RSTALLCARRYIN, -- 1-bit reset input for carry pipeline registers
RSTALUMODE  => RSTALUMODE,       -- 1-bit reset input for ALUMODE pipeline registers
RSTB        => RSTB,             -- 1-bit reset input for B pipeline registers
RSTC        => RSTC,             -- 1-bit reset input for C pipeline registers
RSTCTRL     => RSTCTRL,         -- 1-bit reset input for OPMODE pipeline registers
RSTD        => RSTD,             -- 1-bit reset input for D pipeline registers
RSTM        => RSTM,             -- 1-bit reset input for multiplier registers
RSTP        => RSTP,             -- 1-bit reset input for P pipeline registers
);

-- End of DSP48E1_inst instantiation

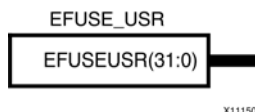
```

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

: 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	タイプ	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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.2

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 User E-Fuse register value output
);

-- End of EFUSE_USR_inst instantiation
```


Verilog 記述 (インスタンス化)

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

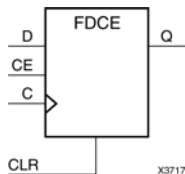
```
Library UNISIM;  
use UNISIM.vcomponents.all;  
  
-- EFUSE_USR: 32-bit non-volatile design ID  
--           Virtex-6  
-- Xilinx HDL Libraries Guide, version 12.2  
  
EFUSE_USR_inst : EFUSE_USR  
generic map (  
    SIM_EFUSE_VALUE => X"00000000" -- Value of the 32-bit non-volatile design ID used in simulation  
)  
port map (  
    EFUSEUSR => EFUSEUSR -- 32-bit 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

: D Flip-Flop with Clock Enable and Asynchronous Clear



概要

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

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

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

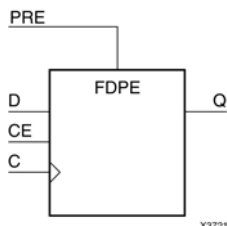
-- End of FDCE_inst instantiation
```

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

: D Flip-Flop with Clock Enable and Asynchronous Preset



概要

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

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

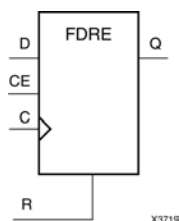
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	1	コンフィギュレーション後の Q 出力の初期値を指定。

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

: D Flip-Flop with Clock Enable and Synchronous Reset



概要

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

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

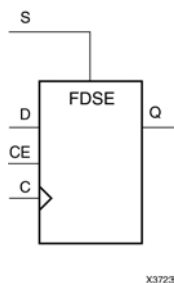
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	0	コンフィギュレーション後の Q 出力の初期値を指定。

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

: 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を追加するとアクティブ Low にできます。

論理表

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

デザインの入力方法

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

使用可能な属性

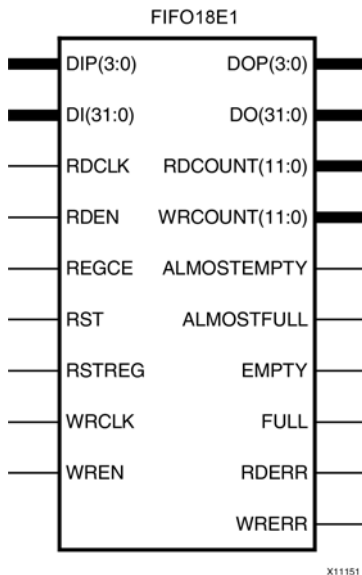
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	1	コンフィギュレーション後の Q 出力の初期値を指定。

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

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

ポート名	タイプ	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"> 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。 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。

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	タイプ	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	タイプ	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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.1

FIFO18E1_inst : FIFO18E1
generic map (
    ALMOST_EMPTY_OFFSET => X"00080", -- Sets the almost empty threshold
    ALMOST_FULL_OFFSET  => X"00080", -- 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"000000000",     -- Initial values on output port
    SRVAL => X"000000000"     -- Set/Reset value for output port
)
port map (
    -- Read Data: 32-bit (each) Read output data
    DO => DO,                -- 32-bit data output
    DOP => DOP,              -- 4-bit parity data output
    -- Status: 1-bit (each) Flags and other FIFO status outputs
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL,   -- 1-bit almost full output flag
    EMPTY => EMPTY,            -- 1-bit empty output flag
    FULL => FULL,              -- 1-bit full output flag
    RDCOUNT => RDCOUNT,         -- 12-bit read count output
    RDERR => RDERR,            -- 1-bit read error output
    WRCOUNT => WRCOUNT,        -- 12-bit write count output
    WRERR => WRERR,           -- 1-bit write error
    -- Read Control Signals: 1-bit (each) Read clock, enable and reset input signals
    RDCLK => RDCLK,           -- 1-bit read clock input
    RDEN => RDEN,             -- 1-bit read enable input
    REGCE => REGCE,          -- 1-bit clock enable input
    RST => RST,               -- 1-bit reset input
    RSTREG => RSTREG,         -- 1-bit output register set/reset
    -- Write Control Signals: 1-bit (each) Write clock and enable input signals
    WRCLK => WRCLK,          -- 1-bit write clock input
    WREN => WREN,            -- 1-bit write enable input
    -- Write Data: 32-bit (each) Write input data
    DI => DI,                -- 32-bit data input
    DIP => DIP,              -- 4-bit parity input
);

-- End of FIFO18E1_inst instantiation

```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.1

FIFO18E1_inst : FIFO18E1
generic map (
    ALMOST_EMPTY_OFFSET => X"00080", -- Sets the almost empty threshold
    ALMOST_FULL_OFFSET => X"00080", -- 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"000000000", -- Initial values on output port
    SRVAL => X"000000000" -- Set/Reset value for output port
)
port map (
    -- Read Data: 32-bit (each) Read output data
    DO => DO, -- 32-bit data output
    DOP => DOP, -- 4-bit parity data output
    -- Status: 1-bit (each) Flags and other FIFO status outputs
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 12-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 12-bit write count output
    WRERR => WRERR, -- 1-bit write error
    -- Read Control Signals: 1-bit (each) Read clock, enable and reset input signals
    RDCLK => RDCLK, -- 1-bit read clock input
    RDEN => RDEN, -- 1-bit read enable input
    REGCE => REGCE, -- 1-bit clock enable input
    RST => RST, -- 1-bit reset input
    RSTREG => RSTREG, -- 1-bit output register set/reset
    -- Write Control Signals: 1-bit (each) Write clock and enable input signals
    WRCLK => WRCLK, -- 1-bit write clock input
    WREN => WREN, -- 1-bit write enable input
    -- Write Data: 32-bit (each) Write input data
    DI => DI, -- 32-bit data input
    DIP => DIP -- 4-bit parity input
);

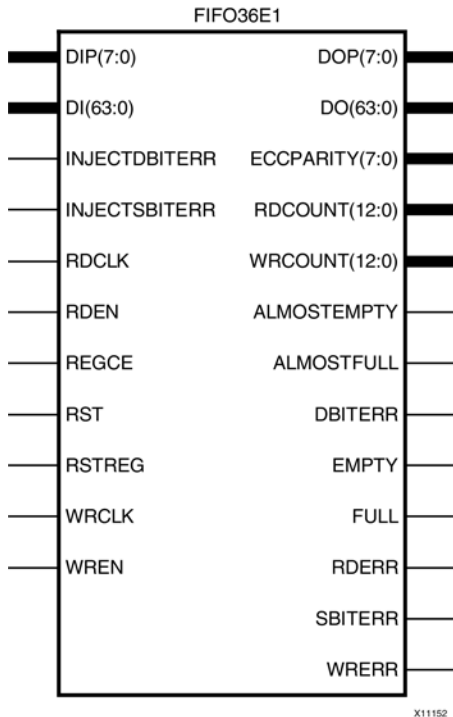
-- End of FIFO18E1_inst instantiation
```

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

: 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要な場合は、FIFO36_72 を使用する必要があります。

ポートの説明

ポート名	タイプ	幅	機能
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示します。このフラグの値は ALMOST_EMPTY_OFFSET 属性で指定します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示します。このフラグの値は 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 デコーダで使用する ECC エンコーダから生成された 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。
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。
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"> 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。 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。

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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.1

FIFO36E1_inst : FIFO36E1
generic map (
    ALMOST_EMPTY_OFFSET => X"00080", -- Sets the almost empty threshold
    ALMOST_FULL_OFFSET => X"00080", -- 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) Error Correction Circuitry ports
    DBITERR => DBITERR, -- 1-bit double bit error status output
    ECCPARITY => ECCPARITY, -- 8-bit generated error correction parity
    SBITERR => SBITERR, -- 1-bit single bit error status output
    -- Read Data: 64-bit (each) Read output data
    DO => DO, -- 64-bit data output
    DOP => DOP, -- 8-bit parity data output
    -- Status: 1-bit (each) Flags and other FIFO status outputs
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 9-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 9-bit write count output
    WRERR => WRERR, -- 1-bit write error
    -- ECC Signals: 1-bit (each) Error Correction Circuitry ports
    INJECTDBITERR => INJECTDBITERR,
    INJECTSBITERR => INJECTSBITERR,
    -- Read Control Signals: 1-bit (each) Read clock, enable and reset input signals
    RDCLK => RDCLK, -- 1-bit read clock input
    RDEN => RDEN, -- 1-bit read enable input
    REGCE => REGCE, -- 1-bit clock enable input
    RST => RST, -- 1-bit reset input
    RSTREG => RSTREG, -- 1-bit output register set/reset
    -- Write Control Signals: 1-bit (each) Write clock and enable input signals
    WRCLK => WRCLK, -- 1-bit write clock input
    WREN => WREN, -- 1-bit write enable input
    -- Write Data: 64-bit (each) Write input data
    DI => DI, -- 64-bit data input
    DIP => DIP, -- 8-bit parity input
);

-- End of FIFO36E1_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.1

FIFO36E1_inst : FIFO36E1
generic map (
    ALMOST_EMPTY_OFFSET => X"00080", -- Sets the almost empty threshold
    ALMOST_FULL_OFFSET => X"00080", -- 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) Error Correction Circuitry ports
    DBITERR => DBITERR, -- 1-bit double bit error status output
    ECCPARITY => ECCPARITY, -- 8-bit generated error correction parity
    SBITERR => SBITERR, -- 1-bit single bit error status output
    -- Read Data: 64-bit (each) Read output data
    DO => DO, -- 64-bit data output
    DOP => DOP, -- 8-bit parity data output
    -- Status: 1-bit (each) Flags and other FIFO status outputs
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 9-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 9-bit write count output
    WRERR => WRERR, -- 1-bit write error
    -- ECC Signals: 1-bit (each) Error Correction Circuitry ports
    INJECTDBITERR => INJECTDBITERR,
    INJECTSBITERR => INJECTSBITERR,
    -- Read Control Signals: 1-bit (each) Read clock, enable and reset input signals
    RDCLK => RDCLK, -- 1-bit read clock input
    RDEN => RDEN, -- 1-bit read enable input
    REGCE => REGCE, -- 1-bit clock enable input
    RST => RST, -- 1-bit reset input
    RSTREG => RSTREG, -- 1-bit output register set/reset
    -- Write Control Signals: 1-bit (each) Write clock and enable input signals
    WRCLK => WRCLK, -- 1-bit write clock input
    WREN => WREN, -- 1-bit write enable input
    -- Write Data: 64-bit (each) Write input data
    DI => DI, -- 64-bit data input
    DIP => DIP, -- 8-bit parity input
);

-- End of FIFO36E1_inst instantiation
```

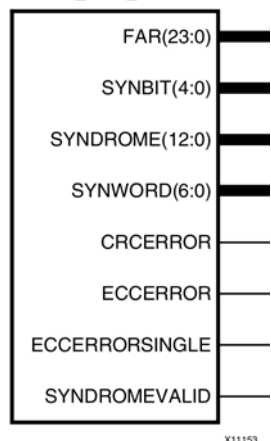
詳細情報

[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

: Virtex®-6 Configuration Frame Error Detection and Correction Circuitry

FRAME_ECC_VIRTEX6



X11153

概要

このデザイン エレメントでは、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。このエレメントには、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	タイプ	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	タイプ	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	0 ビット文字列	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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 indicating a CRC error
    ECCERROR => ECCERROR,            -- 1-bit Output indicating an ECC error
    ECCERRORSINGLE => ECCERRORSINGLE,  -- 1-bit Output Indicating single-bit Frame ECC error detected.
    FAR => FAR,                      -- 24-bit Frame Address Register Value output
    SYNBIT => SYNBIT,                -- 5-bit Output bit address of error
    SYNDROME => SYNDROME,            -- 13-bit Output location of erroneous bit
    SYNDROMEVALID => SYNDROMEVALID,  -- 1-bit Frame ECC output indicating the SYNDROME output is valid.
    SYNWORD => SYNWORD               -- 7-bit Word output in the frame where an ECC error has been detected
);

-- End of FRAME_ECC_VIRTEX6_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 indicating a CRC error  
    ECCERROR => ECCERROR,          -- 1-bit Output indicating an ECC error  
    ECCERRORSINGLE => ECCERRORSINGLE, -- 1-bit Output Indicating single-bit Frame ECC error detected.  
    FAR => FAR,                    -- 24-bit Frame Address Register Value output  
    SYNBIT => SYNBIT,              -- 5-bit Output bit address of error  
    SYNDROME => SYNDROME,          -- 13-bit Output location of erroneous bit  
    SYNDROMEVALID => SYNDROMEVALID, -- 1-bit Frame ECC output indicating the SYNDROME output is valid.  
    SYNWORD => SYNWORD             -- 7-bit 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

: Gigabit Transceiver

DADDR(15:0)	GTHE1_QUAD	DRPDO(15:0)
DI(15:0)		
MGMTPCSLANESEL(3:0)	MGMTPCSRDDATA(15:0)	
MGMTPCSMDADDR(4:0)		RXCODEERR0(7:0)
MGMTPCSREGADDR(15:0)		RXCODEERR1(7:0)
MGMTPCSWORDATA(15:0)		RXCODEERR2(7:0)
PLLPCCLKDIV(5:0)		RXCODEERR3(7:0)
PLREFCLKSEL(2:0)		
RXPPOWERDOWN0(1:0)		RXCTRL0(7:0)
RXPPOWERDOWN1(1:0)		RXCTRL1(7:0)
RXPPOWERDOWN2(1:0)		RXCTRL2(7:0)
RXPPOWERDOWN3(1:0)		RXCTRL3(7:0)
RXRATE0(1:0)		RXDATA0(63:0)
RXRATE1(1:0)		RXDATA1(63:0)
RXRATE2(1:0)		RXDATA2(63:0)
RXRATE3(1:0)		RXDATA3(63:0)
SAMPLERATE0(2:0)		RXDISPERR0(7:0)
SAMPLERATE1(2:0)		RXDISPERR1(7:0)
SAMPLERATE2(2:0)		RXDISPERR2(7:0)
SAMPLERATE3(2:0)		RXDISPERR3(7:0)
TXCTRL0(7:0)		RXVALID0(7:0)
TXCTRL1(7:0)		RXVALID1(7:0)
TXCTRL2(7:0)		RXVALID2(7:0)
TXCTRL3(7:0)		RXVALID3(7:0)
TXDATA0(63:0)		
TXDATA1(63:0)		
TXDATA2(63:0)		
TXDATA3(63:0)		
TXDATAMSB0(7:0)		
TXDATAMSB1(7:0)		
TXDATAMSB2(7:0)		
TXDATAMSB3(7:0)		
TXMARGIN0(2:0)		
TXMARGIN1(2:0)		
TXMARGIN2(2:0)		
TXMARGIN3(2:0)		
TXPOWERDOWN0(1:0)		
TXPOWERDOWN1(1:0)		
TXPOWERDOWN2(1:0)		
TXPOWERDOWN3(1:0)		
TXRATE0(1:0)		
TXRATE1(1:0)		
TXRATE2(1:0)		
TXRATE3(1:0)		
DCLK		DRDY
DEN		
DFETRAINCTRL0	GTHINITDONE	
DFETRAINCTRL1		
DFETRAINCTRL2	MGMTPCSRDACK	
DFETRAINCTRL3		
DISABLEDRP	RXCTRLACK0	
DWE	RXCTRLACK1	
GTHINIT	RXCTRLACK2	
GTHRESET	RXCTRLACK3	
GTHX2LANE01		
GTHX2LANE23		
GTHX4LANE		
MGMTPCSREGRD	RXUSERCLKOUT0	
MGMTPCSREGWR	RXUSERCLKOUT1	
POWERDOWN0	RXUSERCLKOUT2	
POWERDOWN1	RXUSERCLKOUT3	
POWERDOWN2		
POWERDOWN3		
REFCLK		
RXBUFRESET0		
RXBUFRESET1		
RXBUFRESET2		
RXBUFRESET3		
RXENCOMMADET0	TSTPATH	
RXENCOMMADET1	TSTREFCLKFAB	
RXENCOMMADET2	TSTREFCLKOUT	
RXENCOMMADET3		
RXN0	TXCTRLACK0	
RXN1	TXCTRLACK1	
RXN2	TXCTRLACK2	
RXN3	TXCTRLACK3	
RXP0		
RXP1		
RXP2		
RXP3		
RXPOLARITY0	TXN0	
RXPOLARITY1	TXN1	
RXPOLARITY2	TXN2	
RXPOLARITY3	TXN3	
RXSLIP0	TXP0	
RXSLIP1	TXP1	
RXSLIP2	TXP2	
RXSLIP3	TXP3	
RXUSERCLKIN0		
RXUSERCLKIN1		
RXUSERCLKIN2		
RXUSERCLKIN3		
TXBUFRESET0		
TXBUFRESET1		
TXBUFRESET2		
TXBUFRESET3		
TXDEEMPH0	TXUSERCLKOUT0	
TXDEEMPH1	TXUSERCLKOUT1	
TXDEEMPH2	TXUSERCLKOUT2	
TXDEEMPH3	TXUSERCLKOUT3	
TXUSERCLKIN0		
TXUSERCLKIN1		
TXUSERCLKIN2		
TXUSERCLKIN3		

X11000

概要

Virtex®-6 FPGA GTH トランシーバのデザイン 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です。このELEMENTの詳細は、Virtex-6 FPGA GTH トランシーバ 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化してラッパ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をインスタンス化するには、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化しないでください。

詳細情報

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

: 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である Virtex®-6 FPGA RocketIO™ GTX トランシーバが表現されます。このELEMENTの詳細は、Virtex-6 FPGA RocketIO GTX トランシーバ 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化してラッパ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

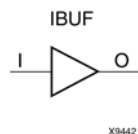
このELEMENTをインスタンス化するには、Virtex-6 FPGA RocketIO GTX Transceiver Wizard またはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンス化しないでください。

詳細情報

- ・ [Virtex-6 FPGA RocketIO GTX トランシーバ 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

: 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エートすることも可能です。インスタンスエートするには、入力ポート (I) を関連する最上位の入力ポートまたは入出力ポートに接続し、出力ポート (O) をそのポートをソースとする FPGA ロジックに接続します。必要なジェネリック マップ (VHDL) またはパラメータ値代入 (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を変更します。

ポートの説明

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

デザインの入力方法

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

このELEMENTは通常、デザインの最上位入力ポートに対して推論されます。通常はソースコードで指定する必要はありませんが、必要に応じてインスタンスエートできます。このコンポーネントをインスタンスエートするには、該当するライブラリ ガイドに含まれるインスタンスエーション コードを最上位エンティティ/モジュールに挿入します。デザイン階層を保つために、すべての I/O コンポーネントを必ずデザインの最上位に配置してください。I ポートをデザインの最上位入力ポートに、O ポートをこの入力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を適切に設定してください。

使用可能な属性

属性	タイプ	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

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

IBUF_inst : IBUF
generic map (
    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        -- 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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

IBUF_inst : IBUF
generic map (
    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        -- Buffer input (connect directly to top-level port)
);

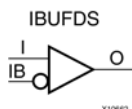
-- End of IBUF_inst instantiation
```

