

Spartan-3 ライブラリ ガイド (HDL 用)

UG607 (v14.1) 2012 年 4 月 24 日



Notice of Disclaimer

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

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

本資料は英語版 (v.14.1) を翻訳したもので、内容に相違が生じる場合には原文を優先します。

資料によっては英語版の更新に対応していないものがあります。

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

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

概要

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

このマニュアルには、次の内容が含まれます。

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

デザイン エLEMENT

このバージョンのライブラリ ガイドでは、このアーキテクチャのザイリンクス ユニファイド ライブラリを構成するプリミティブと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。

プリミティブは、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プロセスを実行すると、変換後のファイルには同一のコンポーネントが含まれます。たとえば、ISERDES_NODELAY という Virtex-5 エLEMENTをユーザープリミティブとしてインスタンス化し、変換 (NGDBuild) を実行すると、ISERDES_NODELAY がそのまま残ります。一方 Virtex-5 デバイスで ISERDES を使用していると、自動的に Virtex-5 用の ISERDES_NODELAY に変換されます。このため、「プリミティブ」の概念は、同じ分野でもユーザーによって異なります。

Xilinx では、さまざまなデバイス アーキテクチャに対応した多数のデザイン エLEMENT (UniMacro およびプリミティブ) を含むソフトウェア ライブラリを提供しています。開発システム ソフトウェアのリリースごとに、新しいデザイン エLEMENT が組み込まれます。このガイドは、デザイン エLEMENTをアーキテクチャごとに説明しているライブラリ ガイド シリーズの 1 つです。

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。

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。

四則演算ファンクション	RAM/ROM
クロック コンポーネント	レジスタおよびラッチ
コンフィギュレーション/BSCAN コンポーネント	シフト レジスタ
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

デザイン エLEMENT	説明
MULT18X18	プリミティブ : 18 x 18 Signed Multiplier
MULT18X18S	プリミティブ : 18 x 18 Signed Multiplier -- Registered Version

クロック コンポーネント

デザイン エLEMENT	説明
BUFG	プリミティブ : Global Clock Buffer
BUFGCE	プリミティブ : Global Clock Buffer with Clock Enable
BUFGCE_1	プリミティブ : Global Clock Buffer with Clock Enable and Output State 1
BUFGMUX	プリミティブ : Global Clock MUX Buffer
BUFGMUX_1	プリミティブ : Global Clock MUX Buffer with Output State 1
DCM	プリミティブ : Digital Clock Manager

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_SPARTAN3	プリミティブ : Spartan®-3 and Spartan-3E JTAG Boundary Scan Logic Access Circuit
CAPTURE_SPARTAN3	プリミティブ : Spartan®-3 Register State Capture for Bitstream Readback
STARTUP_SPARTAN3	プリミティブ : Spartan®-3 User Interface to Global Clock, Reset, and 3-State Controls

I/O コンポーネント

デザイン エLEMENT	説明
IBUF	プリミティブ : Input Buffer
IBUFDS	プリミティブ : Differential Signaling Input Buffer
IBUFG	プリミティブ : Dedicated Input Clock Buffer
IBUFGDS	プリミティブ : Differential Signaling Dedicated Input Clock Buffer and Optional Delay
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
KEEPER	プリミティブ : KEEPER Symbol
OBUF	プリミティブ : Output Buffer
OBUFDS	プリミティブ : Differential Signaling Output Buffer
OBUFT	プリミティブ : 3-State Output Buffer with Active Low Output Enable
OBUFTDS	プリミティブ : 3-State Output Buffer with Differential Signaling, Active-Low Output Enable
PULLDOWN	プリミティブ : Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs
PULLUP	プリミティブ : Resistor to VCC for Input PADS, Open-Drain, and 3-State Outputs

RAM/ROM

デザイン エLEMENT	説明
RAM16X1D	プリミティブ : 16-Deep by 1-Wide Static Dual Port Synchronous RAM
RAM16X1D_1	プリミティブ : 16-Deep by 1-Wide Static Dual Port Synchronous RAM with Negative-Edge Clock
RAM16X1S	プリミティブ : 16-Deep by 1-Wide Static Synchronous RAM
RAM16X1S_1	プリミティブ : 16-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAM16X2S	プリミティブ : 16-Deep by 2-Wide Static Synchronous RAM
RAM32X1S	プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM

デザイン エLEMENT	説明
RAM32X1S_1	プリミティブ：32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAM32X2S	プリミティブ：32-Deep by 2-Wide Static Synchronous RAM
RAM64X1S	プリミティブ：64-Deep by 1-Wide Static Synchronous RAM
RAM64X1S_1	プリミティブ：64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAMB16_S1	プリミティブ：16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 1-bit Port
RAMB16_S1_S1	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit Ports
RAMB16_S1_S18	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 18-bit Ports
RAMB16_S1_S2	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 2-bit Ports
RAMB16_S1_S36	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 36-bit Ports
RAMB16_S1_S4	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 4-bit Ports
RAMB16_S1_S9	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 9-bit Ports
RAMB16_S18	プリミティブ：16K-bit Data + 2K-bit Parity Memory, Single-Port Synchronous Block RAM with 18-bit Port
RAMB16_S18_S18	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit Ports
RAMB16_S18_S36	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit and 36-bit Ports
RAMB16_S2	プリミティブ：16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 2-bit Port
RAMB16_S2_S2	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit Ports
RAMB16_S2_S18	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 18-bit Ports
RAMB16_S2_S36	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 36-bit Ports
RAMB16_S2_S4	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 4-bit Ports
RAMB16_S2_S9	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 9-bit Ports
RAMB16_S36	プリミティブ：16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 36-bit Port
RAMB16_S36_S36	プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with Two 36-bit Ports
RAMB16_S4	プリミティブ：16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 4-bit Port

デザイン エLEMENT	説明
RAMB16_S4_S18	プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 18-bit Ports
RAMB16_S4_S36	プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 36-bit Ports
RAMB16_S4_S4	プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit Ports
RAMB16_S4_S9	プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 9-bit Ports
RAMB16_S9	プリミティブ : 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 9-bit Port
RAMB16_S9_S18	プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 18-bit Ports
RAMB16_S9_S36	プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 36-bit Ports
RAMB16_S9_S9	プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit Ports
ROM128X1	プリミティブ : 128-Deep by 1-Wide ROM
ROM16X1	プリミティブ : 16-Deep by 1-Wide ROM
ROM256X1	プリミティブ : 256-Deep by 1-Wide ROM
ROM32X1	プリミティブ : 32-Deep by 1-Wide ROM
ROM64X1	プリミティブ : 64-Deep by 1-Wide ROM

レジスタおよびラッチ

デザイン エLEMENT	説明
FDCE	プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Clear
FDCPE	プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Preset and Clear
FDRSE	プリミティブ : D Flip-Flop with Synchronous Reset and Set and Clock Enable
IFDDRCPE	プリミティブ : Dual Data Rate Input D Flip-Flop with Clock Enable and Asynchronous Preset and Clear
IFDDRSE	プリミティブ : Dual Data Rate Input D Flip-Flop with Synchronous Reset and Set and Clock Enable
LDCE	プリミティブ : Transparent Data Latch with Asynchronous Clear and Gate Enable
LDCPE	プリミティブ : Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable
LDPE	プリミティブ : Transparent Data Latch with Asynchronous Preset and Gate Enable
OFDDRCPE	プリミティブ : Dual Data Rate Output D Flip-Flop with Clock Enable and Asynchronous Preset and Clear
OFDDRSE	プリミティブ : Dual Data Rate Output D Flip-Flop with Synchronous Reset and Set and Clock Enable

デザイン エLEMENT	説明
OFDDRTCPE	プリミティブ：Dual Data Rate D Flip-Flop with Active-Low 3-State Output Buffer, Clock Enable, and Asynchronous Preset and Clear
OFDDRTRSE	プリミティブ：Dual Data Rate D Flip-Flop with Active-Low 3-State Output Buffer, Synchronous Reset and Set, and Clock Enable

シフトレジスタ

デザイン エLEMENT	説明
SRL16	プリミティブ：16-Bit Shift Register Look-Up Table (LUT)
SRL16_1	プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock
SRL16E	プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Clock Enable
SRL16E_1	プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock and Clock Enable
SRLC16	プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Carry
SRLC16_1	プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Carry and Negative-Edge Clock
SRLC16E	プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Carry and Clock Enable
SRLC16E_1	プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Carry, Negative-Edge Clock, and Clock Enable

スライス/CLB プリミティブ

デザイン エLEMENT	説明
BUFCF	プリミティブ：Fast Connect Buffer
LUT1	プリミティブ：1-Bit Look-Up Table with General Output
LUT1_D	プリミティブ：1-Bit Look-Up Table with Dual Output
LUT1_L	プリミティブ：1-Bit Look-Up Table with Local Output
LUT2	プリミティブ：2-Bit Look-Up Table with General Output
LUT2_D	プリミティブ：2-Bit Look-Up Table with Dual Output
LUT2_L	プリミティブ：2-Bit Look-Up Table with Local Output
LUT3	プリミティブ：3-Bit Look-Up Table with General Output
LUT3_D	プリミティブ：3-Bit Look-Up Table with Dual Output
LUT3_L	プリミティブ：3-Bit Look-Up Table with Local Output
LUT4	プリミティブ：4-Bit Look-Up Table with General Output
LUT4_D	プリミティブ：4-Bit Look-Up Table with Dual Output
LUT4_L	プリミティブ：4-Bit Look-Up Table with Local Output
MULT_AND	プリミティブ：Fast Multiplier AND

デザイン エLEMENT	説明
MUXCY	プリミティブ : 2-to-1 Multiplexer for Carry Logic with General Output
MUXCY_D	プリミティブ : 2-to-1 Multiplexer for Carry Logic with Dual Output
MUXCY_L	プリミティブ : 2-to-1 Multiplexer for Carry Logic with Local Output
MUXF5	プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output
MUXF5_D	プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF5_L	プリミティブ : 2-to-1 Look-Up Table Multiplexer with Local Output
MUXF6	プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output
MUXF6_D	プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF6_L	プリミティブ : 2-to-1 Look-Up Table Multiplexer 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
XORCY	プリミティブ : XOR for Carry Logic with General Output
XORCY_D	プリミティブ : XOR for Carry Logic with Dual Output
XORCY_L	プリミティブ : XOR for Carry Logic with Local Output

デザイン エLEMENT

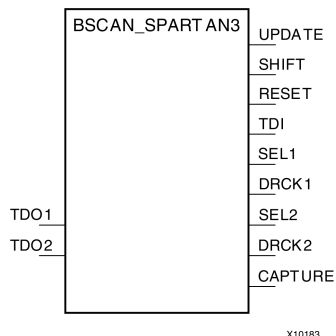
このセクションでは、このアーキテクチャで利用できるデザイン エLEMENTについて説明します。デザイン エLEMENTは、アルファベット順に並べられています。

各ライブラリ エLEMENTについて、次の情報を示します。

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

BSCAN_SPARTAN3

プリミティブ：Spartan®-3 and Spartan-3E JTAG Boundary Scan Logic Access Circuit



概要

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

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

ポートの説明

ポート名	方向	幅	機能
TDI	出力	1	FPGA の TDI 入力ピンと同じ値を出力します。
DRCK1、DRK2	出力	1	JTAG USER 命令が読み込まれるとアクティブになり、JTAG TAP が SHIFT-DR ステートになると TCK ピンと同じ値を出力します。DRK1 は USER1 ロジックに適用され、DRK2 は USER2 に適用されます。
RESET	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP が TEST-LOGIC-RESET ステートになると High にアサートされます。
SEL1、SEL2	出力	1	JTAG 命令レジスタに USER1 または USER2 命令が読み込まれたことを示します。UPDATE-IR ステートになるとアクティブになり、新しい命令が読み込まれるまでアクティブのままになります。
SHIFT	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP が SHIFT-DR ステートになると High にアサートされます。
CAPTURE	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP が CAPTURE-DR ステートになると High にアサートされます。
UPDATE	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP が UPDATE-DR ステートになると High にアサートされます。
TDO1、TDO2	入力	1	USER1 または USER2 命令が読み込まれるとアクティブになり、外部 JTAG TDO ピンにコンポーネントの TDO1 (USER1) または TDO2 (USER2) ピンへのデータ入力の値が反映されます。

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 (インスタンスレーション)

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

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

-- BSCAN_SPARTAN3: Boundary Scan primitive for connecting internal logic to
--                  JTAG interface.
--                  Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

BSCAN_SPARTAN3_inst : BSCAN_SPARTAN3
port map (
    CAPTURE => CAPTURE, -- CAPTURE output from TAP controller
    DRCK1 => DRCK1,      -- Data register output for USER1 functions
    DRCK2 => DRCK2,      -- Data register output for USER2 functions
    RESET => RESET,      -- Reset output from TAP controller
    SEL1 => SEL1,         -- USER1 active output
    SEL2 => SEL2,         -- USER2 active output
    SHIFT => SHIFT,      -- SHIFT output from TAP controller
    TDI => TDI,           -- TDI output from TAP controller
    UPDATE => UPDATE,     -- UPDATE output from TAP controller
    TDO1 => TDO1,         -- Data input for USER1 function
    TDO2 => TDO2          -- Data input for USER2 function
);

-- End of BSCAN_SPARTAN3_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// BSCAN_SPARTAN3: Boundary Scan primitive for connecting internal logic to
//                  JTAG interface.
//                  Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

BSCAN_SPARTAN3 BSCAN_SPARTAN3_inst (
    .CAPTURE(CAPTURE), // CAPTURE output from TAP controller
    .DRCK1(DRCK1),      // Data register output for USER1 functions
    .DRCK2(DRCK2),      // Data register output for USER2 functions
    .RESET(RESET),      // Reset output from TAP controller
    .SEL1(SEL1),         // USER1 active output
    .SEL2(SEL2),         // USER2 active output
    .SHIFT(SHIFT),      // SHIFT output from TAP controller
    .TDI(TDI),           // TDI output from TAP controller
    .UPDATE(UPDATE),    // UPDATE output from TAP controller
    .TDO1(TDO1),         // Data input for USER1 function
    .TDO2(TDO2)          // Data input for USER2 function
);

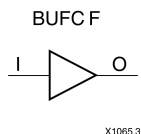
// End of BSCAN_SPARTAN3_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

BUFCF

プリミティブ：Fast Connect Buffer



概要

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

デザインの入力方法

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

VHDL 記述（インスタンス化）

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

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

-- BUFCF: Fast connect buffer used to connect the outputs of the LUTs
--         and some dedicated logic directly to the input of another LUT.
--         Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of BUFCF_inst instantiation
```

Verilog 記述（インスタンス化）

```
// BUFCF: Fast connect buffer used to connect the outputs of the LUTs
//         and some dedicated logic directly to the input of another LUT.
//         Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

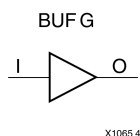
// End of BUFCF_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

BUFG

プリミティブ：Global Clock Buffer



概要

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- BUFG: Global Clock Buffer (source by an internal signal)
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of BUFG_inst instantiation
```


Verilog 記述 (インスタンスレーション)

```
// BUFG: Global Clock Buffer (source by an internal signal)
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

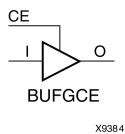
// End of BUFG_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

BUFGCE

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



概要

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

論理表

入力		出力
I	CE	O
X	0	0
I	1	I

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- BUFGCE: Global Clock Buffer with Clock Enable (active high)
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of BUFGCE_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// BUFGCE: Global Clock Buffer with Clock Enable (active high)
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

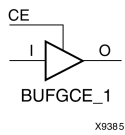
// End of BUFGCE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

BUFGCE_1

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



概要

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

論理表

入力		出力
I	CE	O
X	0	1
I	1	I

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- BUFGCE_1: Global Clock Buffer with Clock Enable (active low)
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of BUFGCE_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// BUFGCE_1: Global Clock Buffer with Clock Enable (active low)
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

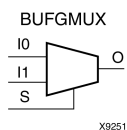
// End of BUFGCE_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

BUFGMUX

プリミティブ：Global Clock MUX Buffer



概要

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

BUFGMUX および BUFGMUX_1 では、S の値が変化した後クロックが切り替わるまでに保持される出力ステートが異なります。BUFGMUX は出力ステートが 0 に、BUFGMUX_1 は出力ステートが 1 に保持されます。

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

論理表

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- BUFGMUX: Global Clock Buffer 2-to-1 MUX
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of BUFGMUX_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// BUFGMUX: Global Clock Buffer 2-to-1 MUX
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

BUFGMUX BUFGMUX_inst (
    .O(O),      // Clock MUX output
    .I0(I0),    // Clock0 input
    .I1(I1),    // Clock1 input
    .S(S)       // Clock select input
);

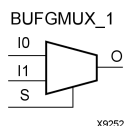
// End of BUFGMUX_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

BUFGMUX_1

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



概要

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

このデザイン エLEMENTと BUFGMUX では、S の値が変化した後クロックが切り替わるまでに保持される出力ステートが異なります。BUFGMUX は出力ステートが 0 に、BUFGMUX_1 は出力ステートが 1 に保持されます。

論理表

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- BUFGMUX_1: Global Clock Buffer 2-to-1 MUX (inverted select)
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

BUFGMUX_1_inst : BUFGMUX_1
port map (
    O => O,      -- Clock MUX output
    I0 => I0,     -- Clock0 input
    I1 => I1,     -- Clock1 input
    S => S       -- Clock select input
);

-- End of BUFGMUX_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// BUFGMUX_1: Global Clock Buffer 2-to-1 MUX (inverted select)
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

BUFGMUX_1 BUFGMUX_1_inst (
    .O(O),       // Clock MUX output
    .I0(I0),     // Clock0 input
    .I1(I1),     // Clock1 input
    .S(S)        // Clock select input
);

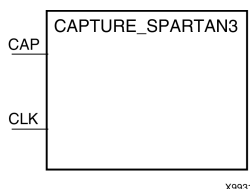
// End of BUFGMUX_1_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

CAPTURE_SPARTAN3

プリミティブ：Spartan®-3 Register State Capture for Bitstream Readback



概要

このデザイン エレメントは、レジスタ (フリップフロップとラッチ) 情報のキャプチャ方法およびそのタイミングを制御します。リードバック機能は、専用のコンフィギュレーション ポート命令により提供されます。このエレメントを使用しない場合は、データはコンフィギュレーション クロックに同期してリードバックされます。このエレメントでは、レジスタ (フリップフロップとラッチ) の値のみをキャプチャできます。LUT RAM、SRL、ブロック RAM の値もリードバックされますが、キャプチャできません。

CAP 信号を High にアサートすると、次にクロックが Low から High に切り替わるときにデバイス内のレジスタがキャプチャされます。デフォルトでは、トリガー (CAP をアサートしているときの CLK の遷移) のたびにデータがキャプチャされます。リードバック処理を 1 回のデータ キャプチャだけに制限するには、このエレメントに ONESHOT=TRUE 属性を追加します。

ポートの説明

ポート名	方向	幅	機能
CAP	入力	1	リードバック キャプチャトリガー
CLK	入力	1	リードバック キャプチャ クロック

デザインの入力方法

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

正しく動作するように、入力および出力をすべてデザインに接続します。

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- CAPTURE_SPARTAN3: Register State Capture for Bitstream Readback
--                      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

CAPTURE_SPARTAN3_inst : CAPTURE_SPARTAN3
port map (
    CAP => CAP,    -- Capture input
    CLK => CLK     -- Clock input
);

-- End of CAPTURE_SPARTAN3_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// CAPTURE_SPARTAN3: Register State Capture for Bitstream Readback
//                      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

CAPTURE_SPARTAN3 CAPTURE_SPARTAN3_inst (
    .CAP(CAP),    // Capture input
    .CLK(CLK)     // Clock input
);

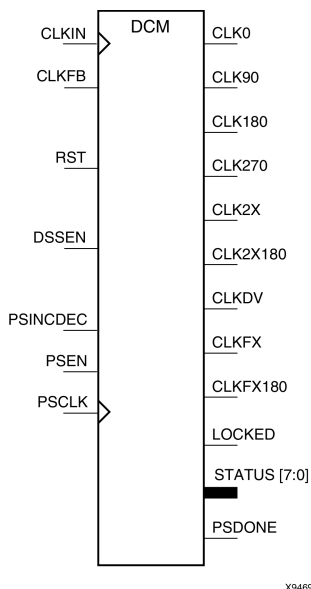
// End of CAPTURE_SPARTAN3_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

DCM

プリミティブ : Digital Clock Manager



概要

このデザイン エLEMENTは、さまざまな機能を備えたデジタル クロック マネージャーで、クロック遅延ロック ループ (DLL)、デジタル周波数合成 (DFS)、デジタル位相シフト (DPS)、デジタル スペクトラム拡散 (DSS) といった機能をインプリメントできます。

メモ: 未使用の入力はすべて Low にしておく必要がありますが、未使用の入力は自動的に Low になります。

クロック遅延ロック ループ (DLL)

Spartan®-3 デバイスの場合、DCM にはクロック スキューを最小限にするクロック遅延ロック ループ (DLL) が含まれます。DCM は、フィードバック クロック入力 (CLKFB) のクロック信号と入力クロック (CLKIN) のクロック信号を同期化させます。この 2 つの信号の位相が一致すると、ロック出力 (LOCKED) が High になります。両信号の立ち上がりエッジの時間差が特定の範囲内 (pn) であれば、これらの信号は同相であると見なされます。

DLL には、2 つの周波数モードがあります。デフォルトでは、DLL_FREQUENCY_MODE 属性は Low に設定されています。Low の場合、CLKIN 入力のクロック信号の周波数を Low (DLL_CLKIN_MIN_LF ~ DLL_CLKIN_MAX_LF) の範囲内にする必要があり、使用できる出力は、CLK0、CLK90、CLK180、CLK270、CLK2X、CLKDV、CLK2X180 です。

High に設定する場合は、CLKIN 入力のクロック信号の周波数を High (DLL_CLKIN_MIN_HF ~ DLL_CLKIN_MAX_HF) の範囲内にする必要があり、使用できる出力は CLK0、CLK180、CLKDV 出力のみです。

グローバル クロック ネットワーク上の BUFG (グローバル クロック バッファ) で駆動されるポイントに CLKFB 入力を接続すると、オンチップで同期化されます。DCM の CLKFB 入力に BUFG が接続されている場合、同じ DCM の CLK0 出力または CLK2X 出力のいずれかをソースにしてください。CLKIN 入力は IBUFG の出力に接続し、IBUFG の入力はシステム クロックで駆動されるパッドに接続します。CLKFB 入力を IBUFG の出力に接続し、IBUFG の入力をパッドに接続すると、オフチップで同期化されます。CLK0 出力と CLK2X 出力はどちらも使用できますが、両方同時には使用できません。CLK0 または CLK2X は、OBUF (出力バッファ) の入力に接続してください。CLK_FEEDBACK 属性は、CLKFB 入力のソースを CLK0 出力 (デフォルト) または CLK2X 出力に指定します。

CLK0 出力のデューティサイクルは、50-50 です。ただし、DUTY_CYCLE_CORRECTION 属性を FALSE に設定した場合は、CLKIN 入力と同じデューティサイクルになります。位相が異なる出力 (CLK90、CLK180、CLK270) のデューティサイクルは、CLK0 出力と同じになります。CLK2X、CLK2X180、CLKDV 出力のデューティサイクルは、CLKDV_DIVIDE の値が整数でない場合、および DLL_FREQUENCY_MODE が High の場合以外は、50-50 です。詳細は、『制約ガイド』の「CLKDV_DIVIDE」を参照してください。CLKDV 出力の周波数は、CLKDV_DIVIDE 属性に割り当てる値によって決定されます。

DCM クロック遅延ロック ループの出力

出力	説明
CLK0	周波数が 1 × CLKIN のクロック
CLK180	周波数が 1 × CLK0 のクロック、CLK0 に対して位相を 180 度シフト
CLK270*	周波数が 1 × CLK0 のクロック、CLK0 に対して位相を 270 度シフト
CLK2X*	周波数が 2 × CLK0 のクロック、CLK0 と同相
CLK2X180*	周波数が 2 × CLK0 のクロック、CLK2X に対して位相を 180 度シフト
CLK90*	周波数が 1 × CLK0 のクロック、CLK0 に対して位相を 90 度シフト
CLKDV	周波数が $(1/n) \times \text{CLK0}$ のクロック ($n = \text{CLKDV_DIVIDE}$ 値)、位相は CLK0 と同じ。
LOCKED	使用可能な DCM の機能がすべてロック状態になったことを示します。
* DLL_FREQUENCY_MODE が High に設定されている場合、CLK90、CLK270、CLK2X、CLK2X180 出力は使用できません。	

デジタル周波数合成 (DFS)

CLKFX および CLKFX180 出力に CLKFX_MULTIPLY および CLKFX_DIVIDE 属性を設定すると、CLKIN を通倍または分周する周波数合成という機能を使用できます。DLL の CLKFB 入力にフィードバックを供給すると、CLKFX の CLKFX_MULTIPLY サイクルまたは CLKIN の CLKFX_DIVIDE サイクルごとに CLKFX と CLKIN の位相が一致します。CLKFX の周波数は次の計算式で定義されます。

$$\text{FrequencyCLKFX} = (\text{CLKFX_MULTIPLY_value} / \text{CLKFX_DIVIDE_value}) * \text{FrequencyCLKIN}$$

CLKFX および CLKFX180 は同時に使用できます。CLKFX180 は周波数が 1 × CLKFX で、CLKFX に対して位相が 180 度シフトされています。また、CLKFX および CLKFX180 のデューティサイクルは常に 50-50 です。DFS_FREQUENCY_MODE 属性は、使用可能な入力クロックおよび出力クロックの周波数の範囲を指定します。CLK_FEEDBACK 属性を NONE に設定すると、DCM はデジタル周波数合成モードになり、CLKFX と CLKFX180 が生成されます。このとき、CLKIN に対して位相は調整されません。DCM の DSSEN 入力ピンは、使用せずに未接続の状態にしておいてください。

デジタル位相シフト (DPS)

CLKIN と CLKFB の立ち上がりエッジ間の位相シフト (スキュー) は、PHASE_SHIFT 属性を使用して、CLKIN 周期の一部としてコンフィギュレーションできます。このようにすると、周囲の状況が変化しても、位相シフトは一定になります。PHASE_SHIFT の値の使用は CLKOUT_PHASE_SHIFT 属性により制御します。CLKOUT_PHASE_SHIFT 属性はデフォルトで NONE に設定されており、PHASE_SHIFT 属性の値は無視されます。

CLKIN と CLKFB 間にスキューを発生させると、そのスキュー量だけ DCM 出力クロックすべての位相がシフトします。CLKOUT_PHASE_SHIFT 属性を FIXED に設定すると、コンフィギュレーションの際に CLKIN および CLKFB の立ち上がりエッジに対して PHASE_SHIFT 属性で設定したスキューが使用されます。この場合、スキューは変化しません。CLKOUT_PHASE_SHIFT 属性を VARIABLE に設定すると、コンフィギュレーションで設定されたスキューが開始点として使用され、動作中に PS* 信号を使用してスキューの値をダイナミックに変化させることができます。このデジタル位相シフトの機能は、同期インターフェイスで制御されます。PSEN (位相シフト イネーブル) および PSINCDEC (位相シフト インクリメント/デクリメント) の入力、PSCLK (位相シフト クロック) の立ち上がりエッジに同期して設定されます。PSDONE (位相シフト終了) 出力は PSCLK の立ち上がりエッジに同期しています。同期インターフェイスをインプリメントするため、PSDONE を接続しておく必要があります。CLKIN と CLKFB の立ち上がりエッジ間のスキューは、LOCKED 出力が High になった後ダイナミックに調整できます。PHASE_SHIFT 属性は、デバイスがコンフィギュレーションされたときの初期の位相シフト量を指定します。PSCLK の 1 周期の間 PSEN がアクティブになると、PHASE_SHIFT の値が 1 ユニット分変更されます。PSEN が High の場合、PHASE_SHIFT の値は PSINCDEC が High のときインクリメントされ、PSINCDEC が Low のときデクリメントされます。

インクリメントまたはデクリメントが完了すると、PSCLK の 1 サイクル分 PSDONE 出力が High になります。この後、再び PHASE_SHIFT 値を変更できます。RST (リセット) が High になると、PHASE_SHIFT 属性値がコンフィギュレーションで設定したスキュー値にリセットされます。CLKOUT_PHASE_SHIFT 属性を FIXED または NONE に設定している場合、PSEN、PSINCDEC、PSCLK 入力は GND に接続する必要があります。これらの入力が接続されていない場合、自動的に GND に接続されます。

ステータス ビット

STATUS 出力ビットについては、次の情報を参照してください。

ビット	説明
0	位相シフト オーバーフロー*
1 = PHASE_SHIFT > 255	
1	DLL CLKIN の停止**
1 = CLKIN のトグル停止	
2	DLL CLKFX の停止**
1 = CLKFX のトグル停止	
3	不可
4	不可
5	不可
6	不可
7	不可
* 位相シフト オーバーフローは、位相シフトの遅延ラインの最後に達した場合にも High になります。最大シフト遅延の最新の値については、製品のデータシートを参照してください。	
** DFS 出力 (CLKFX または CLKFX180) のみを使用している場合、このステータス ビットは CLKIN が停止しても High にはなりません。	

LOCKED

LOCKED が High の場合、使用可能な信号はすべてロック状態です。

RST

マスター リセット入力 (RST) は、DCM を初期値 (電源投入時の状態) にリセットします。RST 入力信号は非同期で、有効な CLKIN の 3 サイクル間 High に保つ必要があります。

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- DCM: Digital Clock Manager Circuit
-- Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

DCM_inst : DCM
generic map (
    CLKDV_DIVIDE => 2.0, -- Divide by: 1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
                        -- 7.0,7.5,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0 or 16.0
    CLKFX_DIVIDE => 1, -- Can be any interger from 1 to 32
    CLKFX_MULTIPLY => 4, -- Can be any integer from 1 to 32
    CLKIN_DIVIDE_BY_2 => FALSE, -- TRUE/FALSE to enable CLKIN divide by two feature
    CLKIN_PERIOD => 0.0, -- Specify period of input clock
    CLKOUT_PHASE_SHIFT => "NONE", -- Specify phase shift of NONE, FIXED or VARIABLE
    CLK_FEEDBACK => "1X", -- Specify clock feedback of NONE, 1X or 2X
    DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS", -- SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                           -- an integer from 0 to 15
    DFS_FREQUENCY_MODE => "LOW", -- HIGH or LOW frequency mode for frequency synthesis
    DLL_FREQUENCY_MODE => "LOW", -- HIGH or LOW frequency mode for DLL
    DUTY_CYCLE_CORRECTION => TRUE, -- Duty cycle correction, TRUE or FALSE
    FACTORY_JF => X"C080", -- FACTORY JF Values
    PHASE_SHIFT => 0, -- Amount of fixed phase shift from -255 to 255
    SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
                        -- Design Guide" for details
    STARTUP_WAIT => FALSE) -- Delay configuration DONE until DCM LOCK, TRUE/FALSE
port map (
    CLK0 => CLK0, -- 0 degree DCM CLK ouptput
    CLK180 => CLK180, -- 180 degree DCM CLK output
    CLK270 => CLK270, -- 270 degree DCM CLK output
    CLK2X => CLK2X, -- 2X DCM CLK output
    CLK2X180 => CLK2X180, -- 2X, 180 degree DCM CLK out
    CLK90 => CLK90, -- 90 degree DCM CLK output
    CLKDV => CLKDV, -- Divided DCM CLK out (CLKDV_DIVIDE)
    CLKFX => CLKFX, -- DCM CLK synthesis out (M/D)
    CLKFX180 => CLKFX180, -- 180 degree CLK synthesis out
    LOCKED => LOCKED, -- DCM LOCK status output
    PSDONE => PSDONE, -- Dynamic phase adjust done output
    STATUS => STATUS, -- 8-bit DCM status bits output
    CLKFB => CLKFB, -- DCM clock feedback
    CLKIN => CLKIN, -- Clock input (from IBUFG, BUFG or DCM)
    PSCLK => PSCLK, -- Dynamic phase adjust clock input
```

