

Spartan-3E Libraries Guide for HDL Designs

UG617 (v 13.2) July 6, 2011



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

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

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

Introduction

This HDL guide is part of the ISE® documentation collection. A separate version of this guide is available if you prefer to work with schematics.

This guide contains the following:

- Introduction.
- A list of design elements supported in this architecture, organized by functional categories.
- Descriptions of each available primitive.

About Design Elements

This version of the Libraries Guide describes the primitives that comprise the Xilinx Unified Libraries for this architecture, and includes examples of instantiation code for each element. Instantiation templates are also supplied in a separate ZIP file, which you can find in your installation directory under ISE/doc/usenglish/isehelp.

Primitives are Xilinx components that are native to the FPGA you are targeting. If you instantiate a primitive in your design, after the translation process you will end up with the exact same component in the back end. For example, if you instantiate the Virtex-5 element known as ISERDES_NODELAY as a user primitive, after you run translate (ngdbuild) you will end up with an ISERDES_NODELAY in the back end as well. If you were using ISERDES in a Virtex-5 device, then this will automatically retarget to an ISERDES_NODELAY for Virtex-5 in the back end. Hence, this concept of a “primitive” differs from other uses of that term in this technology.

Xilinx maintains software libraries with hundreds of functional design elements (UniMacros and primitives) for different device architectures. New functional elements are assembled with each release of development system software. This guide is one in a series of architecture-specific Libraries Guides describing functional design elements.

Design Entry Methods

For each design element in this guide, Xilinx evaluates four options for using the design element, and recommends what we believe is the best solution for you. The four options are:

- **Instantiation** - This component can be instantiated directly into the design. This method is useful if you want to control the exact placement of the individual blocks.
- **Inference** - This component can be inferred by most supported synthesis tools. You should use this method if you want to have complete flexibility and portability of the code to multiple architectures. Inference also gives the tools the ability to optimize for performance, area, or power, as specified by the user to the synthesis tool.
- **Coregen & Wizards** - This component can be used through CORE Generator or other Wizards. You should use this method if you want to build large blocks of any FPGA primitive that cannot be inferred. When using this flow, you will have to re-generate your cores for each architecture that you are targeting.
- **Macro Support** - This component has a UniMacro that can be used. These components are in the UniMacro library in the Xilinx tool, and are used to instantiate primitives that are too complex to instantiate by just using the primitives. The synthesis tools will automatically expand UniMacros to their underlying primitives.

Functional Categories

This section categorizes, by function, the circuit design elements described in detail later in this guide. The elements (*primitives* and *macros*) are listed in alphanumeric order under each functional category.

| | |
|-------------------------|----------------------|
| Arithmetic Functions | RAM/ROM |
| Clock Components | Registers/Latches |
| Config/BSCAN Components | Shift Register LUT |
| I/O Components | Slice/CLB Primitives |

Arithmetic Functions

| Design Element | Description |
|------------------------------|---|
| MULT18X18 | Primitive: 18 x 18 Signed Multiplier |
| MULT18X18S | Primitive: 18 x 18 Signed Multiplier -- Registered Version |
| MULT18X18SIO | Primitive: 18 x 18 Cascadable Signed Multiplier with Optional Input and Output Registers, Clock Enable, and Synchronous Reset |

Clock Components

| Design Element | Description |
|---------------------------|---|
| BUFG | Primitive: Global Clock Buffer |
| BUFGCE | Primitive: Global Clock Buffer with Clock Enable |
| BUFGCE_1 | Primitive: Global Clock Buffer with Clock Enable and Output State 1 |
| BUFGMUX | Primitive: Global Clock MUX Buffer |
| BUFGMUX_1 | Primitive: Global Clock MUX Buffer with Output State 1 |
| DCM_SP | Primitive: Digital Clock Manager |

Config/BSCAN Components

| Design Element | Description |
|-------------------|--|
| BSCAN_SPARTAN3 | Primitive: Spartan®-3 and Spartan-3E JTAG Boundary Scan Logic Access Circuit |
| CAPTURE_SPARTAN3 | Primitive: Spartan®-3 Register State Capture for Bitstream Readback |
| STARTUP_SPARTAN3E | Primitive: Spartan®-3E User Interface to the GSR, GTS, Configuration Startup Sequence and Multi-Boot Trigger Circuitry |