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

: 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。オプションで、プログラム可能な差動終端機能を使用すると、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減らすことができます。

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	タイプ	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に、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 ポートをこの入力に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を適切に設定してください。

使用可能な属性

属性	タイプ	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

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)
);

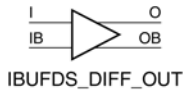
-- End of IBUFDS_inst instantiation
```

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

: 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」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。オプションで、プログラム可能な差動終端機能を使用すると、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減らすことができます。

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に、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/パラメータ値を設定し、バッファのビヘイビアを適切に設定してください。

使用可能な属性

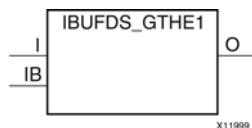
属性	タイプ	値	デフォルト	説明
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	内部差動終端抵抗を使用するかどうかを指定
IOSTANDARD	文字列	データシートを参照	DEFAULT	エレメントに I/O 規格を割り当て
IBUF_LOW_POWER	ブール代数	TRUE、FALSE	FALSE	消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定

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

: Differential Clock Input for the GTH Transceiver Reference Clocks



概要

GTH トランシーバ 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

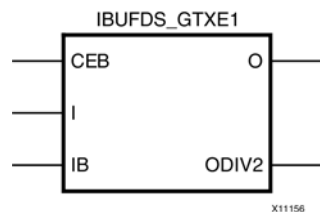
このエレメントをインスタンスエートするには、Virtex-6 FPGA GTH Transceiver Wizard またはこのエレメントを含む関連コアを使用します。このエレメントは直接インスタンスエートしないでください。

詳細情報

- ・ [Virtex-6 FPGA GTH トランシーバ 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

: 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 のマルチプレクサはそれぞれ別々に選択できますが、配線はシリコン上で共有されます。

デザインの入力方法

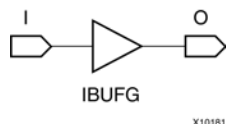
このエレメントをインスタンスエートするには、RocketIO™ Wizard またはこのエレメントを含む関連コアを使用します。このエレメントは直接インスタンスエートしないでください。

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

: 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が最小限に抑えられます。IBUFG の入力は、グローバル ケーパブル (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	タイプ	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

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

IBUFG_inst : IBUFG
generic map (
    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 -- Clock buffer input (connect directly to top-level port)
);

-- End of IBUFG_inst instantiation

```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)
);

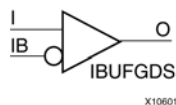
-- End of IBUFG_inst instantiation
```

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

: 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」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。オプションで、プログラム可能な差動終端機能を使用すると、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減らすことができます。デバイスへの入力データの取り込みには、プログラマブル遅延を使用することもできます。

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
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に、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となる入力ポートに、IB ポートを最上位のスレーブとなる入力ポートに、O ポートをこの入力をソースとする MMCM、BUFG、またはロジックに接続してください。一部の合成ツールでは、IBUFG を FPGA のクロック リソースに接続すると、必要に応じて BUFG が自動的に推論されます。generic/defparam 値を設定し、バッファのビヘイビアを適切に設定してください。

使用可能な属性

属性	タイプ	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

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)
);

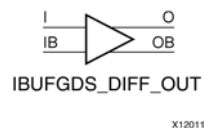
-- End of IBUFGDS_inst instantiation
```

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

: 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」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。マスタとスレーブは MYNET_P と MYNET_N のように、同じ論理信号の反対の状態を示します。IBUFGDS_DIFF_OUT では、差動信号の両方の位相に内部アクセスできる点が IBUFGDS と異なります。オプションで、プログラム可能な差動終端機能を使用すると、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減らすことができます。

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に、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/パラメータ値を設定し、バッファのビヘイビアを適切に設定してください。

使用可能な属性

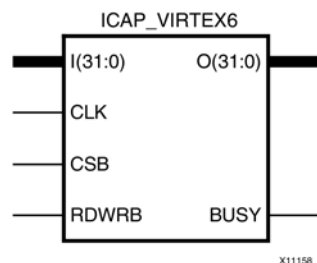
属性	タイプ	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	DEFAULT	エレメントに I/O 規格を割り当てます。
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	内部差動終端抵抗を使用するかどうかを指定
IBUF_LOW_PWR	ブール代数	TRUE、FALSE	FALSE	消費電力を削減することを優先するか、パフォーマンスを向上することを優先するかを指定

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

: 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	タイプ	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	タイプ	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	0 ビット文字列	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 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

-- End of ICAP_VIRTEX6_inst instantiation

```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

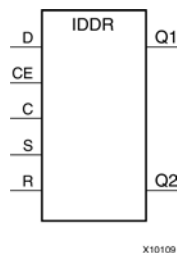
-- End of ICAP_VIRTEX6_inst instantiation
```

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

: Input Dual Data-Rate Register



概要

このデザイン エLEMENTは、ザイリンクス FPGA で外部デュアル データレート (DDR) 信号を受信するための専用入力レジスタです。このプリミティブでは、データが取り込まれるクロック エッジごとにデータを転送するだけでなく、同じクロック エッジで同時にデータを転送することもできます。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受信されますが、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のときに使用します。IDDR_2CLK では、必要なクロック リソース数が増え、IDDR コンポーネントを使用するときには不要な配置制限が発生する可能性があります。

ポートの説明

ポート名	方向	幅	機能
Q1 ~ Q2	出力	1	FPGA に接続する IDDR 出力です。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 を論理値 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 を論理値 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	タイプ	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 エッジに相対させた IDDR 操作モードを選択します。
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 では、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するように指定し、ASYNC では非同期のセット/リセット機能を指定します。

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

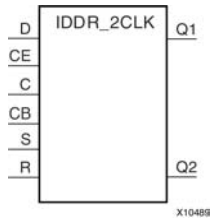
-- End of IDDR_inst instantiation
```

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

: 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。通常は、高速インターフェイスにのみこのプリミティブを使用します。このプリミティブでは、必要なクロック リソース数および消費電力が増加し、IDDR コンポーネントを使用するときには不要な配置配線での制限も発生するためです。代わりに IDDR コンポーネントを使用すると、動作速度は多少遅くなるものの、簡単に使用でき、リソース数も抑えられ、制限も低減します。このプリミティブでは、データが取り込まれるクロック エッジごとにデータを転送するだけでなく、同じクロック エッジで同時にデータを転送することもできます。これにより、タイミングが複雑にならず、追加のリソースも必要ありません。

- OPPOSITE_EDGE モード**：通常の DDR 方式でデータが受信されます。Q1 はクロック C の各立ち上がりエッジの後に変化し、Q2 はクロック CB の各立ち下がりエッジの後に変化します。
- SAME_EDGE モード**：データは、各クロックの立ち上がりエッジで受信されますが、CB クロック データレジスタの前にレジスタが 1 つ追加されます。この追加されたレジスタにもクロック信号 C の立ち上がりエッジでクロックが供給されるため、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受信されますが、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、または適切な入力または双方向バッファに接続します。

ポート名	方向	幅	機能
R	入力	1	アクティブ High のリセットで Q1 および Q2 を論理値 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 を論理値 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 ソースへ、CB ピンを立ち下がりクロック エッジを示すクロック ソースへ接続します。
- ・ D ピンは最上位の入力または双方向ポート、IODELAY、またはインスタンス化された入力または双方向バッファへ接続します。
- ・ Q1 および Q2 ピンは、適切なデータ ソースへ接続する必要があります。
- ・ CE ピンは、未使用のときに High にするか、または適切なクロック イネーブル ロジックに接続します。
- ・ R および S ピンは未使用のときに Low にするか、または適切なセット/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の階層にインスタンス化します。これにより、階層デザイン フローが促進されます。
- ・ 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) から来るようにする必要があります。ローカル反転はクロックにスキューを追加しますが、DCM / PLL はスキューを抑えます。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DDR_CLK_EDGE	文字列	OPPOSITE_EDGE、 SAME_EDGE、 SAME_EDGE_PIPELINED	OPPOSITE_EDGE	DDR のデータ送信モードを選択します。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、ASYNC	SYNC	セット/リセットのタイプを選択します。SYNC では、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するように指定し、ASYNC では非同期のセット/リセット機能を指定します。

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)

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

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

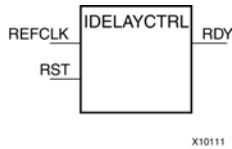
-- IDDR_2CLK: Dual-Clock, Input Double Data Rate Input Register with
--           Set, Reset and Clock Enable.
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 12.2

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
```

IDELAYCTRL

: IDELAY Tap Delay Value Control



概要

このエレメントは、IODELAYE1 を使用する場合にインスタンス化する必要があります。これは、IDELAY または ISERDES プリミティブがインスタンス化されており、IOBDelay_Type 属性が FIXED または VARIABLE に設定されている場合です。このモジュールは、一定の周波数リファレンス 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	タイプ	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assert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assert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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

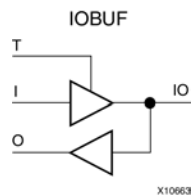
-- End of IDELAYCTRL_inst instantiation
```

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

: 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	タイプ	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	I/O 規格として LVTTTL、LVC MOS12、LVC MOS15、LVC MOS18、LVC MOS25 または LVC MOS33 を使用する SelectIO™ バッファの出力の駆動電流 (mA) を選択
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
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。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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

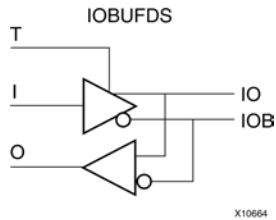
-- End of IOBUF_inst instantiation
```

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

： 3-State Differential Signaling I/O Buffer with Active Low Output Enable



概要

このデザイン エLEMENTは、低電圧差動信号を使用する双方向バッファです。IOBUFDS では、デザイン レベルのインターフェイス信号は、一方が「マスタ」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。オプションで、プログラム可能な差動終端機能を使用すると、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減らすことができます。デバイスへの入力データの取り込みには、プログラマブル遅延を使用することもできます。

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	タイプ	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

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

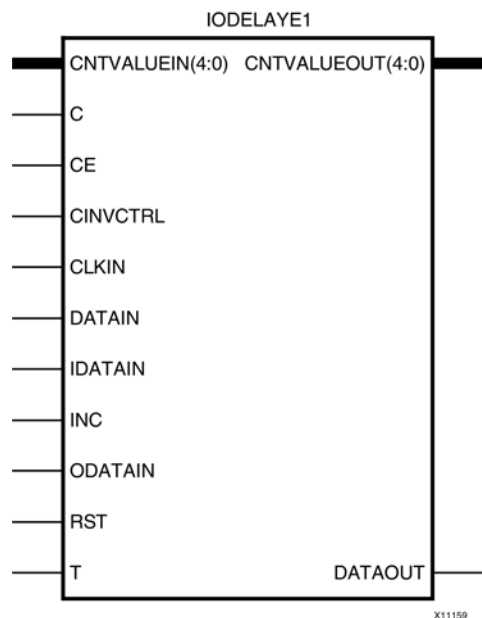
-- End of IOBUFDS_inst instantiation
```

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

: Input and Output Fixed or Variable Delay Element



概要

このデザイン エLEMENTは、Virtex®-6 FPGA の入力パスに固定遅延または可変遅延を、出力パスに固定遅延を追加します。この遅延は、チップへ入力されるデータ、またはチップから出力されるデータをアライメントするのに有効で、プロセス、温度、電圧の影響を受けずにデータ アライメントを監視できます。VARIABLE モードでは、入力パスを調整して遅延の量を増分または減分できます。出力遅延パスは、固定遅延でのみ使用できます。IODELAY は、FPGA の内部パスに一定の遅延または可変遅延を追加する場合にも使用できます。ただし、このように使用する場合、入力パス遅延または出力パス遅延に関連する I/O に使用できなくなります。

ポートの説明

ポート名	タイプ	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 カウンタ値
CNTVALUEOUT[4:0]	出力	5	タップ値モニタ用に FPGA ロジックに送信されるタップ カウンタ値
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。
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、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

ポート名	タイプ	幅	機能
			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 で設定された値で駆動します。
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。VARLOADABLE 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	タイプ	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	I、CLKIN、DATAIN、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"> I は入力ポートまたは IBUF (入力モード) に直接接続します。 O は出力ポートまたは OBUF (出力モード) に接続します。 IO はポートに接続します。 DATAIN はどのポート (内部モード) にも接続しません。
HIGH_PERFORMANCE_MODE	ブール代数	TRUE、FALSE	TRUE	TRUE の場合は、出力ジッタを減少させます。FALSE の場合、消費電力量が減少します。消費電力量の際は、XPower Estimator (XPE) ツールで確認できます。

属性	タイプ	値	デフォルト	説明
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	出力遅延タイプに DEFAULT (ホールド タイムの削除)、FIXED、VARIABLE のいずれかを指定
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	175.0 ~ 225.0	200.0	タップ遅延値を正しく算出するため、関連付けられている IDELAYCTRL コンポーネントで使用される周波数 (MHz) を設定します。
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 ジッタ量が使用されるようにします。

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

IDELAYE1_inst : IDELAYE1
generic map (
    CINCTRL_SEL => "FALSE",      -- Dynamically inverts the Clock (C) polarity.
    DELAY_SRC   => "I",          -- Specifies the source to the IDELAY component. "I" means it will be
                                -- connected directly to an input port or IBUF (input mode), "O" means
                                -- it will be connected to an output port or OBUF (output mode), "IO"
                                -- means it will be connected to a port, and "DATAIN" means it will not
                                -- be connected to any port (internal mode).
    HIGH_PERFORMANCE_MODE => "TRUE", -- When TRUE, this attribute reduces the output jitter.
    IDELAY_TYPE  => "DEFAULT",      -- Specifies a fixed, variable or default (eliminate hold time) input
                                -- delay.
    IDELAY_VALUE => 0,             -- Specifies the number of taps of delay for the input path when in
                                -- fixed mode or the initial delay tap value for variable mode.
    ODELAY_TYPE  => "FIXED",        -- Specifies the number of taps of delay for the output path.
    ODELAY_VALUE => 0,             -- When using an associated IDELAYCTRL, specifies the input reference
    REFCLK_FREQUENCY => 200.0,      -- frequency to the component.
    SIGNAL_PATTERN => "DATA"        -- Used by the delay calculator to determine different propagation
                                -- delays through the IDELAY block based on the setting. DATA will be
                                -- the addition of per tap delay and per tap jitter. No jitter is