```

PSEN => PSEN,      -- Dynamic phase adjust enable input
PSINCDEC => PSINCDEC, -- Dynamic phase adjust increment/decrement
RST => RST          -- DCM asynchronous reset input
);

-- End of DCM_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// DCM: Digital Clock Manager Circuit
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

DCM #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .CLKDV_DIVIDE(2.0), // Divide by: 1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
                        //      7.0,7.5,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0 or 16.0
    .CLKFX_DIVIDE(1),  // Can be any integer from 1 to 32
    .CLKFX_MULTIPLY(4), // Can be any integer from 2 to 32
    .CLKIN_DIVIDE_BY_2("FALSE"), // TRUE/FALSE to enable CLKIN divide by two feature
    .CLKIN_PERIOD(0.0), // Specify period of input clock
    .CLKOUT_PHASE_SHIFT("NONE"), // Specify phase shift of NONE, FIXED or VARIABLE
    .CLK_FEEDBACK("1X"), // Specify clock feedback of NONE, 1X or 2X
    .DESKEW_ADJUST("SYSTEM_SYNCHRONOUS"), // SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                         // an integer from 0 to 15
    .DFS_FREQUENCY_MODE("LOW"), // HIGH or LOW frequency mode for frequency synthesis
    .DLL_FREQUENCY_MODE("LOW"), // HIGH or LOW frequency mode for DLL
    .DUTY_CYCLE_CORRECTION("TRUE"), // Duty cycle correction, TRUE or FALSE
    .FACTORY_JF(16'hC080), // FACTORY JF values
    .PHASE_SHIFT(0), // Amount of fixed phase shift from -255 to 255
    .STARTUP_WAIT("FALSE") // Delay configuration DONE until DCM LOCK, TRUE/FALSE
) DCM_inst (
    .CLK0(CLK0), // 0 degree DCM CLK output
    .CLK180(CLK180), // 180 degree DCM CLK output
    .CLK270(CLK270), // 270 degree DCM CLK output
    .CLK2X(CLK2X), // 2X DCM CLK output
    .CLK2X180(CLK2X180), // 2X, 180 degree DCM CLK out
    .CLK90(CLK90), // 90 degree DCM CLK output
    .CLKDV(CLKDV), // Divided DCM CLK out (CLKDV_DIVIDE)
    .CLKFX(CLKFX), // DCM CLK synthesis out (M/D)
    .CLKFX180(CLKFX180), // 180 degree CLK synthesis out
    .LOCKED(LOCKED), // DCM LOCK status output
    .PSDONE(PSDONE), // Dynamic phase adjust done output
    .STATUS(STATUS), // 8-bit DCM status bits output
    .CLKFB(CLKFB), // DCM clock feedback
    .CLKIN(CLKIN), // Clock input (from IBUFG, BUFG or DCM)
    .PSCLK(PSCLK), // Dynamic phase adjust clock input
    .PSEN(PSEN), // Dynamic phase adjust enable input
    .PSINCDEC(PSINCDEC), // Dynamic phase adjust increment/decrement
    .RST(RST) // DCM asynchronous reset input
);

// End of DCM_inst instantiation

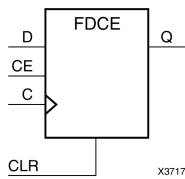
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

FDCE

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



概要

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
--       Clock Enable (posedge clk).
--       Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of FDCE_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
//       Clock Enable (posedge clk).
//       Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

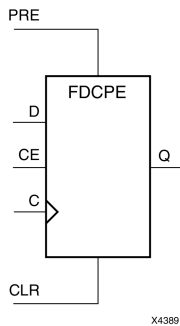
// End of FDCE_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

FDCPE

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



概要

このデザイン エLEMENTは、データ (D)、クロック イネーブル (CE)、非同期プリセット (PRE)、非同期クリア (CLR) の各入力とデータ出力 (Q) がある単一の D タイプ フリップフロップです。非同期 PRE が High になると Q 出力が High にセットされ、CLR が High になると出力が Low にリセットされます (CLR 入力の方が PRE 入力よりも優先される)。PRE と CLR が Low で CE が High の場合、クロック (C) が Low から High に切り替わる時に D 入力の値がフリップフロップにロードされます。CE が Low の場合クロック遷移は無視され、以前の値が保持されます。FDCPE は通常、スライスまたは IOB レジスタとしてインプリメントされます。

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

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

論理表

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

ポートの説明

ポート名	方向	幅	機能
Q	出力	1	データ出力
C	入力	1	クロック入力
CE	入力	1	クロック イネーブル入力
CLR	入力	1	非同期クリア入力
D	入力	1	データ入力
PRE	入力	1	非同期セット入力

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
--       Clock Enable (posedge clk).
--       Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of FDCPE_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
//       Clock Enable (posedge clk).
//       Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

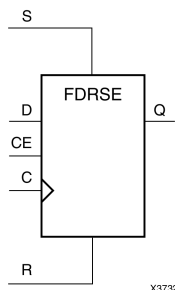
// End of FDCPE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

FDRSE

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



概要

FDRSE は、同期リセット (R)、同期セット (S)、クロック イネーブル (CE) の各入力とデータ出力 (Q) がある単一の D タイプ フリップフロップです。同期リセット入力 (R) が High になると、ほかの入力は無視され、クロック (C) が Low から High に切り替わる時に出力 (Q) が Low にリセットされます (リセットがセットよりも優先される)。セット入力 (S) が High、R が Low の場合、クロック (C) が Low から High に切り替わる時にフリップフロップがセットされ、出力が High になります。R と S が Low、CE が High の場合、クロックが Low から High に切り替わる時に D 入力の値がフリップフロップにロードされます。

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
--       Clock Enable (posedge clk).
--       Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of FDRSE_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//       Clock Enable (posedge clk).
//       Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

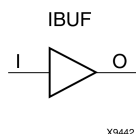
// End of FDRSE_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

IBUF

プリミティブ：Input Buffer



概要

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

ポートの説明

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

デザインの入力方法

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

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- IBUF: Single-ended Input Buffer
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

IBUF_inst : IBUF
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Buffer output
    I => I       -- Buffer input (connect directly to top-level port)
);

-- End of IBUF_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// IBUF: Single-ended Input Buffer
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

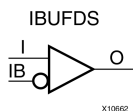
// End of IBUF_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

IBUFDS

プリミティブ : Differential Signaling Input Buffer



概要

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

論理表

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

ポートの説明

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

デザインの入力方法

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

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- IBUFDS: Differential Input Buffer
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

IBUFDS_inst : IBUFDS
generic map (
  CAPACITANCE => "DONT_CARE", -- "LOW", "NORMAL", "DONT_CARE" (Spartan-3 only)
  DIFF_TERM => FALSE, -- Differential Termination
  IOSTANDARD => "DEFAULT")
port map (
  O => O, -- Buffer output
  I => I, -- Diff_p buffer input (connect directly to top-level port)
  IB => IB -- Diff_n buffer input (connect directly to top-level port)
);

-- End of IBUFDS_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// IBUFDS: Differential Input Buffer
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

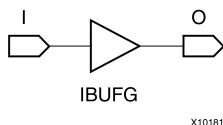
// End of IBUFDS_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

IBUFG

プリミティブ：Dedicated Input Clock Buffer



概要

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述（インスタンス化）

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

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

-- IBUFG: Single-ended global clock input buffer
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

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

Verilog 記述 (インスタンス化)

```
// IBUFG: Single-ended global clock input buffer
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

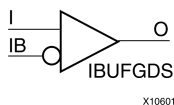
// End of IBUFG_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

IBUFGDS

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



概要

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

論理表

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

ポートの説明

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

デザインの入力方法

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

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- IBUGDS: Differential Global Clock Input Buffer
-- All Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of IBUGDS_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// IBUGDS: Differential Global Clock Input Buffer
// Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

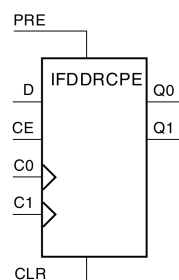
// End of IBUGDS_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

IFDDRCPE

プリミティブ : Dual Data Rate Input D Flip-Flop with Clock Enable and Asynchronous Preset and Clear



X9398

概要

このデザイン エLEMENTは、クロック イネーブル (CE)、非同期プリセット (PRE)、非同期クリア (CLR) があるデュアル データレート (DDR) の入力 D フリップフロップです。このELEMENTは 1 つの入力バッファと、2 つの同種フリップフロップ (FDCPE) から構成されます。

非同期プリセット (PRE) が High で 非同期クリア (CLR) が Low の場合、出力 Q0 と Q1 は High にセットされます。CLR が High になると、出力は Low にリセットされます。PRE と CLR が Low、CE が High の場合、入力 D の値はクロック C0 が Low から High に切り替わる時に出力 Q0、クロック C1 が Low から High に切り替わる時に出力 Q1 にロードされます。

このデザイン ELEMENTのコンポーネントでは、INIT 属性は使用できません。

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

論理表

入力						出力	
C0	C1	CE	D	CLR	PRE	Q0	Q1
X	X	X	X	1	0	0	0
X	X	X	X	0	1	1	1
X	X	X	X	1	1	0	0
X	X	0	X	0	0	変化なし	変化なし
↑	X	1	D	0	0	D	変化なし
X	↑	1	D	0	0	変化なし	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

VHDL 記述 (インスタンスレーション)

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

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

-- IFDDRCPE: Double Data Rate Input Register with Async. Clear, Async. Preset
--           and Clock Enable.
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

IFDDRCPE_inst : IFDDRCPE
port map (
    Q0 => Q0,      -- Posedge data output
    Q1 => Q1,      -- Negedge data output
    C0 => C0,      -- 0 degree clock input
    C1 => C1,      -- 180 degree clock input
    CE => CE,      -- Clock enable input
    CLR => CLR,    -- Asynchronous reset input
    D => D,        -- Data input (connect directly to top-level port)
    PRE => PRE     -- Asynchronous preset input
);

-- End of IFDDRCPE_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// IFDDRCPE: Double Data Rate Input Register with Async. Clear, Async. Preset
//           and Clock Enable.
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

IFDDRCPE IFDDRCPE_inst (
    .Q0(Q0),      // Posedge data output
    .Q1(Q1),      // Negedge data output
    .C0(C0),      // 0 degree clock input
    .C1(C1),      // 180 degree clock input
    .CE(CE),      // Clock enable input
    .CLR(CLR),    // Asynchronous reset input
    .D(D),        // Data input (connect directly to top-level port)
    .PRE(PRE)     // Asynchronous preset input
);

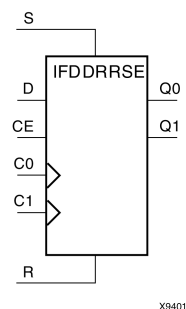
// End of IFDDRCPE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

IFDDRSE

プリミティブ : Dual Data Rate Input D Flip-Flop with Synchronous Reset and Set and Clock Enable



概要

このデザイン エレメントは、同期リセット (R) と同期セット (S) およびクロック イネーブル (CE) があるデュアル データ レート (DDR) の入力 D フリップフロップです。このエレメントは 1 つの入力バッファと、2 つの同種フリップフロップ (FDRSE) から構成されます。

入力 C0 および出力 Q0 では、リセット (R) が優先されます。入力 R が High の場合、クロック Q0 が Low から High に切り替わるときに、出力 Q0 が Low にリセットされます。S が High、R が Low になっている場合、クロック C0 が Low から High に切り替わるときに、出力 Q0 が High にセットされます。入力 C1 および出力 Q1 では、セット (S) が優先されます。入力 R が High の場合、クロック C1 が Low から High に切り替わるときに、出力 Q1 が Low にリセットされます。S が High、R が Low になっている場合、クロック C1 が Low から High に切り替わるときに、出力 Q1 が High になります。

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

このデザイン エレメントでは、INIT 属性は使用できません。

論理表

入力						出力	
C0	C1	CE	D	R	S	Q0	Q1
↑	X	X	X	1	0	0	変化なし
↑	X	X	X	0	1	1	変化なし
↑	X	X	X	1	1	0	変化なし
X	↑	X	X	1	0	変化なし	0
X	↑	X	X	0	1	変化なし	1
X	↑	X	X	1	1	変化なし	0
X	X	0	X	0	0	変化なし	変化なし
↑	X	1	D	0	0	D	変化なし
X	↑	1	D	0	0	変化なし	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

VHDL 記述 (インスタンス化)

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

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

-- IFDDRSE: Double Data Rate Input Register with Sync. Clear, Sync. Preset
--           and Clock Enable.
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

IFDDRSE_inst : IFDDRSE
port map (
  Q0 => Q0,    -- Posedge data output
  Q1 => Q1,    -- Negedge data output
  C0 => C0,    -- 0 degree clock input
  C1 => C1,    -- 180 degree clock input
  CE => CE,    -- Clock enable input
  D => D,      -- Data input (connect directly to top-level port)
  R => R,      -- Synchronous reset input
  S => S       -- Synchronous preset input
);

-- End of IFDDRSE_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// IFDDRSE: Double Data Rate Input Register with Sync. Clear, Sync. Preset
//           and Clock Enable.
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

IFDDRSE IFDDRSE_inst (
  .Q0(Q0),    // Posedge data output
  .Q1(Q1),    // Negedge data output
  .C0(C0),    // 0 degree clock input
  .C1(C1),    // 180 degree clock input
  .CE(CE),    // Clock enable input
  .D(D),      // Data input (connect directly to top-level port)
  .R(R),      // Synchronous reset input
  .S(S)       // Synchronous preset input
);

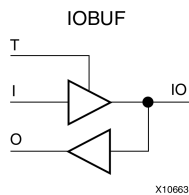
// End of IFDDRSE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

IOBUF

プリミティブ：Bi-Directional Buffer



概要

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

論理表

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- IOBUF: Single-ended Bi-directional Buffer
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of IOBUF_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// IOBUF: Single-ended Bi-directional Buffer
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

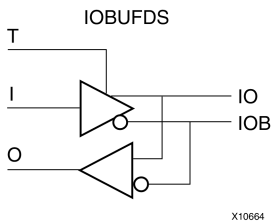
// End of IOBUF_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

IOBUFDS

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



概要

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

論理表

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- IOBUFDS: Differential Bi-directional Buffer
--           Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 14.1

IOBUFDS_inst : IOBUFDS
generic map (
  IOSTANDARD => "DEFAULT")
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)

```
// IOBUFDS: Differential Bi-directional Buffer
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

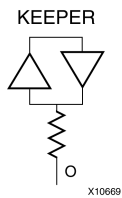
// End of IOBUFDS_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

KEEPER

プリミティブ：KEEPER Symbol



概要

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

ポートの説明

ポート名	方向	幅	機能
O	出力	1 ビット	キーパー出力

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- KEEPER: I/O Buffer Weak Keeper
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of KEEPER_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// KEEPER: I/O Buffer Weak Keeper
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

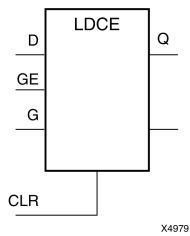
// End of KEEPER_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

LDCE

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



概要

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- LDCE: Transparent latch with Asynchronous Reset and
--       Gate Enable.
--       Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LDCE_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// LDCE: Transparent latch with Asynchronous Reset and Gate Enable.
//       Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

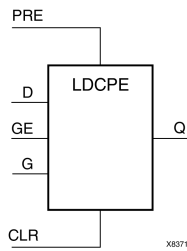
// End of LDCE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

LDCPE

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



概要

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

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

論理表

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- LDCPE: Transparent latch with Asynchronous Reset, Preset and
--       Gate Enable.
--       Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LDCPE_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// LDCPE: Transparent latch with Asynchronous Reset, Preset and
//       Gate Enable.
//       Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

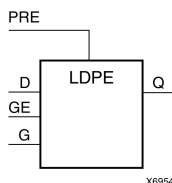
// End of LDCPE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

LDPE

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



概要

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- LDPE: Transparent latch with Asynchronous Set and
--       Gate Enable.
--       Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LDPE_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// LDPE: Transparent latch with Asynchronous Preset and Gate Enable.
//       Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

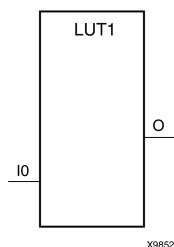
// End of LDPE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

LUT1

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



概要

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

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

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

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- LUT1: 1-input Look-Up Table with general output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// LUT1: 1-input Look-Up Table with general output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

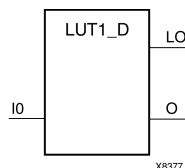
// End of LUT1_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

LUT1_D

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



概要

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

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

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

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- LUT1_D: 1-input Look-Up Table with general and local outputs
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT1_D_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// LUT1_D: 1-input Look-Up Table with general and local outputs
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

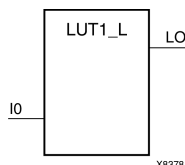
// End of LUT1_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

LUT1_L

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



概要

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

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

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

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- LUT1_L: 1-input Look-Up Table with local output
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT1_L_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// LUT1_L: 1-input Look-Up Table with local output
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

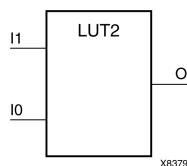
// End of LUT1_L_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

LUT2

プリミティブ : 2-Bit Look-Up Table with General Output



概要

このデザイン エレメントは、汎用出力 (O) を持つ 2 ビットのルックアップ テーブル (LUT) です。

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

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

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- LUT2: 2-input Look-Up Table with general output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT2_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// LUT2: 2-input Look-Up Table with general output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

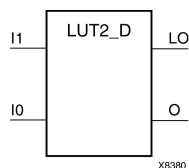
// End of LUT2_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

LUT2_D

プリミティブ : 2-Bit Look-Up Table with Dual Output



概要

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

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

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

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- LUT2_D: 2-input Look-Up Table with general and local outputs
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT2_D_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// LUT2_D: 2-input Look-Up Table with general and local outputs
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

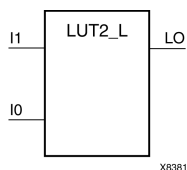
// End of LUT2_D_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

LUT2_L

プリミティブ : 2-Bit Look-Up Table with Local Output



概要

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

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

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

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- LUT2_L: 2-input Look-Up Table with local output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT2_L_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// LUT2_L: 2-input Look-Up Table with local output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

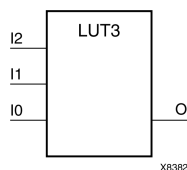
// End of LUT2_L_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

LUT3

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



概要

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

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

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

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- LUT3: 3-input Look-Up Table with general output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT3_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// LUT3: 3-input Look-Up Table with general output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

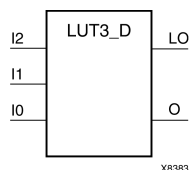
// End of LUT3_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

LUT3_D

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



概要

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

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

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

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- LUT3_D: 3-input Look-Up Table with general and local outputs
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT3_D_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// LUT3_D: 3-input Look-Up Table with general and local outputs
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

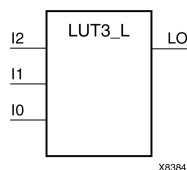
// End of LUT3_D_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

LUT3_L

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



概要

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

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

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

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- LUT3_L: 3-input Look-Up Table with local output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT3_L_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// LUT3_L: 3-input Look-Up Table with local output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

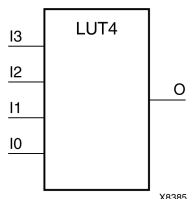
// End of LUT3_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

LUT4

プリミティブ：4-Bit Look-Up-Table with General Output



概要

このデザイン エLEMENTは、汎用出力 (O) を持つ 4 ビットのルックアップ テーブル (LUT) です。

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

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

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

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

論理表

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

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4: 4-input Look-Up Table with general output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT4_inst instantiation

```

Verilog 記述 (インスタンスレーション)

```
// LUT4: 4-input Look-Up Table with general output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

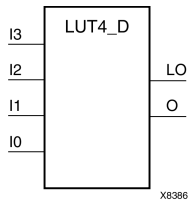
// End of LUT4_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

LUT4_D

プリミティブ：4-Bit Look-Up Table with Dual Output



概要

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

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

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

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

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

論理表

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

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_D: 4-input Look-Up Table with general and local outputs
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT4_D_inst instantiation

```


Verilog 記述 (インスタンス化)

```
// LUT4_D: 4-input Look-Up Table with general and local outputs
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

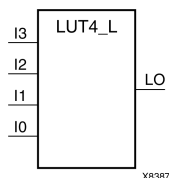
// End of LUT4_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

LUT4_L

プリミティブ : 4-Bit Look-Up Table with Local Output



概要

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

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

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

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

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

論理表

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

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- LUT4_L: 4-input Look-Up Table with local output
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT4_L_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// LUT4_L: 4-input Look-Up Table with local output
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

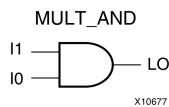
// End of LUT4_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MULT_AND

プリミティブ：Fast Multiplier AND



概要

このデザイン エLEMENTはスライス内にある AND コンポーネントで、2 つの入力は 4 入力 LUT と共有され、出力はキャリー ロジックを駆動します。この追加のロジックはその他の目的にも使用できますが、高速で小型の乗算器の作成に特に有用です。I1 および I0 入力は、対応する LUT の I1 および I0 入力に接続する必要があります。LO 出力は、対応する MUXCY、MUXCY_D、または MUXCY_L の DI 入力に接続する必要があります。

論理表

入力		出力
I1	I0	LO
0	0	0
0	1	0
1	0	0
1	1	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（インスタンス化）

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

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

-- MULT_AND: 2-input AND gate connected to Carry chain
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MULT_AND_inst : MULT_AND
port map (
  LO => LO,    -- MULT_AND output (connect to MUXCY DI)
  I0 => I0,    -- MULT_AND data[0] input
  I1 => I1     -- MULT_AND data[1] input
);

-- End of MULT_AND_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// MULT_AND: 2-input AND gate connected to Carry chain
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MULT_AND MULT_AND_inst (
    .LO(LO),    // MULT_AND output (connect to MUXCY DI)
    .IO(IO),    // MULT_AND data[0] input
    .I1(I1)     // MULT_AND data[1] input
);

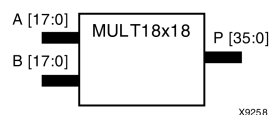
// End of MULT_AND_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MULT18X18

プリミティブ : 18 x 18 Signed Multiplier



概要

MULT18X18 は、組み合わせ符号付き 18 X 18 ビット乗算器です。18 ビット入力 A の値に 18 ビット入力 B の値を掛け合わせた積が、出力 P に 36 ビットで出力されます。

論理表

入力		出力
A	B	P
A	B	A X B
A、B、および P は 2 の補数です。		

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	不可

VHDL 記述 (インスタンス化)

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

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

-- MULT18X18: 18 x 18 signed asynchronous multiplier
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MULT18X18_inst : MULT18X18
port map (
    P => P,      -- 36-bit multiplier output
    A => A,      -- 18-bit multiplier input
    B => B       -- 18-bit multiplier input
);

-- End of MULT18X18_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MULT18X18: 18 x 18 signed asynchronous multiplier
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MULT18X18 MULT18X18_inst (
    .P(P),      // 36-bit multiplier output
    .A(A),      // 18-bit multiplier input
    .B(B)       // 18-bit multiplier input
);

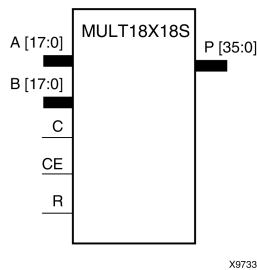
// End of MULT18X18_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MULT18X18S

プリミティブ：18 x 18 Signed Multiplier -- Registered Version



概要

MULT18X18S は、符号付き 18 X 18 乗算器 (MULT18X18) にレジスタを追加したもので、出力 (P)、データ入力 (A、B、C)、クロック イネーブル入力 (CE)、および同期リセット入力 (R) があります。レジスタは、GSR パルス後 0 に初期化されます。

18 ビット入力 A の値に 18 ビット入力 B の値を掛け合わせた積が、出力 P に 36 ビットで出力されます。

論理表

入力					出力
C	CE	Am	Bn	R	P
↑	X	X	X	1	0
↑	1	Am	Bn	0	A X B
X	0	X	X	0	変化なし

A、B、および P は 2 の補数です。

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	不可

VHDL 記述 (インスタンスレーション)

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

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

-- MULT18X18S: 18 x 18 signed synchronous multiplier
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MULT18X18S_inst : MULT18X18S
port map (
  P => P,      -- 36-bit multiplier output
  A => A,      -- 18-bit multiplier input
  B => B,      -- 18-bit multiplier input
  C => C,      -- Clock input
  CE => CE,    -- Clock enable input
  R => R       -- Synchronous reset input
);

-- End of MULT18X18S_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MULT18X18S: 18 x 18 signed synchronous multiplier
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MULT18X18S MULT18X18S_inst (
  .P(P),      // 36-bit multiplier output
  .A(A),      // 18-bit multiplier input
  .B(B),      // 18-bit multiplier input
  .C(C),      // Clock input
  .CE(CE),    // Clock enable input
  .R(R)       // Synchronous reset input
);

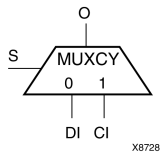
// End of MULT18X18S_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXCY

プリミティブ : 2-to-1 Multiplexer for Carry Logic with General Output



概要

スライスの直接入力 (DI) は、MUXCY の DI 入力に接続します。LC のキャリー入力 (CI) は、MUXCY の CI 入力に接続します。セレクト入力 (S) は、ルックアップ テーブル (LUT) の出力で駆動し、MUX ファンクションとしてコンフィギュレーションします。キャリー出力 (O) には選択された入力の値が出力され、各 LC のキャリー出力ファンクションをインプリメントします。S が Low の場合は DI が選択され、High の場合は CI が選択されます。

このほか、ローカル出力を持つ MUXCY_D および MUXCY_L があり、異なるタイミング モデルでレイアウト前のタイミングをより正確に予測する必要がある場合に使用できます。

論理表

入力			出力
S	DI	CI	O
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- MUXCY: Carry-Chain MUX with general output
-- Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXCY_inst : MUXCY
port map (
    O => O,    -- Carry output signal
    CI => CI,  -- Carry input signal
    DI => DI,  -- Data input signal
    S => S     -- MUX select, tie to '1' or LUT4 out
```

```
);  
-- End of MUXCY_inst instantiation
```

Verilog 記述 (インスタンス化)

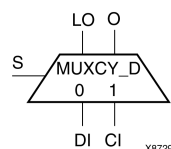
```
// MUXCY: Carry-Chain MUX with general output  
//      Spartan-3  
// Xilinx HDL Libraries Guide, version 14.1  
  
MUXCY MUXCY_inst (  
    .O(O),    // Carry output signal  
    .CI(CI),  // Carry input signal  
    .DI(DI),  // Data input signal  
    .S(S)     // MUX select, tie to '1' or LUT4 out  
);  
  
// End of MUXCY_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXCY_D

プリミティブ：2-to-1 Multiplexer for Carry Logic with Dual Output



概要

このデザイン エLEMENTは、1 ビットの高速キャリー伝搬ファンクションをインプリメントするために使用します。このようなファンクションは、1 つのロジック セル (LC) に 1 つずつインプリメントできるので、1 つの CLB に合計 4 ビットをインプリメントできます。LC の直接入力 (DI) は MUXCY_D の DI 入力に接続し、LC のキャリー入力 (CI) は MUXCY_D の CI 入力に接続します。セレクト入力 (S) は、ルックアップ テーブル (LUT) の出力で駆動し、XOR ファンクションとしてコンフィギュレーションします。キャリー出力 (O と LO) には選択された入力の値が出力され、各 LC のキャリー出力ファンクションをインプリメントします。S が Low の場合は DI が選択され、High の場合は CI が選択されます。

出力 O と LO は、機能的に同じです。出力 O は汎用インターコネクトです。「MUXCY」および「MUXCY_L」も参照してください。

論理表

入力			出力	
S	DI	CI	O	LO
0	1	X	1	1
0	0	X	0	0
1	X	1	1	1
1	X	0	0	0

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- MUXCY_D: Carry-Chain MUX with general and local outputs
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXCY_D_inst : MUXCY_D
port map (
    LO => LO, -- Carry local output signal
    O => O,   -- Carry general output signal
    CI => CI, -- Carry input signal
    DI => DI, -- Data input signal
    S => S    -- MUX select, tie to '1' or LUT4 out
);

-- End of MUXCY_D_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXCY_D: Carry-Chain MUX with general and local outputs
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXCY_D MUXCY_D_inst (
    .LO(LO), // Carry local output signal
    .O(O),   // Carry general output signal
    .CI(CI), // Carry input signal
    .DI(DI), // Data input signal
    .S(S)    // MUX select, tie to '1' or LUT4 out
);

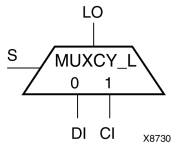
// End of MUXCY_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXCY_L

プリミティブ：2-to-1 Multiplexer for Carry Logic with Local Output



概要

このデザイン エLEMENTは、1 ビットの高速キャリー伝搬ファンクションをインプリメントするために使用します。このようなファンクションは、1 つのロジックセル (LC) に 1 つずつインプリメントできるので、1 つの CLB に合計 4 ビットをインプリメントできます。LC の直接入力 (DI) は MUXCY_L の DI 入力に接続し、LC のキャリー入力 (CI) は MUXCY_L の CI 入力に接続します。セレクト入力 (S) は、ルックアップ テーブル (LUT) の出力で駆動し、XOR ファンクションとしてコンフィギュレーションします。キャリー出力 (LO) には選択された入力の値が出力され、各 LC のキャリー出力ファンクションをインプリメントします。S が Low の場合は DI が選択され、High の場合は CI が選択されます。

「MUXCY」および「MUXCY_D」も参照してください。

論理表

入力			出力
S	DI	CI	LO
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- MUXCY_L: Carry-Chain MUX with local output
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXCY_L_inst : MUXCY_L
port map (
  LO => LO, -- Carry local output signal
  CI => CI, -- Carry input signal
  DI => DI, -- Data input signal
```

```
S => S    -- MUX select, tie to '1' or LUT4 out
);
-- End of MUXCY_L_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXCY_L: Carry-Chain MUX with local output
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXCY_L MUXCY_L_inst (
    .LO(LO), // Carry local output signal
    .CI(CI), // Carry input signal
    .DI(DI), // Data input signal
    .S(S)    // MUX select, tie to '1' or LUT4 out
);