I/O Components

| Design Element | Description |
|----------------|--|
| IBUF | Primitive: Input Buffer |
| IBUFDS | Primitive: Differential Signaling Input Buffer |
| IBUFG | Primitive: Dedicated Input Clock Buffer |
| IBUFGDS | Primitive: Differential Signaling Dedicated Input Clock Buffer and Optional Delay |
| IOBUF | Primitive: Bi-Directional Buffer |
| IOBUFDS | Primitive: 3-State Differential Signaling I/O Buffer with Active Low Output Enable |
| KEEPER | Primitive: KEEPER Symbol |
| OBUF | Primitive: Output Buffer |
| OBUFDS | Primitive: Differential Signaling Output Buffer |
| OBUFT | Primitive: 3-State Output Buffer with Active Low Output Enable |
| OBUFTDS | Primitive: 3-State Output Buffer with Differential Signaling, Active-Low Output Enable |
| PULLDOWN | Primitive: Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs |
| PULLUP | Primitive: Resistor to VCC for Input PADS, Open-Drain, and 3-State Outputs |

RAM/ROM

| Design Element | Description |
|----------------|--|
| RAM16X1D | Primitive: 16-Deep by 1-Wide Static Dual Port Synchronous RAM |
| RAM16X1D_1 | Primitive: 16-Deep by 1-Wide Static Dual Port Synchronous RAM with Negative-Edge Clock |
| RAM16X1S | Primitive: 16-Deep by 1-Wide Static Synchronous RAM |
| RAM16X1S_1 | Primitive: 16-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock |
| RAM16X2S | Primitive: 16-Deep by 2-Wide Static Synchronous RAM |
| RAM32X1S | Primitive: 32-Deep by 1-Wide Static Synchronous RAM |

| Design Element | Description |
|--------------------------------|--|
| RAM32X1S_1 | Primitive: 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock |
| RAM32X2S | Primitive: 32-Deep by 2-Wide Static Synchronous RAM |
| RAM64X1S | Primitive: 64-Deep by 1-Wide Static Synchronous RAM |
| RAM64X1S_1 | Primitive: 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock |
| RAMB16_S1 | Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 1-bit Port |
| RAMB16_S1_S1 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit Ports |
| RAMB16_S1_S18 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 18-bit Ports |
| RAMB16_S1_S2 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 2-bit Ports |
| RAMB16_S1_S36 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 36-bit Ports |
| RAMB16_S1_S4 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 4-bit Ports |
| RAMB16_S1_S9 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 9-bit Ports |
| RAMB16_S18 | Primitive: 16K-bit Data + 2K-bit Parity Memory, Single-Port Synchronous Block RAM with 18-bit Port |
| RAMB16_S18_S18 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit Ports |
| RAMB16_S18_S36 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit and 36-bit Ports |
| RAMB16_S2 | Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 2-bit Port |
| RAMB16_S2_S2 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit Ports |
| RAMB16_S2_S18 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 18-bit Ports |
| RAMB16_S2_S36 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 36-bit Ports |
| RAMB16_S2_S4 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 4-bit Ports |
| RAMB16_S2_S9 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 9-bit Ports |
| RAMB16_S36 | Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 36-bit Port |
| RAMB16_S36_S36 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with Two 36-bit Ports |
| RAMB16_S4 | Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 4-bit Port |
| RAMB16_S4_S18 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 18-bit Ports |

| Design Element | Description |
|-------------------------------|---|
| RAMB16_S4_S36 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 36-bit Ports |
| RAMB16_S4_S4 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit Ports |
| RAMB16_S4_S9 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 9-bit Ports |
| RAMB16_S9 | Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 9-bit Port |
| RAMB16_S9_S18 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 18-bit Ports |
| RAMB16_S9_S36 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 36-bit Ports |
| RAMB16_S9_S9 | Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit Ports |
| ROM128X1 | Primitive: 128-Deep by 1-Wide ROM |
| ROM16X1 | Primitive: 16-Deep by 1-Wide ROM |
| ROM256X1 | Primitive: 256-Deep by 1-Wide ROM |
| ROM32X1 | Primitive: 32-Deep by 1-Wide ROM |
| ROM64X1 | Primitive: 64-Deep by 1-Wide ROM |