```

```

-- introduced for clock-like signals.
)
port map (
  CNTVALUEOUT => CNTVALUEOUT, -- 5-bit Counter value going to fabric for monitoring purpose
  DATAOUT => DATAOUT,        -- 1-bit Delayed data output from input port (connect to input datapath logic)
  C => C,                      -- 1-bit Clock input (Must be connected for variable mode)
  CE => CE,                    -- 1-bit Active high enable increment/decrement function
  CINCTRL => CINCTRL,          -- 1-bit Dynamically inverts the Clock (C) polarity
  CLKIN => CLKIN,              -- 1-bit Clock Access into the IODELAY (from the IO CLKMUX)
  CNTVALUEIN => CNTVALUEIN,    -- 5-bit Counter value from fabric for loadable counter application
  DATAIN => DATAIN,          -- 1-bit Data input for the internal datapath delay. When DATAIN is used,
                                -- IDATAIN and ODATAIN must be tied to a logic zero (ground).

  IDATAIN => IDATAIN,          -- 1-bit Data input to device from the I/O (connect directly to port, I/O
                                -- Buffer). When IDATAIN is used, DATAIN must be tied to a logic zero
                                -- (ground).

  INC => INC,                  -- 1-bit Increment / Decrement tap delay
  ODATAIN => ODATAIN,          -- 1-bit Data input for the output datapath from the device (connect to
                                -- output data source). When ODATAIN is used, DATAIN must be tied to a logic
                                -- zero (ground).

  RST => RST,                  -- 1-bit 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 3-state input control. Tie high for input-only or internal delay or
                                -- tie low for output only.
);

-- End of IODELAYE1_inst instantiation

```


Verilog 記述 (インスタンス化)

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

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

-- IODELAYE1: Input and Output Fixed or Variable Delay Element
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 12.2

IODELAYE1_inst : IODELAYE1
generic map (
    CINVCTRL_SEL => "FALSE",      -- Dynamically inverts the Clock (C) polarity.
    DELAY_SRC => "I",             -- Specifies the source to the IODELAY component. "I" means it will be
                                -- connected directly to an input port or IBUF (input mode), "O" means
                                -- it will be connected to an output port or OBUF (output mode), "IO"
                                -- means it will be connected to a port, and "DATAIN" means it will not
                                -- be connected to any port (internal mode).
    HIGH_PERFORMANCE_MODE => "TRUE", -- When TRUE, this attribute reduces the output jitter.
    IDELAY_TYPE => "DEFAULT",      -- Specifies a fixed, variable or default (eliminate hold time) input
                                -- delay.
    IDELAY_VALUE => 0,             -- Specifies the number of taps of delay for the input path when in
                                -- fixed mode or the initial delay tap value for variable mode.
    ODELAY_TYPE => "FIXED",        -- Specifies the number of taps of delay for the output path.
    ODELAY_VALUE => 0,             -- When using an associated IDELAYCTRL, specifies the input reference
    REFCLK_FREQUENCY => 200.0,    -- frequency to the component.
    SIGNAL_PATTERN => "DATA"      -- Used by the delay calculator to determine different propagation
                                -- delays through the IODELAY block based on the setting. DATA will be
                                -- the addition of per tap delay and per tap jitter. No jitter is
                                -- introduced for clock-like signals.
)
port map (
    CNTVALUEOUT => CNTVALUEOUT, -- 5-bit Counter value going to fabric for monitoring purpose
    DATAOUT => DATAOUT,       -- 1-bit Delayed data output from input port (connect to input datapath logic)
    C => C,                     -- 1-bit Clock input (Must be connected for variable mode)
    CE => CE,                   -- 1-bit Active high enable increment/decrement function
    CINVCTRL => CINVCTRL,       -- 1-bit Dynamically inverts the Clock (C) polarity
    CLKIN => CLKIN,             -- 1-bit Clock Access into the IODELAY (from the IO CLKMUX)
    CNTVALUEIN => CNTVALUEIN,   -- 5-bit Counter value from fabric for loadable counter application
    DATAIN => DATAIN,         -- 1-bit Data input for the internal datapath delay. When DATAIN is used,
                                -- IDATAIN and ODATAIN must be tied to a logic zero (ground).

    IDATAIN => IDATAIN,         -- 1-bit Data input to device from the I/O (connect directly to port, I/O
                                -- Buffer). When IDATAIN is used, DATAIN must be tied to a logic zero
                                -- (ground).

    INC => INC,                 -- 1-bit Increment / Decrement tap delay
    ODATAIN => ODATAIN,         -- 1-bit Data input for the output datapath from the device (connect to
                                -- output data source). When ODATAIN is used, DATAIN must be tied to a logic
                                -- zero (ground).

    RST => RST,                 -- 1-bit 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 3-state input control. Tie high for input-only or internal delay or
                                -- tie low for output only.
);

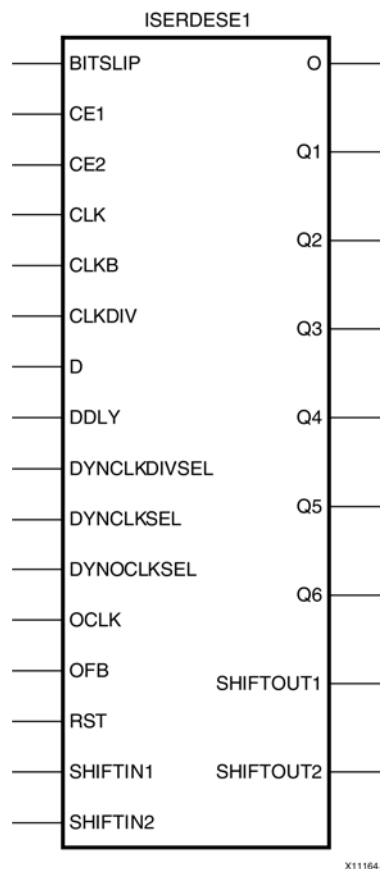
-- End of IODELAYE1_inst instantiation
```

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

: 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 コンバータです。FPGA でデシリアライザを設計する際の複雑なタイミング問題を避けるために使用します。

ポートの説明

ポート名	タイプ	幅	機能
BITSLIP	入力	1	BITSLIP ピンがアサートされると (アクティブ High)、CLKDIV に同期してビットスリップ処理が実行されます。ビットスリップ処理が実行されるごとに、バレル シフタと同様、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	セカンダリ クロック入力 ・ 1 クロック DDR モード (DATA_RATE="DDR") を使用する場合は、CLK ピンに接続されているクロックを反転し、CLKB ピンに接続します。

ポート名	タイプ	幅	機能
			<ul style="list-style-type: none"> 2 クロック DDR モードを使用する場合は、独立した位相シフトクロックを CLKB ピンに接続します。 シングル データレート モード (DATA_RATE="SDR") を使用する場合は、このピンを未接続にするかグランドの接続します。
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。
Q1 ~ Q6	出力	1	ISERDES1 モジュールのレジスタ付き出力。1 つの ISERDES1 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、マスタの SHIFTOUT1 と SHIFTOUT2 出力に接続します。このピンはグラ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	タイプ	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

属性	タイプ	値	デフォルト	説明
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。 <ul style="list-style-type: none"> DATA_RATE = DDR の場合の有効値は 4、6、8、または 10 です。 DATA_RATE = SDR の場合の有効値は 2、3、4、5、6、7、8 です。
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
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
IOBDelay	文字列	NONE、BOTH、IBUF、IFD	NONE	ISERDES モジュールの入力ソースを指定
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	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。 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 OSERDESE1 をマスタ 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。 メモ : 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。
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 モードかスレーブ モードに設定するかどうかを指定

属性	タイプ	値	デフォルト	説明
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

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 Combinatorial output
    -- Q1 - Q6: 1-bit (each) Data outputs
    Q1 => Q1,
    Q2 => Q2,
    Q3 => Q3,
    Q4 => Q4,
    Q5 => Q5,
    Q6 => Q6,
    -- SHIFTOUT1/SHIFTOUT2: 1-bit (each) Cascade outputs for data width expansion
    SHIFTOUT1 => SHIFTOUT1,
    SHIFTOUT2 => SHIFTOUT2,
    BITSLIP => BITSLIP,           -- 1-bit The BITSLIP pin performs a Bitshift operation synchronous to CLKDIV
    -- when asserted (active High). Subsequently, the data seen on the Q1 to Q6
    -- output ports will shift, as in a barrel-shifter operation, one position
    -- every time Bitshift is invoked (DDR operation is different from SDR).

    -- CE1, CE2: 1-bit (each) Data register clock enable inputs
    CE1 => CE1,
    CE2 => CE2,
    CLK => CLK,                   -- 1-bit Clock input
    CLKB => CLKB,                 -- 1-bit Secondary clock input
    CLKDIV => CLKDIV,            -- 1-bit Divided clock input
    D => D,                      -- 1-bit Data input
    DDLY => DDLY,                -- 1-bit Serial input data from IODELAY
    DYNCLKDIVSEL => DYNCLKDIVSEL, -- 1-bit Dynamic CLKDIV inversion input
    DYNCLKSEL => DYNCLKSEL,      -- 1-bit Dynamic CLK/CLKB inversion input
    OCLK => OCLK,                -- 1-bit High speed output clock input
    OFB => OFB,                  -- 1-bit Feedback path input
    RST => RST,                  -- 1-bit Active high asynchronous reset input
    -- SHIFTIN1/SHIFTIN2: 1-bit (each) Cascade inputs for data width expansion
    SHIFTIN1 => SHIFTIN1,
    SHIFTIN2 => SHIFTIN2
);

-- End of ISERDESE1_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 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 => '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 (
    0 => 0,                        -- 1-bit Combinatorial output
    -- Q1 - Q6: 1-bit (each) Data outputs
    Q1 => Q1,
    Q2 => Q2,
    Q3 => Q3,
    Q4 => Q4,
    Q5 => Q5,
    Q6 => Q6,
    -- SHIFTOUT1/SHIFTOUT2: 1-bit (each) Cascade outputs for data width expansion
    SHIFTOUT1 => SHIFTOUT1,
    SHIFTOUT2 => SHIFTOUT2,
    BITSLIP => BITSLIP,           -- 1-bit The BITSLIP pin performs a Bitshift operation synchronous to CLKDIV
    -- when asserted (active High). Subsequently, the data seen on the Q1 to Q6
    -- output ports will shift, as in a barrel-shifter operation, one position
    -- every time Bitshift is invoked (DDR operation is different from SDR).

    -- CE1, CE2: 1-bit (each) Data register clock enable inputs
    CE1 => CE1,
    CE2 => CE2,
    CLK => CLK,                   -- 1-bit Clock input
    CLKB => CLKB,                 -- 1-bit Secondary clock input
    CLKDIV => CLKDIV,             -- 1-bit Divided clock input
    D => D,                       -- 1-bit Data input
    DDLY => DDLY,                 -- 1-bit Serial input data from IODELAY
    DYNCLKDIVSEL => DYNCLKDIVSEL, -- 1-bit Dynamic CLKDIV inversion input
    DYNCLKSEL => DYNCLKSEL,       -- 1-bit Dynamic CLK/CLKB inversion input
    OCLK => OCLK,                 -- 1-bit High speed output clock input
    OFB => OFB,                   -- 1-bit Feedback path input
    RST => RST,                   -- 1-bit Active high asynchronous reset input
    -- SHIFTIN1/SHIFTIN2: 1-bit (each) Cascade inputs for data width expansion
    SHIFTIN1 => SHIFTIN1,
    SHIFTIN2 => SHIFTIN2
);

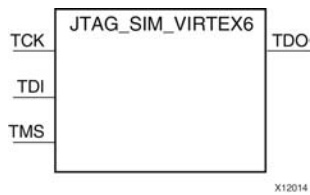
-- End of ISERDESE1_inst instantiation
```

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

: 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 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	タイプ	幅	機能
TDO	出力	1	テスト データ出力： 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 コントローラのステートおよび現在の命令により、特定の操作に TDO に命令またはデータを送信するレジスタ (命令またはデータ) が決定します。TDO のステートは TCK の立ち下がりエッジで変化し、デバイス内を命令またはデータがシフトされている間のみアクティブになります。TDO はアクティブ ドライバ出力です。
TCK	入力	1	テスト クロック： JTAG のテスト クロックです。TAP コントローラおよび JTAG レジスタは TCK に同期して動作します。
TDI	入力	1	テスト データ： 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 コントローラのステートおよび現在の命令により、特定の操作に TDI から命令またはデータを入力するレジスタ (命令またはデータ) が決定します。TDI には内部プルアップ抵抗が含まれており、駆動されない場合はシステムにロジック High を供給します。TDI からの JTAG レジスタへの命令またはデータ供給は、TCK の立ち上がりエッジに同期します。
TMS	入力	1	テスト モード セレクト： TCK の立ち上がりエッジで TAP コントローラ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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.2

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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
```

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

: 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 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 つの構文が存在しない場合はコピーし、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 12.2

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

-- End of KEEPER_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

KEEPER_inst : KEEPER
port map (
  0 => 0      -- Keeper output (connect directly to top-level port)
);

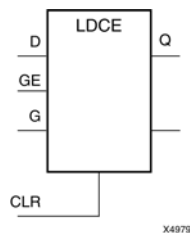
-- End of KEEPER_inst instantiation
```

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

: Transparent Data Latch with Asynchronous Clear and Gate Enable



概要

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

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

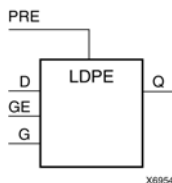
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	0	コンフィギュレーション後の Q 出力の初期値を指定

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

: 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

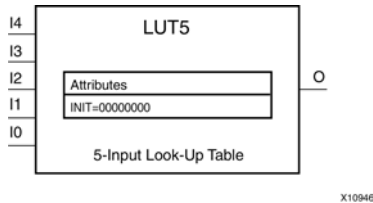
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	1	電源投入時または Q ポートに対する GSR のアサート時の初期値を指定

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

: 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 ファンクションをインプリメントするときに使用します。LUT5 が 1 つの場合はスライス内に含まれる LUT6 に、2 つの場合は 1 つの LUT6 に多少の制限はありますがパックできます。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 に接続することができます。LUT5_L では LUT5 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT5_D では LUT5 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。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 信号の接続を暗黙的に指定する必要がない限り、いつでも使用できます。

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。INIT 値は、関連する入力適用されるときに、対応する INIT ビット値に 1 を割り当てることで計算されます。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に、新しい INIT の値を決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を定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力					出力
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]
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	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	ルックアップ テーブルの論理値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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）

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

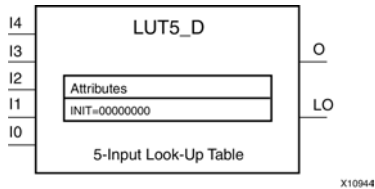
-- End of LUT5_inst instantiation
```

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

: 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 ファンクションをインプリメントするときに使用します。LUT5 が 1 つの場合はスライス内に含まれる LUT6 に、2 つの場合は 1 つの LUT6 に多少の制限はありますがパックできます。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 に接続することができます。LUT5_L では LUT5 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT5_D では LUT5 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。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ない限り、いつでも使用できます。

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。INIT 値は、関連する入力適用されるときに、対応する INIT ビット値に 1 を割り当てることで計算されます。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に、新しい INIT の値を決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を定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力					出力	
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]
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
L0	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	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	ルックアップ テーブルの論理値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

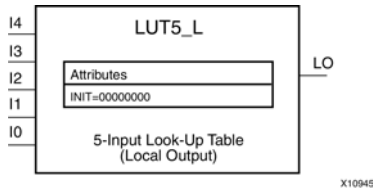
-- End of LUT5_D_inst instantiation
```

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

: 5-Input Lookup Table with Local Output



概要

このデザイン エLEMENTは、入力 5 個、出力 1 個のルックアップ 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 ファンクションをインプリメントするときに使用します。LUT5 が 1 つの場合はスライス内に含まれる LUT6 に、2 つの場合は 1 つの LUT6 に多少の制限はありますがパックできます。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 に接続することができます。LUT5_L では LUT5 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT5_D では LUT5 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。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ない限り、いつでも使用できます。

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。INIT 値は、関連する入力適用されるときに、対応する INIT ビット値に 1 を割り当てることで計算されます。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に、新しい INIT の値を決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を定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力					出力
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]
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	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	ルックアップ テーブルの論理値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

LUT5_L_inst : LUT5_L
generic map (
  INIT => X"00000000" -- Specify LUT Contents
port map (
  L0 => L0, -- 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）