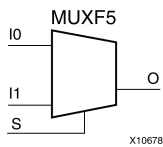
// End of MUXCY_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF5

プリミティブ：2-to-1 Look-Up Table Multiplexer with General Output



概要

このデザイン エLEMENTは、LUT4 ルックアップ テーブルと組み合わせて、5 ファンクションのルックアップ テーブルまたは 4:1 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、2 つの LUT4 ルックアップ テーブルのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O は汎用インターコネクトです。

このほか、ローカル出力を持つ MUXF5_D および MUXF5_L があり、異なるタイミング モデルでレイアウト前のタイミングをより正確に予測する必要がある場合に使用できます。

論理表

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- MUXF5: Slice MUX to tie two LUT4's together with general output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXF5_inst : MUXF5
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- Input (tie directly to the output of LUT4)
    I1 => I1,    -- Input (tie directly to the output of LUT4)
    S => S       -- Input select to MUX
);

-- End of MUXF5_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXF5: Slice MUX to tie two LUT4's together with general output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXF5 MUXF5_inst (
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie directly to the output of LUT4)
    .I1(I1),    // Input (tie directly to the output of LUT4)
    .S(S)       // Input select to MUX
);

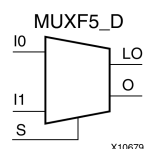
// End of MUXF5_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF5_D

プリミティブ：2-to-1 Look-Up Table Multiplexer with Dual Output



概要

このデザイン エLEMENTは、LUT4 ルックアップ テーブルと組み合わせて、5 ファンクションのルックアップ テーブルまたは 4:1 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、2 つの LUT4 ルックアップ テーブルのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は汎用インターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

「MUXF5」および「MUXF5_L」も参照してください。

論理表

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- MUXF5_D: Slice MUX to tie two LUT4's together with general and local outputs
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXF5_D_inst : MUXF5_D
port map (
    LO => LO,  -- Ouput of MUX to local routing
    O => O,    -- Output of MUX to general routing
    IO => IO,  -- Input (tie directly to the output of LUT4)
    I1 => I1,  -- Input (tie directoy to the output of LUT4)
    S => S     -- Input select to MUX
);

-- End of MUXF5_D_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXF5_D: Slice MUX to tie two LUT4's together with general and local outputs
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXF5_D MUXF5_D_inst (
    .LO(LO),  // Ouput of MUX to local routing
    .O(O),    // Output of MUX to general routing
    .IO(IO),  // Input (tie directly to the output of LUT4)
    .I1(I1),  // Input (tie directoy to the output of LUT4)
    .S(S)     // Input select to MUX
);

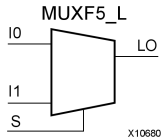
// End of MUXF5_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF5_L

プリミティブ：2-to-1 Look-Up Table Multiplexer with Local Output



概要

このデザイン エLEMENTは、LUT4 ルックアップ テーブルと組み合わせて、5 ファンクションのルックアップ テーブルまたは 4:1 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、2 つの LUT4 ルックアップ テーブルのローカル出力 (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。

「MUXF5」および「MUXF5_D」も参照してください。

論理表

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

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- MUXF5_L: Slice MUX to tie two LUT4's together with local output
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXF5_L_inst : MUXF5_L
port map (
    LO => LO,  -- Output of MUX to local routing
    I0 => I0,  -- Input (tie directly to the output of LUT4)
    I1 => I1,  -- Input (tie directly to the output of LUT4)
    S  => S    -- Input select to MUX
```

```
);  
  
-- End of MUXF5_L_inst instantiation
```

Verilog 記述 (インスタンス化)

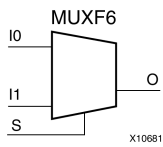
```
// MUXF5_L: Slice MUX to tie two LUT4's together with local output  
//           Spartan-3  
// Xilinx HDL Libraries Guide, version 14.1  
  
MUXF5_L MUXF5_L_inst (  
    .LO(LO), // Output of MUX to local routing  
    .IO(IO), // Input (tie directly to the output of LUT4)  
    .I1(I1), // Input (tie directly to the output of LUT4)  
    .S(S)    // Input select to MUX  
);  
  
// End of MUXF5_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF6

プリミティブ：2-to-1 Look-Up Table Multiplexer with General Output



概要

このデザイン エLEMENTは、4 つの LUT4 ルックアップ テーブルおよび 2 つの MUXF5 マルチプレクサーと組み合わせて、スライス 2 つに 6 ファンクションのルックアップ テーブルまたは 8:1 マルチプレクサーを作成するための 2 入力 マルチプレクサーです。I0 および I1 入力には、同じ CLB 内にある 2 つの MUXF5 マルチプレクサーのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O は汎用インターコネクトです。

このほか、ローカル出力を持つ MUXF6_D および MUXF6_L があり、異なるタイミング モデルでレイアウト前のタイミング をより正確に予測する場合に使用できます。

論理表

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- MUXF6: CLB MUX to tie two MUXF5's together with general output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXF6_inst : MUXF6
port map (
    O => O,      -- Output of MUX to general routing
    IO => IO,    -- Input (tie to MUXF5 LO out)
    I1 => I1,    -- Input (tie to MUXF5 LO out)
    S => S       -- Input select to MUX
);

-- End of MUXF6_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXF6: CLB MUX to tie two MUXF5's together with general output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXF6 MUXF6_inst (
    .O(O),      // Output of MUX to general routing
    .IO(IO),    // Input (tie to MUXF5 LO out)
    .I1(I1),    // Input (tie to MUXF5 LO out)
    .S(S)       // Input select to MUX
);

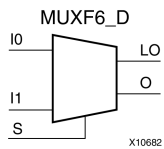
// End of MUXF6_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF6_D

プリミティブ：2-to-1 Look-Up Table Multiplexer with Dual Output



概要

このデザイン エLEMENTは、4 つの LUT4 ルックアップ テーブルおよび 2 つの MUXF5 マルチプレクサーと組み合わせて、スライス 2 つに 6 ファンクションのルックアップ テーブルまたは 8:1 マルチプレクサーを作成するための 2 入力 マルチプレクサーです。I0 および I1 入力には、同じ CLB 内にある 2 つの MUXF5 マルチプレクサーのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は汎用インターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

「MUXF6」および「MUXF6_L」も参照してください。

論理表

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

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- MUXF6_D: CLB MUX to tie two MUXF5's together with general and local outputs
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXF6_D_inst : MUXF6_D
port map (
    LO => LO,  -- Output of MUX to local routing
    O  => O,   -- Output of MUX to general routing
    IO => IO,  -- Input (tie to MUXF5 LO out)
    I1 => I1,  -- Input (tie to MUXF5 LO out)
    S  => S    -- Input select to MUX
);

-- End of MUXF6_D_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// MUXF6_D: CLB MUX to tie two MUXF5's together with general and local outputs
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXF6_D MUXF6_D_inst (
    .LO(LO), // Output of MUX to local routing
    .O(O),  // Output of MUX to general routing
    .IO(IO), // Input (tie to MUXF5 LO out)
    .I1(I1), // Input (tie to MUXF5 LO out)
    .S(S)   // Input select to MUX
);

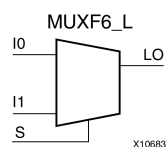
// End of MUXF6_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF6_L

プリミティブ：2-to-1 Look-Up Table Multiplexer with Local Output



概要

このデザイン エLEMENTは、4 つの LUT4 ルックアップ テーブルおよび 2 つの MUXF5 マルチプレクサーと組み合わせて、スライス 2 つに 6 ファンクションのルックアップ テーブルまたは 8:1 マルチプレクサーを作成するための 2 入力 マルチプレクサーです。I0 および I1 入力には、同じ CLB 内にある 2 つの MUXF5 マルチプレクサーのローカル出力 (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。

「MUXF6」および「MUXF6_D」も参照してください。

論理表

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- MUXF6_L: CLB MUX to tie two MUXF5's together with local output
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXF6_L_inst : MUXF6_L
port map (
    LO => LO,  -- Output of MUX to local routing
    IO => IO,  -- Input (tie to MUXF5 LO out)
    I1 => I1,  -- Input (tie to MUXF5 LO out)
    S => S     -- Input select to MUX
);

-- End of MUXF6_L_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXF6_L: CLB MUX to tie two MUXF5's together with local output
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXF6_L MUXF6_L_inst (
    .LO(IO), // Output of MUX to local routing
    .IO(IO), // Input (tie to MUXF5 LO out)
    .I1(I1), // Input (tie to MUXF5 LO out)
    .S(S)    // Input select to MUX
);

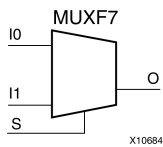
// End of MUXF6_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF7

プリミティブ：2-to-1 Look-Up Table Multiplexer with General Output



概要

このデザイン エLEMENTは、MUXF6 および MUXF5 マルチプレクサー、LUT4 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 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、MUXF6 の 2 つのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O は汎用インターコネクトです。

このほか、ローカル出力を持つ MUXF7_D および MUXF7_L があり、異なるタイミング モデルでレイアウト前のタイミングをより正確に予測する必要がある場合に使用できます。

論理表

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

ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
I0	入力	1	入力 (MUXF6 LO 出力に接続)
I1	入力	1	入力 (MUXF6 LO 出力に接続)
S	入力	1	MUX への入力セレクト

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- MUXF7: CLB MUX to tie two MUXF6's together with general output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXF7_inst : MUXF7
port map (
    O => O,      -- Output of MUX to general routing
    IO => IO,    -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,    -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S       -- Input select to MUX
);

-- End of MUXF7_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXF7: CLB MUX to tie two MUXF6's together with general output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXF7 MUXF7_inst (
    .O(O),       // Output of MUX to general routing
    .IO(IO),    // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .I1(I1),    // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .S(S)       // Input select to MUX
);

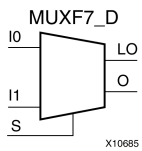
// End of MUXF7_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF7_D

プリミティブ：2-to-1 Look-Up Table Multiplexer with Dual Output



概要

このデザイン エLEMENTは、MUXF6 および MUXF5 マルチプレクサー、LUT4 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 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、MUXF6 の 2 つのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は汎用インターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

「MUXF7」および「MUXF7_L」も参照してください。

論理表

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

ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF6 LO 出力に接続)
I1	入力	1	入力 (MUXF6 LO 出力に接続)
S	入力	1	MUX への入力セレクト

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- MUXF7_D: CLB MUX to tie two MUXF6's together with general and local outputs
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

MUXF7_D_inst : MUXF7_D
port map (
    LO => LO,  -- Ouput of MUX to local routing
    O => O,    -- Output of MUX to general routing
    IO => IO,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S     -- Input select to MUX
);

-- End of MUXF7_D_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXF7_D: CLB MUX to tie two MUXF6's together with general and local outputs
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXF7_D MUXF7_D_inst (
    .LO(LO),  // Ouput of MUX to local routing
    .O(O),    // Output of MUX to general routing
    .IO(IO),  // Input (tie to 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
);

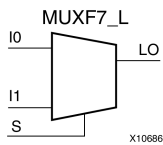
// End of MUXF7_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF7_L

プリミティブ：2-to-1 look-up table Multiplexer with Local Output



概要

このデザイン エLEMENTは、MUXF6 および MUXF5 マルチプレクサー、LUT4 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 マルチプレクサーを作成するための 2 入力マルチプレクサーです。I0 および I1 入力には、MUXF6 の 2 つのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

「MUXF7」および「MUXF7_D」も参照してください。

論理表

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

ポートの説明

ポート名	方向	幅	機能
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力
I1	入力	1	入力
S	入力	1	MUX への入力セレクト

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- MUXF7_L: CLB MUX to tie two MUXF6's together with local output
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of MUXF7_L_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXF7_L: CLB MUX to tie two MUXF6's together with local output
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXF7_L MUXF7_L_inst (
    .LO(IO), // Output of MUX to local routing
    .IO(IO), // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .I1(I1), // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .S(S)    // Input select to MUX
);

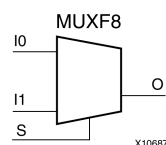
// End of MUXF7_L_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

MUXF8

プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output



概要

このデザイン エLEMENTは、ルックアップ テーブル、MUXF5、MUXF6、および MUXF7 と組み合わせて、スライス 8 個に 8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサーを作成するためのマルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

論理表

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

ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- MUXF8: CLB MUX to tie two MUXF7's together with general output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of MUXF8_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXF8: CLB MUX to tie two MUXF7's together with general output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

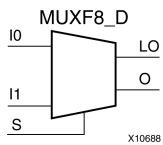
// End of MUXF8_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF8_D

プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output



概要

このデザイン エLEMENTは、ルックアップ テーブル、MUXF5、MUXF6、および MUXF7 を組み合わせて、スライス 8 個に 8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサーを作成するためのマルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は汎用インターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

論理表

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

ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of MUXF8_D_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

MUXF8_D MUXF8_D_inst (
    .LO(LO), // Output of MUX to local routing
    .O(O),   // Output of MUX to general routing
    .IO(IO), // Input (tie to MUXF7 LO out)
    .I1(I1), // Input (tie to MUXF7 LO out)
    .S(S)    // Input select to MUX
);

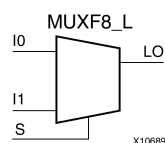
// End of MUXF8_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

MUXF8_L

プリミティブ : 2-to-1 Look-Up Table Multiplexer with Local Output



概要

このデザイン エLEMENTは、ルックアップ テーブル、MUXF5、MUXF6、および MUXF7 を組み合わせて、スライス 8 個に 8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサーを作成するためのマルチプレクサーです。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

論理表

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

ポートの説明

ポート名	方向	幅	機能
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

-- MUXF8_L: CLB MUX to tie two MUXF7's together with local output
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of MUXF8_L_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// MUXF8_L: CLB MUX to tie two MUXF7's together with local output
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

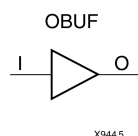
// End of MUXF8_L_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

OBUF

プリミティブ：Output Buffer



概要

このデザイン エLEMENTは単純な出力バッファで、トライステートにならない (常に駆動される) FPGA デバイス ピンへの出力信号を駆動するために使用します。デザインのすべての出力ポートに OBUF、OBUFT、OBUFDS、OBUFTDS のいずれかを接続する必要があります。

このELEMENTは内部回路を外部から分離し、チップから出力する信号の駆動電流を供給します。I/O ブロック (IOB) 内にあります。出力 (O) は、OPAD または IOPAD に接続されます。このELEMENTでは、LVTTL 規格が使用され、DRIVE 制約と SLOW または FAST 制約を使用して駆動電流とスルー レートを選択できます。デフォルトでは、DRIVE は 12mA、スルー レートは SLOW に設定されています。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	最上位出力ポートに直接接続される OBUF の出力
I	入力	1	OBUF の入力。出力ポートを駆動するロジックに接続

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

-- OBUF: Single-ended Output Buffer
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

OBUF_inst : OBUF
generic map (
    DRIVE => 12,
    IOSTANDARD => "DEFAULT",
    SLEW => "SLOW")
port map (
    O => O,      -- Buffer output (connect directly to top-level port)
    I => I       -- Buffer input
);

-- End of OBUF_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// OBUF: Single-ended Output Buffer
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

OBUF #(
    .DRIVE(12),    // Specify the output drive strength
    .IOSTANDARD("DEFAULT"), // Specify the output I/O standard
    .SLEW("SLOW") // Specify the output slew rate
) OBUF_inst (
    .O(O),        // Buffer output (connect directly to top-level port)
    .I(I)         // Buffer input
);

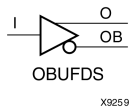
// End of OBUF_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

OBUFDS

プリミティブ : Differential Signaling Output Buffer



概要

このデザイン エLEMENTは、低電圧の差動信号 (1.8V CMOS) をサポートする単一の出力バッファです。内部回路を外部から分離し、チップから出力する信号の駆動電流を供給します。出力には 2 つの異なるポート (O および OB) があり、これらのポートをそれぞれ「マスター」および「スレーブ」と呼びます。マスターとスレーブは MYNET と MYNETB のように、同じ論理信号の反対の状態を示します。

論理表

入力	出力	
I	O	OB
0	0	1
1	1	0

ポートの説明

ポート名	方向	幅	機能
O	出力	1	Diff_p 出力 (最上位ポートに直接接続)
OB	出力	1	Diff_n 出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- OBUFDS: Differential Output Buffer
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

OBUFDS_inst : OBUFDS
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Diff_p output (connect directly to top-level port)
    OB => OB,    -- Diff_n output (connect directly to top-level port)
    I => I       -- Buffer input
);

-- End of OBUFDS_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// OBUFDS: Differential Output Buffer
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

OBUFDS #(
    .IOSTANDARD("DEFAULT") // Specify the output I/O standard
) OBUFDS_inst (
    .O(O),      // Diff_p output (connect directly to top-level port)
    .OB(OB),    // Diff_n output (connect directly to top-level port)
    .I(I)       // Buffer input
);

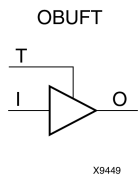
// End of OBUFDS_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

OBUFT

プリミティブ : 3-State Output Buffer with Active Low Output Enable



概要

このデザイン エLEMENTは、入力 (I)、出力 (O)、アクティブ Low 出力イネーブル (T) を持つ単一のトライステート出力バッファです。このELEMENTでは、LVTTTL 規格が使用され、DRIVE 制約と SLOW または FAST 制約を使用して駆動電流とスルー レートを選択できます。デフォルトでは、DRIVE=12mA、スルー レートは SLOW に設定されています。

T が Low の場合、バッファに入力された値が対応する出力に送られます。T が High の場合は、出力がハイインピーダンス (オフまたは Z ステート) になります。OBUFT は、双方向 I/O を作成するなど、トライステート機能にシングルエンド出力を使用する必要がある場合に使用します。

論理表

入力		出力
T	I	O
1	X	Z
0	1	1
0	0	0

ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファ出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力
T	入力	1	トライステート イネーブル入力

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	出力の駆動電流を指定します。デザインの駆動電流およびタイミング要件で許容される最も低い値を使用してください。
IOSTANDARD	文字列	データシートを参照	"DEFAULT"	エレメントに I/O 規格を割り当てます。
SLEW	文字列	"SLOW"、"FAST"	"SLOW"	出力ドライバーのスループートを指定します。この属性の最適な設定方法は、データシートを参照してください。

VHDL 記述 (インスタンス化)

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

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

-- OBUFT: Single-ended 3-state Output Buffer
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of OBUFT_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// OBUFT: Single-ended 3-state Output Buffer
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

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

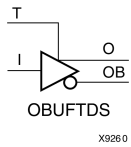
// End of OBUFT_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

OBUFTDS

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



概要

このデザイン エLEMENTは、低電圧差動信号をサポートする出力バッファです。OBUFTDS では、デザイン レベルのインターフェイス信号は、一方が「マスター」で、もう一方が「スレーブ」となる 2 つの異なるポート (O、OB) で表されます。マスターとスレーブは MYNET_P と MYNET_N のように、同じ論理信号の反対の状態を示します。

論理表

入力		出力	
I	T	O	OB
X	1	Z	Z
0	0	0	1
1	0	1	0

ポートの説明

ポート名	方向	幅	機能
O	出力	1	Diff_p 出力 (最上位ポートに直接接続)
OB	出力	1	Diff_n 出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力
T	入力	1	トライステート イネーブル入力

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- OBUFTDS: Differential 3-state Output Buffer
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

OBUFTDS_inst : OBUFTDS
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Diff_p output (connect directly to top-level port)
    OB => OB,    -- Diff_n output (connect directly to top-level port)
    I => I,      -- Buffer input
    T => T       -- 3-state enable input
);

-- End of OBUFTDS_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// OBUFTDS: Differential 3-state Output Buffer
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

OBUFTDS #(
    .IOSTANDARD("DEFAULT") // Specify the output I/O standard
) OBUFTDS_inst (
    .O(O),      // Diff_p output (connect directly to top-level port)
    .OB(OB),    // Diff_n output (connect directly to top-level port)
    .I(I),      // Buffer input
    .T(T)       // 3-state enable input
);

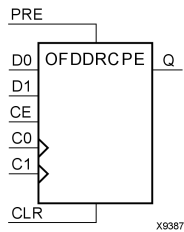
// End of OBUFTDS_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

OFDDRCPE

プリミティブ : Dual Data Rate Output D Flip-Flop with Clock Enable and Asynchronous Preset and Clear



概要

このデザイン エLEMENTは、クロック イネーブル (CE)、非同期プリセット (PRE)、非同期クリア (CLR) があるデュアル データレート (DDR) の出力 D フリップフロップです。このフリップフロップは、出力バッファ 1 つとデュアル データレートのフリップフロップ (FDDRCPE) 1 つで構成されています。PRE が High で CLR が Low の場合、出力 Q は High にプリセットされます。

CLR が High の場合、Q が Low にセットされます。PRE と CLR が Low で CE が High の場合、クロック (C0) が Low から High に切り替わるたびに D0 入力の値がフリップフロップにロードされ、クロック (C1) が Low から High に切り替わるたびに D1 入力の値がフリップフロップにロードされます。OFDDRCPE コンポーネントでは、INIT 属性は使用できません。

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

論理表

入力							出力
C0	C1	CE	D0	D1	CLR	PRE	Q
X	X	X	X	X	1	0	0
X	X	X	X	X	0	1	1
X	X	X	X	X	1	1	0
X	X	0	X	X	0	0	変化なし
↑	X	1	D0	X	0	0	D0
X	↑	1	X	D1	0	0	D1

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- OFDDRCPE: Double Data Rate Output Register with Async. Clear, Async. Preset
--           and Clock Enable.
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

OFDDRCPE_inst : OFDDRCPE
port map (
    Q => Q,          -- Data output (connect directly to top-level port)
    C0 => C0,         -- 0 degree clock input
    C1 => C1,         -- 180 degree clock input
    CE => CE,         -- Clock enable input
    CLR => CLR,       -- Asynchronous reset input
    D0 => D0,         -- Posedge data input
    D1 => D1,         -- Negedge data input
    PRE => PRE        -- Asynchronous preset input
);

-- End of OFDDRCPE_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// OFDDRCPE: Double Data Rate Output Register with Async. Clear, Async. Preset
//           and Clock Enable.
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

OFDDRCPE OFDDRCPE_inst (
    .Q(Q),           // Data output (connect directly to top-level port)
    .C0(C0),         // 0 degree clock input
    .C1(C1),         // 180 degree clock input
    .CE(CE),         // Clock enable input
    .CLR(CLR),       // Asynchronous reset input
    .D0(D0),         // Posedge data input
    .D1(D1),         // Negedge data input
    .PRE(PRE)        // Asynchronous preset input
);

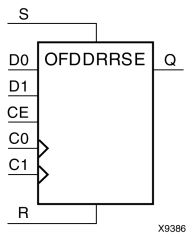
// End of OFDDRCPE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

OFDDRSE

プリミティブ：Dual Data Rate Output D Flip-Flop with Synchronous Reset and Set and Clock Enable



概要

このデザイン エLEMENTは、同期リセット (R)、同期セット (S)、およびクロック イネーブル (CE) があるデュアル データレート (DDR) の出力 D フリップフロップです。このフリップフロップは、出力バッファ 1 つとデュアル データレートのフリップフロップ (FDDRSE) 1 つで構成されています。

クロック (C0 または C1) が Low から High に切り替わるときに、入力 R が High になっていると出力 Q が Low にリセットされ、入力 R が Low で入力 S が High になっていると出力 Q が High にセットされます。R と S の両方が Low で CE が High の場合、クロック (C0) が Low から High に切り替わるときに D0 入力の値がフリップフロップにロードされ、クロック (C1) が Low から High に切り替わるときに D1 入力の値がフリップフロップにロードされます。

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

このデザイン エLEMENTでは、INIT 属性は使用できません。

論理表

入力							出力
C0	C1	CE	D0	D1	R	S	Q
↑	X	X	X	X	1	0	0
↑	X	X	X	X	0	1	1
↑	X	X	X	X	1	1	0
X	↑	X	X	X	1	0	0
X	↑	X	X	X	0	1	1
X	↑	X	X	X	1	1	0
X	X	0	X	X	0	0	変化なし
↑	X	1	D0	X	0	0	D0
X	↑	1	X	D1	0	0	D1

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- OFDDRSE: Double Data Rate Input Register with Sync. Clear, Sync. Preset
--           and Clock Enable.
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

OFDDRSE_inst : OFDDRSE
port map (
  Q => Q,      -- Data output (connect directly to top-level port)
  C0 => C0,    -- 0 degree clock input
  C1 => C1,    -- 180 degree clock input
  CE => CE,    -- Clock enable input
  D0 => D0,    -- Posedge data input
  D1 => D1,    -- Negedge data input
  R => R,      -- Synchronous reset input
  S => S       -- Synchronous preset input
);

-- End of OFDDRSE_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// OFDDRSE: Double Data Rate Input Register with Sync. Clear, Sync. Preset
//           and Clock Enable.
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

OFDDRSE OFDDRSE_inst (
  .Q(Q),        // Data output (connect directly to top-level port)
  .C0(C0),      // 0 degree clock input
  .C1(C1),      // 180 degree clock input
  .CE(CE),      // Clock enable input
  .D0(D0),      // Posedge data input
  .D1(D1),      // Negedge data input
  .R(R),        // Synchronous reset input
  .S(S)         // Synchronous preset input
);

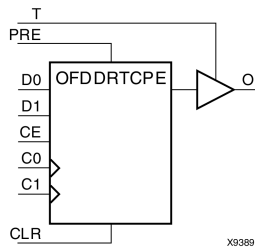
// End of OFDDRSE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

OFDDRTCPE

プリミティブ : Dual Data Rate D Flip-Flop with Active-Low 3--State Output Buffer, Clock Enable, and Asynchro-nous Preset and Clear



概要

このデザイン エLEMENTは、クロック イネーブル (CE)、非同期プリセット、非同期クリアがあるデュアル データレート (DDR) の出力 D フリップフロップで、出力はトリステート バッファでイネーブル制御されます。このフリップフロップは、デュアル データレートのフリップフロップ (FDDRCPE) 1 つとトリステート出力バッファ (OBUFT) 1 つで構成されています。フリップフロップのデータ出力 (O) は、出力バッファ (OBUFT) の入力に接続されます。OBUFT の出力は、OPAD または IOPAD に接続されます。

アクティブ Low のイネーブル入力 (T) が Low の場合、出力がイネーブルになり、フリップフロップの出力 (Q) の値が OBUFT の O に出力されます。非同期プリセット (PRE) が High で 非同期クリア (CLR) が Low の場合、出力 O は High にプリセットされます。CLR が High になると、出力 O は Low にリセットされます。PRE と CLR が Low で CE が High の場合、クロック (C0) が Low から High に切り替わる時に D0 入力の値がフリップフロップにロードされ、クロック (C1) が Low から High に切り替わる時に D1 入力の値がフリップフロップにロードされます。

T が High になると、出力はハイ インピーダンス (オフ) になります。クロック イネーブル (CE) と T が Low の場合は、出力は変化しません。

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

このデザイン エLEMENTでは、INIT 属性は使用できません。

論理表

入力								出力
C0	C1	CE	D0	D1	CLR	PRE	T	O
X	X	X	X	X	X	X	1	Z
X	X	X	X	X	1	0	0	0
X	X	X	X	X	0	1	0	1
X	X	X	X	X	1	1	0	0
X	X	0	X	X	0	0	0	変化なし
↑	X	1	D0	X	0	0	0	D0
X	↑	1	X	D1	0	0	0	D1

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- OFDDRTCPE: Double Data Rate Output Register with Async. Clear, Async. Preset
--             and Clock Enable with 3-state.
--             Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

OFDDRTCPE_inst : OFDDRTCPE
port map (
  O => O,      -- Data output (connect directly to top-level port)
  C0 => C0,     -- 0 degree clock input
  C1 => C1,     -- 180 degree clock input
  CE => CE,     -- Clock enable input
  CLR => CLR,   -- Asynchronous reset input
  D0 => D0,     -- Posedge data input
  D1 => D1,     -- Negedge data input
  PRE => PRE,   -- Asynchronous preset input
  T => T       -- 3-state enable input
);

-- End of OFDDRTCPE_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// OFDDRTCPE: Double Data Rate Output Register with Async. Clear, Async. Preset
//             and Clock Enable with 3-state.
//             Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

OFDDRTCPE OFDDRTCPE_inst (
  .O(O),        // Data output (connect directly to top-level port)
  .C0(C0),      // 0 degree clock input
  .C1(C1),      // 180 degree clock input
  .CE(CE),      // Clock enable input
  .CLR(CLR),    // Asynchronous reset input
  .D0(D0),      // Posedge data input
  .D1(D1),      // Negedge data input
  .PRE(PRE),    // Asynchronous preset input
  .T(T)         // 3-state enable input
);

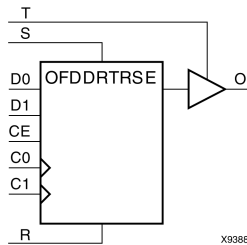
// End of OFDDRTCPE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

OFDDRTRSE

プリミティブ : Dual Data Rate D Flip-Flop with Active-Low 3-State Output Buffer, Synchronous Reset and Set, and Clock Enable



概要

このデザイン エLEMENTは、クロック イネーブル (CE)、同期リセット (R)、同期セット (S) があるデュアル データ レート (DDR) の出力 D フリップフロップであり、出力はトライステート バッファでイネーブル制御されます。このフリップフロップは、デュアル データ レートのフリップフロップ (FDDRSE) とトライステート出力バッファ (OBUFT) で構成されています。フリップフロップのデータ出力 (O) は、出力バッファ (OBUFT) の入力に接続されます。OBUFT の出力は、OPAD または IOPAD に接続されます。

アクティブ Low のイネーブル入力 (T) が Low の場合、出力がイネーブルになり、フリップフロップの出力 (Q) の値が OBUFT の O に出力されます。クロック (C0 または C1) が Low から High に切り替わるときに、入力 R が High になっていると出力 Q が Low にリセットされ、入力 R が Low で入力 S が High になっていると出力 O が High にセットされます。R と S の両方が Low で CE が High の場合、クロック (C0) が Low から High に切り替わるときに D0 入力の値がフリップフロップにロードされ、クロック (C1) が Low から High に切り替わるときに D1 入力の値がフリップフロップにロードされます。

T が High になると、出力はハイ インピーダンス (オフ) になります。クロック イネーブル (CE) と T が Low の場合は、出力は変化しません。

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

このデザイン エLEMENTのコンポーネントでは、INIT 属性は使用できません。

論理表

入力								出力
C0	C1	CE	D0	D1	R	S	T	O
X	X	X	X	X	X	X	1	Z
↑	X	X	X	X	1	0	0	0
↑	X	X	X	X	0	1	0	1
↑	X	X	X	X	1	1	0	0
X	↑	X	X	X	1	0	0	0
X	↑	X	X	X	0	1	0	1
X	↑	X	X	X	1	1	0	0
X	X	0	X	X	0	0	0	変化なし