Registers/Latches

| Design Element | Description |
|-----------------------|---|
| FDCE | Primitive: D Flip-Flop with Clock Enable and Asynchronous Clear |
| FDCPE | Primitive: D Flip-Flop with Clock Enable and Asynchronous Preset and Clear |
| FDRSE | Primitive: D Flip-Flop with Synchronous Reset and Set and Clock Enable |
| IDDR2 | Primitive: Double Data Rate Input D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset |
| LDCE | Primitive: Transparent Data Latch with Asynchronous Clear and Gate Enable |
| LDCPE | Primitive: Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable |
| LDPE | Primitive: Transparent Data Latch with Asynchronous Preset and Gate Enable |
| ODDR2 | Primitive: Dual Data Rate Output D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset |

Shift Register LUT

| Design Element | Description |
|---------------------------|--|
| SRL16 | Primitive: 16-Bit Shift Register Look-Up Table (LUT) |
| SRL16_1 | Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock |
| SRL16E | Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Clock Enable |
| SRL16E_1 | Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock and Clock Enable |
| SRLC16 | Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry |
| SRLC16_1 | Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry and Negative-Edge Clock |
| SRLC16E | Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry and Clock Enable |
| SRLC16E_1 | Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry, Negative-Edge Clock, and Clock Enable |

Slice/CLB Primitives

| Design Element | Description |
|--------------------------|---|
| BUFCF | Primitive: Fast Connect Buffer |
| LUT1 | Primitive: 1-Bit Look-Up Table with General Output |
| LUT1_D | Primitive: 1-Bit Look-Up Table with Dual Output |
| LUT1_L | Primitive: 1-Bit Look-Up Table with Local Output |
| LUT2 | Primitive: 2-Bit Look-Up Table with General Output |
| LUT2_D | Primitive: 2-Bit Look-Up Table with Dual Output |
| LUT2_L | Primitive: 2-Bit Look-Up Table with Local Output |
| LUT3 | Primitive: 3-Bit Look-Up Table with General Output |
| LUT3_D | Primitive: 3-Bit Look-Up Table with Dual Output |
| LUT3_L | Primitive: 3-Bit Look-Up Table with Local Output |
| LUT4 | Primitive: 4-Bit Look-Up-Table with General Output |
| LUT4_D | Primitive: 4-Bit Look-Up Table with Dual Output |
| LUT4_L | Primitive: 4-Bit Look-Up Table with Local Output |
| MULT_AND | Primitive: Fast Multiplier AND |
| MUXCY | Primitive: 2-to-1 Multiplexer for Carry Logic with General Output |
| MUXCY_D | Primitive: 2-to-1 Multiplexer for Carry Logic with Dual Output |
| MUXCY_L | Primitive: 2-to-1 Multiplexer for Carry Logic with Local Output |
| MUXF5 | Primitive: 2-to-1 Look-Up Table Multiplexer with General Output |
| MUXF5_D | Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output |

| Design Element | Description |
|-------------------------|---|
| MUXF5_L | Primitive: 2-to-1 Look-Up Table Multiplexer with Local Output |
| MUXF6 | Primitive: 2-to-1 Look-Up Table Multiplexer with General Output |
| MUXF6_D | Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output |
| MUXF6_L | Primitive: 2-to-1 Look-Up Table Multiplexer with Local Output |
| MUXF7 | Primitive: 2-to-1 Look-Up Table Multiplexer with General Output |
| MUXF7_D | Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output |
| MUXF7_L | Primitive: 2-to-1 look-up table Multiplexer with Local Output |
| MUXF8 | Primitive: 2-to-1 Look-Up Table Multiplexer with General Output |
| MUXF8_D | Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output |
| MUXF8_L | Primitive: 2-to-1 Look-Up Table Multiplexer with Local Output |
| XORCY | Primitive: XOR for Carry Logic with General Output |
| XORCY_D | Primitive: XOR for Carry Logic with Dual Output |
| XORCY_L | Primitive: XOR for Carry Logic with Local Output |

About Design Elements

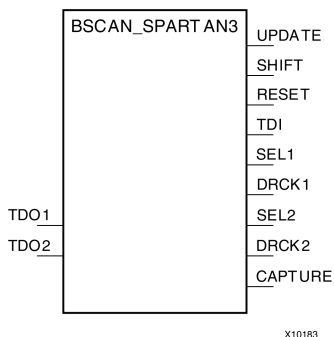
This section describes the design elements that can be used with this architecture. The design elements are organized alphabetically.