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

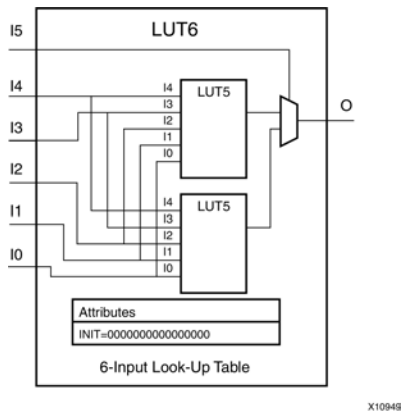
-- End of LUT5_L_inst instantiation
```

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

: 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 ファンクションをインプリメントするときに使用します。LUT6 はルックアップ テーブル (LUT) 4 個のうちの 1 つにマップされます。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 に接続することができます。LUT6_L では LUT6 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT6_D では LUT6 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がない限り、いつでも使用できます。

LUT のロジック ファンクションを指定するために 64 ビットの 16 進数で INIT 属性を設定する必要があります。INIT 値は、関連する入力に適用されるときに、対応する INIT ビット値に 1 を割り当てることで算出されます。たとえば Verilog で INIT 値が `64'h8000000000000000` (VHDL では `X"8000000000000000"`) の場合は、入力すべてが 1 ではない限り、出力が 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値が `64'hffffffffffffff` (VHDL では `X"FFFFFFFFFFFFFFFF"`) の場合は、入力がすべてゼロではない限り、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に、新しい INIT の値を決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を定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力						出力
I5	I4	I3	I2	I1	I0	O
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]

入力						出力
I5	I4	I3	I2	I1	I0	O
0	0	0	0	1	0	INIT[2]
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]

入力						出力
I5	I4	I3	I2	I1	I0	O
1	0	0	1	0	1	INIT[37]
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	タイプ	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

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

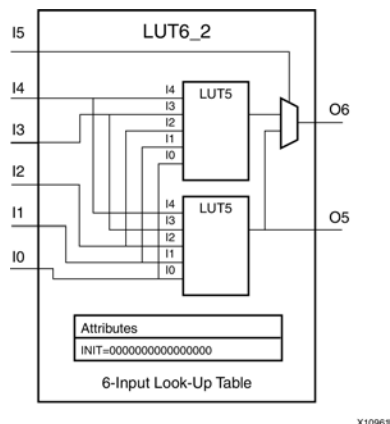
-- End of LUT6_inst instantiation
```

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

: 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、共有入力の付き 5 入力のロジック ファンクションのいずれか 2 つをインプリメントできます。または、共有入力とロジック値の付いた 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することもできます。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 ファンクションをインプリメントするときに使用します。LUT6_2 は、Virtex-5 スライスに含まれるルックアップ テーブル (LUT) 4 個のうちの 1 つにマップされます。

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。INIT 値は、関連する入力 that 適用されるときに、対応する INIT ビット値に 1 を割り当てることで算出されます。たとえば、Verilog で INIT 値を 64'hffffffff (VHDL では X"FFFFFFFFFFFFFFFF") に設定すると、入力がすべてゼロの場合以外は O6 出力は 1 になり、I[4:0] がすべてゼロの場合以外は O5 出力は 1 になります (5 または 6 入力の OR ゲート)。INIT 値の下位半分 (ビット 31:0) は O5 出力のロジック ファンクションに適用されます。

FPGA LUT プリミティブでは、INIT パラメータ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に、新しい INIT の値を決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を定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力						出力	
I5	I4	I3	I2	I1	I0	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]

入力						出力	
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[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]

入力						出力	
1	0	1	0	0	0	INIT[8]	INIT[40]
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	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

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

LUT6_2_inst : LUT6_2
generic map (
  INIT => X"0000000000000000") -- Specify LUT Contents
port map (
  O6 => O6, -- 6/5-LUT output (1-bit)
  O5 => O5, -- 5-LUT output (1-bit)
  I0 => I0, -- LUT input (1-bit)
  I1 => I1, -- LUT input (1-bit)
  I2 => I2, -- LUT input (1-bit)
  I3 => I3, -- LUT input (1-bit)
  I4 => I4, -- LUT input (1-bit)
  I5 => I5  -- LUT input (1-bit)
);

-- End of LUT6_2_inst instantiation
```

Verilog 記述（インスタンス化）

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)
);

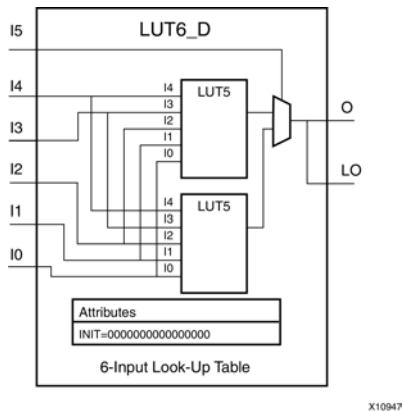
-- End of LUT6_2_inst instantiation
```

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

: 6-Input Lookup Table with General and Local Outputs



概要

このデザイン エLEMENTは、入力 6 個、出力 1 個のルックアップ 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 ファンクションをインプリメントするときに使用します。LUT6 はルックアップ テーブル (LUT) 4 個のうちの 1 つにマップされます。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 に接続することができます。LUT6_L では LUT6 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT6_D では LUT6 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がない限り、いつでも使用できます。

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。INIT 値は、関連する入力に適用されるときに、対応する INIT ビット値に 1 を割り当てることで算出されます。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、入力がすべて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に、新しい INIT の値を決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を定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
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]
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]

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
1	0	0	1	0	1	INIT[37]	INIT[37]
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	タイプ	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

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

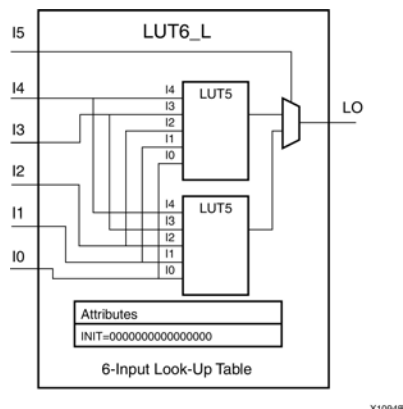
-- End of LUT6_D_inst instantiation
```

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

: 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 ファンクションをインプリメントするときに使用します。LUT6 はルックアップ テーブル (LUT) 4 個のうちの 1 つにマップされます。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 に接続することができます。LUT6_L では LUT6 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT6_D では LUT6 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がない限り、いつでも使用できます。

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。INIT 値は、関連する入力に適用されるときに、対応する INIT ビット値に 1 を割り当てることで計算されます。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'hfffffffffe` (VHDL では `X"FFFFFFFFFFFFFFFFFE"`) に設定すると、入力がすべてゼロの場合以外は出力は 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に、新しい INIT の値を決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を定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力						出力
I5	I4	I3	I2	I1	I0	LO
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]

入力						出力
I5	I4	I3	I2	I1	I0	LO
0	0	0	0	1	0	INIT[2]
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]

入力						出力
I5	I4	I3	I2	I1	I0	LO
1	0	0	1	0	1	INIT[37]
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	タイプ	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

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

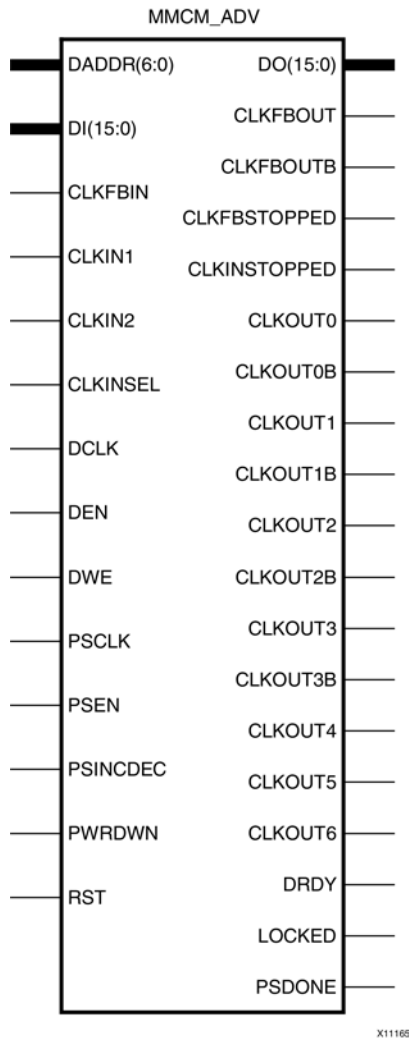
-- End of LUT6_L_inst instantiation
```

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

: 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	タイプ	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 ピン

ポート名	タイプ	幅	機能
CLKINSEL	入力	1	入力マルチプレクサのステートを制御する信号で、High = CLKIN1、Low = CLKIN2 です。
CLKINSTOPPED	出力	1	入力クロックが停止したことを示すステータス ピン
CLKIN1	入力	1	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	ユーザー コンフィギュラブル 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です。このバスを使用しない場合は、すべてのビットを 0 にする必要があります。
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 クロックです。
DEN	入力	1	ダイナミック リコンフィギュレーション機能にアクセスするためのイネーブル制御信号です。ダイナミック リコンフィギュレーションを使用しない場合は、DEN を Low に固定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です。このバスを使用しない場合は、すべてのビットを 0 にする必要があります。
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 データ出力バスです。
DRDY	出力	1	MMCM のダイナミック リコンフィギュレーション機能の DEN 信号への応答を供給する READY 出力です。
DWE	入力	1	DADDR アドレスへの DI データの書き込みを制御するライト イネーブル信号です。使用しない場合は、Low に固定する必要があります。
LOCKED	出力	1	MMCM で定義されている時間枠内で位相アライメントが完了し、PPM 範囲内で周波数が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	パワー ダウンがインスタンスシートされますが、MMCM は未使用です。
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。入力クロックのコンディション (周波数など) が変わる場合、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	タイプ	値	デフォルト	説明
BANDWIDTH	文字列	OPTIMIZED、HIGH、LOW	OPTIMIZED	ジッタ、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	1.0	別の周波数を使用する場合に、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、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。
CLKIN_PERIOD	フロート (ns)	1,000 ～ 1000.000	0.000	MMCM CLKIN1 入力の入力周期を指定します。精度は ps までです。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	別の周波数を使用する場合に、CLKOUT クロック出力を分周する値を指定します。この値、CLKFBOUT_MULT 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	別の周波数を使用する場合に、CLKOUT クロック出力を分周する値を指定します。この値、CLKFBOUT_MULT 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 の場合、VCO 周波数を CLKIN を損失する前の近似値に保持します。

属性	タイプ	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"> ・ 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。 ・ 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。 ・ EXTERNAL : FPGA の外部ネットワークが調整されます。 ・ CASCADE : 2 つの MMCM がカスケード接続されます。 ・ 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。
DIVCLK_DIVIDE	整数	1 ~ 128	1	入力クロックに対するすべての出力クロックの分周比を指定し、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値を指定します。バンド幅が OPTIMIZED に設定されていると、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値を指定します。バンド幅が OPTIMIZED に設定されていると、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。
USE_FINE_PS	ブール代数	FALSE、TRUE	FALSE	カウンタ可変ファイン位相シフト イネーブル

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 33 MHz).
    CLKIN1_PERIOD => 0.0,
    CLKIN2_PERIOD => 0.0,
    -- CLKOUT0_DIVIDE - CLKOUT6_DIVIDE: Divide amount for CLKOUT (1-128)
    CLKOUT1_DIVIDE => 1,
    CLKOUT2_DIVIDE => 1,
    CLKOUT3_DIVIDE => 1,
    CLKOUT4_DIVIDE => 1,
    CLKOUT5_DIVIDE => 1,
    CLKOUT6_DIVIDE => 1,
    CLKOUT0_DIVIDE_F => 1.0,      -- Divide amount for CLKOUT0 (1.000-128.000).
    -- CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for CLKOUT outputs (0.01-0.99).
    CLKOUT0_DUTY_CYCLE => 0.5,
    CLKOUT1_DUTY_CYCLE => 0.5,
    CLKOUT2_DUTY_CYCLE => 0.5,
    CLKOUT3_DUTY_CYCLE => 0.5,
    CLKOUT4_DUTY_CYCLE => 0.5,
    CLKOUT5_DUTY_CYCLE => 0.5,
    CLKOUT6_DUTY_CYCLE => 0.5,
    -- CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for CLKOUT outputs (-360.000-360.000).
    CLKOUT0_PHASE => 0.0,
    CLKOUT1_PHASE => 0.0,
    CLKOUT2_PHASE => 0.0,
    CLKOUT3_PHASE => 0.0,
    CLKOUT4_PHASE => 0.0,
    CLKOUT5_PHASE => 0.0,
    CLKOUT6_PHASE => 0.0,
    CLKOUT4_CASCADE => FALSE,    -- Cascade CLKOUT4 counter with CLKOUT6 (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) User configurable clock outputs
    CLKOUT0 => CLKOUT0,        -- 1-bit CLKOUT0 output
    CLKOUT0B => CLKOUT0B,      -- 1-bit Inverted CLKOUT0 output
    CLKOUT1 => CLKOUT1,        -- 1-bit CLKOUT1 output
    CLKOUT1B => CLKOUT1B,      -- 1-bit Inverted CLKOUT1 output
    CLKOUT2 => CLKOUT2,        -- 1-bit CLKOUT2 output
    CLKOUT2B => CLKOUT2B,      -- 1-bit Inverted CLKOUT2 output
    CLKOUT3 => CLKOUT3,        -- 1-bit CLKOUT3 output
    CLKOUT3B => CLKOUT3B,      -- 1-bit Inverted CLKOUT3 output
    CLKOUT4 => CLKOUT4,        -- 1-bit CLKOUT4 output
    CLKOUT5 => CLKOUT5,        -- 1-bit CLKOUT5 output
    CLKOUT6 => CLKOUT6,        -- 1-bit CLKOUT6 output
    -- DRP Ports: 16-bit (each) Dynamic reconfiguration ports
    DO => DO,                  -- 16-bit DRP data output
    DRDY => DRDY,              -- 1-bit DRP ready output
    -- Dynamic Phase Shift Ports: 1-bit (each) Ports used for dynamic phase shifting of the outputs
    PSDONE => PSDONE,          -- 1-bit Phase shift done output
    -- Feedback Clocks: 1-bit (each) Clock feedback ports
    CLKFBOUT => CLKFBOUT,      -- 1-bit Feedback clock output
    CLKFBOUTB => CLKFBOUTB,    -- 1-bit Inverted CLKFBOUT
    -- Status Ports: 1-bit (each) MMCM status ports
    CLKFBSTOPPED => CLKFBSTOPPED, -- 1-bit Feedback clock stopped output
    CLKINSTOPPED => CLKINSTOPPED, -- 1-bit Input clock stopped output

```

```

LOCKED => LOCKED,           -- 1-bit LOCK output
-- Clock Inputs: 1-bit (each) Clock inputs
CLKIN1 => CLKIN1,           -- 1-bit Primary clock input
CLKIN2 => CLKIN2,           -- 1-bit Secondary clock input
-- Control Ports: 1-bit (each) MMCM control ports
CLKINSEL => CLKINSEL,       -- 1-bit Clock select input
PWRDWN => PWRDWN,           -- 1-bit Power-down input
RST => RST,                  -- 1-bit Reset input
-- DRP Ports: 7-bit (each) Dynamic reconfiguration ports
DADDR => DADDR,             -- 7-bit DRP address input
DCLK => DCLK,               -- 1-bit DRP clock input
DEN => DEN,                 -- 1-bit DRP enable input
DI => DI,                   -- 16-bit DRP data input
DWE => DWE,                 -- 1-bit DRP write enable input
-- Dynamic Phase Shift Ports: 1-bit (each) Ports used for dynamic phase shifting of the outputs
PSCLK => PSCLK,             -- 1-bit Phase shift clock input
PSEN => PSEN,               -- 1-bit Phase shift enable input
PSINCDEC => PSINCDEC,       -- 1-bit Phase shift increment/decrement input
-- Feedback Clocks: 1-bit (each) Clock feedback ports
CLKFBIN => CLKFBIN         -- 1-bit Feedback clock input
);