入力								出力
C0	C1	CE	D0	D1	R	S	T	O
↑	X	1	D0	X	0	0	0	D0
X	↑	1	X	D1	0	0	0	D1

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- OFDDRTRSE: Double Data Rate Input Register with Sync. Clear, Sync. Preset
--             and Clock Enable with 3-state.
--             Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

OFDDRTRSE_inst : OFDDRTRSE
port map (
    O => O,           -- Data output (connect directly to top-level port)
    C0 => C0,         -- 0 degree clock input
    C1 => C1,         -- 180 degree clock input
    CE => CE,         -- Clock enable input
    D0 => D0,         -- Posedge data input
    D1 => D1,         -- Negedge data input
    R => R,           -- Synchronous reset input
    S => S,           -- Synchronous preset input
    T => T,           -- 3-state enable input
);

-- End of OFDDRTRSE_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// OFDDRTRSE: Double Data Rate Input Register with Sync. Clear, Sync. Preset
//             and Clock Enable with 3-state.
//             Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

OFDDRTRSE OFDDRTRSE_inst (
    .Q(Q),           // Data output (connect directly to top-level port)
    .C0(C0),         // 0 degree clock input
    .C1(C1),         // 180 degree clock input
    .CE(CE),         // Clock enable input
    .D0(D0),         // Posedge data input
    .D1(D1),         // Negedge data input
    .R(R),           // Synchronous reset input
    .S(S),           // Synchronous preset input
    .T(T),           // 3-state enable input
);

// End of OFDDRTRSE_inst instantiation
```

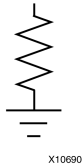

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

PULLDOWN

プリミティブ : Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs

PULLDOWN



X10690

概要

この抵抗エレメントは、入力、出力、双方向のパッドに接続し、フロートする可能性のあるノードのロジックレベルを Low にします。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	プルダウン出力 (最上位ポートに直接接続)

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- PULLDOWN: I/O Buffer Weak Pull-down
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of PULLDOWN_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// PULLDOWN: I/O Buffer Weak Pull-down
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

PULLDOWN PULLDOWN_inst (
    .O(0)      // Pulldown output (connect directly to top-level port)
);

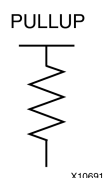
// End of PULLDOWN_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

PULLUP

プリミティブ : Resistor to VCC for Input PADs, Open-Drain, and 3-State Outputs



概要

このデザイン エLEMENTは、入力、トライステート出力、または双方向ポートが内部または外部ソースで駆動されないときに、弱い High で駆動します。すべてのドライバーがオフのときに、オープンドレイン エLEMENTおよびマクロのロジックレベルを High にします。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	プルアップ出力 (最上位ポートに直接接続)

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- PULLUP: I/O Buffer Weak Pull-up
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of PULLUP_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// PULLUP: I/O Buffer Weak Pull-up
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

PULLUP PULLUP_inst (
    .O(0)      // Pullup output (connect directly to top-level port)
);

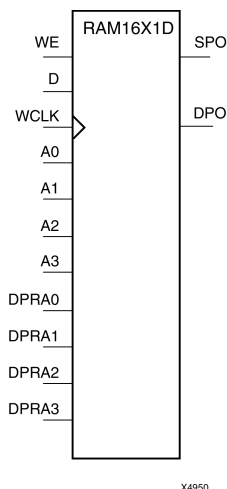
// End of PULLUP_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAM16X1D

プリミティブ：16-Deep by 1-Wide Static Dual Port Synchronous RAM



概要

このエレメントは 16 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 (DPRA3 ~ DPRA0) と書き込みアドレス (A3 ~ A0) の 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非同期です。読み出しアドレスによって出力ピン (DPO) に出力される値が指定され、書き込みアドレスによって書き込みを行う位置が指定されます。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。

WE が High の場合、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 4 ビットの書き込みアドレスで選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

SPO 出力には、A3 ~ 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、DPRA3 ~ 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。

INIT 属性を使用すると、RAM を直接初期化できます。値は、INIT=ABAC のように、16 進数で指定してください。INIT 属性を指定しない場合は、RAM は 0 に初期化されます。

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	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 = A3 ～ A0 で指定されたワード				
data_d = DPRA3 ～ DPRA0 で指定されたワード				

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 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 つの文が存在しない場合は、コピーしてエンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM16X1D: 16 x 1 positive edge write, asynchronous read dual-port distributed RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAM16X1D_inst : RAM16X1D
generic map (
  INIT => X"0000")
port map (
  DPO => DPO,      -- Read-only 1-bit data output for DPRA
  SPO => SPO,      -- R/W 1-bit data output for A0-A3
  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
  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
  WCLK => WCLK,    -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM16X1D_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// RAM16X1D: 16 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAM16X1D #(
    .INIT(16'h0000) // Initial contents of RAM
) RAM16X1D_inst (
    .DPO(DPO),      // Read-only 1-bit data output for DPRA
    .SPO(SPO),      // Rw/ 1-bit data output for A0-A3
    .A0(A0),        // Rw/ address[0] input bit
    .A1(A1),        // Rw/ address[1] input bit
    .A2(A2),        // Rw/ address[2] input bit
    .A3(A3),        // Rw/ address[3] input bit
    .D(D),          // Write 1-bit data input
    .DPRA0(DPRA0),  // Read address[0] input bit
    .DPRA1(DPRA1),  // Read address[1] input bit
    .DPRA2(DPRA2),  // Read address[2] input bit
    .DPRA3(DPRA3),  // Read address[3] input bit
    .WCLK(WCLK),    // Write clock input
    .WE(WE)         // Write enable input
);

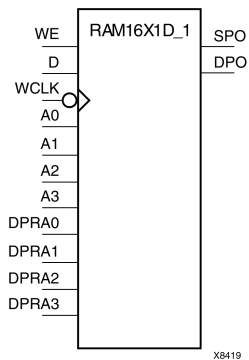
// End of RAM16X1D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAM16X1D_1

プリミティブ：16-Deep by 1-Wide Static Dual Port Synchronous RAM with Negative-Edge Clock



概要

このエレメントは、クロックのネガティブ エッジで動作する 16 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 (DPRA3 ~ DPRA0) と書き込みアドレス (A3 ~ 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 ポートは非同期です。読み出しアドレスによって出力ピン (DPO) に出力される値が指定され、書き込みアドレスによって書き込みを行う位置が指定されます。

ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High の場合、WCLK が High から Low に切り替わるときに、4 ビットの書き込みアドレスで選択されているワードにデータ入力 (D) の値が書き込まれます。書き込みを正しく行うには、WCLK が High から Low に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ Low ですが、インバーターを使用してアクティブ High にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

INIT 属性を使用すると、コンフィギュレーション中に RAM16X1D_1 を初期化できます。

SPO 出力には、A3 ~ 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、DPRA3 ~ 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

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	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 = A3 ~ A0 で指定されたワード				
data_d = DPRA3 ~ DPRA0 で指定されたワード				

ポートの説明

ポート名	方向	幅	機能
DPO	出力	1	読み出し専用 1 ビット データ出力
SPO	出力	1	読み出し/書き込み 1 ビット データ出力
A0	入力	1	読み出し/書き込み address[0] 入力
A1	入力	1	読み出し/書き込み address[1] 入力
A2	入力	1	読み出し/書き込み address[2] 入力
A3	入力	1	読み出し/書き込み address[3] 入力
D	入力	1	書き込み 1 ビット データ入力
DPRA0	入力	1	読み出し専用 address[0] 入力
DPRA1	入力	1	読み出し専用 address[1] 入力
DPRA2	入力	1	読み出し専用 address[2] 入力
DPRA3	入力	1	読み出し専用 address[3] 入力
WCLK	入力	1	書き込みクロック入力
WE	入力	1	書き込みイネーブル入力

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンス化)

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

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

-- RAM16X1D_1: 16 x 1 negative edge write, asynchronous read dual-port distributed RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAM16X1D_1_inst : RAM16X1D_1
generic map (
  INIT => X"0000")
port map (
  DPO => DPO,      -- Read-only 1-bit data output for DPRA
  SPO => SPO,      -- R/W 1-bit data output for A0-A3
  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
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
WCLK => WCLK,      -- Write clock input
WE => WE           -- Write enable input
);

-- End of RAM16X1D_1_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAM16X1D_1: 16 x 1 negative edge write, asynchronous read dual-port distributed RAM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAM16X1D_1 #(
    .INIT(16'h0000) // Initial contents of RAM
) RAM16X1D_1_inst (
    .DPO(DPO),      // Read-only 1-bit data output
    .SPO(SPO),      // Rw/ 1-bit data output
    .A0(A0),        // Rw/ address[0] input bit
    .A1(A1),        // Rw/ address[1] input bit
    .A2(A2),        // Rw/ address[2] input bit
    .A3(A3),        // Rw/ address[3] input bit
    .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
    .WCLK(WCLK),    // Write clock input
    .WE(WE)         // Write enable input
);

// End of RAM16X1D_1_inst instantiation

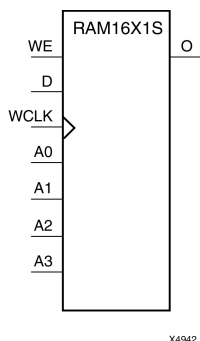
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAM16X1S

プリミティブ：16-Deep by 1-Wide Static Synchronous RAM



概要

このエレメントは 16 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わる時に、データ入力 (D) の値が 4 ビットのアドレス (A3 ~ A0) で選択されたワードに書き込まれます。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。INIT 属性を使用すると、コンフィギュレーション中に RAM16X1S を初期化できます。

論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D	D
1 (読み出し)	↓	X	データ
データ = A3 ~ A0 で指定されたワード			

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンスレーション)

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

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

-- RAM16X1S: 16 x 1 posedge write distributed (LUT) RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAM16X1S_inst : RAM16X1S
generic map (
  INIT => X"0000")
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
  D => D,          -- RAM data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM16X1S_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM16X1S: 16 x 1 posedge write distributed (LUT) RAM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAM16X1S #(
  .INIT(16'h0000) // Initial contents of RAM
) RAM16X1S_inst (
  .O(O),          // RAM output
  .A0(A0),        // RAM address[0] input
  .A1(A1),        // RAM address[1] input
  .A2(A2),        // RAM address[2] input
  .A3(A3),        // RAM address[3] input
  .D(D),          // RAM data input
  .WCLK(WCLK),    // Write clock input
  .WE(WE)         // Write enable input
);

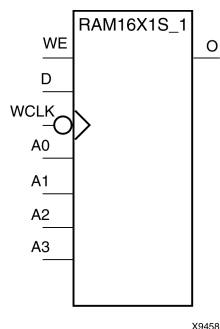
// End of RAM16X1S_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

RAM16X1S_1

プリミティブ : 16-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



概要

このエレメントはクロックの立ち下がりエッジで動作する 16 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 の場合、WCLK が High から Low に切り替わる時に、データ入力 (D) の値が 4 ビットのアドレス (A3 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が High から Low に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ Low ですが、インバーターを使用してアクティブ High にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

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

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

論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↓	D	D
1 (読み出し)	↑	X	データ
データ = A3 ~ A0 で指定されたワード			

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンスレーション)

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

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

-- RAM16X1S_1: 16 x 1 negedge write distributed (LUT) RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAM16X1S_1_inst : RAM16X1S_1
generic map (
  INIT => X"0000")
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
  D => D,       -- RAM data input
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);

-- End of RAM16X1S_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM16X1S_1: 16 x 1 negedge write distributed (LUT) RAM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAM16X1S_1 #(
  .INIT(16'h0000) // Initial contents of RAM
) RAM16X1S_1_inst (
  .O(O),          // RAM output
  .A0(A0),        // RAM address[0] input
  .A1(A1),        // RAM address[1] input
  .A2(A2),        // RAM address[2] input
  .A3(A3),        // RAM address[3] input
  .D(D),          // RAM data input
  .WCLK(WCLK),    // Write clock input
  .WE(WE)         // Write enable input
);

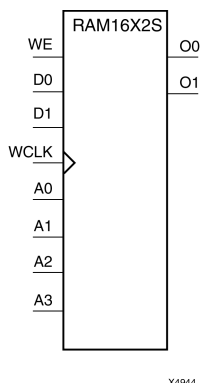
// End of RAM16X1S_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAM16X2S

プリミティブ : 16-Deep by 2-Wide Static Synchronous RAM



概要

このエレメントは 16 ワード X 2 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High の場合、WCLK が Low から High に切り替わるときに、入力 (D1 ~ D0) の値が 4 ビットのアドレス (A3 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O1 ~ O0) に出力される値は、アドレスピンで指定された RAM 内の位置に格納されている値です。

INIT_xx 属性を使用すると、RAM の初期値を指定できます。INIT_00 は出力 (O0) に対応する RAM のセルを初期化し、INIT_01 は出力 (O1) に対応するセルを初期化します。たとえば、RAM16X2S インスタンスは、INIT_00 および INIT_01 にそれぞれ 4 個の 16 進数値を指定して初期化します。RAM16X8S インスタンスは、INIT_00 ~ INIT_07 の 8 個の属性にそれぞれ 4 個の 16 進数値を指定して初期化します。RAM64X2S インスタンスは、INIT_00 および INIT_01 にそれぞれ 16 個の 16 進数値を指定して初期化します。

Virtex-4 デバイス以外では、このエレメントの初期値を直接指定することはできません。

論理表

入力			出力
WE (モード)	WCLK	D1 ~ D0	O1 ~ O0
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D1 ~ D0	D1 ~ D0
1 (読み出し)	↓	X	データ
データ = A3 ~ A0 で指定されたワード			

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_01	16 進数	16 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンス化)

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

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

-- RAM16X2S: 16 x 2 posedge write distributed (LUT) RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAM16X2S_inst : RAM16X2S
generic map (
  INIT_00 => X"0000", -- INIT for bit 0 of RAM
  INIT_01 => X"0000") -- INIT for bit 1 of RAM
port map (
  O0 => O0,      -- RAM data[0] output
  O1 => O1,      -- RAM data[1] output
  A0 => A0,      -- RAM address[0] input
  A1 => A1,      -- RAM address[1] input
  A2 => A2,      -- RAM address[2] input
  A3 => A3,      -- RAM address[3] input
  D0 => D0,      -- RAM data[0] input
  D1 => D1,      -- RAM data[1] input
  WCLK => WCLK,  -- Write clock input
  WE => WE       -- Write enable input
);

-- End of RAM16X2S_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// RAM16X2S: 16 x 2 posedge write distributed (LUT) RAM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAM16X2S #(
  .INIT_00(16'h0000), // Initial contents of bit 0 of RAM
  .INIT_01(16'h0000) // Initial contents of bit 1 of RAM
) RAM16X2S_inst (
  .O0(O0),           // RAM data[0] output
  .O1(O1),           // RAM data[1] output
  .A0(A0),           // RAM address[0] input
  .A1(A1),           // RAM address[1] input
  .A2(A2),           // RAM address[2] input
  .A3(A3),           // RAM address[3] input
  .D0(D0),           // RAM data[0] input
  .D1(D1),           // RAM data[1] input
  .WCLK(WCLK),       // Write clock input
```

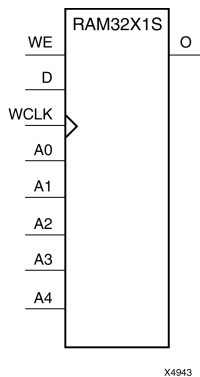
```
.WE(WE)      // Write enable input  
);  
  
// End of RAM16X2S_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAM32X1S

プリミティブ：32-Deep by 1-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 32 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わる時に、データ入力 (D) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。INIT 属性を使用すると、コンフィギュレーション中に RAM32X1S を初期化できます。

論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D	D
1 (読み出し)	↓	X	データ

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンスレーション)

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

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

-- RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAM32X1S_inst : RAM32X1S
generic map (
  INIT => X"00000000")
port map (
  O => O,          -- RAM output
  A0 => A0,         -- RAM address[0] input
  A1 => A1,         -- RAM address[1] input
  A2 => A2,         -- RAM address[2] input
  A3 => A3,         -- RAM address[3] input
  A4 => A4,         -- RAM address[4] input
  D => D,          -- RAM data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM32X1S_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAM32X1S #(
  .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1S_inst (
  .O(O),              // RAM output
  .A0(A0),            // RAM address[0] input
  .A1(A1),            // RAM address[1] input
  .A2(A2),            // RAM address[2] input
  .A3(A3),            // RAM address[3] input
  .A4(A4),            // RAM address[4] input
  .D(D),              // RAM data input
  .WCLK(WCLK),        // Write clock input
  .WE(WE)             // Write enable input
);

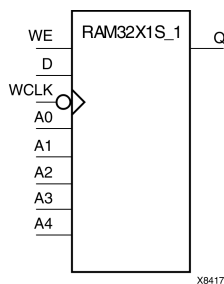
// End of RAM32X1S_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAM32X1S_1

プリミティブ：32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



概要

このデザイン エLEMENTは 32 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が High から Low に切り替わるときに、データ入力 (D) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が High から Low に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ Low ですが、インバーターを使用してアクティブ High にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。INIT 属性を使用すると、コンフィギュレーション中に RAM32X1S_1 を初期化できます。

論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↓	D	D
1 (読み出し)	↑	X	データ
データ = A4 ~ A0 で指定されたワード			

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	0	RAM の初期値を指定

VHDL 記述 (インスタンスレーション)

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

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

-- RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAM32X1S_1_inst : RAM32X1S_1
generic map (
  INIT => X"00000000")
port map (
  O => O,          -- RAM output
  A0 => A0,         -- RAM address[0] input
  A1 => A1,         -- RAM address[1] input
  A2 => A2,         -- RAM address[2] input
  A3 => A3,         -- RAM address[3] input
  A4 => A4,         -- RAM address[4] input
  D => D,          -- RAM data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM32X1S_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAM32X1S_1 #(
  .INIT(32'h00000000) // Initial contents of RAM
)RAM32X1S_1_inst (
  .O(O),              // RAM output
  .A0(A0),            // RAM address[0] input
  .A1(A1),            // RAM address[1] input
  .A2(A2),            // RAM address[2] input
  .A3(A3),            // RAM address[3] input
  .A4(A4),            // RAM address[4] input
  .D(D),              // RAM data input
  .WCLK(WCLK),        // Write clock input
  .WE(WE)             // Write enable input
);

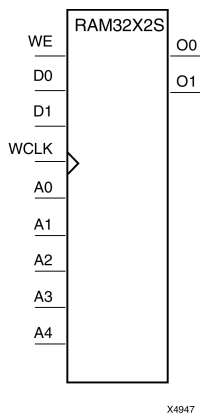
// End of RAM32X1S_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAM32X2S

プリミティブ：32-Deep by 2-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 32 ワード X 2 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、入力 (D1 ~ D0) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。出力ピン (O1 ~ O0) に出力される値は、アドレスピンで指定された RAM 内の位置に格納されている値です。

INIT_00 および INIT_01 属性を使用して RAM32X2S の初期値を指定できます。

論理表

入力			出力
WE (モード)	WCLK	D	O0 ~ O1
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D1 ~ D0	D1 ~ D0
1 (読み出し)	↓	X	データ
データ = A4 ~ A0 で指定されたワード			

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT_00	16 進数	32 ビット値	すべてゼロ	RAM のビット 0 の初期値を指定
INIT_01	16 進数	32 ビット値	すべてゼロ	RAM のビット 1 の初期値を指定

VHDL 記述 (インスタンス化)

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

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

-- RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAM32X2S_inst : RAM32X2S
generic map (
  INIT_00 => X"00000000", -- INIT for bit 0 of RAM
  INIT_01 => X"00000000") -- INIT for bit 1 of RAM
port map (
  O0 => O0,      -- RAM data[0] output
  O1 => O1,      -- RAM data[1] output
  A0 => A0,      -- RAM address[0] input
  A1 => A1,      -- RAM address[1] input
  A2 => A2,      -- RAM address[2] input
  A3 => A3,      -- RAM address[3] input
  A4 => A4,      -- RAM address[4] input
  D0 => D0,      -- RAM data[0] input
  D1 => D1,      -- RAM data[1] input
  WCLK => WCLK,  -- Write clock input
  WE => WE       -- Write enable input
);

-- End of RAM32X2S_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAM32X2S #(
  .INIT_00(32'h00000000), // INIT for bit 0 of RAM
  .INIT_01(32'h00000000) // INIT for bit 1 of RAM
) RAM32X2S_inst (
  .O0(O0),      // RAM data[0] output
  .O1(O1),      // RAM data[1] output
  .A0(A0),      // RAM address[0] input
  .A1(A1),      // RAM address[1] input
  .A2(A2),      // RAM address[2] input
  .A3(A3),      // RAM address[3] input
  .A4(A4),      // RAM address[4] input
  .D0(D0),      // RAM data[0] input
  .D1(D1),      // RAM data[1] input
  .WCLK(WCLK),  // Write clock input
  .WE(WE)       // Write enable input
);

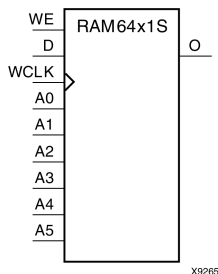
// End of RAM32X2S_inst instantiation
```


詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

RAM64X1S

プリミティブ：64-Deep by 1-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 64 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 6 ビットのアドレス (A5 ～ A0) で選択されたワードに書き込まれます。WCLK はデフォルトではアクティブ High ですが、インバーターを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

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

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

論理表

モード選択を次の論理表に示します。

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

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンスレーション)

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

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

-- RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAM64X1S_inst : RAM64X1S
generic map (
  INIT => X"0000000000000000")
port map (
  O => O,          -- 1-bit data output
  A0 => A0,         -- Address[0] input bit
  A1 => A1,         -- Address[1] input bit
  A2 => A2,         -- Address[2] input bit
  A3 => A3,         -- Address[3] input bit
  A4 => A4,         -- Address[4] input bit
  A5 => A5,         -- Address[5] input bit
  D => D,          -- 1-bit data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM64X1S_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAM64X1S #(
  .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_inst (
  .O(O),          // 1-bit data output
  .A0(A0),        // Address[0] input bit
  .A1(A1),        // Address[1] input bit
  .A2(A2),        // Address[2] input bit
  .A3(A3),        // Address[3] input bit
  .A4(A4),        // Address[4] input bit
  .A5(A5),        // Address[5] input bit
  .D(D),          // 1-bit data input
  .WCLK(WCLK),    // Write clock input
  .WE(WE)         // Write enable input
);

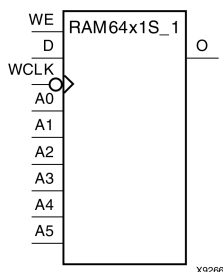
// End of RAM64X1S_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAM64X1S_1

プリミティブ：64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



概要

このデザイン エLEMENTは 64 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が High から Low に切り替わるときに、データ入力 (D) の値が 6 ビットのアドレス (A5 ~ A0) で選択されたワードに書き込まれます。書き込みを正しく行うには、WCLK が High から Low に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ Low ですが、インバーターを使用してアクティブ High にすることもできます。WCLK の入力ネットに配置されたインバーターは、RAM ブロック内に組み込まれます。

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンスレーション)

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

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

-- RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAM64X1S_1_inst : RAM64X1S_1
generic map (
  INIT => X"0000000000000000")
port map (
  O => O,          -- 1-bit data output
  A0 => A0,         -- Address[0] input bit
  A1 => A1,         -- Address[1] input bit
  A2 => A2,         -- Address[2] input bit
  A3 => A3,         -- Address[3] input bit
  A4 => A4,         -- Address[4] input bit
  A5 => A5,         -- Address[5] input bit
  D => D,          -- 1-bit data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM64X1S_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAM64X1S_1 #(
  .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_1_inst (
  .O(O),          // 1-bit data output
  .A0(A0),        // Address[0] input bit
  .A1(A1),        // Address[1] input bit
  .A2(A2),        // Address[2] input bit
  .A3(A3),        // Address[3] input bit
  .A4(A4),        // Address[4] input bit
  .A5(A5),        // Address[5] input bit
  .D(D),          // 1-bit data input
  .WCLK(WCLK),    // Write clock input
  .WE(WE)         // Write enable input
);

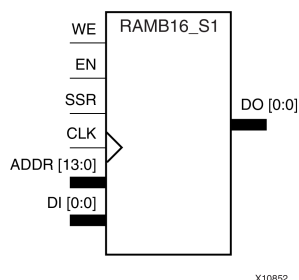
// End of RAM64X1S_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S1

プリミティブ : 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 1-bit Port



概要

このデザイン エLEMENTは、同期書き込み機能を備えた専用 RAM ブロックです。ブロック RAM のポートには 16384 ビットのデータ メモリがあります。セルの構成を、次の表に示します。

データ セル		パリティ セル				
ワード数	幅	ワード数	幅	アドレス バス	データ バス	パリティ バス
16384	1	–	–	(13:0)	(0:0)	–

イネーブル ピン EN は、読み出し、書き込み、リセットを制御します。EN が Low の場合、データは書き込まれず、出力 (DO および DOP) の値は変化しません。EN とリセット (SSR) が High の場合、クロック (CLK) が Low から High に切り替わる時に、DO および DOP が SRVAL にセットされます。ライト イネーブル (WE) も High の場合は、DI および DIP の値が RAM に書き込まれます。SSR と WE が Low で EN が High の場合、クロックが Low から High に切り替わる時に、RAM アドレス (ADDR) に格納されているデータが DO および DOP に出力されます。デフォルトでは WRITE_MODE=WRITE_FIRST に設定されており、SSR が Low で EN および WE が High の場合、クロックが Low から High に切り替わる時に、データ入力 (DI および DIP) のデータが書き込みアドレス (ADDR) で選択されているワードに書き込まれ、データ出力 (DO および DOP) にその値が出力されます。

前述の説明では、制御ピン (EN、WE、SSR、CLK) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

論理表

入力								出力			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT	INIT	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR=グローバル セットリセット信号

INIT = データ メモリ用に INIT 属性で設定された値。デフォルトはすべて 0 です。

SRVAL = SRVAL 属性で設定された SSR のアサート後の値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹ WRITE_MODE=NO_CHANGE

² WRITE_MODE=READ_FIRST

³ WRITE_MODE=WRITE_FIRST

初期化

メモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16 の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティメモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティメモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。シングルポート RAMB16 の出力レジスタの初期化に使用する属性には、INIT と SRVAL の 2 種類があります。INIT 属性は、電源投入時の出力レジスタの値を指定します。SRVAL 属性は、SSR (セット/リセット) 入力のアサートしたときの値を指定します。

INIT および SRVAL 属性は、初期化値を、出力ポートの各ビットを 1 ビット含む 16 進数文字列として指定します。たとえば、ポート幅が 1 の RAMB16_S1 の場合、出力レジスタは 1 ビットなので、INIT および SRVAL には 1 または 0 を指定します。ポート幅が 4 の RAMB16_S4 の場合は、出力レジスタは 4 ビットなので、0 ~ F の 16 進数値を指定します。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT または SRVAL の値の上位ビットで指定します。

書き込みモードの選択

WRITE_MODE 属性は、RAMB16 メモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE は WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。WRITE_MODE を READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。WRITE_MODE を NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DO 出力ポートの初期値を指定。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定
SRVAL	2 進数/ 16 進数	任意の値	すべてゼロ	SSR ピンをアサートすると、DO 出力ポートがセット (1) またはリセット (0) に設定されるよう指定できます。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
WRITE_MODE	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	書き込みコマンドが DO ポートで実行されるときポートの動作を指定します。 ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> ・ "NO_CHANGE": 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1: Single-Port RAM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1_inst : RAMB16_S1
generic map (
  INIT => X"0", -- Value of output RAM registers at startup
  SRVAL => X"0", -- Output value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 4095
  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",
  -- Address 4096 to 8191
  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",
  -- Address 8192 to 12287
  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",
-- Address 12288 to 16383
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")
port map (
DO => DO,      -- 1-bit Data Output
ADDR => ADDR,  -- 14-bit Address Input
CLK => CLK,    -- Clock
DI => DI,      -- 1-bit Data Input
EN => EN,      -- RAM Enable Input
SSR => SSR,    -- Synchronous Set/Reset Input
WE => WE       -- Write Enable Input
);

-- End of RAMB16_S1_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAMB16_S1: 16Kx1 Single-Port RAM
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1 #(
    .INIT(1'b0), // Value of output RAM registers at startup
    .SRVAL(1'b0), // Output value upon SSR assertion
    .WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 4095
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Address 4096 to 8191
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),

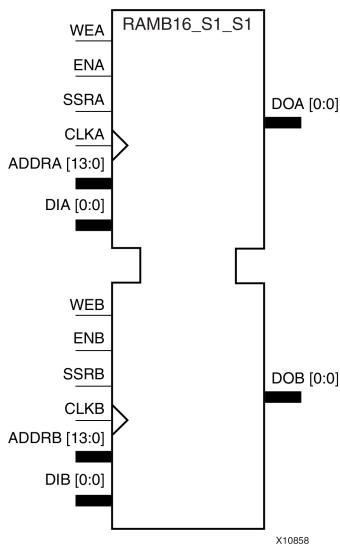
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S1_S1

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S1_S1	16384 x 1	—	(13:0)	(0:0)	—	16384 x 1	—	(13:0)	(0:0)	—

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるときに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるときに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるときに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるときに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるときに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるときに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マッピング

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリセルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A は 1 または 0 にしか設定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリセルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリセルにアクセスできます。ただし、一方のポートがあるメモリセルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリセルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE"：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S1: 16k x 1 Dual-Port RAM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1_S1_inst : RAMB16_S1_S1
generic map (
  INIT_A => "0", -- Value of output RAM registers on Port A at startup
  INIT_B => "0", -- Value of output RAM registers on Port B at startup
  SRVAL_A => "0", -- Port A output value upon SSR assertion
  SRVAL_B => "0", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 4095
  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",
  -- Address 4096 to 8191
  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",

```

```
-- Address 8192 to 12287
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",
-- Address 12288 to 16383
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",
port map (
  DOA => DOA,      -- Port A 1-bit Data Output
  DOB => DOB,      -- Port B 1-bit Data Output
  ADDRA => ADDRA,   -- Port A 14-bit Address Input
  ADDR_B => ADDR_B, -- Port B 14-bit Address Input
  CLKA => CLKA,     -- Port A Clock
  CLKB => CLKB,     -- Port B Clock
  DIA => DIA,      -- Port A 1-bit Data Input
  DIB => DIB,      -- Port B 1-bit Data Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- Port B RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S1_S1_inst instantiation
```

Verilog 記述 (インスタンスエーション)

```
// RAMB16_S1_S1: 16k x 1 Dual-Port RAM
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1_S1 #(
  .INIT_A(1'b0), // Value of output RAM registers on Port A at startup
  .INIT_B(1'b0), // Value of output RAM registers on Port B at startup
  .SRVAL_A(1'b0), // Port A output value upon SSR assertion
  .SRVAL_B(1'b0), // Port B output value upon SSR assertion
  .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