-- End of MMCM_ADV_inst instantiation

```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 33 MHz).
    CLKIN1_PERIOD => 0.0,
    CLKIN2_PERIOD => 0.0,
    -- CLKOUT0_DIVIDE - CLKOUT6_DIVIDE: Divide amount for CLKOUT (1-128)
    CLKOUT1_DIVIDE => 1,
    CLKOUT2_DIVIDE => 1,
    CLKOUT3_DIVIDE => 1,
    CLKOUT4_DIVIDE => 1,
    CLKOUT5_DIVIDE => 1,
    CLKOUT6_DIVIDE => 1,
    CLKOUT0_DIVIDE_F => 1.0,        -- Divide amount for CLKOUT0 (1.000-128.000).
    -- CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for CLKOUT outputs (0.01-0.99).
    CLKOUT0_DUTY_CYCLE => 0.5,
    CLKOUT1_DUTY_CYCLE => 0.5,
    CLKOUT2_DUTY_CYCLE => 0.5,
    CLKOUT3_DUTY_CYCLE => 0.5,
    CLKOUT4_DUTY_CYCLE => 0.5,
    CLKOUT5_DUTY_CYCLE => 0.5,
    CLKOUT6_DUTY_CYCLE => 0.5,
    -- CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for CLKOUT outputs (-360.000-360.000).
    CLKOUT0_PHASE => 0.0,
    CLKOUT1_PHASE => 0.0,
    CLKOUT2_PHASE => 0.0,
    CLKOUT3_PHASE => 0.0,
    CLKOUT4_PHASE => 0.0,
    CLKOUT5_PHASE => 0.0,
    CLKOUT6_PHASE => 0.0,
    CLKOUT4_CASCADE => FALSE,       -- Cascade CLKOUT4 counter with CLKOUT6 (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) User configurable clock outputs
  CLKOUT0 => CLKOUT0,           -- 1-bit CLKOUT0 output
  CLKOUT0B => CLKOUT0B,         -- 1-bit Inverted CLKOUT0 output
  CLKOUT1 => CLKOUT1,           -- 1-bit CLKOUT1 output
  CLKOUT1B => CLKOUT1B,         -- 1-bit Inverted CLKOUT1 output
  CLKOUT2 => CLKOUT2,           -- 1-bit CLKOUT2 output
  CLKOUT2B => CLKOUT2B,         -- 1-bit Inverted CLKOUT2 output
  CLKOUT3 => CLKOUT3,           -- 1-bit CLKOUT3 output
  CLKOUT3B => CLKOUT3B,         -- 1-bit Inverted CLKOUT3 output
  CLKOUT4 => CLKOUT4,           -- 1-bit CLKOUT4 output
  CLKOUT5 => CLKOUT5,           -- 1-bit CLKOUT5 output
  CLKOUT6 => CLKOUT6,           -- 1-bit CLKOUT6 output
  -- DRP Ports: 16-bit (each) Dynamic reconfiguration ports
  DO => DO,                     -- 16-bit DRP data output
  DRDY => DRDY,                 -- 1-bit DRP ready output
  -- Dynamic Phase Shift Ports: 1-bit (each) Ports used for dynamic phase shifting of the outputs
  PSDONE => PSDONE,             -- 1-bit Phase shift done output
  -- Feedback Clocks: 1-bit (each) Clock feedback ports
  CLKFBOUT => CLKFBOUT,         -- 1-bit Feedback clock output
  CLKFBOUTB => CLKFBOUTB,       -- 1-bit Inverted CLKFBOUT
  -- Status Ports: 1-bit (each) MMCM status ports
  CLKFBSTOPPED => CLKFBSTOPPED, -- 1-bit Feedback clock stopped output
  CLKINSTOPPED => CLKINSTOPPED, -- 1-bit Input clock stopped output
  LOCKED => LOCKED,             -- 1-bit LOCK output
  -- Clock Inputs: 1-bit (each) Clock inputs
  CLKIN1 => CLKIN1,             -- 1-bit Primary clock input
  CLKIN2 => CLKIN2,             -- 1-bit Secondary clock input
  -- Control Ports: 1-bit (each) MMCM control ports
  CLKINSEL => CLKINSEL,         -- 1-bit Clock select input
  PWRDWN => PWRDWN,             -- 1-bit Power-down input
  RST => RST,                   -- 1-bit Reset input
  -- DRP Ports: 7-bit (each) Dynamic reconfiguration ports
  DADDR => DADDR,               -- 7-bit DRP address input
  DCLK => DCLK,                 -- 1-bit DRP clock input
  DEN => DEN,                   -- 1-bit DRP enable input
  DI => DI,                     -- 16-bit DRP data input
  DWE => DWE,                   -- 1-bit DRP write enable input
  -- Dynamic Phase Shift Ports: 1-bit (each) Ports used for dynamic phase shifting of the outputs
  PSCLK => PSCLK,               -- 1-bit Phase shift clock input
  PSEN => PSEN,                 -- 1-bit Phase shift enable input
  PSINCDEC => PSINCDEC,         -- 1-bit Phase shift increment/decrement input
  -- Feedback Clocks: 1-bit (each) Clock feedback ports
  CLKFBIN => CLKFBIN           -- 1-bit Feedback clock input
);

-- End of MMCM_ADV_inst instantiation

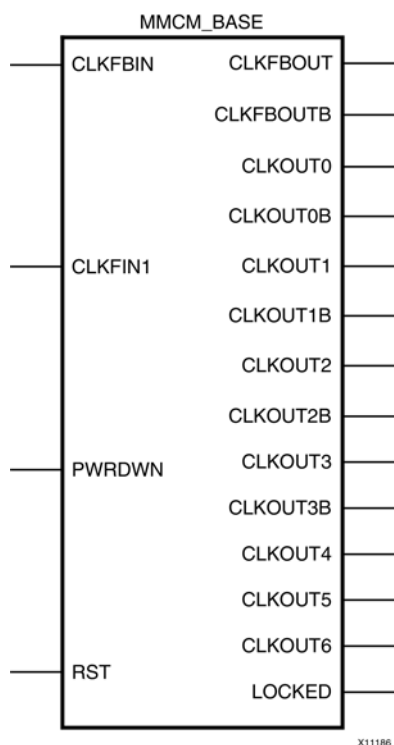
```

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

: 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、ジッタ低減をサポートするための混合信号ブロックです。7 つの O カウンタを独立してプログラムすることができます。つまり O0 は 2、O1 は 3 で分周するよう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を駆動するため、VCO 動作周波数をすべての出力カウンタと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	タイプ	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	ユーザー コンフィギュラブル 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。
LOCKED	出力	1	MMCM で定義されている時間枠内で位相アライメントが完了し、PPM 範囲内で周波数が一致したことを示します。MMCM は電源投入時

ポート名	タイプ	幅	機能
			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	パワー ダウンがインスタンスシートされますが、MMCM は未使用です。
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。入力クロックのコンディション（周波数など）が変わる場合、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	タイプ	値	デフォルト	説明
BANDWIDTH	文字列	OPTIMIZED、HIGH、LOW	OPTIMIZED	ジッタ、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	1.0	別の周波数を使用する場合に、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、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	MMCM CLKIN1 入力の入力周期を指定します。精度は ps までです。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	別の周波数を使用する場合に、CLKOUT クロック出力を分周する値を指定します。この値、CLKFBOUT_MULT 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 度または 4 分の 1 サイクルの位相オフセット、180 は 180 度または 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 の場合、VCO 周波数を CLKIN を損失する前の近似値に保持します。

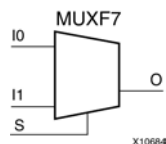
属性	タイプ	値	デフォルト	説明
DIVCLK_DIVIDE	整数	1 ~ 128	1	入力クロックに対するすべての出力クロックの分周比を指定し、PFD に入力される CLKIN を分周します。
REF_JITTER1	3 上位ビット 浮動小数点	0.000 ~ 0.999	0.010	MMCM パフォーマンスを最適化するため、リファレンス クロックに予測されるジッタ値を指定します。バンド幅が OPTIMIZED に設定されていると、値が既知でない場合は入力クロックに最適なパラメータが選択されます。値が既知である場合は、値を入力クロックに予測されるジッタの UI パーセント (最大ピークトゥピーク値) 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。

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

: 2-to-1 Look-Up Table Multiplexer with General Output



概要

このデザイン エレメントは、対応するルックアップ テーブルと組み合わせて、7 ファンクションのルックアップ テーブルまたは 8:1 マルチプレクサを作成するためのマルチプレクサ 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予測をより正確に行う必要がある場合に使用できます。

論理表

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

ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
I0	入力	1	入力 (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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

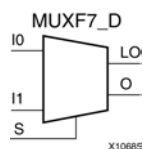
-- End of MUXF7_inst instantiation
```

詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](http://www.xilinx.com/products/fpga/virtex6/index.html)

MUXF7_D

: 2-to-1 Look-Up Table Multiplexer with Dual Output



概要

このデザイン エLEMENTは、対応するルックアップ テーブルと組み合わせて、7 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサを作成するためのマルチプレクサ 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 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

MUXF7_D_inst : MUXF7_D
port map (
    LO => LO,  -- Ouput of MUX to local routing
    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_D_inst instantiation
```

Verilog 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

MUXF7_D_inst : MUXF7_D
port map (
    LO => LO,  -- Ouput of MUX to local routing
    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
);

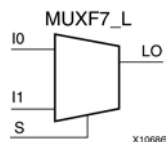
-- End of MUXF7_D_inst instantiation
```

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

: 2-to-1 look-up table Multiplexer with Local Output



概要

このデザイン エレメントは、対応するルックアップ テーブルと組み合わせて、7 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサを作成するためのマルチプレクサ 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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

MUXF7_L_inst : MUXF7_L
port map (
    LO => LO,  -- Output of MUX to local routing
    IO => IO,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S     -- Input select to MUX
);

-- End of MUXF7_L_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

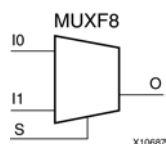
-- End of MUXF7_L_inst instantiation
```

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

: 2-to-1 Look-Up Table Multiplexer with General Output



概要

このデザイン エLEMENTは、対応するルックアップ 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 マルチプレクサを作成するためのマルチプレクサ 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。

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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

MUXF8_inst : MUXF8
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,     -- 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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

MUXF8_inst : MUXF8
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,     -- Input (tie to MUXF7 LO out)
    I1 => I1,     -- Input (tie to MUXF7 LO out)
    S => S       -- Input select to MUX
);

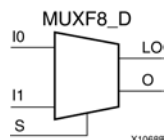
-- End of MUXF8_inst instantiation
```

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

: 2-to-1 Look-Up Table Multiplexer with Dual Output



概要

このデザイン エLEMENTは、対応するルックアップ テーブル 4 つと MUXF8 を 2 つ組み合わせて、8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサを作成するためのマルチプレクサ 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。

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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

MUXF8_D_inst : MUXF8_D
port map (
    LO => LO,  -- Output of MUX to local routing
    O  => O,  -- Output of MUX to general routing
    I0 => I0,  -- 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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

MUXF8_D_inst : MUXF8_D
port map (
    LO => LO,  -- Output of MUX to local routing
    O  => O,  -- Output of MUX to general routing
    I0 => I0,  -- Input (tie to MUXF7 LO out)
    I1 => I1,  -- Input (tie to MUXF7 LO out)
    S  => S    -- Input select to MUX
);

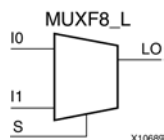
-- End of MUXF8_D_inst instantiation
```

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

: 2-to-1 Look-Up Table Multiplexer with Local Output



概要

このデザイン エLEMENTは、対応するルックアップ テーブル 4 つと MUXF8 を 2 つ組み合わせて、8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサを作成するためのマルチプレクサ 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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

MUXF8_L_inst : MUXF8_L
port map (
    LO => LO,  -- Output of MUX to local routing
    I0 => I0,  -- 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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

MUXF8_L_inst : MUXF8_L
port map (
    LO => LO,  -- Output of MUX to local routing
    I0 => I0,  -- Input (tie to MUXF7 LO out)
    I1 => I1,  -- Input (tie to MUXF7 LO out)
    S => S     -- Input select to MUX
);

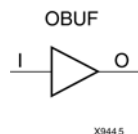
-- End of MUXF8_L_inst instantiation
```

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

: Output Buffer



概要

このデザイン エLEMENTは単純な出力バッファで、出力信号を、トライステートでない FPGA デバイス ピンに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=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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

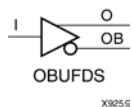
-- End of OBUF_inst instantiation
```

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

: 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」、「スレーブ」と呼びます。マスタとスレーブは 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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

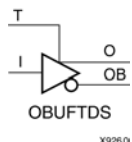
-- End of OBUFDS_inst instantiation
```

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

: 3-State Output Buffer with Differential Signaling, Active-Low Output Enable



概要

このデザイン エLEMENTは、低電圧差動信号を使用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」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	タイプ	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をELEMENTに割り当て

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

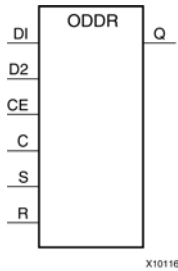
-- End of OBUFTDS_inst instantiation
```

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

: 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 入力 がクロック信号 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	タイプ	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 がイ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	タイプ	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	整数	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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

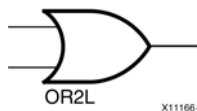
-- End of ODDR_inst instantiation
```

詳細情報

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

OR2L

: Two input OR gate implemented in place of a Slice Latch



概要

このデザイン エLEMENTでは、コンフィギュレーション可能なスライス ラッチで 2 入力 OR ゲートのファンクションが使用されます (論理表を参照)。このELEMENTを使用すると、ロジックのレジスタ/ラッチ リソース数をトレードオフにすることで、ロジック レベルを削減して、デバイスのロジック集積度を高めることができます。このELEMENTはレジスタのパックおよび集積度に影響を与えるので注意してください。AND2B1L または OR2L ELEMENTをスライスに指定すると、残りのレジスタおよびラッチが使用できなくなります。

論理表

入力		出力
DI	SRI	O
0	0	0
0	1	1
1	0	1
1	1	1

ポートの説明

ポート名	タイプ	幅	機能
O	出力	1	OR ゲートの出力
DI	入力	1	同じスライスにあるソース LUT に通常接続されるアクティブ High の入力
SRI	入力	1	通常スライス外からソースされるアクティブ Low の入力 メモ: 複数の AND2B1L または OR2B1L を 1 つのスライスにパックするには、この入力に共通の信号を接続する必要があります。

デザインの入力方法

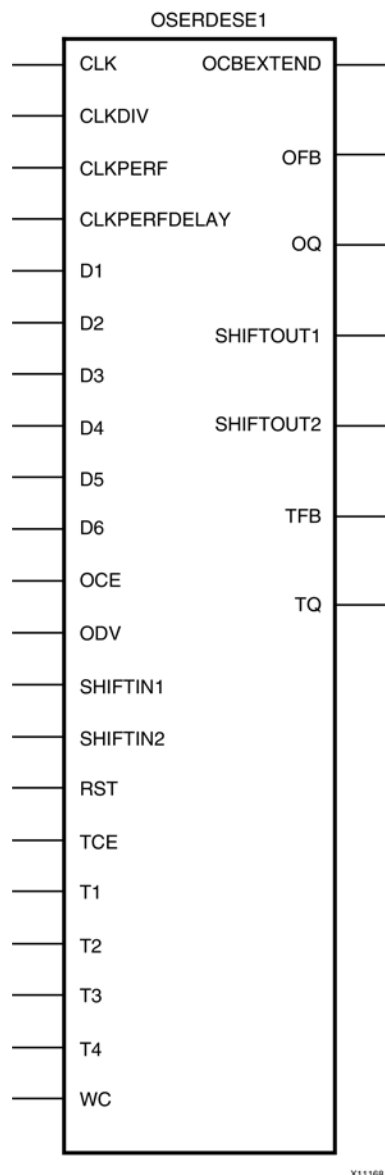
インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

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

: Dedicated IOB Output Serializer



X11168

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が含まれています。データシリアライザおよびトライステートシリアライザは、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 (OSERDES 幅を拡張している場合は 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かけられます。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	タイプ	幅	機能
CLK	入力	1	高速クロック入力。パラレル/シリアル コンバータを駆動するのに使用するクロック入力です。
CLKDIV	入力	1	高速分周クロック入力。パラレル/シリアル コンバータを駆動するのに使用します。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 モードで使し、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拡大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、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です。入力ポート D1 の データが OQ に一番最初に出力されます。このポートは、データ パラレル/シリアル コンバータの出力と 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 パルスを供給するだけですみます。RST は、CLKDIV の 1 サイクル以上 High に駆動する必要があります。複数の OSERDES ポートを含むインターフェイスを構築する場合は、すべての OSERDES ポートを同期化する必要があります。RST 入力の内部タイミングは、同じリセット パルスを受信するすべての OSERDES ブロックがお互いに同期してリセット状態から戻るように調整されます。

ポート名	タイプ	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です。スレーブの SHIFTOUT1、SHIFTOUT2 に接続します。
SHIFTOUT1/ SHIFTOUT2	出力	1	データ入力を拡張するためのカスケード出力です。マスタの SHIFTIN1、SHIFTIN2 に接続します。
TCE	入力	1	トライステート制御パス用のアクティブ High のクロック イネーブル
TFB	出力	1	IODELAY に送信される OSERDES モジュールのトライステート制御出力。トライステート パラレル/シリアル コンバータの出力を IODELAY の制御/トライステート入力に接続します。
TQ	出力	1	OSERDES モジュールのトライステート制御出力。トライステート パラレル/シリアル コンバータの出力を IOB の制御/トライステート入力に接続します。
T1 ~ T4	入力	1	パラレルトライステート入力。パラレルトライステート信号は、T1 ~ T4 から OSERDES モジュールに入力されます。このポートは FPGA に接続され、1、2、または 4 ビットにコンフィギュレーションできます。
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時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	タイプ	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
DATA_RATE_TQ	文字列	DDR、BUF、SDR	DDR	トライステート (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 コンバータの幅を指定。有効な値は、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	1, 0	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
INIT_TQ	2 進数	1'b0 ~ 1'b1	1'b0	TQ 出力の初期値を指定
INTERFACE_TYPE	文字列	DEFAULT、MEMORY_DDR3	DEFAULT	OSERDESE1 の使用モデルを選択します。

属性	タイプ	値	デフォルト	説明
ODELAY_USED	整数	0、1	0	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。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 OSERDESE1 をマスタ 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。 メモ： 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。
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かスレーブかを指定
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
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
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の幅を指定。DATA_RATE_TQ = DDR、DATA_WIDTH = 4、および DATA_RATE_OQ = DDR の場合、値は 1 または 4 に制限されます。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、値は 1 になります。

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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,              -- DDR3 data mode DQ/DQS=0, other=1
    INIT_OQ => '0',              -- Initial value of OQ output
    INIT_TQ => '0',              -- Initial value of TQ output
    INTERFACE_TYPE => "DEFAULT", -- DEFAULT or MEMORY_DDR3
    ODELAY_USED => 0,            -- DDR3 mode attribute (0/1)
    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          -- 3-state converter width (1 or 4)
)
port map (
    OCBEXTEND => OCBEXTEND,      -- 1-bit Used only for DDR3 applications
    OFB => OFB,                  -- 1-bit Feedback path for data output

```

```
OQ => OQ, -- 1-bit Data path output
-- SHIFTOUT1/SHIFTOUT2: 1-bit (each) 1-bit Data output expansion
SHIFTOUT1 => SHIFTOUT1,
SHIFTOUT2 => SHIFTOUT2,
TFB => TFB, -- 1-bit 3-state control output
TQ => TQ, -- 1-bit 3-state path output
CLK => CLK, -- 1-bit High speed clock input
CLKDIV => CLKDIV, -- 1-bit Divided high-speed clock input
CLKPERF => CLKPERF, -- 1-bit High-peformance clock input for DDR3 applications
CLKPERFDELAY => CLKPERFDELAY, -- 1-bit Delayed high-peformance clock input for DDR3 applications
-- D1 - D6: 1-bit (each) 1-bit Parallel data inputs
D1 => D1,
D2 => D2,
D3 => D3,
D4 => D4,
D5 => D5,
D6 => D6,
OCE => OCE, -- 1-bit Parallel to serial converter (data) clock enable input
ODV => ODV, -- 1-bit Used only for DDR3 applications
RST => RST, -- 1-bit Reset input
-- SHIFTIN1/SHIFTIN2: 1-bit (each) 1-bit Data input expansion
SHIFTIN1 => SHIFTIN1,
SHIFTIN2 => SHIFTIN2,
-- T1 - T4: 1-bit (each) Parallel 3-state inputs
T1 => T1,
T2 => T2,
T3 => T3,
T4 => T4,
TCE => TCE, -- 1-bit Parallel to serial converter (3-state) clock enable input
WC => WC -- 1-bit Write command input
);

-- End of OSERDESE1_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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,              -- DDR3 data mode DQ/DQS=0, other=1
  INIT_OQ => '0',              -- Initial value of OQ output
  INIT_TQ => '0',              -- Initial value of TQ output
  INTERFACE_TYPE => "DEFAULT", -- DEFAULT or MEMORY_DDR3
  ODELAY_USED => 0,            -- DDR3 mode attribute (0/1)
  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          -- 3-state converter width (1 or 4)
)
port map (
  OCBEXTEND => OCBEXTEND,      -- 1-bit Used only for DDR3 applications
  OFB => OFB,                  -- 1-bit Feedback path for data output
  OQ => OQ,                    -- 1-bit Data path output
  -- SHIFTOUT1/SHIFTOUT2: 1-bit (each) 1-bit Data output expansion
  SHIFTOUT1 => SHIFTOUT1,
  SHIFTOUT2 => SHIFTOUT2,
  TFB => TFB,                  -- 1-bit 3-state control output
  TQ => TQ,                    -- 1-bit 3-state path output
  CLK => CLK,                  -- 1-bit High speed clock input
  CLKDIV => CLKDIV,            -- 1-bit Divided high-speed clock input
  CLKPERF => CLKPERF,         -- 1-bit High-peformance clock input for DDR3 applications
  CLKPERFDELAY => CLKPERFDELAY, -- 1-bit Delayed high-peformance clock input for DDR3 applications
  -- D1 - D6: 1-bit (each) 1-bit Parallel data inputs
  D1 => D1,
  D2 => D2,
  D3 => D3,
  D4 => D4,
  D5 => D5,
  D6 => D6,
  OCE => OCE,                  -- 1-bit Parallel to serial converter (data) clock enable input
  ODV => ODV,                  -- 1-bit Used only for DDR3 applications
  RST => RST,                  -- 1-bit Reset input
  -- SHIFTIN1/SHIFTIN2: 1-bit (each) 1-bit Data input expansion
  SHIFTIN1 => SHIFTIN1,
  SHIFTIN2 => SHIFTIN2,
  -- T1 - T4: 1-bit (each) Parallel 3-state inputs
  T1 => T1,
  T2 => T2,
  T3 => T3,
  T4 => T4,
  TCE => TCE,                  -- 1-bit Parallel to serial converter (3-state) clock enable input
  WC => WC                     -- 1-bit Write command input
);

-- 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

: PCI Express version 2.0 Compliant.



概要

このELEMENTは、RocketIO™ トランシーバ、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、PCIE_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ELEMENTをインスタンス化するには、PCI EXPRESS コアまたはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンス化しないでください。

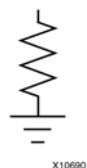
詳細情報

- ・ [Virtex-6 FPGA RocketIO GTX トランシーバ 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

: Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs

PULLDOWN



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 つの構文が存在しない場合はコピーし、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 12.2

PULLDOWN_inst : PULLDOWN
port map (
  O => O      -- Pulldown output (connect directly to top-level port)
);

-- End of PULLDOWN_inst instantiation
```

Verilog 記述（インスタンス化）

次の 2 つの構文が存在しない場合はコピーし、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 12.2  
  
PULLDOWN_inst : PULLDOWN  
port map (  
    O => O      -- 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

: Resistor to VCC for Input PADs, Open-Drain, and 3-State Outputs



概要

このデザイン エLEMENTは、1 つの入力、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、値、weak High で駆動できます。このELEMENTは、すべてのドライバが使用されていないときにオープンドレイン ELEMENTおよびマクロのロジック レベルを 1 (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 つの構文が存在しない場合はコピーし、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 12.2

PULLUP_inst : PULLUP
port map (
  O => O      -- Pullup output (connect directly to top-level port)
);

-- End of PULLUP_inst instantiation
```

Verilog 記述（インスタンス化）

次の 2 つの構文が存在しない場合はコピーし、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 12.2

PULLUP_inst : PULLUP
port map (
  O => O      -- Pullup output (connect directly to top-level port)
);

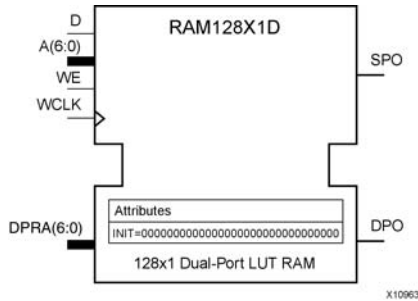
-- End of PULLUP_inst instantiation
```

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

: 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 で指定された読み出し/書き込みポートのデータ出力
DPO	出力	1	アドレス バス DPRA で指定された読み出しポートのデータ出力
D	入力	1	アドレス バス A で指定された書き込みデータ入力
A	入力	7	読み出し/書き込みポートのアドレス バス
DPRA	入力	7	読み出しポートのアドレス バス
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

インスタンスエートする場合は、このコンポーネントを次のように接続します。

- WCLK 入力をクロック ソースに、D 入力を格納するデータ ソースに、DPO 出力を FDCE の D 入力などの適切なデスティネーションに接続します。
- オプションで、SPO 出力を適切なデスティネーションに接続するか、または未接続にすることもできます。
- クロック イネーブル ピン (WE) は、適切なライト イネーブル ソースに接続します。
- 7 ビット バス A は読み出し/書き込みアドレスに、7 ビット バス DPRA は読み出しアドレスに接続する必要があります。
- 128 ビットの 16 進数で構成される INIT 属性で、RAM の初期値を指定できます。

指定しない場合は、初期値はすべて 0 になります。

デザインの入力方法

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

使用可能な属性

属性	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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

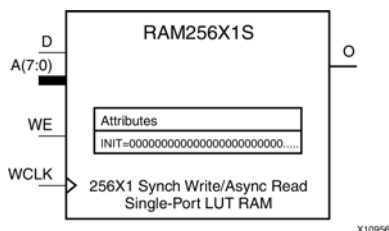
-- End of RAM128X1D_inst instantiation
```

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

: 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。ただし、この場合は 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。RAM256X1S には、アクティブ High のライト イネーブル (WE) があり、この信号が 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、出力の値が更新されます。

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 で指定された読み出し/書き込みポートのデータ出力
D	入力	1	アドレス バス A で指定された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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

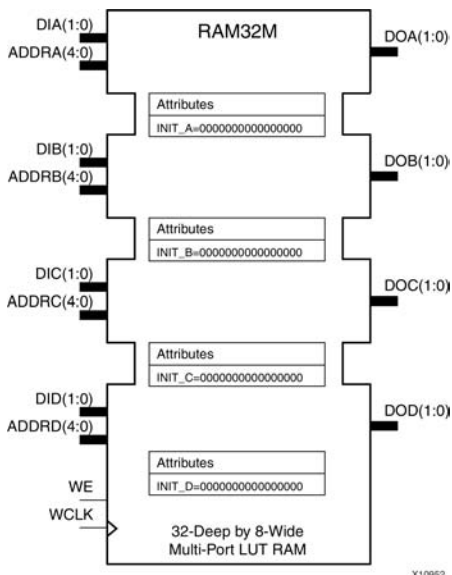
-- End of RAM256X1S_inst instantiation
```

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

: 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)
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 ADDRC 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 ADDR_D は読み出し/書き込みアドレスに、5 ビットバス ADDR_A、ADDR_B、ADDR_C 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、 $ADDR_y[z] = INIT_y[2*z+1:2*z]$ で計算されます。たとえば、RAM の ADDR_C 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	タイプ	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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
  ADDR_A => ADDR_A, -- 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 ADDR_A
  DIB => DIB, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_B
  DIC => DIC, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_C
  DID => DID, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_D
  WCLK => WCLK, -- Write clock input
  WE => WE, -- Write enable input
);
-- End of RAM32M_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

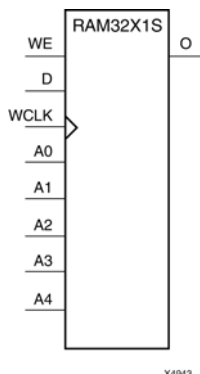
RAM32M_inst : RAM32M
generic map (
  INIT_A => X"0000000000000000",  -- Initial contents of A port
  INIT_B => X"0000000000000000",  -- Initial contents of B port
  INIT_C => X"0000000000000000",  -- Initial contents of C port
  INIT_D => X"0000000000000000")  -- Initial contents of D port
port map (
  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
```

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

: 32-Deep by 1-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 32 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードにロードされます。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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

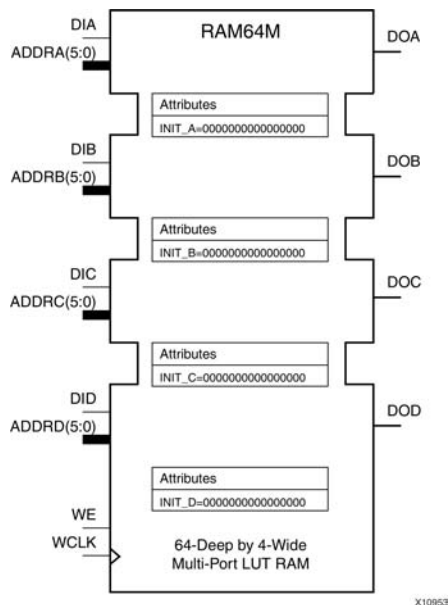
-- End of RAM32X1S_inst instantiation
```

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

： 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、1 ビット読み出しのポート 1 つと、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 をグランドに接続した場合、DOD は使用されません。ADDRA、ADDRb、ADDRc を同じアドレスに接続すると、64x3 のシンプルデュアル ポート RAM になります。ADDRd を ADDRA、ADDRb、ADDRc に接続すると、64x4 のシングル ポート RAM になります。この RAM には、ほかにも可能なコンフィギュレーションがあります。

ポートの説明

ポート名	方向	幅	機能
DOA	出力	1	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	1	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	1	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	1	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	1	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	1	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)
DIC	入力	1	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRC で指定)
DID	入力	1	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	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 ADDR_D は読み出し/書き込みアドレスに、5 ビットバス ADDR_A、ADDR_B、ADDR_C は読み出しアドレスに接続する必要があります。オプションで INIT_A、INIT_B、INIT_C、INIT_D 属性を使用すると、各ポートの初期メモリ内容を 64 ビット (16 進数) で指定できます。RAM の INIT 値は、ADDRy[z] = INIT_y[z] で計算されます。

たとえば、RAM の ADDR_C 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	タイプ	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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
  ADDR_A => ADDR_A, -- Read port A 6-bit address input
  ADDR_B => ADDR_B, -- Read port B 6-bit address input
  ADDR_C => ADDR_C, -- Read port C 6-bit address input
  ADDR_D => ADDR_D, -- Read/Write port D 6-bit address input
  DIA => DIA, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_A
  DIB => DIB, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_B
  DIC => DIC, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_C
  DID => DID, -- RAM 1-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_D
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);
-- End of RAM64M_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

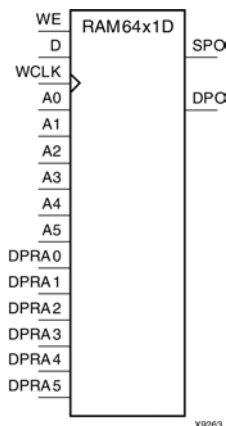
RAM64M_inst : RAM64M
generic map (
  INIT_A => X"0000000000000000",  -- Initial contents of A port
  INIT_B => X"0000000000000000",  -- Initial contents of B port
  INIT_C => X"0000000000000000",  -- Initial contents of C port
  INIT_D => X"0000000000000000")  -- Initial contents of D port
port map (
  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
```

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

: 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 に切り替わるときに、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ロードされます。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	64 ビット値	すべてゼロ	RAM、レジスタ、LUT の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

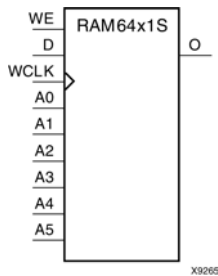
-- End of RAM64X1D_1_inst instantiation
```

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

: 64-Deep by 1-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 64 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 6 ビットのアドレス (A5 ～ A0) で選択されたワードにロードされます。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は、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。

INIT 属性を使用すると、コンフィギュレーション中にこのELEMENTを初期化できます。

論理表

モード選択を次の論理表に示します。

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D	D
1 (読み出し)	↓	X	データ
データ = A5 ～ A0 で指定されたワード			

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	ROM、RAM、レジスタ、LUT の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

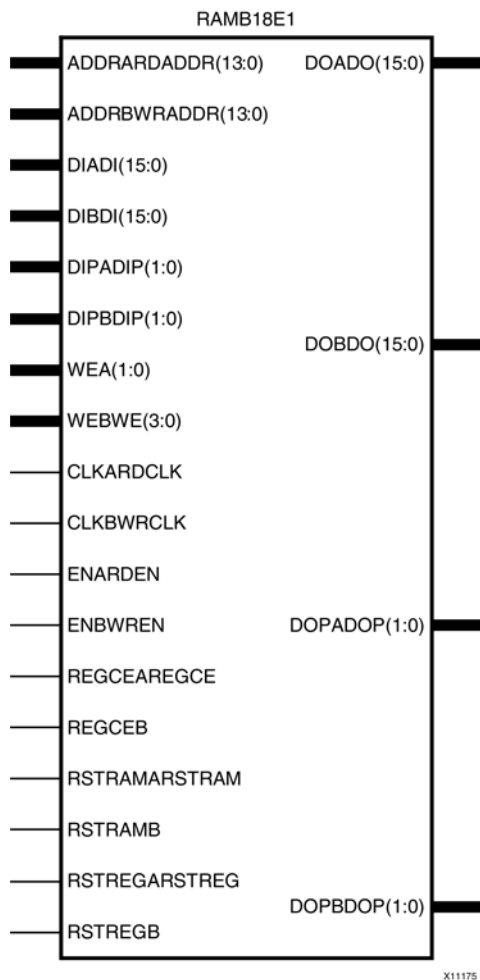
-- End of RAM64X1S_inst instantiation
```

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

: 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。またこのデザイン エLEMENTを使用すると、18kb の FIFO のブロック RAM へアクセスできます。この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 X 512 ワードの単純デュアル 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、READ と WRITE 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	タイプ	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
ADDRBWRADDR[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	タイプ	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	SRVAL_A で示される値に同期データ ラッチ セット/リセット。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	SRVAL_B で示される値に同期データ ラッチ セット/リセット。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	SRVAL_A で示される値に同期出力レジスタ セット/リセット。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	タイプ	幅	機能
RSTREGB	入力	1	SRVAL_B で示される値に同期出力レジスタ セット/リセット。 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	タイプ	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にシミュレーションの動作を変更できます。</p> <ul style="list-style-type: none"> ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 GENERATE_X_ONLY に設定すると、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。 NONE に設定すると、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。 <p>メモ： 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>

属性	タイプ	値	デフォルト	説明
DOA_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。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、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。
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	0 ビット文字列	なし	初期 RAM の内容を指定するファイル名
INIT_00 ~ INIT_3F	16 進数	256'h0000000000 00000000000000 00000000000000 00000000000000 000 ~ 256'hffffffff ffffffffffffffff ffffffffffffffff	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	256'h0000000000 00000000000000 00000000000000 00000000000000 000 ~ 256'hffffffff ffffffffffffffff ffffffffffffffff	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	シングル デュアル ポート (SDP) または真のデュアル ポート (TDP) を選択します。
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。

属性	タイプ	値	デフォルト	説明
READ_WIDTH_B	整数	0、1、2、4、9、18	0	ポート B の読み出しのデータ幅を指定します (パリティビットを含む)。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。
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	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"> WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 READ_FIRST に設定すると、そのメモリロケーションに直前に格納されていた値が出力ポートに出力されます。 NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
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、任意のデータ幅に設定してください。SDP 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、任意のデータ幅に設定してください。SDP の場合は、パリティビットを含む書き込み幅です。

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.1

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"00000000",
INIT_B => X"00000000",
INIT_FILE => "NONE",
RAM_MODE => "TDP",
-- 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"00000000",
SRVAL_B => X"00000000",
-- 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 (
  DOADO => DOADO,
  DOBDO => DOBDO,
  DOPADOP => DOPADOP,
  DOPBDOP => DOPBDOP,
  ADDRARDADDR => ADDRARDADDR,
  ADDRBRWADDR => ADDRBRWADDR,
  CLKARDCLK => CLKARDCLK,
  CLKBWRCLK => CLKBWRCLK,
  DIADI => DIADI,
  DIBDI => DIBDI,
  DIPADIP => DIPADIP,
  DIPBDIP => DIPBDIP,
  ENARDEN => ENARDEN,
  ENBWREN => ENBWREN,
  REGCEAREGCE => REGCEAREGCE,
  REGCEB => REGCEB,
  RSTRAMARSTRAM => RSTRAMARSTRAM,
  RSTRAMB => RSTRAMB,
  RSTREGARSTREG => RSTREGARSTREG,
  RSTREGB => RSTREGB,
  WEA => WEA,
  WEBWE => WEBWE
);
-- End of RAMB18E1_inst instantiation