  // The following INIT_xx declarations specify the initial contents of the RAM
  // Address 0 to 4095
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日

```
.DIA(DIA),      // Port A 1-bit Data Input
.DIB(DIB),      // Port B 1-bit Data Input
.ENA(ENA),      // Port A RAM Enable Input
.ENB(ENB),      // Port B RAM Enable Input
.SSRA(SSRA),    // Port A Synchronous Set/Reset Input
.SSRB(SSRB),    // Port B Synchronous Set/Reset Input
.WEA(WEA),      // Port A Write Enable Input
.WEB(WEB)       // Port B Write Enable Input
);

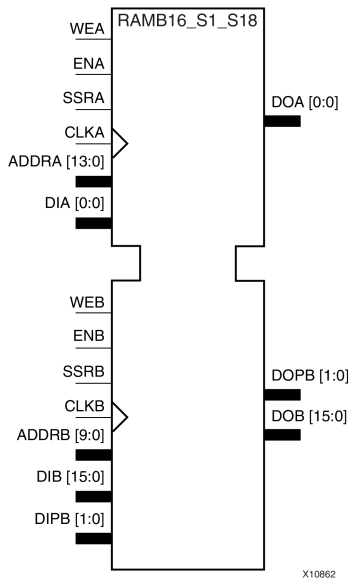
// End of RAMB16_S1_S1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S1_S18

プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 18-bit Ports



概要

このデザイン エレメントは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このエレメントのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDR_A	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S1_S18	16384 x 1	—	(13:0)	(0:0)	—	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マッピング

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリセルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																			
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0			
4	4096	<--	7				6				5				4				3				2				1				0					
8	2048	<--	3								2								1								0									
16	1024	<--	1																0																	
32	512	<--	0																																	

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリセルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリセルにアクセスできます。ただし、一方のポートがあるメモリセルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリセルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化なし	X	変化なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	変化なし	変化なし	変化なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE"：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S18: 16k/1k x 1/16 + 0/2 Parity bits Dual-Port RAM
--
-- Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1_S18_inst : RAMB16_S1_S18
generic map (
    INIT_A => "0", -- Value of output RAM registers on Port A at startup
    INIT_B => X"00000", -- Value of output RAM registers on Port B at startup
    SRVAL_A => "0", -- Port A output value upon SSR assertion
    SRVAL_B => X"00000", -- Port B output value upon SSR assertion
    WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
    WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
    SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
    -- The following INIT_xx declarations specify the initial contents of the RAM
    -- Port A Address 0 to 4095, Port B Address 0 to 255
    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",
    -- Port A Address 4096 to 8191, Port B Address 256 to 511
    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",

```

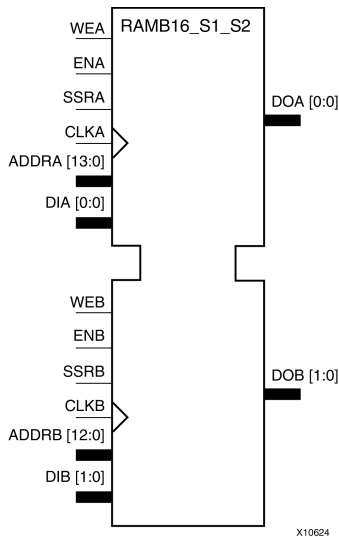

Verilog 記述 (インスタンス化)

Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日

Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日

RAMB16_S1_S2

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 2-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号
 INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。
 SRVAL_A = レジスタの値
 addr = RAM アドレス
 RAM(addr) = アドレス ADDR の RAM の内容
 data = RAM の入力データ
 pdata = RAM のパリティ データ
¹WRITE_MODE_A=NO_CHANGE
²WRITE_MODE_A=READ_FIRST
³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S1_S2	16384 x 1	—	(13:0)	(0:0)	—	8192 x 2	—	(12:0)	(1:0)	—

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan-3A の場合、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロックの SelectRAM は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化なし	X	変化なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	変化なし	変化なし	変化なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S2: 16k/8k x 1/2 Dual-Port RAM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1_S2_inst : RAMB16_S1_S2
generic map (
  INIT_A => "0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"0", -- Value of output RAM registers on Port B at startup
  SRVAL_A => "0", -- Port A output value upon SSR assertion
  SRVAL_B => X"0", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 4095, Port B Address 0 to 2047
  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",
  -- Port A Address 4096 to 8191, Port B Address 2048 to 4095
  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",

```

```

--Port A Address 8192 to 12287, Port B Address 4095 to 6143
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",
-- Port A Address 12288 to 16383, Port B Address 6144 to 8091
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")
port map (
  DOA => DOA,      -- Port A 1-bit Data Output
  DOB => DOB,      -- Port B 2-bit Data Output
  ADDRA => ADDRA,   -- Port A 14-bit Address Input
  ADDR_B => ADDR_B, -- Port B 13-bit Address Input
  CLKA => CLKA,     -- Port A Clock
  CLKB => CLKB,     -- Port B Clock
  DIA => DIA,      -- Port A 1-bit Data Input
  DIB => DIB,      -- Port B 2-bit Data Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- Port B RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S1_S2_inst instantiation

```

Verilog 記述 (インスタンスエーション)

```

// RAMB16_S1_S2: 16k/8k x 1/2 Dual-Port RAM
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1_S2 #(
  .INIT_A(1'b0),    // Value of output RAM registers on Port A at startup
  .INIT_B(2'b00),   // Value of output RAM registers on Port B at startup
  .SRVAL_A(1'b0),   // Port A output value upon SSR assertion
  .SRVAL_B(2'b00),  // Port B output value upon SSR assertion
  .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

  // The following INIT_xx declarations specify the initial contents of the RAM
  // Port A Address 0 to 4095, Port B Address 0 to 2047
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

```
) RAMB16_S1_S2_inst (
    .DOA(DOA),           // Port A 1-bit Data Output
    .DOB DOB),           // Port B 2-bit Data Output
    .ADDRA(ADDRA),       // Port A 14-bit Address Input
    .ADDRB(ADDRB),       // Port B 13-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
```

```
.DIA(DIA),      // Port A 1-bit Data Input
.DIB(DIB),      // Port B 2-bit Data Input
.ENA(ENA),      // Port A RAM Enable Input
.ENB(ENB),      // Port B RAM Enable Input
.SSRA(SSRA),    // Port A Synchronous Set/Reset Input
.SSRB(SSRB),    // Port B Synchronous Set/Reset Input
.WEA(WEA),      // Port A Write Enable Input
.WEB(WEB)       // Port B Write Enable Input
);

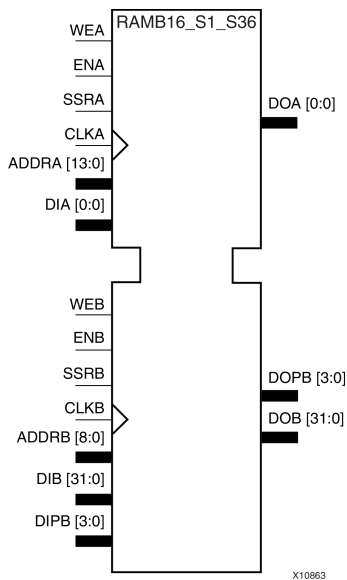
// End of RAMB16_S1_S2_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S1_S36

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 36-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレ ス バス	データ バス	パリティ バス
RAMB16_S1_S36	16384 x 1	—	(13:0)	(0:0)	—	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マッピング

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリセルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリセルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリセルにアクセスできます。ただし、一方のポートがあるメモリセルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリセルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化なし	X	変化なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	変化なし	変化なし	変化なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE"：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S36: 16k/512 x 1/32 + 0/4 Parity bits Dual-Port RAM
--                Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1_S36_inst : RAMB16_S1_S36
generic map (
  INIT_A => "0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"000000000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => "0", -- Port A output value upon SSR assertion
  SRVAL_B => X"000000000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 4095, Port B Address 0 to 127
  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",
  -- Port A Address 4096 to 8191, Port B Address 128 to 255
  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",

```



```
-- Port A Address 8192 to 12287, Port B Address 256 to 383
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",
-- Port A Address 12288 to 16383, Port B Address 384 to 512
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
--Port B Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 384 to 512
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 1-bit Data Output
  DOB => DOB,      -- Port B 32-bit Data Output
  DOPB => DOPB,    -- Port B 4-bit Parity Output
  ADDRA => ADDRA,  -- Port A 14-bit Address Input
  ADDRb => ADDRb,  -- Port B 9-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- Port A 1-bit Data Input
  DIB => DIB,      -- Port B 32-bit Data Input
  DIPB => DIPB,    -- Port-B 4-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- PortB RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S1_S36_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// RAMB16_S1_S36: 16k/512 x 1/32 + 0/4 Parity bits Dual-Port RAM
```

Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日

```
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port B Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 384 to 512
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S1_S36_inst (
.DOA(DOA), // Port A 1-bit Data Output
.DOB DOB), // Port B 32-bit Data Output
.DOPB(DOPB), // Port B 4-bit Parity Output
.ADDRA(ADDRA), // Port A 14-bit Address Input
.ADDRB(ADDRB), // Port B 9-bit Address Input
.CLKA(CLKA), // Port A Clock
.CLKB(CLKB), // Port B Clock
.DIA(DIA), // Port A 1-bit Data Input
.DIB(DIB), // Port B 32-bit Data Input
.DIPB(DIPB), // Port-B 4-bit parity Input
.ENA(ENA), // Port A RAM Enable Input
.ENB(ENB), // Port B RAM Enable Input
.SSRA(SSRA), // Port A Synchronous Set/Reset Input
.SSRB(SSRB), // Port B Synchronous Set/Reset Input
.WEA(WEA), // Port A Write Enable Input
.WEB(WEB) // Port B Write Enable Input
);

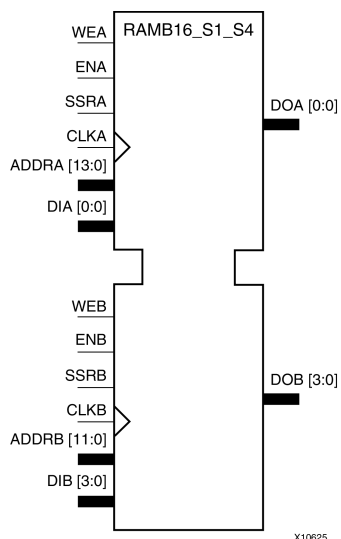
// End of RAMB16 S1 S36 inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S1_S4

プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 4-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S1_S4	16384 x 1	—	(13:0)	(0:0)	—	4096 x 4	—	(11:0)	(3:0)	—

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1															0																	
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリセルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリセルにアクセスできます。ただし、一方のポートがあるメモリセルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリセルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化なし	X	変化なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	変化なし	変化なし	変化なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE"：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S4: 16k/4k x 1/4 Dual-Port RAM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1_S4_inst : RAMB16_S1_S4
generic map (
  INIT_A => "0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"0", -- Value of output RAM registers on Port B at startup
  SRVAL_A => "0", -- Port A output value upon SSR assertion
  SRVAL_B => X"0", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 4095, Port B Address 0 to 1023
  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",
  -- Port A Address 4096 to 8191, Port B Address 1024 to 2047
  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",

```

```

-- Port A Address 8192 to 12287, Port B Address 2048 to 3071
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",
-- Port A Address 12288 to 16383, Port B Address 3072 to 4095
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")
port map (
  DOA => DOA,      -- Port A 1-bit Data Output
  DOB => DOB,      -- Port B 4-bit Data Output
  ADDRA => ADDRA,   -- Port A 14-bit Address Input
  ADDR_B => ADDR_B, -- Port B 12-bit Address Input
  CLKA => CLKA,     -- Port A Clock
  CLKB => CLKB,     -- Port B Clock
  DIA => DIA,      -- Port A 1-bit Data Input
  DIB => DIB,      -- Port B 4-bit Data Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- Port B RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S1_S4_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAMB16_S1_S4: 16k/4k x 1/4 Dual-Port RAM
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1_S4 #(
  .INIT_A(1'b0), // Value of output RAM registers on Port A at startup
  .INIT_B(4'h0), // Value of output RAM registers on Port B at startup
  .SRVAL_A(1'b0), // Port A output value upon SSR assertion
  .SRVAL_B(4'h0), // Port B output value upon SSR assertion
  .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

  // The following INIT_xx declarations specify the initial contents of the RAM
  // Port A Address 0 to 4095, Port B Address 0 to 1023
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

```
) RAMB16_S1_S4_inst (
    .DOA(DOA),           // Port A 1-bit Data Output
    .DOB DOB),           // Port B 4-bit Data Output
    .ADDRA(ADDRA),       // Port A 14-bit Address Input
    .ADDRB(ADDRB),       // Port B 12-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
```

```
.DIA(DIA),      // Port A 1-bit Data Input
.DIB(DIB),      // Port B 4-bit Data Input
.ENA(ENA),      // Port A RAM Enable Input
.ENB(ENB),      // Port B RAM Enable Input
.SSRA(SSRA),    // Port A Synchronous Set/Reset Input
.SSRB(SSRB),    // Port B Synchronous Set/Reset Input
.WEA(WEA),      // Port A Write Enable Input
.WEB(WEB)       // Port B Write Enable Input
);

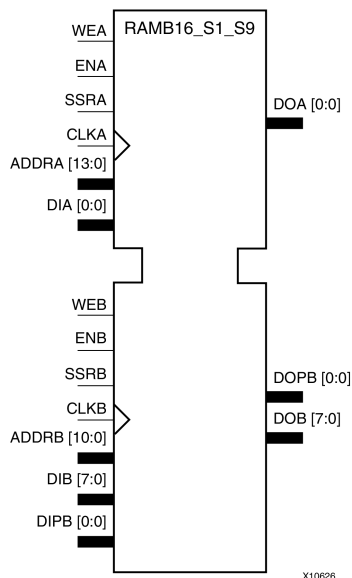
// End of RAMB16_S1_S4_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S1_S9

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 9-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDR_A	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号
 INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。
 SRVAL_A = レジスタの値
 addr = RAM アドレス
 RAM(addr) = アドレス ADDR の RAM の内容
 data = RAM の入力データ
 pdata = RAM のパリティ データ
¹WRITE_MODE_A=NO_CHANGE
²WRITE_MODE_A=READ_FIRST
³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S1_S9	16384 x 1	—	(13:0)	(0:0)	—	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるときに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるときに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるときに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるときに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるときに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるときに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マッピング

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリセルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A の場合、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力のアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力のアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロックの SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化なし	X	変化なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	変化なし	変化なし	変化なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE"：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S9: 16k/2k x 1/8 + 0/1 Parity bit Dual-Port RAM
--
-- Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S1_S9_inst : RAMB16_S1_S9
generic map (
  INIT_A => "0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => "0", -- Port A output value upon SSR assertion
  SRVAL_B => X"000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 4095, Port B Address 0 to 511
  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",
  -- Port A Address 4096 to 8191, Port B Address 512 to 1023
  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",

```



```
-- Port A Address 8192 to 12287, Port B Address 1024 to 1535
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",
-- Port A Address 12288 to 16383, Port B Address 1535 to 2047
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port B Address 0 to 511
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 512 to 1023
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 1024 to 1535
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 1535 to 2047
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 1-bit Data Output
  DOB => DOB,      -- Port B 8-bit Data Output
  DOPB => DOPB,    -- Port B 1-bit Parity Output
  ADDRA => ADDRA,  -- Port A 14-bit Address Input
  ADDRb => ADDRb,  -- Port B 11-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- Port A 1-bit Data Input
  DIB => DIB,      -- Port B 8-bit Data Input
  DIPB => DIPB,    -- Port-B 1-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- PortB RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S1_S9_inst instantiation
```

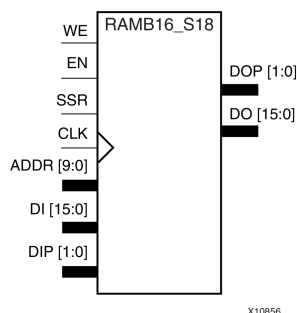
Verilog 記述 (インスタンス化)

```
// RAMB16_S1_S9: 16k/2k x 1/8 + 0/1 Parity bit Dual-Port RAM
```

Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日

RAMB16_S18

プリミティブ : 16K-bit Data + 2K-bit Parity Memory, Single-Port Synchronous Block RAM with 18-bit Port



概要

このデザイン エLEMENTは、同期書き込み機能を備えた専用 RAM ブロックです。ブロック RAM のポートには 16384 ビットのデータ メモリがあります。セルの構成を、次の表に示します。

データ セル		パリティ セル				
ワード数	幅	ワード数	幅	アドレス バス	データ バス	パリティ バス
1024	16	1024	2	(9:0)	(15:0)	(1:0)

イネーブル ピン EN は、読み出し、書き込み、リセットを制御します。EN が Low の場合、データは書き込まれず、出力 (DO および DOP) の値は変化しません。EN とリセット (SSR) が High の場合、クロック (CLK) が Low から High に切り替わる時に、DO および DOP が SRVAL にセットされます。ライト イネーブル (WE) も High の場合は、DI および DIP の値が RAM に書き込まれます。SSR と WE が Low で EN が High の場合、クロックが Low から High に切り替わる時に、RAM アドレス (ADDR) に格納されているデータが DO および DOP に出力されます。デフォルトでは WRITE_MODE=WRITE_FIRST に設定されており、SSR が Low で EN および WE が High の場合、クロックが Low から High に切り替わる時に、データ入力 (DI および DIP) のデータが書き込みアドレス (ADDR) で選択されているワードに書き込まれ、データ出力 (DO および DOP) にその値が出力されます。

前述の説明では、制御ピン (EN、WE、SSR、CLK) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

論理表

入力								出力			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT	INIT	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR=グローバル セットリセット信号

INIT = データ メモリ用に INIT 属性で設定された値。デフォルトはすべて 0 です。

SRVAL = SRVAL 属性で設定された SSR のアサート後の値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹ WRITE_MODE=NO_CHANGE

² WRITE_MODE=READ_FIRST

³ WRITE_MODE=WRITE_FIRST

初期化

メモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16 の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティメモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティメモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。シングルポート RAMB16 の出力レジスタの初期化に使用する属性には、INIT と SRVAL の 2 種類があります。INIT 属性は、電源投入時の出力レジスタの値を指定します。SRVAL 属性は、SSR (セット/リセット) 入力のアサートしたときの値を指定します。

INIT および SRVAL 属性は、初期化値を、出力ポートの各ビットを 1 ビット含む 16 進数文字列として指定します。たとえば、ポート幅が 1 の RAMB16_S1 の場合、出力レジスタは 1 ビットなので、INIT および SRVAL には 1 または 0 を指定します。ポート幅が 4 の RAMB16_S4 の場合は、出力レジスタは 4 ビットなので、0 ~ F の 16 進数値を指定します。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT または SRVAL の値の上位ビットで指定します。

書き込みモードの選択

WRITE_MODE 属性は、RAMB16 メモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE は WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。WRITE_MODE を READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。WRITE_MODE を NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DO 出力ポートの初期値を指定。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定
SRVAL	2 進数/ 16 進数	任意の値	すべてゼロ	SSR ピンをアサートすると、DO 出力ポートがセット (1) またはリセット (0) に設定されるよう指定できます。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
WRITE_MODE	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	書き込みコマンドが DO ポートで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"：値がポートに書き込まれてから出力されます。 "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> ・ "NO_CHANGE": 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S18: 1k x 16 + 2 Parity bits Single-Port RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S18_inst : RAMB16_S18
generic map (
  INIT => X"00000", -- Value of output RAM registers at startup
  SRVAL => X"00000", -- Output value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 255
  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",
  -- Address 256 to 511
  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",
  -- Address 512 to 767
  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",
-- Address 768 to 1023
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 255
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 511
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 512 to 767
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 768 to 1023
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DO => DO,      -- 16-bit Data Output
DOP => DOP,     -- 2-bit parity Output
ADDR => ADDR,   -- 10-bit Address Input
CLK => CLK,     -- Clock
DI => DI,       -- 16-bit Data Input
DIP => DIP,     -- 2-bit parity Input
EN => EN,       -- RAM Enable Input
SSR => SSR,     -- Synchronous Set/Reset Input
WE => WE        -- Write Enable Input
);

-- End of RAMB16_S18_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAMB16_S18: 1k x 16 + 2 Parity bits Single-Port RAM
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S18 #(
    .INIT(18'h00000), // Value of output RAM registers at startup
    .SRVAL(18'h000000), // Output value upon SSR assertion
    .WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 255
    .INIT_00(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_01(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_02(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_03(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),

```


Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日

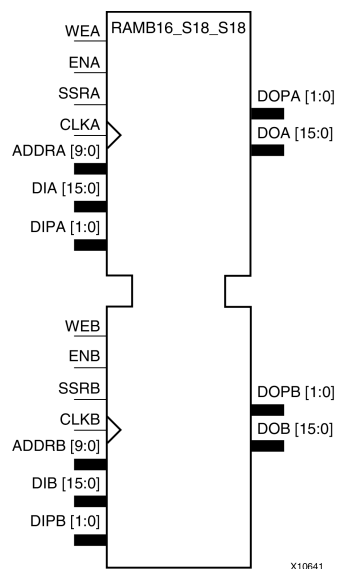
```
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),  
// Address 768 to 1023  
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),  
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)  
);  
RAMB16_S18_inst (  
    .DO(DO),           // 16-bit Data Output  
    .DOP(DOP),         // 2-bit parity Output  
    .ADDR(ADDR),       // 10-bit Address Input  
    .CLK(CLK),         // Clock  
    .DI(DI),           // 16-bit Data Input  
    .DIP(DIP),         // 2-bit parity Input  
    .EN(EN),           // RAM Enable Input  
    .SSR(SSR),         // Synchronous Set/Reset Input  
    .WE(WE)            // Write Enable Input  
);  
  
// End of RAMB16_S18_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S18_S18

プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレ ス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレ ス バス	データ バス	パリティ バス
RAMB16_S18_S18	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるときに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるときに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるときに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるときに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるときに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるときに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マッピング

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリセルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリセルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリセルにアクセスできます。ただし、一方のポートがあるメモリセルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリセルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S18_S18: 1k x 16 + 2 Parity bits Dual-Port RAM
--                Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S18_S18_inst : RAMB16_S18_S18
generic map (
  INIT_A => X"00000", -- Value of output RAM registers on Port A at startup
  INIT_B => X"00000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"00000", -- Port A output value upon SSR assertion
  SRVAL_B => X"00000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 255
  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",
  -- Address 256 to 511
  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",

```

268 <http://japan.xilinx.com> Spartan-3 ライブラリ ガイド (HDL 用) UG607 (v14.1) 2012 年 4 月 24 日

Verilog 記述 (インスタンス化)

```
// RAMB16_S18_S18: 1k x 16 + 2 Parity bits Dual-Port RAM
//                      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S18_S18 #(
    .INIT_A(18'h00000), // Value of output RAM registers on Port A at startup
    .INIT_B(18'h00000), // Value of output RAM registers on Port B at startup
    .SRVAL_A(18'h00000), // Port A output value upon SSR assertion
    .SRVAL_B(18'h00000), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 255
    .INIT_00(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_01(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_02(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_03(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_04(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_05(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_06(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_07(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_08(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_09(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    // Address 256 to 511
    .INIT_10(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_11(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_12(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_13(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_14(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_15(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_16(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_17(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_18(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_19(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    // Address 512 to 767
    .INIT_20(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_21(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_22(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_23(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_24(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_25(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_26(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_27(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_28(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_29(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_2A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_2B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_2C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_2D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_2E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_2F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    // Address 768 to 1023
    .INIT_30(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_31(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_32(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_33(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
```

```
.INIT_34(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_35(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_36(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_37(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_38(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_39(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),

// The next set of INITP_xx are for the parity bits
// Address 0 to 255
.INITP_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 256 to 511
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 512 to 767
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 768 to 1023
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S18_S18_inst (
.DOA(DOA), // Port A 16-bit Data Output
.DOB DOB), // Port B 16-bit Data Output
.DOPA(DOPA), // Port A 2-bit Parity Output
.DOPB(DOPB), // Port B 2-bit Parity Output
.ADDRA(ADDR_A), // Port A 10-bit Address Input
.ADDRB(ADDR_B), // Port B 10-bit Address Input
.CLKA(CLKA), // Port A Clock
.CLKB(CLKB), // Port B Clock
.DIA(DIA), // Port A 16-bit Data Input
.DIB(DIB), // Port B 16-bit Data Input
.DIPA(DIPA), // Port A 2-bit parity Input
.DIPB(DIPB), // Port-B 2-bit parity Input
.ENA(ENA), // Port A RAM Enable Input
.ENB(ENB), // Port B RAM Enable Input
.SSRA(SSRA), // Port A Synchronous Set/Reset Input
.SSRB(SSRB), // Port B Synchronous Set/Reset Input
.WEA(WEA), // Port A Write Enable Input
.WEB(WEB) // Port B Write Enable Input
);

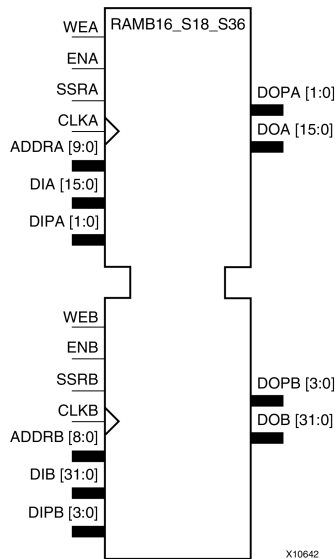
// End of RAMB16_S18_S18_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S18_S36

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit and 36-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレ ス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレ ス バス	データ バス	パリティ バス
RAMB16_S18_S36	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マッピング

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリセルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリセルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリセルにアクセスできます。ただし、一方のポートがあるメモリセルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリセルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S18_S36: 1k/512 x 16/32 + 2/4 Parity bits Dual-Port RAM
--                      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S18_S36_inst : RAMB16_S18_S36
generic map (
  INIT_A => X"00000", -- Value of output RAM registers on Port A at startup
  INIT_B => X"000000000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"00000", -- Port A output value upon SSR assertion
  SRVAL_B => X"000000000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 255, Port B Address 0 to 127
  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",
  -- Port A Address 256 to 511, Port B Address 128 to 255
  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",

```

```
-- Port A Address 512 to 767, Port B Address 256 to 383
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",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port A Address 0 to 255, Port B Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 16-bit Data Output
  DOB => DOB,      -- Port B 32-bit Data Output
  DOPA => DOPA,    -- Port A 2-bit Parity Output
  DOPB => DOPB,    -- Port B 4-bit Parity Output
  ADDRA => ADDRA,  -- Port A 10-bit Address Input
  ADDR8 => ADDR8,  -- Port B 9-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- Port A 16-bit Data Input
  DIB => DIB,      -- Port B 32-bit Data Input
  DIPA => DIPA,    -- Port A 2-bit parity Input
  DIPB => DIPB,    -- Port-B 4-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- PortB RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S18_S36_inst instantiation
```


[illegible]

```
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 255, Port B Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 256 to 511, Port B Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 767, Port B Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 768 to 1023, Port B Address 384 to 511
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S18_S36_inst (
.DOA(DOA),           // Port A 16-bit Data Output
.DOB DOB),           // Port B 32-bit Data Output
.DOPA(DOPA),         // Port A 2-bit Parity Output
.DOPB(DOPB),         // Port B 4-bit Parity Output
.ADDRA(ADDRA),       // Port A 10-bit Address Input
.ADDRB(ADDRB),       // Port B 9-bit Address Input
.CLKA(CLKA),         // Port A Clock
.CLKB(CLKB),         // Port B Clock
.DIA(DIA),           // Port A 16-bit Data Input
.DIB(DIB),           // Port B 32-bit Data Input
.DIPA(DIPA),         // Port A 2-bit parity Input
.DIPB(DIPB),         // Port-B 4-bit parity Input
.ENA(ENA),           // Port A RAM Enable Input
.ENB(ENB),           // Port B RAM Enable Input
.SSRA(SSRA),         // Port A Synchronous Set/Reset Input
.SSRB(SSRB),        // Port B Synchronous Set/Reset Input
.WEA(WEA),           // Port A Write Enable Input
.WEB(WEB),           // Port B Write Enable Input
);

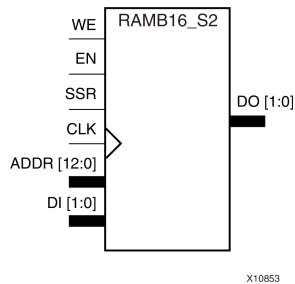
// End of RAMB16_S18_S36_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

RAMB16_S2

プリミティブ：16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 2-bit Port



概要

このデザイン エLEMENTは、同期書き込み機能を備えた専用 RAM ブロックです。ブロック RAM のポートには 16384 ビットのデータ メモリがあります。セルの構成を、次の表に示します。

データ セル		パリティ セル				
ワード数	幅	ワード数	幅	アドレス バス	データ バス	パリティ バス
8192	2	–	–	(12:0)	(1:0)	–

イネーブル ピン EN は、読み出し、書き込み、リセットを制御します。EN が Low の場合、データは書き込まれず、出力 (DO および DOP) の値は変化しません。EN とリセット (SSR) が High の場合、クロック (CLK) が Low から High に切り替わるときに、DO および DOP が SRVAL にセットされます。ライト イネーブル (WE) も High の場合は、DI および DIP の値が RAM に書き込まれます。SSR と WE が Low で EN が High の場合、クロックが Low から High に切り替わるときに、RAM アドレス (ADDR) に格納されているデータが DO および DOP に出力されます。デフォルトでは WRITE_MODE=WRITE_FIRST に設定されており、SSR が Low で EN および WE が High の場合、クロックが Low から High に切り替わるときに、データ入力 (DI および DIP) のデータが書き込みアドレス (ADDR) で選択されているワードに書き込まれ、データ出力 (DO および DOP) にその値が出力されます。

前述の説明では、制御ピン (EN、WE、SSR、CLK) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

論理表

入力								出力			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT	INIT	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR=グローバル セットリセット信号

INIT = データ メモリ用に INIT 属性で設定された値。デフォルトはすべて 0 です。

SRVAL = SRVAL 属性で設定された SSR のアサート後の値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹ WRITE_MODE=NO_CHANGE

² WRITE_MODE=READ_FIRST

³ WRITE_MODE=WRITE_FIRST

初期化

メモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16 の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティメモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティメモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。シングルポート RAMB16 の出力レジスタの初期化に使用する属性には、INIT と SRVAL の 2 種類があります。INIT 属性は、電源投入時の出力レジスタの値を指定します。SRVAL 属性は、SSR (セット/リセット) 入力のアサートしたときの値を指定します。

INIT および SRVAL 属性は、初期化値を、出力ポートの各ビットを 1 ビット含む 16 進数文字列として指定します。たとえば、ポート幅が 1 の RAMB16_S1 の場合、出力レジスタは 1 ビットなので、INIT および SRVAL には 1 または 0 を指定します。ポート幅が 4 の RAMB16_S4 の場合は、出力レジスタは 4 ビットなので、0 ~ F の 16 進数値を指定します。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT または SRVAL の値の上位ビットで指定します。

書き込みモードの選択

WRITE_MODE 属性は、RAMB16 メモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE は WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。WRITE_MODE を READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。WRITE_MODE を NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DO 出力ポートの初期値を指定。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定
SRVAL	2 進数/ 16 進数	任意の値	すべてゼロ	SSR ピンをアサートすると、DO 出力ポートがセット (1) またはリセット (0) に設定されるよう指定できます。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
WRITE_MODE	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	書き込みコマンドが DO ポートで実行されるときのパートの動作を指定します。 <ul style="list-style-type: none"> "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。

属性	データ型	値	デフォルト	説明
				・ "NO_CHANGE": 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2: 8k x 2 Single-Port RAM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S2_inst : RAMB16_S2
generic map (
  INIT => X"0", -- Value of output RAM registers at startup
  SRVAL => X"0", -- Output value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 2047
  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",
  -- Address 2048 to 4095
  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",
  -- Address 4096 to 6143
  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",
-- Address 6143 to 8191
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")
port map (
  DO => DO,      -- 2-bit Data Output
  ADDR => ADDR,   -- 13-bit Address Input
  CLK => CLK,     -- Clock
  DI => DI,       -- 2-bit Data Input
  EN => EN,       -- RAM Enable Input
  SSR => SSR,     -- Synchronous Set/Reset Input
  WE => WE        -- Write Enable Input
);

-- End of RAMB16_S2_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// RAMB16_S2: 8k x 2 Single-Port RAM
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S2 #(
  .INIT(2'b00), // Value of output RAM registers at startup
  .SRVAL(2'b00), // Output value upon SSR assertion
  .WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

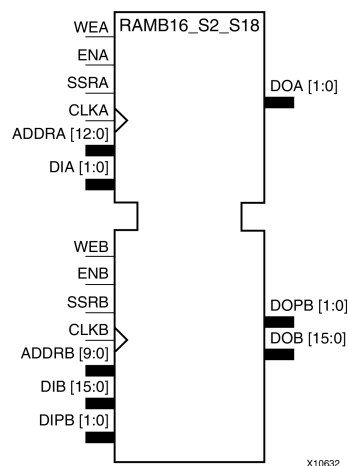
  // The following INIT_xx declarations specify the initial contents of the RAM
  // Address 0 to 2047
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
  // Address 2048 to 4095
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S2_S18

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 18-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号
INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。
SRVAL_B = レジスタの値
addr = RAM アドレス
RAM(addr) = アドレス ADDR の RAM の内容
data = RAM の入力データ
pdata = RAM のパリティ データ
¹WRITE_MODE_B=NO_CHANGE
²WRITE_MODE_B=READ_FIRST
³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S2_S18	8192 x 2	—	(12:0)	(1:0)	—	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロックの SelectIO™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S18: 8k/1k x 2/16 + 0/2 Parity bit Dual-Port RAM
-- Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S2_S18_inst : RAMB16_S2_S18
generic map (
  INIT_A => X"0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"0", -- Port A output value upon SSR assertion
  SRVAL_B => X"000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 2047, Port B Address 0 to 255
  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",
  -- Port A Address 2048 to 4095, Port B Address 256 to 511
  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",