```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.1

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"00000000",
INIT_B => X"00000000",
INIT_FILE => "NONE", -- RAM init file
RAM_MODE => "TDP", -- SDP or TDP
-- READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per port
READ_WIDTH_A => 0, -- 0,1,2,4,9,18,36
READ_WIDTH_B => 0, -- 0,1,2,4,9,18
WRITE_WIDTH_A => 0, -- 0,1,2,4,9,18
WRITE_WIDTH_B => 0, -- 0,1,2,4,9,18,36
-- 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"00000000",
SRVAL_B => X"00000000",
-- 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 (
  DOADO => DOADO, -- 16-bit A port data/LSB data output
  DOBDO => DOBDO, -- 16-bit B port data/MSB data output
  DOPADOP => DOPADOP, -- 2-bit A port parity/LSB parity output
  DOPBDOP => DOPBDOP, -- 2-bit B port parity/MSB parity output
  ADDRARDADDR => ADDRARDADDR, -- 14-bit A port address/Read address input
  ADDRBRWADDR => ADDRBRWADDR, -- 14-bit B port address/Write address input
  CLKARDCLK => CLKARDCLK, -- 1-bit A port clock/Read clock input
  CLKBWRCLK => CLKBWRCLK, -- 1-bit B port clock/Write clock input
  DIADI => DIADI, -- 16-bit A port data/LSB data input
  DIBDI => DIBDI, -- 16-bit B port data/MSB data input
  DIPADIP => DIPADIP, -- 2-bit A port parity/LSB parity input
  DIPBDIP => DIPBDIP, -- 2-bit B port parity/MSB parity input
  ENARDEN => ENARDEN, -- 1-bit A port enable/Read enable input
  ENBWREN => ENBWREN, -- 1-bit B port enable/Write enable input
  REGCEAREGCE => REGCEAREGCE, -- 1-bit A port register enable/Register enable input
  REGCEB => REGCEB, -- 1-bit B port register enable input
  RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit A port set/reset input
  RSTRAMB => RSTRAMB, -- 1-bit B port set/reset input
  RSTREGARSTREG => RSTREGARSTREG, -- 1-bit A port register set/reset input
  RSTREGB => RSTREGB, -- 1-bit B port register set/reset input
  WEA => WEA, -- 2-bit A port write enable input
  WEBWE => WEBWE -- 4-bit B port write enable/Write enable input
);
-- End of RAMB18E1_inst instantiation

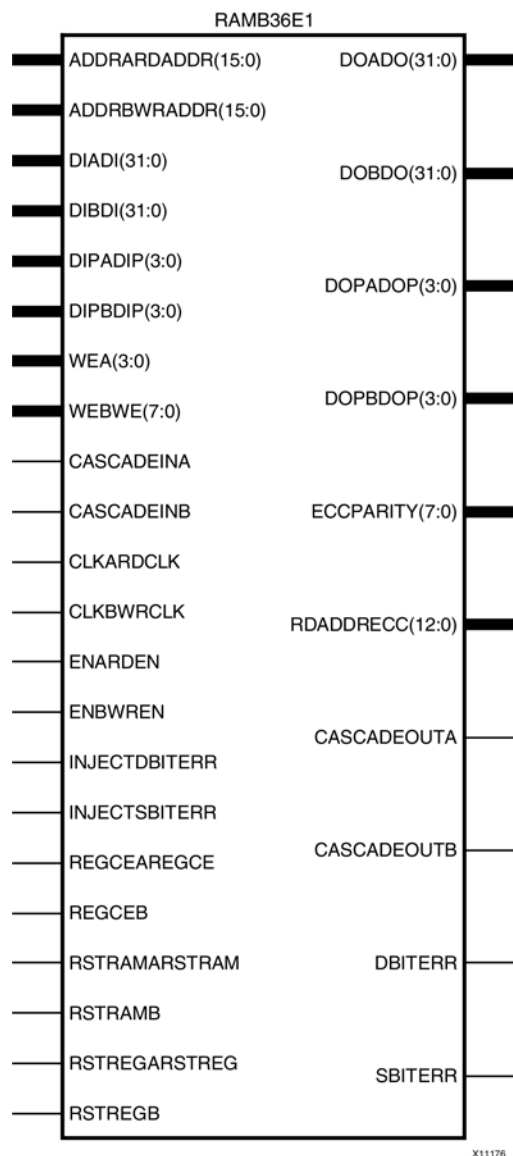
```

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

: 36K-bit Configurable Synchronous Block RAM



X11176

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。またこのデザイン エレメントを使用すると、36kb の FIFO のブロック RAM へアクセスできます。このエレメントをカスケード 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、READ と WRITE は互いに独立しており、同じメモリ アレイにアクセスする間は非同期になります。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	タイプ	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	メモリ エラー検出と訂正を行う ECC デコーダで使用される ECC エンコーダから生成された 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	タイプ	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	SRVAL_A で示される値に同期データ ラッチ セット/リセット。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	SRVAL_B で示される値に同期データ ラッチ セット/リセット。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	SRVAL_A で示される値に同期出力レジスタ セット/リセット。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	SRVAL_B で示される値に同期出力レジスタ セット/リセット。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

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

使用可能な属性

属性	タイプ	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にシミュレーションの動作を変更できます。</p> <ul style="list-style-type: none"> ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 GENERATE_X_ONLY に設定すると、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。 NONE に設定すると、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。 <p>メモ： 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。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。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	タイプ	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 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	0 ビット文字列	NONE	初期 RAM の内容を指定するファイル名
INIT_00 ~ INIT_7F	16 進数	256'h00000000 000000000000 000000000000 000000000000 0000 ~ 256'hffffffff ffffffffffffff ffffffffffffff	すべてゼロ	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'h00000000 000000000000 000000000000 000000000000 0000 ~ 256'hffffffff ffffffffffffff ffffffffffffff	すべてゼロ	4kb のパリティ メモリ アレイの初期値を指定
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	シンプル デュアル ポート (SDP) または真のデュアル ポート (TDP) を選択します。
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 (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 (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。

属性	タイプ	値	デフォルト	説明
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のポートの動作を指定します。 <ul style="list-style-type: none"> WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 READ_FIRST に設定すると、そのメモリロケーションに直前に格納されていた値が出力ポートに出力されます。 NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
WRITE_WIDTH_A	整数	0、1、2、4、9、18、 36、72	0	ポート A の書き込みのデータ幅を指定 (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 (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 つの構文が存在しない場合はコピーし、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 12.1

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)

```

Virtex-6 ライブラリ ガイド (HDL 用)
UG623 (v12.2) 2010 年 7 月 23 日

Virtex-6 ライブラリ ガイド (HDL 用)
UG623 (v12.2) 2010 年 7 月 23 日

```

INIT_7E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- INIT_A, INIT_B: Initial values on output ports
INIT_A => X"000000000",
INIT_B => X"000000000",
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
-- 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"000000000",
SRVAL_B => X"000000000",
-- 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 (
  CASCADEOUTA => CASCADEOUTA, -- 1-bit A port cascade output
  CASCADEOUTB => CASCADEOUTB, -- 1-bit B port cascade output
  DBITERR => DBITERR, -- 1-bit double bit error status output
  DOADO => DOADO, -- 32-bit A port data/LSB data output
  DOBDO => DOBDO, -- 32-bit B port data/MSB data output
  DOPADOP => DOPADOP, -- 4-bit A port parity/LSB parity output
  DOPBDOP => DOPBDOP, -- 4-bit B port parity/MSB parity output
  ECCPARITY => ECCPARITY, -- 8-bit generated error correction parity
  RDADDRECC => RDADDRECC, -- 9-bit ECC read address. Not used when RAM_MODE=TDP.
  SBITERR => SBITERR, -- 1-bit Single bit error status output
  ADDRARDADDR => ADDRARDADDR, -- 16-bit A port address/Read address input
  ADDRBRWADDR => ADDRBRWADDR, -- 16-bit B port address/Write address input
  CASCADEINA => CASCADEINA, -- 1-bit A port cascade input
  CASCADEINB => CASCADEINB, -- 1-bit B port cascade input
  CLKARDCLK => CLKARDCLK, -- 1-bit A port clock/Read clock input
  CLKBWRCLK => CLKBWRCLK, -- 1-bit B port clock/Write clock input
  DIADI => DIADI, -- 32-bit A port data/LSB data input
  DIBDI => DIBDI, -- 32-bit B port data/MSB data input
  DIPADIP => DIPADIP, -- 4-bit A port parity/LSB parity input
  DIPBDIP => DIPBDIP, -- 4-bit B port parity/MSB parity input
  ENARDEN => ENARDEN, -- 1-bit A port enable/Read enable input
  ENBWREN => ENBWREN, -- 1-bit B port enable/Write enable input
  INJECTDBITERR => INJECTDBITERR, -- 1-bit Inject a double bit error if ECC feature is used.
  INJECTSBITERR => INJECTSBITERR, -- 1-bit Inject a single bit error if ECC feature is used.
  REGCEAREGCE => REGCEAREGCE, -- 1-bit A port register enable/Register enable input
  REGCEB => REGCEB, -- 1-bit B port register enable input
  RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit A port set/reset input
  RSTRAMB => RSTRAMB, -- 1-bit B port set/reset input
  RSTREGARSTREG => RSTREGARSTREG, -- 1-bit A port register set/reset input
  RSTREGB => RSTREGB, -- 1-bit B port register set/reset input
  WEA => WEA, -- 4-bit A port write enable input
  WEBWE => WEBWE -- 8-bit B port write enable/Write enable input
);

-- End of RAMB36E1_inst instantiation

```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.1

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",
```

Virtex-6 ライブラリ ガイド (HDL 用)
UG623 (v12.2) 2010 年 7 月 23 日

```

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",
-- INIT_A, INIT_B: Initial values on output ports
INIT_A => X"000000000",
INIT_B => X"000000000",
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
-- 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"000000000",
SRVAL_B => X"000000000",
-- 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 (
  CASCADEOUTA => CASCADEOUTA, -- 1-bit A port cascade output
  CASCADEOUTB => CASCADEOUTB, -- 1-bit B port cascade output
  DBITERR => DBITERR, -- 1-bit double bit error status output
  DOADO => DOADO, -- 32-bit A port data/LSB data output
  DOBDO => DOBDO, -- 32-bit B port data/MSB data output
  DOPADOP => DOPADOP, -- 4-bit A port parity/LSB parity output
  DOPBDOP => DOPBDOP, -- 4-bit B port parity/MSB parity output
  ECCPARITY => ECCPARITY, -- 8-bit generated error correction parity
  RDADDRECC => RDADDRECC, -- 9-bit ECC read address. Not used when RAM_MODE=TDP.
  SBITERR => SBITERR, -- 1-bit Single bit error status output
  ADDRARDADDR => ADDRARDADDR, -- 16-bit A port address/Read address input
  ADDRBRWADDR => ADDRBRWADDR, -- 16-bit B port address/Write address input
  CASCADEINA => CASCADEINA, -- 1-bit A port cascade input
  CASCADEINB => CASCADEINB, -- 1-bit B port cascade input
  CLKARDCLK => CLKARDCLK, -- 1-bit A port clock/Read clock input
  CLKBWRCLK => CLKBWRCLK, -- 1-bit B port clock/Write clock input
  DIADI => DIADI, -- 32-bit A port data/LSB data input
  DIBDI => DIBDI, -- 32-bit B port data/MSB data input
  DIPADIP => DIPADIP, -- 4-bit A port parity/LSB parity input
  DIPBDIP => DIPBDIP, -- 4-bit B port parity/MSB parity input
  ENARDEN => ENARDEN, -- 1-bit A port enable/Read enable input
  ENBWREN => ENBWREN, -- 1-bit B port enable/Write enable input
  INJECTDBITERR => INJECTDBITERR, -- 1-bit Inject a double bit error if ECC feature is used.
  INJECTSBITERR => INJECTSBITERR, -- 1-bit Inject a single bit error if ECC feature is used.
  REGCEAREGCE => REGCEAREGCE, -- 1-bit A port register enable/Register enable input
  REGCEB => REGCEB, -- 1-bit B port register enable input
  RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit A port set/reset input

```

```
RSTRAMB => RSTRAMB,          -- 1-bit B port set/reset input
RSTREGARSTREG => RSTREGARSTREG, -- 1-bit A port register set/reset input
RSTREGB => RSTREGB,          -- 1-bit B port register set/reset input
WEA => WEA,                  -- 4-bit A port write enable input
WEBWE => WEBWE               -- 8-bit B port write enable/Write enable input
);

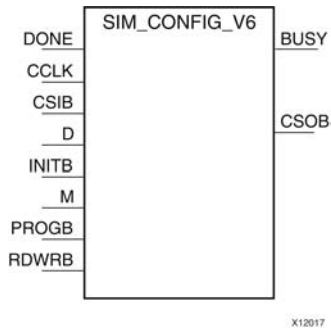
-- End of RAMB36E1_inst instantiation
```

詳細情報

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

SIM_CONFIG_V6

: 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	タイプ	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"> 0 = FPGA のコンフィギュレーションは完了していない 1 = FPGA のコンフィギュレーション完了
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"> 0 = SelectMAP データ バスをイネーブル 1 = SelectMAP データ バスをディスエーブル
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"> 0 = CRC エラー

ポート名	タイプ	幅	機能
			<ul style="list-style-type: none"> 1 = CRC エラーなし 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"> 0 = 入力 1 = 出力 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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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

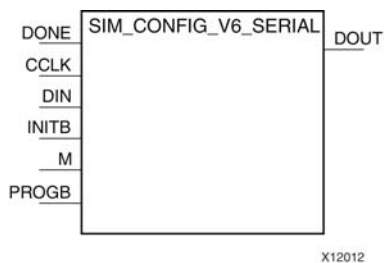
```

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

: 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	タイプ	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"> 0 = FPGA のコンフィギュレーションは完了していない 1 = FPGA のコンフィギュレーション完了
DOUT	出力	1	デイジー チェーンのダウンストリーム デバイス用のシリアル データ出力。データは CCLK の立ち下がりエッジで供給されます。
CCLK	入力	1	JTAG を除くすべてのコンフィギュレーション モードのコンフィギュレーション クロック ソース
DIN	入力	1	シリアル コンフィギュレーション データ入力 (CCLK の立ち上がりエッジに同期)
INITB	入力	1	モード ピンが読み込まれる前は、Low に保持することでコンフィギュレーションを遅延できます。モード ピンが読み込まれた後は、オープンドレインのアクティブ Low 出力となり、コンフィギュレーション中の CRC エラーの有無を示します。 <ul style="list-style-type: none"> 0 = CRC エラー 1 = CRC エラーなし 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 ファイルで使います。

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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
  PROG => PROG,     -- 1-bit input Program pin
);

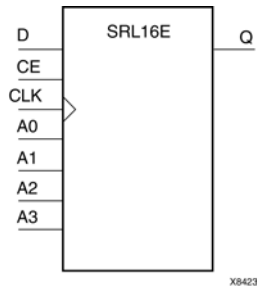
-- End of SIM_CONFIG_V6_SERIAL_inst instantiation
```

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

： 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 の値を一定の値にします。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 <ul style="list-style-type: none"> ・ 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	タイプ	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 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

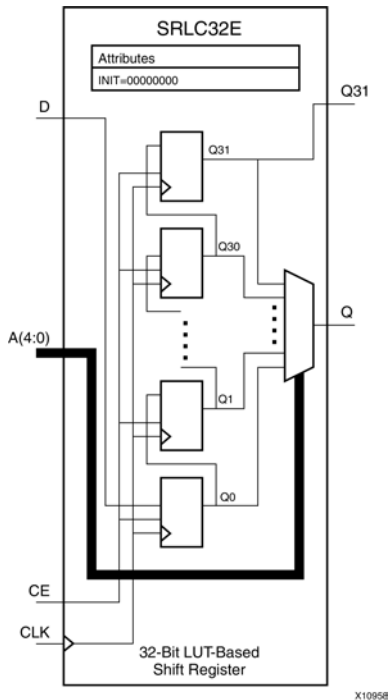
-- End of SRL16E_inst instantiation
```

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

: 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ている、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、より大きな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 のワード数のダイナミック選択 A=00000 ==> 1 ビット シフト長 A=11111 ==> 32 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