```

```

-- Port A Address 4096 to 6143, Port B Address 512 to 767
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",
-- Port A Address 6144 to 8191, Port B Address 768 to 1023
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port B Address 0 to 255
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 256 to 511
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 512 to 767
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 768 to 1023
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 2-bit Data Output
  DOB => DOB,      -- Port B 16-bit Data Output
  DOPB => DOPB,    -- Port B 2-bit Parity Output
  ADDRA => ADDRA,  -- Port A 13-bit Address Input
  ADDRb => ADDRb,  -- Port B 10-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- Port A 2-bit Data Input
  DIB => DIB,      -- Port B 16-bit Data Input
  DIPB => DIPB,    -- Port-B 2-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- PortB RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S2_S18_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// RAMB16_S2_S18: 8k/1k x 2/16 + 0/2 Parity bits Dual-Port RAM
```

299

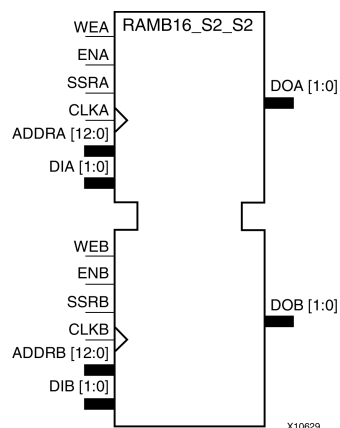
```
.INIT_38(256'h0000000000000000000000000000000000000000000000000),  
.INIT_39(256'h0000000000000000000000000000000000000000000000000),  
.INIT_3A(256'h0000000000000000000000000000000000000000000000000),  
.INIT_3B(256'h0000000000000000000000000000000000000000000000000),  
.INIT_3C(256'h0000000000000000000000000000000000000000000000000),  
.INIT_3D(256'h0000000000000000000000000000000000000000000000000),  
.INIT_3E(256'h0000000000000000000000000000000000000000000000000),  
.INIT_3F(256'h0000000000000000000000000000000000000000000000000),  
  
// The next set of INITP_xx are for the parity bits  
// Port B Address 0 to 255  
.INITP_00(256'h0000000000000000000000000000000000000000000000000),  
.INITP_01(256'h0000000000000000000000000000000000000000000000000),  
// Port B Address 256 to 511  
.INITP_02(256'h0000000000000000000000000000000000000000000000000),  
.INITP_03(256'h0000000000000000000000000000000000000000000000000),  
// Port B Address 512 to 767  
.INITP_04(256'h0000000000000000000000000000000000000000000000000),  
.INITP_05(256'h0000000000000000000000000000000000000000000000000),  
// Port B Address 768 to 1023  
.INITP_06(256'h0000000000000000000000000000000000000000000000000),  
.INITP_07(256'h0000000000000000000000000000000000000000000000000)  
) RAMB16_S2_S18_inst (  
    .DOA(DOA),        // Port A 2-bit Data Output  
    .DOB DOB),        // Port B 16-bit Data Output  
    .DOPB DOPB),      // Port B 2-bit Parity Output  
    .ADDRA(ADDR_A),   // Port A 13-bit Address Input  
    .ADDRB(ADDR_B),   // Port B 10-bit Address Input  
    .CLKA(CLK_A),     // Port A Clock  
    .CLKB(CLK_B),     // Port B Clock  
    .DIA(DI_A),       // Port A 2-bit Data Input  
    .DIB(DI_B),       // Port B 16-bit Data Input  
    .DIPB(DIP_B),     // Port-B 2-bit parity Input  
    .ENA(EN_A),       // Port A RAM Enable Input  
    .ENB(EN_B),       // Port B RAM Enable Input  
    .SSRA(SSR_A),     // Port A Synchronous Set/Reset Input  
    .SSRB(SSR_B),     // Port B Synchronous Set/Reset Input  
    .WEA(WEA),        // Port A Write Enable Input  
    .WEB(WEB),        // Port B Write Enable Input  
);  
  
// End of RAMB16_S2_S18_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S2_S2

プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、RAM(addr) ² 、data ³	変化なし ¹ 、RAM(addr) ² 、pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S2_S2	8192 x 2	—	(12:0)	(1:0)	—	8192 x 2	—	(12:0)	(1:0)	—

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロックの SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S2: 8k x 2 Dual-Port RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S2_S2_inst : RAMB16_S2_S2
generic map (
  INIT_A => X"0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"0", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"0", -- Port A output value upon SSR assertion
  SRVAL_B => X"0", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 2047
  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",
  -- Address 2048 to 4095
  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",

```

```

-- Address 4096 to 6143
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",
-- Address 6143 to 8191
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")
port map (
  DOA => DOA,      -- Port A 2-bit Data Output
  DOB => DOB,      -- Port B 2-bit Data Output
  ADDRA => ADDRA,   -- Port A 13-bit Address Input
  ADDR_B => ADDR_B, -- Port B 13-bit Address Input
  CLKA => CLKA,     -- Port A Clock
  CLKB => CLKB,     -- Port B Clock
  DIA => DIA,      -- Port A 2-bit Data Input
  DIB => DIB,      -- Port B 2-bit Data Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- Port B RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S2_S2_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAMB16_S2_S2: 8k x 2 Dual-Port RAM
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S2_S2 #(
  .INIT_A(2'b00), // Value of output RAM registers on Port A at startup
  .INIT_B(2'b00), // Value of output RAM registers on Port B at startup
  .SRVAL_A(2'b00), // Port A output value upon SSR assertion
  .SRVAL_B(2'b00), // Port B output value upon SSR assertion
  .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

  // Address 0 to 2047
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日

```
.DIB(DIB),      // Port B 2-bit Data Input
.ENA(ENA),      // Port A RAM Enable Input
.ENB(ENB),      // Port B RAM Enable Input
.SSRA(SSRA),    // Port A Synchronous Set/Reset Input
.SSRB(SSRB),    // Port B Synchronous Set/Reset Input
.WEA(WEA),      // Port A Write Enable Input
.WEB(WEB)       // Port B Write Enable Input
);

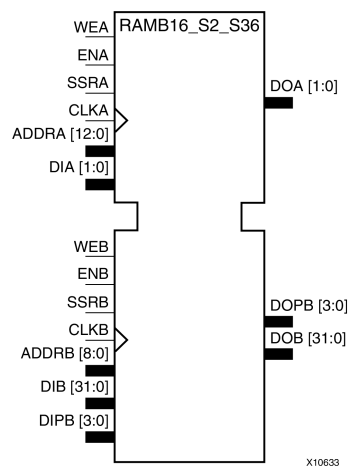
// End of RAMB16_S2_S2_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S2_S36

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 36-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S2_S36	8192 x 2	—	(12:0)	(1:0)	—	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロックの SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S36: 8k/512 x 2/32 + 0/4 Parity bits Dual-Port RAM
--                Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S2_S36_inst : RAMB16_S2_S36
generic map (
    INIT_A => X"0", -- Value of output RAM registers on Port A at startup
    INIT_B => X"000000000", -- Value of output RAM registers on Port B at startup
    SRVAL_A => X"0", -- Port A output value upon SSR assertion
    SRVAL_B => X"000000000", -- Port B output value upon SSR assertion
    WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
    WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
    SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
    -- The following INIT_xx declarations specify the initial contents of the RAM
    -- Port A Address 0 to 2047, Port B Address 0 to 127
    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",
    -- Port A Address 2048 to 4095, Port B Address 128 to 255
    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",

```

```

-- Port A Address 4096 to 6143, Port B Address 256 to 383
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",
-- Port A Address 6144 to 8191, Port B Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port B Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 2-bit Data Output
  DOB => DOB,      -- Port B 32-bit Data Output
  DOPB => DOPB,    -- Port B 4-bit Parity Output
  ADDRA => ADDRA,  -- Port A 13-bit Address Input
  ADDRb => ADDRb,  -- Port B 9-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- Port A 2-bit Data Input
  DIB => DIB,      -- Port B 32-bit Data Input
  DIPB => DIPB,    -- Port-B 4-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- PortB RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S2_S36_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// RAMB16_S2_S36: 8k/512 x 2/32 + 0/4 Parity bits Dual-Port RAM
```



```
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port B Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 384 to 511
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S2_S36_inst (
.DOA(DOA), // Port A 2-bit Data Output
.DOB DOB), // Port B 32-bit Data Output
.DOPB(DOPB), // Port B 4-bit Parity Output
.ADDRA(ADDR A), // Port A 13-bit Address Input
.ADDRB(ADDRB), // Port B 9-bit Address Input
.CLKA(CLKA), // Port A Clock
.CLKB(CLKB), // Port B Clock
.DIA(DIA), // Port A 2-bit Data Input
.DIB(DIB), // Port B 32-bit Data Input
.DIPB(DIPB), // Port-B 4-bit parity Input
.ENA(ENA), // Port A RAM Enable Input
.ENB(ENB), // Port B RAM Enable Input
.SSRA(SSRA), // Port A Synchronous Set/Reset Input
.SSRB(SSRB), // Port B Synchronous Set/Reset Input
.WEA(WEA), // Port A Write Enable Input
.WEB(WEB) // Port B Write Enable Input
);

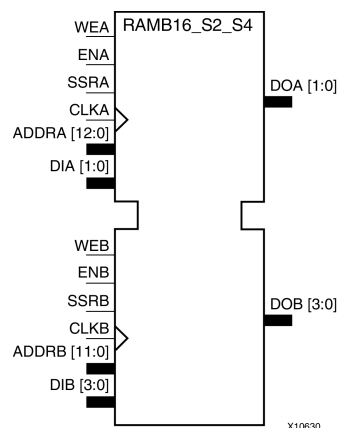
// End of RAMB16_S2_S36_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

RAMB16_S2_S4

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 4-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S2_S4	8192 x 2	—	(12:0)	(1:0)	—	4096 x 4	—	(11:0)	(3:0)	—

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロックの SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S4: 8k/4k x 2/4 Dual-Port RAM
-- Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S2_S4_inst : RAMB16_S2_S4
generic map (
  INIT_A => X"0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"0", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"0", -- Port A output value upon SSR assertion
  SRVAL_B => X"0", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 2047, Port B Address 0 to 1023
  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",
  -- Port A Address 2048 to 4095, Port B Address 1024 to 2047
  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",

```

```

-- Port A Address 4096 to 6143, Port B Address 2048 to 3071
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",
-- Port A Address 6144 to 8191, Port B Address 3072 to 4095
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")
port map (
  DOA => DOA,      -- Port A 2-bit Data Output
  DOB => DOB,      -- Port B 4-bit Data Output
  ADDRA => ADDRA,   -- Port A 13-bit Address Input
  ADDR_B => ADDR_B, -- Port B 12-bit Address Input
  CLKA => CLKA,     -- Port A Clock
  CLKB => CLKB,     -- Port B Clock
  DIA => DIA,      -- Port A 2-bit Data Input
  DIB => DIB,      -- Port B 4-bit Data Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- Port B RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S2_S4_inst instantiation

```

Verilog 記述 (インスタンスエーション)

```

// RAMB16_S2_S4: 8k/4k x 2/4 Dual-Port RAM
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S2_S4 #(
  .INIT_A(2'b00), // Value of output RAM registers on Port A at startup
  .INIT_B(4'h0),  // Value of output RAM registers on Port B at startup
  .SRVAL_A(2'b00), // Port A output value upon SSR assertion
  .SRVAL_B(4'h0),  // Port B output value upon SSR assertion
  .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .SIM_COLLISION_CHECK("ALL"),  // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

  // The following INIT_xx declarations specify the initial contents of the RAM
  // Port A Address 0 to 2047, Port B Address 0 to 1023
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

335

```
.DIA(DIA),      // Port A 2-bit Data Input
.DIB(DIB),      // Port B 4-bit Data Input
.ENA(ENA),      // Port A RAM Enable Input
.ENB(ENB),      // Port B RAM Enable Input
.SSRA(SSRA),    // Port A Synchronous Set/Reset Input
.SSRB(SSRB),    // Port B Synchronous Set/Reset Input
.WEA(WEA),      // Port A Write Enable Input
.WEB(WEB)       // Port B Write Enable Input
);

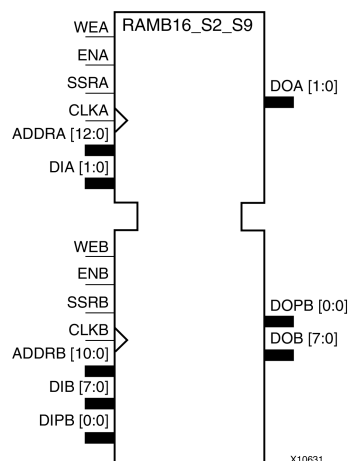
// End of RAMB16_S2_S4_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S2_S9

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 9-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S2_S9	8192 x 2	—	(12:0)	(1:0)	—	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan®-3A ブロックの SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S9: 8k/2k x 2/8 + 0/1 Parity bit Dual-Port RAM
--              Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S2_S9_inst : RAMB16_S2_S9
generic map (
  INIT_A => X"0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"0", -- Port A output value upon SSR assertion
  SRVAL_B => X"000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 2047, Port B Address 0 to 511
  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",
  -- Port A Address 2048 to 4095, Port B Address 512 to 1023
  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",

```

```

-- Port A Address 4096 to 6143, Port B Address 1024 to 1535
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",
-- Port A Address 6144 to 8191, Port B Address 1536 to 2047
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port B Address 0 to 511
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 512 to 1023
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 1024 to 1535
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 6144 to 8191, Port B Address 1536 to 2047
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 2-bit Data Output
  DOB => DOB,      -- Port B 8-bit Data Output
  DOPB => DOPB,    -- Port B 1-bit Parity Output
  ADDRA => ADDRA,  -- Port A 13-bit Address Input
  ADDRb => ADDRb,  -- Port B 11-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- Port A 2-bit Data Input
  DIB => DIB,      -- Port B 8-bit Data Input
  DIPB => DIPB,    -- Port-B 1-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- Port B RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S2_S9_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// RAMB16_S2_S9: 8k/2k x 2/8 + 0/1 Parity bit Dual-Port RAM
```

347

```
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port B Address 0 to 511
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 512 to 1023
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 1024 to 1535
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 1536 to 2047
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S2_S9_inst (
.DOA(DOA), // Port A 2-bit Data Output
.DOB DOB), // Port B 8-bit Data Output
.DOPB(DOPB), // Port B 1-bit Parity Output
.ADDRA(ADDRA), // Port A 13-bit Address Input
.ADDRB(ADDRB), // Port B 11-bit Address Input
.CLKA(CLKA), // Port A Clock
.CLKB(CLKB), // Port B Clock
.DIA(DIA), // Port A 2-bit Data Input
.DIB(DIB), // Port B 8-bit Data Input
.DIPB(DIPB), // Port-B 1-bit parity Input
.ENA(ENA), // Port A RAM Enable Input
.ENB(ENB), // Port B RAM Enable Input
.SSRA(SSRA), // Port A Synchronous Set/Reset Input
.SSRB(SSRB), // Port B Synchronous Set/Reset Input
.WEA(WEA), // Port A Write Enable Input
.WEB(WEB) // Port B Write Enable Input
);

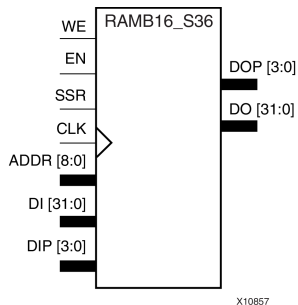
// End of RAMB16_S2_S9_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S36

プリミティブ：16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 36-bit Port



概要

このデザイン エLEMENTは、同期書き込み機能を備えた専用 RAM ブロックです。ブロック RAM のポートには 16384 ビットのデータ メモリがあります。セルの構成を、次の表に示します。

データ セル		パリティ セル				
ワード数	幅	ワード数	幅	アドレス バス	データ バス	パリティ バス
512	32	512	4	(8:0)	(31:0)	(3:0)

イネーブル ピン EN は、読み出し、書き込み、リセットを制御します。EN が Low の場合、データは書き込まれず、出力 (DO および DOP) の値は変化しません。EN とリセット (SSR) が High の場合、クロック (CLK) が Low から High に切り替わるときに、DO および DOP が SRVAL にセットされます。ライト イネーブル (WE) も High の場合は、DI および DIP の値が RAM に書き込まれます。SSR と WE が Low で EN が High の場合、クロックが Low から High に切り替わるときに、RAM アドレス (ADDR) に格納されているデータが DO および DOP に出力されます。デフォルトでは WRITE_MODE=WRITE_FIRST に設定されており、SSR が Low で EN および WE が High の場合、クロックが Low から High に切り替わるときに、データ入力 (DI および DIP) のデータが書き込みアドレス (ADDR) で選択されているワードに書き込まれ、データ出力 (DO および DOP) にその値が出力されます。

前述の説明では、制御ピン (EN、WE、SSR、CLK) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

論理表

入力								出力			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT	INIT	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR=グローバル セットリセット信号

INIT = データ メモリ用に INIT 属性で設定された値。デフォルトはすべて 0 です。

SRVAL = SRVAL 属性で設定された SSR のアサート後の値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹ WRITE_MODE=NO_CHANGE

² WRITE_MODE=READ_FIRST

³ WRITE_MODE=WRITE_FIRST

初期化

メモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16 の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティメモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティメモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。シングルポート RAMB16 の出力レジスタの初期化に使用する属性には、INIT と SRVAL の 2 種類があります。INIT 属性は、電源投入時の出力レジスタの値を指定します。SRVAL 属性は、SSR (セット/リセット) 入力のアサートしたときの値を指定します。

INIT および SRVAL 属性は、初期化値を、出力ポートの各ビットを 1 ビット含む 16 進数文字列として指定します。たとえば、ポート幅が 1 の RAMB16_S1 の場合、出力レジスタは 1 ビットなので、INIT および SRVAL には 1 または 0 を指定します。ポート幅が 4 の RAMB16_S4 の場合は、出力レジスタは 4 ビットなので、0 ~ F の 16 進数値を指定します。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT または SRVAL の値の上位ビットで指定します。

書き込みモードの選択

WRITE_MODE 属性は、RAMB16 メモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE は WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。WRITE_MODE を READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。WRITE_MODE を NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DO 出力ポートの初期値を指定。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定
SRVAL	2 進数/ 16 進数	任意の値	すべてゼロ	SSR ピンをアサートすると、DO 出力ポートがセット (1) またはリセット (0) に設定されるよう指定できます。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
WRITE_MODE	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	書き込みコマンドが DO ポートで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"：値がポートに書き込まれてから出力されます。 "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。


```

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",
-- Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DO => DO,      -- 32-bit Data Output
DOP => DOP,    -- 4-bit parity Output
ADDR => ADDR,  -- 9-bit Address Input
CLK => CLK,    -- Clock
DI => DI,      -- 32-bit Data Input
DIP => DIP,    -- 4-bit parity Input
EN => EN,      -- RAM Enable Input
SSR => SSR,    -- Synchronous Set/Reset Input
WE => WE       -- Write Enable Input
);

-- End of RAMB16_S36_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAMB16_S36: 512 x 32 + 4 Parity bits Single-Port RAM
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S36 #(
    .INIT(36'h00000000), // Value of output RAM registers at startup
    .SRVAL(36'h00000000), // Output value upon SSR assertion
    .WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

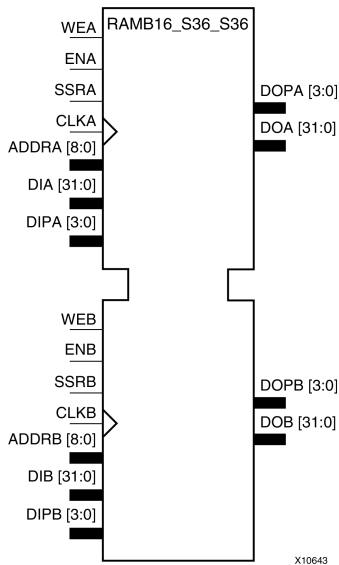
    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 127
    .INIT_00(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_01(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_02(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_03(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),

```

Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日

RAMB16_S36_S36

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with Two 36-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポート には、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレ ス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレ ス バス	データ バス	パリティ バス
RAMB16_S36_S36	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マッピング

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリセルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリセルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリセルにアクセスできます。ただし、一方のポートがあるメモリセルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリセルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化なし	X	変化なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	変化なし	変化なし	変化なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE"：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S36_S36: 512 x 32 + 4 Parity bits Dual-Port RAM
--                Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S36_S36_inst : RAMB16_S36_S36
generic map (
  INIT_A => X"0000000000", -- Value of output RAM registers on Port A at startup
  INIT_B => X"0000000000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"0000000000", -- Port A output value upon SSR assertion
  SRVAL_B => X"0000000000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 127
  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",
  -- Address 128 to 255
  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",

```

```
-- Address 256 to 383
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",
-- Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 32-bit Data Output
  DOB => DOB,      -- Port B 32-bit Data Output
  DOPA => DOPA,    -- Port A 4-bit Parity Output
  DOPB => DOPB,    -- Port B 4-bit Parity Output
  ADDRA => ADDRA,  -- Port A 9-bit Address Input
  ADDR8 => ADDR8,  -- Port B 9-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- Port A 32-bit Data Input
  DIB => DIB,      -- Port B 32-bit Data Input
  DIPA => DIPA,    -- Port A 4-bit parity Input
  DIPB => DIPB,    -- Port-B 4-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- PortB RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S36_S36_inst instantiation
```

Verilog 記述 (インスタンス化)

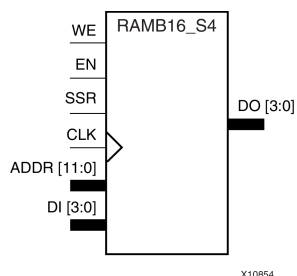
```
// RAMB16_S36_S36: 512 x 32 + 4 Parity bits Dual-Port RAM
//                      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S36_S36 #(
    .INIT_A(36'h00000000), // Value of output RAM registers on Port A at startup
    .INIT_B(36'h00000000), // Value of output RAM registers on Port B at startup
    .SRVAL_A(36'h00000000), // Port A output value upon SSR assertion
    .SRVAL_B(36'h00000000), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 127
    .INIT_00(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_01(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_02(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_03(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_04(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_05(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_06(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_07(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_08(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_09(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    // Address 128 to 255
    .INIT_10(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_11(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_12(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_13(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_14(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_15(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_16(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_17(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_18(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_19(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    // Address 256 to 383
    .INIT_20(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_21(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_22(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_23(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_24(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_25(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_26(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_27(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_28(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_29(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_2A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_2B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_2C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_2D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_2E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_2F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    // Address 384 to 511
    .INIT_30(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_31(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_32(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_33(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
```


RAMB16_S4

プリミティブ : 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 4-bit Port



概要

このデザイン エLEMENTは、同期書き込み機能を備えた専用 RAM ブロックです。ブロック RAM のポートには 16384 ビットのデータ メモリがあります。セルの構成を、次の表に示します。

データ セル		パリティ セル				
ワード数	幅	ワード数	幅	アドレス バス	データ バス	パリティ バス
4096	4	–	–	(11:0)	(3:0)	–

イネーブル ピン EN は、読み出し、書き込み、リセットを制御します。EN が Low の場合、データは書き込まれず、出力 (DO および DOP) の値は変化しません。EN とリセット (SSR) が High の場合、クロック (CLK) が Low から High に切り替わるときに、DO および DOP が SRVAL にセットされます。ライト イネーブル (WE) も High の場合は、DI および DIP の値が RAM に書き込まれます。SSR と WE が Low で EN が High の場合、クロックが Low から High に切り替わるときに、RAM アドレス (ADDR) に格納されているデータが DO および DOP に出力されます。デフォルトでは WRITE_MODE=WRITE_FIRST に設定されており、SSR が Low で EN および WE が High の場合、クロックが Low から High に切り替わるときに、データ入力 (DI および DIP) のデータが書き込みアドレス (ADDR) で選択されているワードに書き込まれ、データ出力 (DO および DOP) にその値が出力されます。

前述の説明では、制御ピン (EN、WE、SSR、CLK) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

論理表

入力								出力			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT	INIT	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR=グローバル セットリセット信号

INIT = データ メモリ用に INIT 属性で設定された値。デフォルトはすべて 0 です。

SRVAL = SRVAL 属性で設定された SSR のアサート後の値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹ WRITE_MODE=NO_CHANGE

² WRITE_MODE=READ_FIRST

³ WRITE_MODE=WRITE_FIRST

初期化

メモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16 の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティメモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティメモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。シングルポート RAMB16 の出力レジスタの初期化に使用する属性には、INIT と SRVAL の 2 種類があります。INIT 属性は、電源投入時の出力レジスタの値を指定します。SRVAL 属性は、SSR (セット/リセット) 入力のアサートしたときの値を指定します。

INIT および SRVAL 属性は、初期化値を、出力ポートの各ビットを 1 ビット含む 16 進数文字列として指定します。たとえば、ポート幅が 1 の RAMB16_S1 の場合、出力レジスタは 1 ビットなので、INIT および SRVAL には 1 または 0 を指定します。ポート幅が 4 の RAMB16_S4 の場合は、出力レジスタは 4 ビットなので、0 ~ F の 16 進数値を指定します。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT または SRVAL の値の上位ビットで指定します。

書き込みモードの選択

WRITE_MODE 属性は、RAMB16 メモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE は WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。WRITE_MODE を READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。WRITE_MODE を NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DO 出力ポートの初期値を指定。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定
SRVAL	2 進数/ 16 進数	任意の値	すべてゼロ	SSR ピンをアサートすると、DO 出力ポートがセット (1) またはリセット (0) に設定されるよう指定できます。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
WRITE_MODE	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	書き込みコマンドが DO ポートで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"：値がポートに書き込まれてから出力されます。 "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> ・ "NO_CHANGE": 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S4: 4k x 4 Single-Port RAM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S4_inst : RAMB16_S4
generic map (
  INIT => X"0", -- Value of output RAM registers at startup
  SRVAL => X"0", -- Output value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 1023
  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",
  -- Address 1024 to 2047
  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",
  -- Address 2048 to 3071
  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",
-- Address 3072 to 4095
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")
port map (
  DO => DO,      -- 4-bit Data Output
  ADDR => ADDR,  -- 12-bit Address Input
  CLK => CLK,    -- Clock
  DI => DI,      -- 4-bit Data Input
  EN => EN,      -- RAM Enable Input
  SSR => SSR,    -- Synchronous Set/Reset Input
  WE => WE       -- Write Enable Input
);

-- End of RAMB16_S4_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAMB16_S4: 4k x 4 Single-Port RAM
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S4 #(
  .INIT(4'h0), // Value of output RAM registers at startup
  .SRVAL(4'h0), // Output value upon SSR assertion
  .WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

  // The following INIT_xx declarations specify the initial contents of the RAM
  // Address 0 to 1023
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
  // Address 1024 to 2047
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),

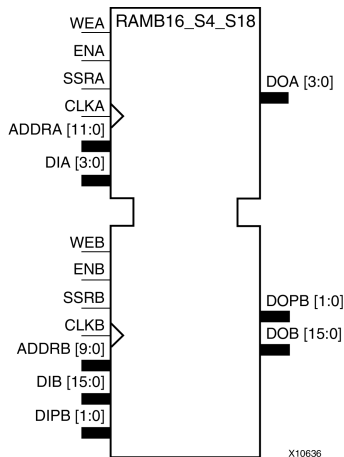
```

詳細情報

- 373

RAMB16_S4_S18

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 18-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号
 INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。
 SRVAL_A = レジスタの値
 addr = RAM アドレス
 RAM(addr) = アドレス ADDR の RAM の内容
 data = RAM の入力データ
 pdata = RAM のパリティ データ
¹WRITE_MODE_A=NO_CHANGE
²WRITE_MODE_A=READ_FIRST
³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S4_S18	4096 x 4	—	(11:0)	(3:0)	—	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDRA) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マッピング

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティ メモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																			
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0			
4	4096	<--	7				6				5				4				3				2				1				0					
8	2048	<--	3								2								1								0									
16	1024	<--	1																0																	
32	512	<--	0																																	

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE"：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S4_S18: 4k/1k x 4/16 + 0/2 Parity bits Dual-Port RAM
--
-- Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S4_S18_inst : RAMB16_S4_S18
generic map (
  INIT_A => X"0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"00000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"0", -- Port A output value upon SSR assertion
  SRVAL_B => X"00000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 1023, Port B Address 0 to 255
  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",
  -- Port A Address 1024 to 2047, Port B Address 256 to 511
  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",

```

```

-- Port A Address 2048 to 3071, Port B Address 512 to 767
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",
-- Port A Address 3072 to 4095, Port B Address 768 to 1023
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port A Address 0 to 1023, Port B Address 0 to 255
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 1024 to 2047, Port B Address 256 to 511
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 2048 to 3071, Port B Address 512 to 767
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 3072 to 4095, Port B Address 768 to 1023
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 4-bit Data Output
  DOB => DOB,      -- Port B 16-bit Data Output
  DOPB => DOPB,    -- Port B 2-bit Parity Output
  ADDRA => ADDRA,  -- Port A 12-bit Address Input
  ADDRb => ADDRb,  -- Port B 10-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- Port A 4-bit Data Input
  DIB => DIB,      -- Port B 16-bit Data Input
  DIPB => DIPB,    -- Port-B 2-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- PortB RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S4_S18_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// RAMB16_S4_S18: 4k/1k x 4/16 + 0/2 Parity bits Dual-Port RAM
```

Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日


```
.INIT_38(256'h00000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 1023, Port B Address 0 to 255
.INITP_00(256'h00000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000),
// Port A Address 1024 to 2047, Port B Address 256 to 511
.INITP_02(256'h00000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000),
// Port A Address 2048 to 3071, Port B Address 512 to 767
.INITP_04(256'h00000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000),
// Port A Address 3072 to 4095, Port B Address 768 to 1023
.INITP_06(256'h00000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000)
) RAMB16_S4_S18_inst (
.DOA(DOA), // Port A 4-bit Data Output
.DOB DOB), // Port B 16-bit Data Output
.DOPB(DOPB), // Port B 2-bit Parity Output
.ADDRA(ADDRA), // Port A 12-bit Address Input
.ADDRB(ADDRB), // Port B 10-bit Address Input
.CLKA(CLKA), // Port A Clock
.CLKB(CLKB), // Port B Clock
.DIA(DIA), // Port A 4-bit Data Input
.DIB(DIB), // Port B 16-bit Data Input
.DIPB(DIPB), // Port-B 2-bit parity Input
.ENA(ENA), // Port A RAM Enable Input
.ENB(ENB), // Port B RAM Enable Input
.SSRA(SSRA), // Port A Synchronous Set/Reset Input
.SSRB(SSRB), // Port B Synchronous Set/Reset Input
.WEA(WEA), // Port A Write Enable Input
.WEB(WEB) // Port B Write Enable Input
);

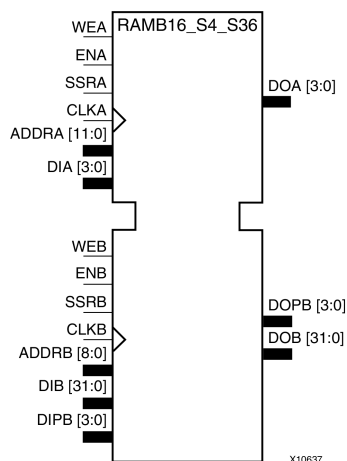
// End of RAMB16_S4_S18_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S4_S36

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 36-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号
 INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。
 SRVAL_A = レジスタの値
 addr = RAM アドレス
 RAM(addr) = アドレス ADDR の RAM の内容
 data = RAM の入力データ
 pdata = RAM のパリティ データ
¹WRITE_MODE_A=NO_CHANGE
²WRITE_MODE_A=READ_FIRST
³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S4_S36	4096 x 4	—	(11:0)	(3:0)	—	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリセルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロックの SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化なし	X	変化なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	変化なし	変化なし	変化なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE"：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S4_S36: 4k/512 x 4/32 + 0/4 Parity bits Dual-Port RAM
--                Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S4_S36_inst : RAMB16_S4_S36
generic map (
  INIT_A => X"0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"000000000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"0", -- Port A output value upon SSR assertion
  SRVAL_B => X"000000000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 1023, Port B Address 0 to 127
  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",
  -- Port A Address 1024 to 2047, Port B Address 128 to 255
  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",

```

```
-- Port A Address 2048 to 3071, Port B Address 256 to 383
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",
-- Port A Address 3072 to 4095, Port B Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port A Address 0 to 1023, Port B Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 1024 to 2047, Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 2048 to 3071, Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 3072 to 4095, Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 4-bit Data Output
  DOB => DOB,      -- Port B 32-bit Data Output
  DOPB => DOPB,    -- Port B 4-bit Parity Output
  ADDRA => ADDRA,  -- Port A 12-bit Address Input
  ADDRb => ADDRb,  -- Port B 9-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- Port A 4-bit Data Input
  DIB => DIB,      -- Port B 32-bit Data Input
  DIPB => DIPB,    -- Port-B 4-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- PortB RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S4_S36_inst instantiation
```

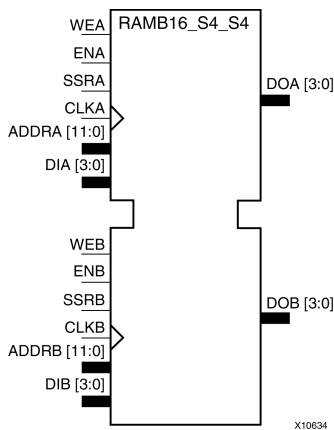
Verilog 記述 (インスタンス化)

```
// RAMB16_S4_S36: 4k/512 x 4/32 + 0/4 Parity bits Dual-Port RAM
```

Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日

RAMB16_S4_S4

プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号
 INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。
 SRVAL_A = レジスタの値
 addr = RAM アドレス
 RAM(addr) = アドレス ADDR の RAM の内容
 data = RAM の入力データ
 pdata = RAM のパリティ データ
¹WRITE_MODE_A=NO_CHANGE
²WRITE_MODE_A=READ_FIRST
³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S4_S4	4096 x 4	—	(11:0)	(3:0)	—	4096 x 4	—	(11:0)	(3:0)	—

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																			
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0			
4	4096	<--	7				6				5				4				3				2				1				0					
8	2048	<--	3								2								1								0									
16	1024	<--	1																0																	
32	512	<--	0																																	

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロックの SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化なし	X	変化なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	変化なし	変化なし	変化なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化なし	X	変化なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE"：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S4_S4: 4k x 4 Dual-Port RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S4_S4_inst : RAMB16_S4_S4
generic map (
  INIT_A => X"0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"0", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"0", -- Port A output value upon SSR assertion
  SRVAL_B => X"0", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 1023
  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",
  -- Address 1024 to 2047
  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",

```

```
-- Address 2048 to 3071
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",
-- Address 3072 to 4095
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")
port map (
  DOA => DOA,      -- Port A 4-bit Data Output
  DOB => DOB,      -- Port B 4-bit Data Output
  ADDRA => ADDRA,   -- Port A 12-bit Address Input
  ADDR_B => ADDR_B, -- Port B 12-bit Address Input
  CLKA => CLKA,     -- Port A Clock
  CLKB => CLKB,     -- Port B Clock
  DIA => DIA,      -- Port A 4-bit Data Input
  DIB => DIB,      -- Port B 4-bit Data Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- Port B RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S4_S4_inst instantiation
```

Verilog 記述 (インスタンスエーション)

```
// RAMB16_S4_S4: 4k x 4 Dual-Port RAM
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

RAMB16_S4_S4 #(
  .INIT_A(4'h0), // Value of output RAM registers on Port A at startup
  .INIT_B(4'h0), // Value of output RAM registers on Port B at startup
  .SRVAL_A(4'h0), // Port A output value upon SSR assertion
  .SRVAL_B(4'h0), // Port B output value upon SSR assertion
  .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

  // The following INIT_xx declarations specify the initial contents of the RAM
  // Address 0 to 1023
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

408
<http://japan.xilinx.com>
Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日


```
.DIA(DIA),      // Port A 4-bit Data Input
.DIB(DIB),      // Port B 4-bit Data Input
.ENA(ENA),      // Port A RAM Enable Input
.ENB(ENB),      // Port B RAM Enable Input
.SSRA(SSRA),    // Port A Synchronous Set/Reset Input
.SSRB(SSRB),    // Port B Synchronous Set/Reset Input
.WEA(WEA),      // Port A Write Enable Input
.WEB(WEB)       // Port B Write Enable Input
);

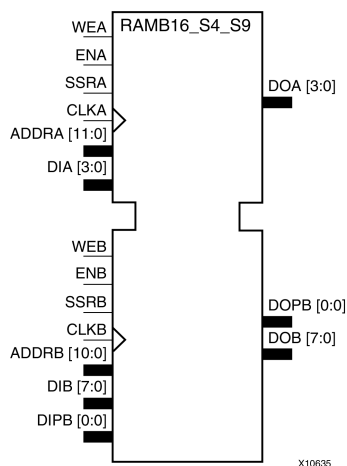
// End of RAMB16_S4_S4_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S4_S9

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 9-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S4_S9	4096 x 4	—	(11:0)	(3:0)	—	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロックの SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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"：警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY"：警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY"：警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE"：警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 ・ "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE"：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST"：値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S4_S9: 4k/2k x 4/8 + 0/1 Parity bit Dual-Port RAM
--              Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S4_S9_inst : RAMB16_S4_S9
generic map (
  INIT_A => X"0", -- Value of output RAM registers on Port A at startup
  INIT_B => X"000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"0", -- Port A output value upon SSR assertion
  SRVAL_B => X"000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 1023, Port B Address 0 to 511
  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",
  -- Port A Address 1024 to 2047, Port B Address 512 to 1023
  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",

```

```
-- Port A Address 2048 to 3071, Port B Address 1024 to 1535
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",
-- Port A Address 3072 to 4095, Port B Address 1536 to 2047
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port B Address 0 to 511
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 512 to 1023
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 1024 to 1535
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port B Address 1536 to 2047
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,    -- Port A 4-bit Data Output
  DOB => DOB,    -- Port B 8-bit Data Output
  DOPB => DOPB,  -- Port B 1-bit Parity Output
  ADDRA => ADDRA, -- Port A 12-bit Address Input
  ADDRb => ADDRb, -- Port B 11-bit Address Input
  CLKA => CLKA,   -- Port A Clock
  CLKB => CLKB,   -- Port B Clock
  DIA => DIA,     -- Port A 4-bit Data Input
  DIB => DIB,     -- Port B 8-bit Data Input
  DIPB => DIPB,   -- Port-B 1-bit parity Input
  ENA => ENA,     -- Port A RAM Enable Input
  ENB => ENB,     -- PortB RAM Enable Input
  SSRA => SSRA,   -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,   -- Port B Synchronous Set/Reset Input
  WEA => WEA,     -- Port A Write Enable Input
  WEB => WEB      -- Port B Write Enable Input
);

-- End of RAMB16_S4_S9_inst instantiation
```

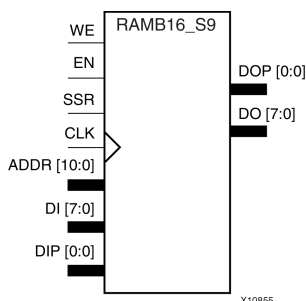
Verilog 記述 (インスタンス化)

```
// RAMB16_S4_S9: 4k/2k x 4/8 + 0/1 Parity bit Dual-Port RAM
```

Spartan-3 ライブラリ ガイド (HDL 用)
UG607 (v14.1) 2012 年 4 月 24 日

RAMB16_S9

プリミティブ : 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 9-bit Port



概要

このデザイン エLEMENTは、同期書き込み機能を備えた専用 RAM ブロックです。ブロック RAM のポートには 16384 ビットのデータ メモリがあります。セルの構成を、次の表に示します。

データ セル		パリティ セル				
ワード数	幅	ワード数	幅	アドレス バス	データ バス	パリティ バス
2048	8	2048	1	(10:0)	(7:0)	(0:0)

イネーブル ピン EN は、読み出し、書き込み、リセットを制御します。EN が Low の場合、データは書き込まれず、出力 (DO および DOP) の値は変化しません。EN とリセット (SSR) が High の場合、クロック (CLK) が Low から High に切り替わるときに、DO および DOP が SRVAL にセットされます。ライト イネーブル (WE) も High の場合は、DI および DIP の値が RAM に書き込まれます。SSR と WE が Low で EN が High の場合、クロックが Low から High に切り替わるときに、RAM アドレス (ADDR) に格納されているデータが DO および DOP に出力されます。デフォルトでは WRITE_MODE=WRITE_FIRST に設定されており、SSR が Low で EN および WE が High の場合、クロックが Low から High に切り替わるときに、データ入力 (DI および DIP) のデータが書き込みアドレス (ADDR) で選択されているワードに書き込まれ、データ出力 (DO および DOP) にその値が出力されます。

前述の説明では、制御ピン (EN、WE、SSR、CLK) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

論理表

入力								出力			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT	INIT	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR=グローバル セットリセット信号

INIT = データ メモリ用に INIT 属性で設定された値。デフォルトはすべて 0 です。

SRVAL = SRVAL 属性で設定された SSR のアサート後の値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹ WRITE_MODE=NO_CHANGE

² WRITE_MODE=READ_FIRST

³ WRITE_MODE=WRITE_FIRST

初期化

メモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16 の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティメモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティメモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。シングルポート RAMB16 の出力レジスタの初期化に使用する属性には、INIT と SRVAL の 2 種類があります。INIT 属性は、電源投入時の出力レジスタの値を指定します。SRVAL 属性は、SSR (セット/リセット) 入力のアサートしたときの値を指定します。

INIT および SRVAL 属性は、初期化値を、出力ポートの各ビットを 1 ビット含む 16 進数文字列として指定します。たとえば、ポート幅が 1 の RAMB16_S1 の場合、出力レジスタは 1 ビットなので、INIT および SRVAL には 1 または 0 を指定します。ポート幅が 4 の RAMB16_S4 の場合は、出力レジスタは 4 ビットなので、0 ~ F の 16 進数値を指定します。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT または SRVAL の値の上位ビットで指定します。

書き込みモードの選択

WRITE_MODE 属性は、RAMB16 メモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE は WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。WRITE_MODE を READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。WRITE_MODE を NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DO 出力ポートの初期値を指定。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定
SRVAL	2 進数/ 16 進数	任意の値	すべてゼロ	SSR ピンをアサートすると、DO 出力ポートがセット (1) またはリセット (0) に設定されるよう指定できます。ビット幅は、RAM の A ポートまたは B ポートの幅によって決まります。
WRITE_MODE	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	書き込みコマンドが DO ポートで実行されるときのパートの動作を指定します。 <ul style="list-style-type: none"> "WRITE_FIRST"：値がポートに書き込まれてから出力されます。 "READ_FIRST"：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。

属性	データ型	値	デフォルト	説明
				・ "NO_CHANGE": 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

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

-- RAMB16_S9: 2k x 8 + 1 Parity bit Single-Port RAM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S9_inst : RAMB16_S9
generic map (
  INIT => X"000", -- Value of output RAM registers at startup
  SRVAL => X"000", -- Output value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 511
  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",
  -- Address 512 to 1023
  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",
  -- Address 1024 to 1535
  INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

Verilog 記述 (インスタンス化)

[illegible]

427

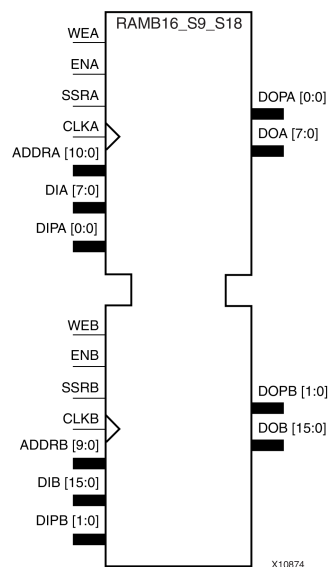
```
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),  
// Address 1536 to 2047  
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),  
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)  
);  
RAMB16_S9_inst (  
    .DO(DO),           // 8-bit Data Output  
    .DOP(DOP),         // 1-bit parity Output  
    .ADDR(ADDR),       // 11-bit Address Input  
    .CLK(CLK),         // Clock  
    .DI(DI),           // 8-bit Data Input  
    .DIP(DIP),         // 1-bit parity Input  
    .EN(EN),           // RAM Enable Input  
    .SSR(SSR),         // Synchronous Set/Reset Input  
    .WE(WE)            // Write Enable Input  
);  
  
// End of RAMB16_S9_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S9_S18

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 18-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S9_S18	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるときに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるときに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるときに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるときに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるときに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるときに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マッピング

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリセルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリセルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリセルにアクセスできます。ただし、一方のポートがあるメモリセルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリセルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S9_S18: 2k/1k x 8/16 + 1/2 Parity bits Parity bits Dual-Port RAM
--                Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S9_S18_inst : RAMB16_S9_S18
generic map (
  INIT_A => X"000", -- Value of output RAM registers on Port A at startup
  INIT_B => X"00000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"000", -- Port A output value upon SSR assertion
  SRVAL_B => X"00000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 511, Port B Address 0 to 255
  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",
  -- Port A Address 512 to 1023, Port B Address 256 to 511
  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",

```

```
-- Port A Address 1024 to 1535, Port B Address 512 to 767
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",
-- Port A Address 1536 to 2047, Port B Address 768 to 1024
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port A Address 0 to 511, Port B Address 0 to 255
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 1023, Port B Address 256 to 511
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 1024 to 1535, Port B Address 512 to 767
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 1536 to 2047, Port B Address 768 to 1024
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 8-bit Data Output
  DOB => DOB,      -- Port B 16-bit Data Output
  DOPA => DOPA,    -- Port A 1-bit Parity Output
  DOPB => DOPB,    -- Port B 2-bit Parity Output
  ADDRA => ADDRA,  -- Port A 11-bit Address Input
  ADDR8 => ADDR8,  -- Port B 10-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- Port A 8-bit Data Input
  DIB => DIB,      -- Port B 16-bit Data Input
  DIPA => DIPA,    -- Port A 1-bit parity Input
  DIPB => DIPB,    -- Port B 2-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- Port B RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S9_S18_inst instantiation
```

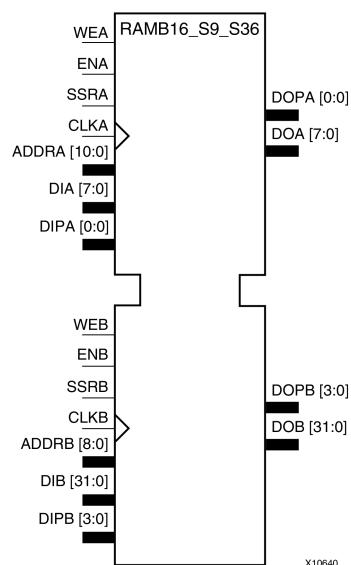
[illegible]

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

RAMB16_S9_S36

プリミティブ : 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 36-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S9_S36	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA、WEA、SSRA、CLKA、ENB、WEB、SSRB、CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マッピング

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティメモリセルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリ セルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリ セルにアクセスできます。ただし、一方のポートがあるメモリ セルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリ セルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S9_S36: 2k/512 x 8/32 + 1/4 Parity bits Parity bits Dual-Port RAM
--                Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S9_S36_inst : RAMB16_S9_S36
generic map (
  INIT_A => X"000", -- Value of output RAM registers on Port A at startup
  INIT_B => X"000000000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"000", -- Port A output value upon SSR assertion
  SRVAL_B => X"000000000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Port A Address 0 to 511, Port B Address 0 to 127
  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",
  -- Port A Address 512 to 1023, Port B Address 128 to 255
  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",

```

```
-- Port A Address 1024 to 1535, Port B Address 255 to 383
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",
-- Port A Address 1536 to 2047, Port B Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port A Address 0 to 511, Port B Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 1023, Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 1024 to 1535, Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 1536 to 2047, Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 8-bit Data Output
  DOB => DOB,      -- Port B 32-bit Data Output
  DOPA => DOPA,     -- Port A 1-bit Parity Output
  DOPB => DOPB,     -- Port B 4-bit Parity Output
  ADDRA => ADDRA,   -- Port A 11-bit Address Input
  ADDR8 => ADDR8,   -- Port B 9-bit Address Input
  CLKA => CLKA,     -- Port A Clock
  CLKB => CLKB,     -- Port B Clock
  DIA => DIA,      -- Port A 8-bit Data Input
  DIB => DIB,      -- Port B 32-bit Data Input
  DIPA => DIPA,     -- Port A 1-bit parity Input
  DIPB => DIPB,     -- Port-B 4-bit parity Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- PortB RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB       -- Port B Write Enable Input
);

-- End of RAMB16_S9_S36_inst instantiation
```

[illegible]

```
.INIT_34(256'h00000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 511, Port B Address 0 to 127
.INITP_00(256'h00000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 1023, Port B Address 128 to 255
.INITP_02(256'h00000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000),
// Port A Address 1024 to 1535, Port B Address 256 to 383
.INITP_04(256'h00000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000),
// Port A Address 1536 to 2047, Port B Address 384 to 511
.INITP_06(256'h00000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000)
) RAMB16_S9_S36_inst (
.DOA(DOA), // Port A 8-bit Data Output
.DOB DOB), // Port B 32-bit Data Output
.DOPA(DOPA), // Port A 1-bit Parity Output
.DOPB(DOPB), // Port B 4-bit Parity Output
.ADDRA(ADDRA), // Port A 11-bit Address Input
.ADDRB(ADDRB), // Port B 9-bit Address Input
.CLKA(CLKA), // Port A Clock
.CLKB(CLKB), // Port B Clock
.DIA(DIA), // Port A 8-bit Data Input
.DIB(DIB), // Port B 32-bit Data Input
.DIPA(DIPA), // Port A 1-bit parity Input
.DIPB(DIPB), // Port-B 4-bit parity Input
.ENA(ENA), // Port A RAM Enable Input
.ENB(ENB), // Port B RAM Enable Input
.SSRA(SSRA), // Port A Synchronous Set/Reset Input
.SSRB(SSRB), // Port B Synchronous Set/Reset Input
.WEA(WEA), // Port A Write Enable Input
.WEB(WEB) // Port B Write Enable Input
);

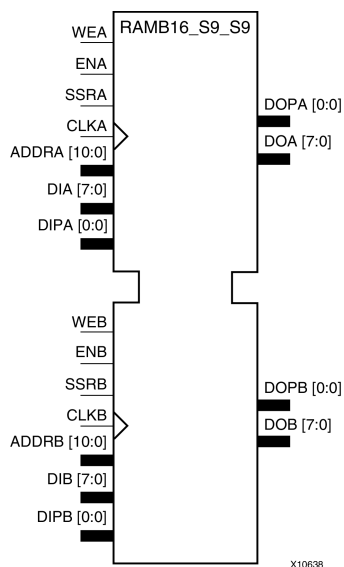
// End of RAMB16_S9_S36_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

RAMB16_S9_S9

プリミティブ：16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit Ports



概要

このデザイン エLEMENTは、同期書き込み機能を備えた、デュアル ポートの専用 RAM ブロックです。各ブロック RAM のポートには 16384 ビットのデータ メモリがあります。ポート幅が 9、18、36 ビットにコンフィギュレーションされたポートには、さらに 2048 ビットのパリティ メモリがあります。各ポートは、同じ 16384 個のデータ メモリ セルに独立してアクセスします。各ポートのデータ幅は、それぞれ個別に設定できます。このELEMENTのポートおよびセルの構成は「ポートの説明」の表に示されています。

論理表

論理表 A

入力								出力			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セットリセット信号

INIT_A = 出力レジスタ用に INIT_A 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_A = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_A=NO_CHANGE

²WRITE_MODE_A=READ_FIRST

³WRITE_MODE_A=WRITE_FIRST

論理表 B

入力								出力			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM の内容	
										データ RAM	パリティ RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	変化なし	変化なし
0	0	X	X	X	X	X	X	変化なし	変化なし	変化なし	変化なし
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	変化なし	変化なし
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	変化なし	変化なし
0	1	0	1	↑	addr	data	pdata	変化なし ¹ 、 RAM(addr) ² 、 data ³	変化なし ¹ 、 RAM(addr) ² 、 pdata ³	RAM(addr) =>data	RAM(addr) =>pdata

GSR = グローバル セット リセット信号

INIT_B = 出力レジスタ用に INIT_B 属性で設定された値。デフォルトはすべて 0 です。

SRVAL_B = レジスタの値

addr = RAM アドレス

RAM(addr) = アドレス ADDR の RAM の内容

data = RAM の入力データ

pdata = RAM のパリティ データ

¹WRITE_MODE_B=NO_CHANGE

²WRITE_MODE_B=READ_FIRST

³WRITE_MODE_B=WRITE_FIRST

ポートの説明

ポート A						ポート B				
デザイン エレメント	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス	データ セル ¹	パリティ セル ¹	アドレス バス	データ バス	パリティ バス
RAMB16_S9_S9	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)

¹ ワード数 X 幅

各ポートは、それぞれのクロックに完全に同期します。ポート A の各データ入力ピン (DIA) のセットアップ タイム、およびデータ出力バス (DOA) の clock-to-out タイムは、CLKA を基準とします。ポート B の各データ入力ピン (DIB) のセットアップ タイム、およびデータ出力バス (DOB) の clock-to-out タイムは、CLKB を基準とします。イネーブル ピン ENA は、ポート A の読み出し、書き込み、リセットを制御します。ENA が Low の場合、データは書き込まれず、出力 (DOA および DOPA) は変化しません。ENA とリセット (SSRA) が High の場合、クロック (CLKA) が Low から High に切り替わるたびに DOA および DOPA が SRVAL_A にセットされます。ライト イネーブル (WEA) も High の場合は、DIA および DIPA の値が RAM に書き込まれます。ENA が High で SSRA と WEA が Low の場合、クロックが Low から High に切り替わるたびに、RAM アドレス (ADDR_A) に格納されているデータが読み出されます。SSRA が Low で ENA と WEA が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDR_A) で選択されているワードにデータ入力 (DIA および DIPA) の値が読み込まれます。データ出力 (DOA および DOPA) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。

イネーブル ピン ENB は、ポート B の読み出し、書き込み、リセットを制御します。ENB が Low の場合、データは書き込まれず、出力 (DOB および DOPB) は変化しません。ENB とリセット (SSRB) が High の場合、クロック (CLKB) が Low から High に切り替わるたびに、DOB および DOPB が SRVAL_B にセットされます。ライト イネーブル (WEB) も High の場合は、DIB および DIPB の値が RAM に書き込まれます。ENB が High で SSRB と WEB が Low の場合、クロックが Low から High に切り替わるたびに RAM アドレス (ADDRB) に格納されているデータが出力されます。SSRB が Low で ENB と WEB が High の場合、クロックが Low から High に切り替わるたびに、書き込みアドレス (ADDRB) で選択されているワードにデータ入力 (DIB および DIPB) の値が読み込まれます。データ出力 (DOB および DOPB) に出力される値は、書き込みモードによって異なります。書き込みモードは、デフォルトでは WRITE_MODE=WRITE_FIRST に設定されています。前述の説明では、制御ピン (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, CLKB) がアクティブ High であると想定していますが、ポートにインバーターを配置してアクティブ Low にすることもできます。RAMB16 のポートに配置したインバーターはブロック内に組み込まれるので、CLB リソースは使用されません。

アドレス マップ

各ポートは、ポートの幅によって異なるアドレス指定方法を使用して、同じ 18432 個のメモリ セルにアクセスします。「データ用のポート アドレス マップ」に示すように、すべてのポート幅で 16384 個のメモリ セルをデータの格納に使用できます。9、18、および 36 ビット幅のポートには、「パリティ用のポート アドレス マップ」に示すように、2408 個のパリティ メモリ セルもあります。特定のポート幅での物理的な RAM の位置は、次の式によって決定されます。

$$\text{Start} = ((\text{ADDR_port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

次の表に、デュアル ポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

次の表に、各ポート幅のアドレス マップを示します。

データ用のポート アドレス マップ

データ 幅	ポートのデータ アドレス																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6				5				4				3				2				1				0				
8	2048	<--	3								2								1								0								
16	1024	<--	1																0																
32	512	<--	0																																

パリティ用のポート アドレス マップ

パリティ幅	ポートのパリティ アドレス														
1	2048	<-----	3				2				1				0
2	1024	<-----	1								0				
4	512	<-----	0												

デュアル ポート RAMB16 のメモリ内容の初期化

INIT_xx 属性を使用すると、デバイスのコンフィギュレーション中に RAMB16 のメモリの内容を初期化できます。各 RAMB16_Sm_Sn の初期値は、64 種類の初期化属性 (INIT_00 ~ INIT_3F) で指定します。各属性は、64 の 16 進数値から成り、合計 16384 ビットを指定できます。

INITP_xx 属性を使用すると、デバイスのコンフィギュレーションまたはアサート中にパリティ メモリを初期化できます。9、18、36 ビット幅にコンフィギュレーションされたポートのパリティ メモリの初期値は、8 個の初期化属性 (INITP_00 ~ INITP_07) で指定します。各属性は 64 個の 16 進数値から成り、合計 2048 ビットを指定できます。

INIT_xx または INITP_xx 属性を指定しない場合は、そのアドレスの値は 0 に設定されます。属性を一部だけ指定すると、上位ビットが 0 になります。

デュアル ポート RAMB16 の出力レジスタの初期化

Spartan®-3A 以降のデバイスでは、電源投入時 (GSR が High のとき) に出力レジスタの各ビットを 0 または 1 に初期化できます。また、セット/リセットをアサートした後の値を、電源投入時の初期値とは異なる値に設定できます。デュアル ポート RAMB16 の出力レジスタを初期化する属性には、INIT_A、INIT_B、SRVAL_A、SRVAL_B の 4 種類があります。INIT_A 属性は電源投入時のポート A の出力レジスタの初期値を指定し、INIT_B 属性は電源投入時のポート B の出力レジスタの初期値を指定します。SRVAL_A 属性はポート A で SSRA (セット/リセット) 入力をアサートしたときの初期値を指定し、SRVAL_B 属性はポート B で SSRB (セット/リセット) 入力をアサートしたときの初期値を指定します。

INIT_A、INIT_B、SRVAL_A、SRVAL_B 属性は、16 進数で指定します。この初期値はポート幅によって異なります。たとえば、ポート A の幅が 1、ポート B の幅が 4 の RAMB16_S1_S4 の場合、ポート A の出力レジスタは 1 ビットなので、INIT_A または SRVAL_A には 1 か 0 しか指定できません。ポート B の出力レジスタは 4 ビットなので、INIT_B または SRVAL_B に 0 ~ F の 16 進数値を指定できます。

パリティビットを含むポートでは、出力レジスタのパリティ部分は、INIT_A、INIT_B、SRVAL_A、SRVAL_B の値の上位ビットで指定します。

INIT および SRVAL 属性を指定しない場合は、デフォルトで 0 に初期化されます。

書き込みモードの選択

WRITE_MODE_A 属性は、デュアル ポート RAMB16 のポート A のメモリおよび出力の内容を制御し、WRITE_MODE_B 属性は、ポート B のメモリおよび出力の内容を制御します。デフォルトでは、WRITE_MODE_A と WRITE_MODE_B は両方とも WRITE_FIRST に設定されています。この場合、入力値がメモリに書き込まれた後にその値が出力されます。READ_FIRST に設定すると、メモリの内容が出力された後、入力値がメモリに書き込まれます。NO_CHANGE に設定すると、入力値はメモリに書き込まれますが、出力の値は変化しません。ポート A とポート B で同じメモリセルに読み出し/書き込みを行おうとした場合の競合の解決方法については、「ポート A とポート B が競合する場合の対処方法」を参照してください。