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 入力を任意のクロック 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値を 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することもできます。
- ・ シフトレジスタの長さを 32 ビットより大きくする場合は、Q31 出力ピンを後続の SRLC32E の D 入力に接続してカスケード接続します。
- ・ Q31 出力を SRLC32E 以外に接続することはできません。
- ・ 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。
- ・ INIT[0] は、シフトアウトされる最初の値です。

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 の初期のシフト パターンを指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

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
);

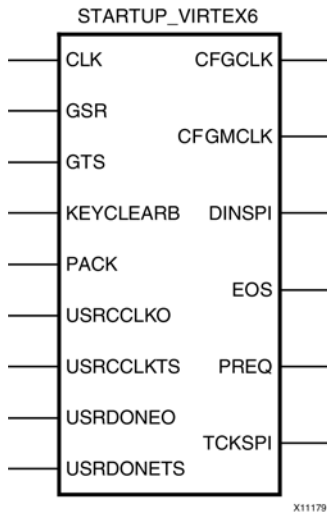
-- End of SRLC32E_inst instantiation
```

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

: 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	タイプ	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
CFGMCLK	出力	1	コンフィギュレーションの内部オシレータのクロック出力
CLK	入力	1	ユーザー スタートアップ クロック
DINSPI	出力	1	DIN SPI PROM アクセス出力
EOS	出力	1	コンフィギュレーションの終了を示すアクティブ High の信号
GSR	入力	1	グローバル セット/リセット (GSR) 入力 (ポート名に GSR は使用不可)
GTS	入力	1	グローバル トライステート (GTS) 入力 (ポート名に GTS は使用不可)
KEYCLEARB	入力	1	バッテリー充電 RAM (BBRAM) からのクリア AES デクリプタ
PACK	入力	1	PROGRAM 確認入力
PREQ	出力	1	デバイス出力への PROGRAM リクエスト
TCKSPI	出力	1	TCK コンフィギュレーション ピン アクセス出力
USRCCLKO	入力	1	ユーザー CCLK 入力
USRCCLKTS	入力	1	内部ユーザー CCLK のトライステート イネーブル
USRDONEO	入力	1	内部ユーザー DONE ピンの出力を制御
USRDONETS	入力	1	ユーザー DONE ピンのトライステート イネーブル

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 エレメントの CLK ピンに接続します。CFGCLK 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する場合に、コンフィギュレーション後の SPI PROM へのアクセスが必要であれば、このコンポーネントの TCK_SPI ピンと DIN_SPI ピンを使用して、ほかの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	タイプ	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。

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

STARTUP_VIRTEX6_inst : STARTUP_VIRTEX6
generic map (
    PROG_USR => FALSE -- Activate program event security feature
)
port map (
    CFGCLK => CFGCLK,      -- 1-bit Configuration main clock output
    CFGMCLK => CFGMCLK,    -- 1-bit Configuration internal oscillator clock output
    DINSPI => DINSPI,      -- 1-bit DIN SPI PROM access output
    EOS => EOS,            -- 1-bit Active high output signal indicating the End Of Configuration.
    PREQ => PREQ,          -- 1-bit PROGRAM request to fabric output
    TCKSPI => TCKSPI,      -- 1-bit TCK configuration pin access output
    CLK => CLK,            -- 1-bit User start-up clock input
    GSR => GSR,            -- 1-bit Global Set/Reset input (GSR cannot be used for the port name)
    GTS => GTS,            -- 1-bit Global 3-state input (GTS cannot be used for the port name)
    KEYCLEARB => KEYCLEARB, -- 1-bit Clear AES Decrypter Key input from Battery-Backed RAM (BDRAM)
    PACK => PACK,          -- 1-bit PROGRAM acknowledge input
    USRCCLKO => USRCCLKO,  -- 1-bit User CCLK input
    USRCCLKTS => USRCCLKTS, -- 1-bit User CCLK 3-state enable input
    USRDONEO => USRDONEO,  -- 1-bit User DONE pin output control
    USRDONETS => USRDONETS -- 1-bit User DONE 3-state enable output
);

-- End of STARTUP_VIRTEX6_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

STARTUP_VIRTEX6_inst : STARTUP_VIRTEX6
generic map (
    PROG_USR => FALSE -- Activate program event security feature
)
port map (
    CFGCLK => CFGCLK,      -- 1-bit Configuration main clock output
    CFGMCLK => CFGMCLK,    -- 1-bit Configuration internal oscillator clock output
    DINSPI => DINSPI,      -- 1-bit DIN SPI PROM access output
    EOS => EOS,            -- 1-bit Active high output signal indicating the End Of Configuration.
    PREQ => PREQ,          -- 1-bit PROGRAM request to fabric output
    TCKSPI => TCKSPI,      -- 1-bit TCK configuration pin access output
    CLK => CLK,            -- 1-bit User start-up clock input
    GSR => GSR,            -- 1-bit Global Set/Reset input (GSR cannot be used for the port name)
    GTS => GTS,            -- 1-bit Global 3-state input (GTS cannot be used for the port name)
    KEYCLEARB => KEYCLEARB, -- 1-bit Clear AES Decrypter Key input from Battery-Backed RAM (BBRAM)
    PACK => PACK,          -- 1-bit PROGRAM acknowledge input
    USRCCLKO => USRCCLKO,  -- 1-bit User CCLK input
    USRCCLKTS => USRCCLKTS, -- 1-bit User CCLK 3-state enable input
    USRDONEO => USRDONEO,  -- 1-bit User DONE pin output control
    USRDONETS => USRDONETS -- 1-bit User DONE 3-state enable output
);

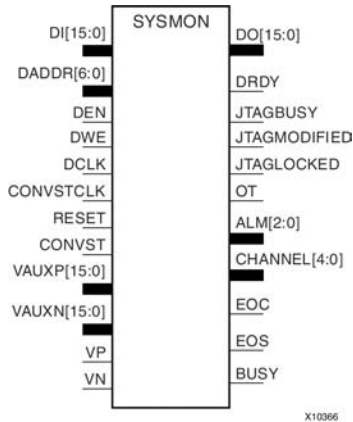
-- End of STARTUP_VIRTEX6_inst instantiation
```

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

: 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	タイプ	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
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	タイプ	値	デフォルト	説明
INIT_40	16 進数	16'h0000 ~ 16'hffff	16'h0000	コンフィギュレーション レジスタ 0
INIT_41	16 進数	16'h0000 ~ 16'hffff	16'h0000	コンフィギュレーション レジスタ 1
INIT_42	16 進数	16'h0000 ~ 16'hffff	16'h0800	コンフィギュレーション レジスタ 2
INIT_43	16 進数	16'h0000 ~ 16'hffff	16'h0000	テスト レジスタ 0
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

属性	タイプ	値	デフォルト	説明
INIT_4E	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 6
INIT_4F	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 7
INIT_50	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 0
INIT_51	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 1
INIT_52	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 2
INIT_53	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 3
INIT_54	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 4
INIT_55	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 5
INIT_56	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 6
INIT_57	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 7
SIM_DEVICE	文字列	VIRTEX5、 VIRTEX6	VIRTEX5	シミュレーション用のターゲット デバイス ファミリ
SIM_MONITOR_FILE	文字列	0 ビット文字列	design.txt	シミュレーション アナログ入力ファイル

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

SYSMON_inst : SYSMON
generic map (
    INIT_40 => X"0000",      -- Configuration register 0
    INIT_41 => X"0000",      -- Configuration register 1
    INIT_42 => X"0800",      -- Configuration register 2
    INIT_43 => X"0000",      -- Test register 0
    INIT_44 => X"0000",      -- Test register 1
    INIT_45 => X"0000",      -- Test register 2
    INIT_46 => X"0000",      -- Test register 3
    INIT_47 => X"0000",      -- Test register 4
    INIT_48 => X"0000",      -- Sequence register 0
    INIT_49 => X"0000",      -- Sequence register 1
    INIT_4A => X"0000",      -- Sequence register 2
    INIT_4B => X"0000",      -- Sequence register 3
    INIT_4C => X"0000",      -- Sequence register 4
    INIT_4D => X"0000",      -- Sequence register 5
    INIT_4E => X"0000",      -- Sequence register 6
    INIT_4F => X"0000",      -- Sequence register 7
    INIT_50 => X"0000",      -- Alarm limit register 0
    INIT_51 => X"0000",      -- Alarm limit register 1
    INIT_52 => X"0000",      -- Alarm limit register 2
    INIT_53 => X"0000",      -- Alarm limit register 3
    INIT_54 => X"0000",      -- Alarm limit register 4
    INIT_55 => X"0000",      -- Alarm limit register 5
    INIT_56 => X"0000",      -- Alarm limit register 6
    INIT_57 => X"0000",      -- Alarm limit register 7
    SIM_DEVICE => "VIRTEX5", -- Device family selection
    SIM_MONITOR_FILE => "design.txt" -- Simulation analog entry file
)
port map (
    ALM => ALM,              -- 3-bit output alarm for temp, Vccint and Vccaux
    BUSY => BUSY,            -- 1-bit output ADC busy signal
    CHANNEL => CHANNEL,      -- 5-bit output channel selection
    DO => DO,                -- 16-bit output data bus for dynamic reconfig
    DRDY => DRDY,            -- 1-bit output data ready for dynamic reconfig
    EOC => EOC,              -- 1-bit output end of conversion
    EOS => EOS,              -- 1-bit output end of sequence
    JTAGBUSY => JTAGBUSY,    -- 1-bit output JTAG DRP busy
    JTAGLOCKED => JTAGLOCKED, -- 1-bit output DRP port lock
    JTAGMODIFIED => JTAGMODIFIED, -- 1-bit output JTAG write to DRP
    OT => OT,                -- 1-bit output over temperature alarm
    CONVST => CONVST,        -- 1-bit input convert start
    CONVSTCLK => CONVSTCLK,  -- 1-bit input convert start clock
    DADDR => DADDR,          -- 7-bit input address bus for dynamic reconfig
    DCLK => DCLK,            -- 1-bit input clock for dynamic reconfig
    DEN => DEN,              -- 1-bit input enable for dynamic reconfig
    DI => DI,                -- 16-bit input data bus for dynamic reconfig
    DWE => DWE,              -- 1-bit input write enable for dynamic reconfig
    RESET => RESET,          -- 1-bit input active high reset
    VAUXN => VAUXN,          -- 16-bit input N-side auxiliary analog input
    VAUXP => VAUXP,          -- 16-bit input P-side auxiliary analog input
    VN => VN,                -- 1-bit input N-side analog input
    VP => VP,                -- 1-bit input P-side analog input
);

-- End of SYSMON_inst instantiation

```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

SYSMON_inst : SYSMON
generic map (
    INIT_40 => X"0000",      -- Configuration register 0
    INIT_41 => X"0000",      -- Configuration register 1
    INIT_42 => X"0800",      -- Configuration register 2
    INIT_43 => X"0000",      -- Test register 0
    INIT_44 => X"0000",      -- Test register 1
    INIT_45 => X"0000",      -- Test register 2
    INIT_46 => X"0000",      -- Test register 3
    INIT_47 => X"0000",      -- Test register 4
    INIT_48 => X"0000",      -- Sequence register 0
    INIT_49 => X"0000",      -- Sequence register 1
    INIT_4A => X"0000",      -- Sequence register 2
    INIT_4B => X"0000",      -- Sequence register 3
    INIT_4C => X"0000",      -- Sequence register 4
    INIT_4D => X"0000",      -- Sequence register 5
    INIT_4E => X"0000",      -- Sequence register 6
    INIT_4F => X"0000",      -- Sequence register 7
    INIT_50 => X"0000",      -- Alarm limit register 0
    INIT_51 => X"0000",      -- Alarm limit register 1
    INIT_52 => X"0000",      -- Alarm limit register 2
    INIT_53 => X"0000",      -- Alarm limit register 3
    INIT_54 => X"0000",      -- Alarm limit register 4
    INIT_55 => X"0000",      -- Alarm limit register 5
    INIT_56 => X"0000",      -- Alarm limit register 6
    INIT_57 => X"0000",      -- Alarm limit register 7
    SIM_DEVICE => "VIRTEX5", -- Device family selection
    SIM_MONITOR_FILE => "design.txt" -- Simulation analog entry file
)
port map (
    ALM => ALM,              -- 3-bit output alarm for temp, Vccint and Vccaux
    BUSY => BUSY,            -- 1-bit output ADC busy signal
    CHANNEL => CHANNEL,      -- 5-bit output channel selection
    DO => DO,                -- 16-bit output data bus for dynamic reconfig
    DRDY => DRDY,            -- 1-bit output data ready for dynamic reconfig
    EOC => EOC,              -- 1-bit output end of conversion
    EOS => EOS,              -- 1-bit output end of sequence
    JTAGBUSY => JTAGBUSY,    -- 1-bit output JTAG DRP busy
    JTAGLOCKED => JTAGLOCKED, -- 1-bit output DRP port lock
    JTAGMODIFIED => JTAGMODIFIED, -- 1-bit output JTAG write to DRP
    OT => OT,                -- 1-bit output over temperature alarm
    CONVST => CONVST,        -- 1-bit input convert start
    CONVSTCLK => CONVSTCLK,  -- 1-bit input convert start clock
    DADDR => DADDR,          -- 7-bit input address bus for dynamic reconfig
    DCLK => DCLK,            -- 1-bit input clock for dynamic reconfig
    DEN => DEN,              -- 1-bit input enable for dynamic reconfig
    DI => DI,                -- 16-bit input data bus for dynamic reconfig
    DWE => DWE,              -- 1-bit input write enable for dynamic reconfig
    RESET => RESET,          -- 1-bit input active high reset
    VAUXN => VAUXN,          -- 16-bit input N-side auxiliary analog input
    VAUXP => VAUXP,          -- 16-bit input P-side auxiliary analog input
    VN => VN,                -- 1-bit input N-side analog input
    VP => VP                 -- 1-bit input P-side analog input
);

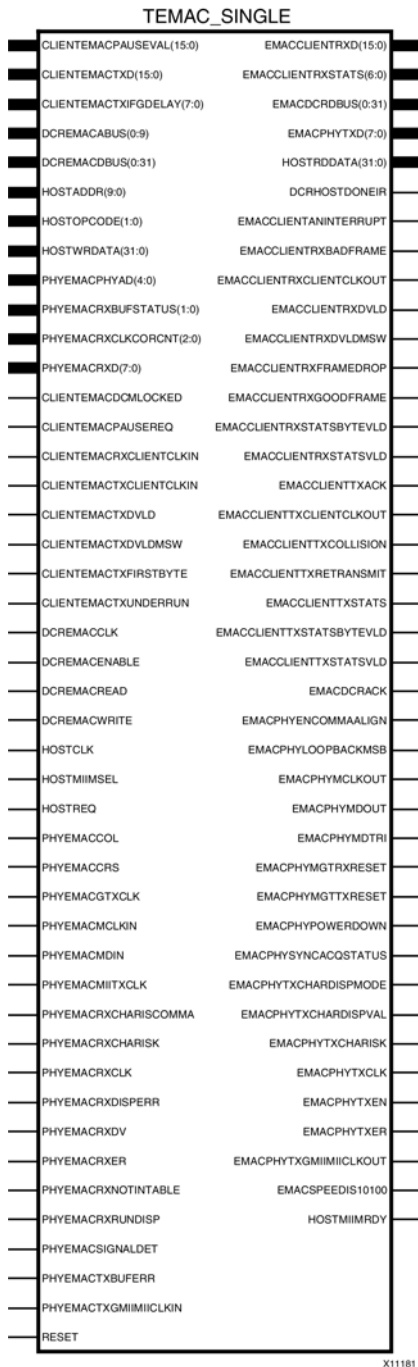
-- End of SYSMON_inst instantiation
```

詳細情報

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

TEMAC_SINGLE

: Tri-mode Ethernet Media Access Controller (MAC)



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

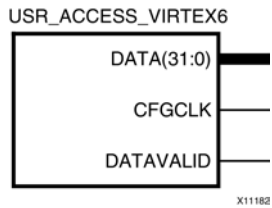
このエレメントをインスタンスエートするには、エンベデッド開発キット (EDK) またはこのエレメントを含む関連コアを使用します。このエレメントは直接インスタンスエートしないでください。

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

: Virtex-6 User Access Register



概要

このデザイン エLEMENTを使用すると、コンフィギュレーション ロジック内の 32 ビットのレジスタにアクセスでき、ビットストリームからのデータを読み出すことができます。たとえば、コンフィギュレーション後に FPGA デザインからビットストリーム格納ソースに保存されたデータにアクセスすることができます。

ポートの説明

ポート名	タイプ	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	有効なデータを含むアクティブ High の DATA ポート

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 つの構文が存在しない場合はコピーし、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 12.2

USR_ACCESS_VIRTEX6_inst : USR_ACCESS_VIRTEX6
port map (
  CFGCLK => CFGCLK,      -- 1-bit Configuration Clock output
  DATA  => DATA,        -- 32-bit Configuration Data output
  DATAVALID => DATAVALID -- 1-bit Active high data valid output
);

-- End of USR_ACCESS_VIRTEX6_inst instantiation
```

Verilog 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、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 12.2

USR_ACCESS_VIRTEX6_inst : USR_ACCESS_VIRTEX6
port map (
    CFGCLK => CFGCLK,          -- 1-bit Configuration Clock output
    DATA => DATA,            -- 32-bit Configuration Data output
    DATAVALID => DATAVALID  -- 1-bit Active high data valid output
);

-- End of USR_ACCESS_VIRTEX6_inst instantiation
```

詳細情報

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