ポート A とポート B が競合する場合の対処方法

Spartan-3A ブロック SelectRAM™ は、完全なデュアル ポート RAM で、2 つのポートが同時に同じメモリセルにアクセスできます。ただし、一方のポートがあるメモリセルに書き込みを行っている場合は、もう一方のポートで clock-to-clock セットアップ タイム内に、そのメモリセルに対して書き込みまたは読み出しを実行しないようにする必要があります。

次の表に、デュアルポート RAMB16 で競合が発生した場合の動作を、WRITE_MODE_A と WRITE_MODE_B の設定別に示します。

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=NO_CHANGE の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	変化 なし	X	変化 なし	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	変化 なし	変化 なし	変化 なし	X	X

WRITE_MODE_A=READ_FIRST、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE_MODE_A=WRITE_FIRST、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=READ_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIB	DIPB

WRITE_MODE_A=NO_CHANGE、WRITE_MODE_B=WRITE_FIRST の場合

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	変化 なし	X	変化 なし	X	X	X

WRITE_MODE_A=READ_FIRST および WRITE_MODE_B=WRITE_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	データ RAM	パリティ RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	変化なし	変化なし
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

デザインの入力方法

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	不可

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT_00 ~ INIT_3F	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のデータ部分の初期値を指定します。
INIT_A	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOA 出力ポートの初期値を指定します。ビット幅は、RAM の A ポートの幅によって決まります。
INIT_B	2 進数/ 16 進数	任意の値	すべてゼロ	コンフィギュレーション後の DOB 出力ポートの初期値を指定します。ビット幅は、RAM の B ポートの幅によって決まります。
INITP_00 ~ INITP_07	2 進数/ 16 進数	任意の値	すべてゼロ	RAM 配列のパリティ部分の初期値を指定します。

属性	データ型	値	デフォルト	説明
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" : 警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ "WARNING_ONLY" : 警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ "GENERATE_X_ONLY" : 警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ "NONE" : 警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <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>
SRVAL_A	2 進数/ 16 進数	任意の値	すべてゼロ	RSTA ピンをアサートしたときに、DOA 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の A ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
SRVAL_B	2 進数/ 16 進数	任意の値	すべてゼロ	RSTB ピンをアサートしたときに、DOB 出力ポートをセット (1) するかリセット (0) するかをビットごとに設定します。ビット幅は、RAM の B ポートの幅によって決まります。デフォルトではすべてのビットがリセットされます。
WRITE_MODE_A	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOA ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。 ・ "READ_FIRST" : 新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 ・ "NO_CHANGE" : 出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。
WRITE_MODE_B	文字列	"WRITE_FIRST"、 "READ_FIRST"、 "NO_CHANGE"	"WRITE_FIRST"	<p>書き込みコマンドが DOB ポートで実行されるときのポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ "WRITE_FIRST" : 値がポートに書き込まれてから出力されます。

属性	データ型	値	デフォルト	説明
				<ul style="list-style-type: none"> “READ_FIRST”：新たに値が書き込まれる前に、RAM の以前の値が出力ポートに出力されます。 “NO_CHANGE”：出力ポートの以前の値が保持され、出力ポートの値は変化しません。RAM のポートから値を読み出さない場合は、このモードに設定することをお勧めします。

VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S9_S9: 2k x 8 + 1 Parity bit Dual-Port RAM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16_S9_S9_inst : RAMB16_S9_S9
generic map (
  INIT_A => X"000", -- Value of output RAM registers on Port A at startup
  INIT_B => X"000", -- Value of output RAM registers on Port B at startup
  SRVAL_A => X"000", -- Port A output value upon SSR assertion
  SRVAL_B => X"000", -- Port B output value upon SSR assertion
  WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 511
  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",
  -- Address 512 to 1023
  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",

```

462 <http://japan.xilinx.com> Spartan-3 ライブラリ ガイド (HDL 用) UG607 (v14.1) 2012 年 4 月 24 日

[illegible]

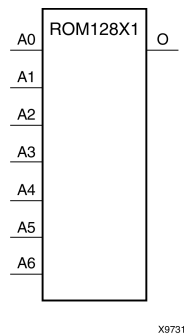
[illegible]

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

ROM128X1

プリミティブ：128-Deep by 1-Wide ROM



X9731

概要

このデザイン エLEMENTは 128 ワード X 1 ビットの ROM です。データ出力 (O) には、7 ビットのアドレス (A6 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 32 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。INIT を指定しないと、エラーが発生します。

論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	128 ビット値	すべてゼロ	ROM の値を指定

VHDL 記述 (インスタンス化)

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

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

-- ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

ROM128X1_inst : ROM128X1
generic map (
  INIT => X"00000000000000000000000000000000")
port map (
  O => O,    -- ROM output
  A0 => A0,  -- ROM address[0]
  A1 => A1,  -- ROM address[1]
  A2 => A2,  -- ROM address[2]
  A3 => A3,  -- ROM address[3]
  A4 => A4,  -- ROM address[4]
  A5 => A5,  -- ROM address[5]
  A6 => A6,  -- ROM address[6]
);

-- End of ROM128X1_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

ROM128X1 #(
  .INIT(128'h00000000000000000000000000000000) // Contents of ROM
) ROM128X1_inst (
  .O(O),    // ROM output
  .A0(A0), // ROM address[0]
  .A1(A1), // ROM address[1]
  .A2(A2), // ROM address[2]
  .A3(A3), // ROM address[3]
  .A4(A4), // ROM address[4]
  .A5(A5), // ROM address[5]
  .A6(A6)  // ROM address[6]
);

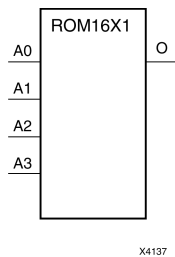
// End of ROM128X1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

ROM16X1

プリミティブ：16-Deep by 1-Wide ROM



概要

このデザイン エLEMENTは 16 ワード X 1 ビットの ROM です。データ出力 (O) には、4 ビットのアドレス (A3 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 4 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。たとえば、INIT=10A7 と指定すると、「0001 0000 1010 0111」というデータストリームが生成されます。INIT を指定しないと、エラーが発生します。

論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	ROM の値を指定

VHDL 記述 (インスタンス化)

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

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

-- ROM16X1: 16 x 1 Asynchronous Distributed (LUT) ROM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

ROM16X1_inst : ROM16X1
generic map (
  INIT => X"0000")
port map (
  O => O,    -- ROM output
  A0 => A0,  -- ROM address[0]
  A1 => A1,  -- ROM address[1]
  A2 => A2,  -- ROM address[2]
  A3 => A3   -- ROM address[3]
);

-- End of ROM16X1_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// ROM16X1: 16 x 1 Asynchronous Distributed (LUT) ROM
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

ROM16X1 #(
  .INIT(16'h0000) // Contents of ROM
) ROM16X1_inst (
  .O(O),          // ROM output
  .A0(A0),        // ROM address[0]
  .A1(A1),        // ROM address[1]
  .A2(A2),        // ROM address[2]
  .A3(A3)         // ROM address[3]
);

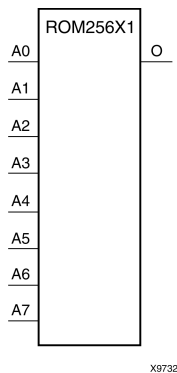
// End of ROM16X1_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3 FPGA ファミリー データシート](#)

ROM256X1

プリミティブ：256-Deep by 1-Wide ROM



概要

このデザイン エLEMENTは 256 ワード X 1 ビットの ROM です。データ出力 (O) には、8 ビットのアドレス (A7 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 64 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。

INIT を指定しないと、エラーが発生します。

論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	256 ビット値	すべてゼロ	ROM の値を指定

VHDL 記述 (インスタンス化)

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

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

-- ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

ROM256X1_inst : ROM256X1
generic map (
  INIT => X"0000000000000000000000000000000000000000000000000000000000000000"
)
port map (
  O => O,    -- ROM output
  A0 => A0,   -- ROM address[0]
  A1 => A1,   -- ROM address[1]
  A2 => A2,   -- ROM address[2]
  A3 => A3,   -- ROM address[3]
  A4 => A4,   -- ROM address[4]
  A5 => A5,   -- ROM address[5]
  A6 => A6,   -- ROM address[6]
  A7 => A7    -- ROM address[7]
);

-- End of ROM256X1_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

ROM256X1 #(
  .INIT(256'h0000000000000000000000000000000000000000000000000000000000000000) // Contents of ROM
) ROM256X1_inst (
  .O(O),    // ROM output
  .A0(A0),  // ROM address[0]
  .A1(A1),  // ROM address[1]
  .A2(A2),  // ROM address[2]
  .A3(A3),  // ROM address[3]
  .A4(A4),  // ROM address[4]
  .A5(A5),  // ROM address[5]
  .A6(A6),  // ROM address[6]
  .A7(A7)   // ROM address[7]
);

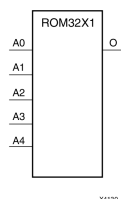
// End of ROM256X1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

ROM32X1

プリミティブ：32-Deep by 1-Wide ROM



概要

このデザイン エLEMENTは 32 ワード X 1 ビットの ROM です。データ出力 (O) には、5 ビットのアドレス (A4 ～ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 8 桁の 16 進数で、ROM には最上位ビット A=1FH から最下位ビット A=00H の順に書き込まれます。

たとえば INIT=10A78F39 と指定すると、「0001 0000 1010 0111 1000 1111 0011」というデータストリームが生成されます。INIT を指定しないと、エラーが発生します。

論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべてゼロ	ROM の値を指定

VHDL 記述 (インスタンス化)

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

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

-- ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

ROM32X1_inst : ROM32X1
generic map (
  INIT => X"00000000")
port map (
  O => O,    -- ROM output
  A0 => A0,  -- ROM address[0]
  A1 => A1,  -- ROM address[1]
  A2 => A2,  -- ROM address[2]
  A3 => A3,  -- ROM address[3]
  A4 => A4   -- ROM address[4]
);
-- End of ROM32X1_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

ROM32X1 #(
  .INIT(32'h00000000) // Contents of ROM
) ROM32X1_inst (
  .O(O),    // ROM output
  .A0(A0),  // ROM address[0]
  .A1(A1),  // ROM address[1]
  .A2(A2),  // ROM address[2]
  .A3(A3),  // ROM address[3]
  .A4(A4)   // ROM address[4]
);

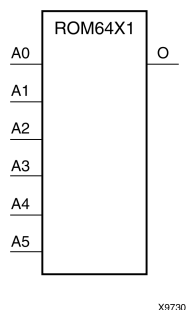
// End of ROM32X1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

ROM64X1

プリミティブ : 64-Deep by 1-Wide ROM



概要

このデザイン エLEMENTは 64 ワード X 1 ビットの ROM です。データ出力 (O) には、6 ビットのアドレス (A5 ~ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT で指定された値に初期化されます。初期値は 16 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。INIT を指定しないと、エラーが発生します。

論理表

入力				出力
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	ROM の値を指定

VHDL 記述 (インスタンス化)

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

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

-- ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

ROM64X1_inst : ROM64X1
generic map (
  INIT => X"0000000000000000")
port map (
  O => O,    -- ROM output
  A0 => A0,  -- ROM address[0]
  A1 => A1,  -- ROM address[1]
  A2 => A2,  -- ROM address[2]
  A3 => A3,  -- ROM address[3]
  A4 => A4,  -- ROM address[4]
  A5 => A5   -- ROM address[5]
);

-- End of ROM64X1_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

ROM64X1 #(
  .INIT(64'h0000000000000000) // Contents of ROM
) ROM64X1_inst (
  .O(O),    // ROM output
  .A0(A0), // ROM address[0]
  .A1(A1), // ROM address[1]
  .A2(A2), // ROM address[2]
  .A3(A3), // ROM address[3]
  .A4(A4), // ROM address[4]
  .A5(A5)  // ROM address[5]
);

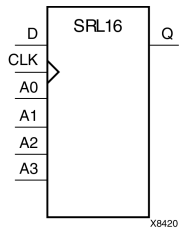
// End of ROM64X1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

SRL16

プリミティブ：16-Bit Shift Register Look-Up Table (LUT)



概要

このデザイン エLEMENTは、シフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには：** 入力 A3 ～ A0 の値を一定にします。シフトレジスタは 1 ～ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには：** 入力 A3 ～ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ～ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

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、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。

論理表

入力			出力
Am	CLK	D	Q
Am	X	X	Q(Am)
Am	↑	D	Q(Am - 1)
m = 0、1、2、3			

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後の Q 出力の初期値を指定

VHDL 記述 (インスタンス化)

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

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

-- SRL16: 16-bit shift register LUT operating on posedge of clock
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

SRL16_inst : SRL16
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
  CLK => CLK,       -- Clock input
  D => D            -- SRL data input
);

-- End of SRL16_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// SRL16: 16-bit shift register LUT operating on posedge of clock
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

SRL16 #(
  .INIT(16'h0000) // Initial Value of Shift Register
) SRL16_inst (
  .Q(Q),          // SRL data output
  .A0(A0),        // Select[0] input
  .A1(A1),        // Select[1] input
  .A2(A2),        // Select[2] input
  .A3(A3),        // Select[3] input
  .CLK(CLK),      // Clock input
  .D(D)           // SRL data input
);

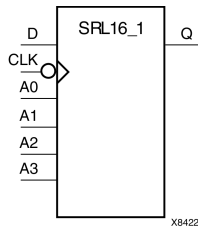
// End of SRL16_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

SRL16_1

プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock



概要

このデザイン エLEMENTは、シフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには：** 入力 A3 ～ A0 の値を一定にします。シフトレジスタは 1 ～ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには：** 入力 A3 ～ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ～ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

クロック (CLK) が High から Low 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 High から Low に切り替わると、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。

論理表

入力			出力
A _m	CLK	D	Q
A _m	X	X	Q(A _m)
A _m	↓	D	Q(A _m - 1)
m = 0、1、2、3			

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後の Q 出力の初期値を指定

VHDL 記述 (インスタンスレーション)

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

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

-- SRL16_1: 16-bit shift register LUT operating on negedge of clock
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

SRL16_1_inst : SRL16_1
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
  CLK => CLK,       -- Clock input
  D => D            -- SRL data input
);

-- End of SRL16_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// SRL16_1: 16-bit shift register LUT operating on negedge of clock
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

SRL16_1 #(
  .INIT(16'h0000) // Initial Value of Shift Register
) SRL16_1_inst (
  .Q(Q),          // SRL data output
  .A0(A0),        // Select[0] input
  .A1(A1),        // Select[1] input
  .A2(A2),        // Select[2] input
  .A3(A3),        // Select[3] input
  .CLK(CLK),      // Clock input
  .D(D)           // SRL data input
);

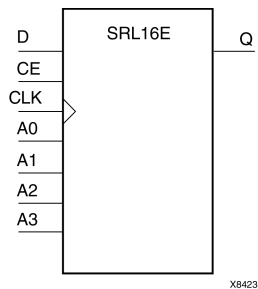
// End of SRL16_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

SRL16E

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Clock Enable



概要

このデザイン エLEMENTは、シフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには：** 入力 A3 ～ A0 の値を一定にします。シフトレジスタは 1 ～ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには：** 入力 A3 ～ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ～ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

CE が High の場合、クロック (CLK) が Low から High に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが Low から High に切り替わるときに CE が High の場合、シフトレジスタの値は次の高位ビットにシフトされ、新しい値がロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。CE が Low の場合、クロック遷移は無視されます。

論理表

入力				出力
A _m	CE	CLK	D	Q
A _m	0	X	X	Q(A _m)
A _m	1	↑	D	Q(A _m - 1)
m = 0、1、2、3				

ポートの説明

ポート名	方向	幅	機能
Q	出力	1	シフトレジスタ データ出力
D	入力	1	シフトレジスタ データ入力
CLK	入力	1	クロック
CE	入力	1	アクティブ High のクロック イネーブル
A	入力	4	SRL のワード数のダイナミック選択 ・ A=0000 ==> 1 ビットシフト長 ・ A=1111 ==> 16 ビットシフト長

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	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 (インスタンシエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

SRL16E_inst : SRL16E
generic map (
  INIT => X"0000")
port map (
  Q => Q,          -- SRL data output
  A0 => A0,         -- Select[0] input
  A1 => A1,         -- Select[1] input
  A2 => A2,         -- Select[2] input
  A3 => A3,         -- Select[3] input
  CE => CE,         -- Clock enable input
  CLK => CLK,       -- Clock input
  D => D           -- SRL data input
);

-- End of SRL16E_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

SRL16E #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRL16E_inst (
    .Q(Q),           // SRL data output
    .A0(A0),         // Select[0] input
    .A1(A1),         // Select[1] input
    .A2(A2),         // Select[2] input
    .A3(A3),         // Select[3] input
    .CE(CE),         // Clock enable input
    .CLK(CLK),       // Clock input
    .D(D)            // SRL data input
);

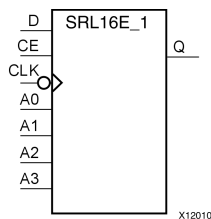
// End of SRL16E_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

SRL16E_1

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock and 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 (CE) があるシフトレジスタ ルックアップ テーブル (LUT) で、クロックの立ち下がりエッジで動作します。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- 固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

CE が High の場合、クロック (CLK) が High から Low 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 High から Low 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
A _m	CE	CLK	D	Q
A _m	0	X	X	Q(A _m)
A _m	1	↓	D	Q(A _m - 1)
m = 0、1、2、3				

デザインの入力方法

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

使用可能な属性

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

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

-- SRL16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

SRL16E_1_inst : SRL16E_1
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_1_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// SRL16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

SRL16E_1 #(
  .INIT(16'h0000) // Initial Value of Shift Register
) SRL16E_1_inst (
  .Q(Q),          // SRL data output
  .A0(A0),        // Select[0] input
  .A1(A1),        // Select[1] input
  .A2(A2),        // Select[2] input
  .A3(A3),        // Select[3] input
  .CE(CE),        // Clock enable input
  .CLK(CLK),      // Clock input
  .D(D)           // SRL data input
);

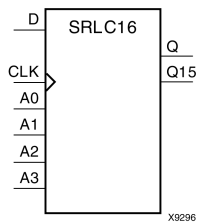
// End of SRL16E_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

SRLC16

プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Carry



概要

このデザイン エLEMENTは、キャリーがあるシフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには：** 入力 A3 ～ A0 の値を一定にします。シフトレジスタは 1 ～ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには：** 入力 A3 ～ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ～ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

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、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。

メモ： Q15 の出力を使用すると、複数のシフトレジスタ LUT 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。

論理表

入力			出力
A _m	CLK	D	Q
A _m	X	X	Q(A _m)
A _m	↑	D	Q(A _m - 1)
m = 0、1、2、3			

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンス化)

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

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

-- SRLC16: 16-bit cascadable shift register LUT operating on posedge of clock
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

SRLC16_inst : SRLC16
generic map (
  INIT => X"0000")
port map (
  Q => Q,          -- SRL data output
  Q15 => Q15,      -- Carry output (connect to next SRL)
  A0 => A0,        -- Select[0] input
  A1 => A1,        -- Select[1] input
  A2 => A2,        -- Select[2] input
  A3 => A3,        -- Select[3] input
  CLK => CLK,      -- Clock input
  D => D           -- SRL data input
);

-- End of SRLC16_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// SRLC16: 16-bit cascadable shift register LUT operating on posedge of clock
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

SRLC16 #(
  .INIT(16'h0000) // Initial Value of Shift Register
) SRLC16_inst (
  .Q(Q),          // SRL data output
  .Q15(Q15),      // Carry output (connect to next SRL)
  .A0(A0),        // Select[0] input
  .A1(A1),        // Select[1] input
  .A2(A2),        // Select[2] input
  .A3(A3),        // Select[3] input
  .CLK(CLK),      // Clock input
  .D(D)           // SRL data input
);

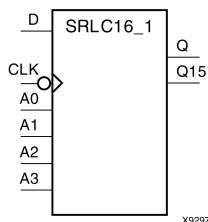
// End of SRLC16_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

SRLC16_1

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Carry and Negative-Edge Clock



概要

このデザイン エレメントは、キャリーがあるシフトレジスタ ルックアップ テーブル (LUT) で、クロックの立ち下がりエッジで動作します。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

メモ : Q15 の出力を使用すると、複数のシフトレジスタ LUT 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。

論理表

入力			出力	
A _m	CLK	D	Q	Q15
A _m	X	X	Q(A _m)	変化なし
A _m	↓	D	Q(A _m - 1)	Q14
m = 0、1、2、3				

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンスレーション)

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

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

-- SRLC16_1: 16-bit cascadable shift register LUT operating on negedge of clock
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

SRLC16_1_inst : SRLC16_1
generic map (
    INIT => X"0000")
port map (
    Q => Q,      -- SRL data output
    Q15 => Q15,   -- Carry output (connect to next SRL)
    A0 => A0,     -- Select[0] input
    A1 => A1,     -- Select[1] input
    A2 => A2,     -- Select[2] input
    A3 => A3,     -- Select[3] input
    CLK => CLK,   -- Clock input
    D => D        -- SRL data input
);

-- End of SRLC16_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// SRLC16_1: 16-bit cascadable shift register LUT operating on negedge of clock
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

SRLC16_1 #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRLC16_1_inst (
    .Q(Q),          // SRL data output
    .Q15(Q15),     // Carry output (connect to next SRL)
    .A0(A0),       // Select[0] input
    .A1(A1),       // Select[1] input
    .A2(A2),       // Select[2] input
    .A3(A3),       // Select[3] input
    .CLK(CLK),     // Clock input
    .D(D)          // SRL data input
);

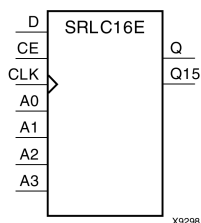
// End of SRLC16_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

SRLC16E

プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Carry and Clock Enable



概要

このデザイン エレメントは、キャリーとクロック イネーブルがあるシフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには：** 入力 A3 ～ A0 の値を一定にします。シフトレジスタは 1 ～ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには：** 入力 A3 ～ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ～ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

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、新しいデータがロードされます。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。

メモ： Q15 の出力を使用すると、複数のシフトレジスタ LUT 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。

論理表

入力				出力	
Am	CLK	CE	D	Q	Q15
Am	X	0	X	Q(Am)	Q(15)
Am	X	1	X	Q(Am)	Q(15)
Am	↑	1	D	Q(Am - 1)	Q15
m = 0、1、2、3					

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンス化)

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

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

-- SRLC16E: 16-bit cascable shift register LUT with clock enable operating on posedge of clock
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

SRLC16E_inst : SRLC16E
generic map (
  INIT => X"0000")
port map (
  Q => Q,          -- SRL data output
  Q15 => Q15,      -- Carry output (connect to next SRL)
  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 SRLC16E_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// SRLC16E: 16-bit cascable shift register LUT with clock enable operating on posedge of clock
//          Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

SRLC16E #(
  .INIT(16'h0000) // Initial Value of Shift Register
) SRLC16E_inst (
  .Q(Q),          // SRL data output
  .Q15(Q15),      // Carry output (connect to next SRL)
  .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
);
```

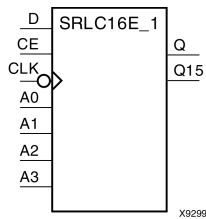
```
// End of SRLC16E_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリ データシート](#)

SRLC16E_1

プリミティブ：16-Bit Shift Register Look-Up Table (LUT) with Carry, Negative-Edge Clock, and Clock Enable



概要

このデザイン エLEMENTは、キャリーおよびクロック イネーブルがあるシフトレジスタ ルックアップ テーブル (LUT) で、クロックの立ち下がりエッジで動作します。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには：** 入力 A3 ～ A0 の値を一定にします。シフトレジスタは 1 ～ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには：** 入力 A3 ～ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ～ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

CE が High の場合、クロック (CLK) が High から Low 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 High から Low 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。

メモ： Q15 の出力を使用すると、複数のシフトレジスタ LUT 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。

論理表

入力				出力	
Am	CE	CLK	D	Q	Q15
Am	0	X	X	Q(Am)	変化なし
Am	1	X	X	Q(Am)	変化なし
Am	1	↓	D	Q(Am-1)	Q14
m = 0、1、2、3					

デザインの入力方法

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

使用可能な属性

属性	データ型	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンス化)

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

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

-- SRLC16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
--           Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

SRLC16E_1_inst : SRLC16E_1
generic map (
  INIT => X"0000")
port map (
  Q => Q,          -- SRL data output
  Q15 => Q15,       -- Carry output (connect to next SRL)
  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 SRLC16E_1_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// SRLC16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

SRLC16E_1 #(
  .INIT(16'h0000) // Initial Value of Shift Register
) SRLC16E_1_inst (
  .Q(Q),          // SRL data output
  .Q15(Q15),      // Carry output (connect to next SRL)
  .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
);
```

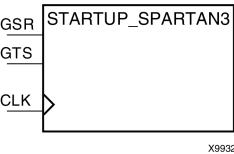
```
// End of SRLC16E_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

STARTUP_SPARTAN3

プリミティブ : Spartan®-3 User Interface to Global Clock, Reset, and 3-State Controls



概要

このデザイン エLEMENTは、グローバル セット/リセット、グローバル トライステート信号、ユーザー コンフィギュレーション クロックの制御に使用します。グローバル セット/リセット入力 (GSR) が High になると、デバイス内にあるすべてのフリップフロップ、ラッチ、ブロック RAM (RAMB16) 出力レジスタが、コンポーネントの初期値 (INIT=1 または INIT=0) によって、セットまたはリセットされます。

メモ : ブロック RAM、LUT RAM、デジタル クロック マネージャー (DCM)、シフト レジスタ LUT (SRL16、SRL16_1、SRL16E、SRL16E_1、SRLC16、SRLC16_1、SRLC16E、SRLC16E_1) は、セットもリセットもされません。

BSCAN がディスエーブルで EXTEST 命令が実行されていない場合、コンフィギュレーションの後にグローバル トライステート信号 (GTS) が High になると、すべての IOB 出力がハイ インピーダンスになり、デバイスの出力が回路から切り離されます。ただし、入力 is アクティブのままです。

メモ : GTS = グローバル トライステート

デザインに STARTUP_SPARTAN3 プリミティブを含めるかどうかはオプションですが、次の状況で含める必要があります。

- ・ グローバル セット/リセットを外部から制御する場合は、GSR ピンを最上位のポートと IBUF に接続します。
- ・ グローバル トライステート信号を外部から制御する場合は、GTS ピンを最上位のポートと IBUF に接続します。
- ・ スタートアップをユーザー クロックに同期させる場合は、ユーザー クロック信号を CLK 入力に接続します。ユーザー クロックは BitGen で選択する必要があります。

ロケーション制約を使用すると、GSR、GTS にアクセスするピンを指定できます。

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 (インスタンスレーション)

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

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

-- STARTUP_SPARTAN3: Startup primitive for GSR, GTS or startup sequence
--                      control.
--                      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

STARTUP_SPARTAN3_inst : STARTUP_SPARTAN3
port map (
    CLK => CLK,          -- Clock input for start-up sequence
    GSR => GSR_PORT,     -- Global Set/Reset input (GSR cannot be used for the port name)
    GTS => GTS_PORT      -- Global 3-state input (GTS cannot be used for the port name)
);

-- End of STARTUP_SPARTAN3_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// STARTUP_SPARTAN3: Startup primitive for GSR, GTS or startup sequence control
//                      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

STARTUP_SPARTAN3 STARTUP_SPARTAN3_inst (
    .CLK(CLK),           // Clock input for start-up sequence
    .GSR(GSR_PORT),     // Global Set/Reset input (GSR can not be used as a port name)
    .GTS(GTS_PORT)      // Global 3-state input (GTS can not be used as a port name)
);

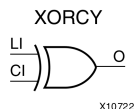
// End of STARTUP_SPARTAN3_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

XORCY

プリミティブ : XOR for Carry Logic with General Output



概要

このデザイン エレメントは、汎用出力 (O) を持つ特殊な XOR ゲートで、高速で小型の演算ファンクションを生成するために使用します。このプリミティブは、スライスのキャリーチェーン ロジック内の専用 XOR ファンクションで、演算ファンクション (加算または除算) または多入力ロジック ファンクション (多入力 AND または OR ゲート) を高速かつ効率的に作成できます。

論理表

入力		出力
LI	CI	O
0	0	0
0	1	1
1	0	1
1	1	0

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- XORCY: Carry-Chain XOR-gate with general output
--      Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

XORCY_inst : XORCY
port map (
    O => O,    -- XOR output signal
    CI => CI,  -- Carry input signal
    LI => LI   -- LUT4 input signal
);

-- End of XORCY_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// XORCY: Carry-Chain XOR-gate with general output
//      Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

XORCY XORCY_inst (
    .O(O), // XOR output signal
    .CI(CI), // Carry input signal
    .LI(LI) // LUT4 input signal
);

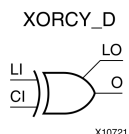
// End of XORCY_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

XORCY_D

プリミティブ : XOR for Carry Logic with Dual Output



概要

このデザイン エLEMENTは、汎用出力 (O) とローカル出力 (LO) を持つ特殊な XOR ゲートで、高速で小型の演算ファンクションを生成するために使用します。

論理表

入力		出力
LI	CI	O および LO
0	0	0
0	1	1
1	0	1
1	1	0

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- XORCY_D: Carry-Chain XOR-gate with local and general outputs
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

XORCY_D_inst : XORCY_D
port map (
    LO => LO, -- XOR local output signal
    O  => O,  -- XOR general output signal
    CI => CI, -- Carry input signal
    LI => LI  -- LUT4 input signal
);

-- End of XORCY_D_inst instantiation
```


Verilog 記述 (インスタンス化)

```
// XORCY_D: Carry-Chain XOR-gate with local and general outputs
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

XORCY_D XORCY_D_inst (
    .LO(LO), // XOR local output signal
    .O(O),   // XOR general output signal
    .CI(CI), // Carry input signal
    .LI(LI)  // LUT4 input signal
);

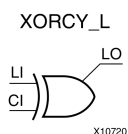
// End of XORCY_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)

XORCY_L

プリミティブ : XOR for Carry Logic with Local Output



概要

このデザイン エLEMENTは、ローカル出力 (LO) のある特殊な XOR ゲートで、高速で小型の演算ファンクションを生成するために使用します。

論理表

入力		出力
LI	CI	LO
0	0	0
0	1	1
1	0	1
1	1	0

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- XORCY_L: Carry-Chain XOR-gate with local  => direct-connect output
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 14.1

XORCY_L_inst : XORCY_L
port map (
    LO => LO, -- XOR local output signal
    CI => CI, -- Carry input signal
    LI => LI  -- LUT4 input signal
);

-- End of XORCY_L_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// XORCY_L: Carry-Chain XOR-gate with local (direct-connect) output
//           Spartan-3
// Xilinx HDL Libraries Guide, version 14.1

XORCY_L XORCY_L_inst (
    .LO(LO), // XOR local output signal
    .CI(CI), // Carry input signal
    .LI(LI)  // LUT4 input signal
);

// End of XORCY_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3 FPGA ファミリー データシート](#)