The following information is provided for each design element, where applicable:

- Name of element
- Brief description
- Schematic symbol (if any)
- Logic table (if any)
- Port descriptions
- Design Entry Method
- Available attributes (if any)
- Example instantiation code
- For more information

BSCAN_SPARTAN3

Primitive: Spartan®-3 and Spartan-3E JTAG Boundary Scan Logic Access Circuit



Introduction

This design element allows access to and from internal logic by the JTAG Boundary Scan logic controller. This allows for communication between the internal running design and the dedicated JTAG pins of the FPGA.

Note For specific information on boundary scan for an architecture, see the Programmable Logic Data Sheet for this element.

Port Descriptions

| Port | Direction | Width | Function |
|-------------|-----------|-------|---|
| TDI | Output | 1 | A mirror of the TDI input pin to the FPGA. |
| DRCK1, DRK2 | Output | 1 | A mirror of the TCK input pin to the FPGA when the JTAG USER instruction is loaded and the JTAG TAP controller is in the SHIFT-DR state. DRK1 applies to the USER1 logic while DRK2 applies to USER2. |
| RESET | Output | 1 | Active upon the loading of the USER instruction. It asserts High when the JTAG TAP controller is in the TEST-LOGIC-RESET state. |
| SEL1, SEL2 | Output | 1 | Indicates when the USER1 or USER2 instruction has been loaded into the JTAG Instruction Register. SEL1 or SEL2 becomes active in the UPDATE-IR state, and stays active until a new instruction is loaded. |
| SHIFT | Output | 1 | Active upon the loading of the USER instruction. It asserts High when the JTAG TAP controller is in the SHIFT-DR state. |
| CAPTURE | Output | 1 | Active upon the loading of the USER instruction. Asserts High when the JTAG TAP controller is in the CAPTURE-DR state. |
| UPDATE | Output | 1 | Active upon the loading of the USER instruction. It asserts High when the JTAG TAP controller is in the UPDATE-DR state. |
| TDO1, TDO2 | Input | 1 | Active upon the loading of the USER1 or USER2 instruction. External JTAG TDO pin reflects data input to the component's TDO1 (USER1) or TDO2 (USER2) pin. |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Recommended |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- BSCAN_SPARTAN3: Boundary Scan primitive for connecting internal logic to
--                   JTAG interface.
--                   Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// BSCAN_SPARTAN3: Boundary Scan primitive for connecting internal logic to
//                   JTAG interface.
//                   Spartan-3E/3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

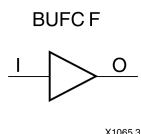
// End of BSCAN_SPARTAN3_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

BUFCF

Primitive: Fast Connect Buffer



Introduction

This design element is a single fast connect buffer used to connect the outputs of the LUTs and some dedicated logic directly to the input of another LUT. Using this buffer implies CLB packing. No more than four LUTs may be connected together as a group.

Design Entry Method

| Instantiation | Recommended |
|-----------------------------|-------------|
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
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-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// BUFCF: Fast connect buffer used to connect the outputs of the LUTs
//       and some dedicated logic directly to the input of another LUT.
//       Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

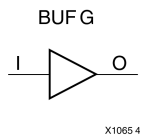
// End of BUFCF_inst instantiation
```

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

BUFG

Primitive: Global Clock Buffer



Introduction

This design element is a high-fanout buffer that connects signals to the global routing resources for low skew distribution of the signal. BUFGs are typically used on clock nets as well other high fanout nets like sets/resets and clock enables.

Port Descriptions

| Port | Type | Width | Function |
|------|--------|-------|---------------------|
| I | Input | 1 | Clock buffer input |
| O | Output | 1 | Clock buffer output |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFG: Global Clock Buffer (source by an internal signal)
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of BUFG_inst instantiation

```


Verilog Instantiation Template

```
// BUFG: Global Clock Buffer (source by an internal signal)
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

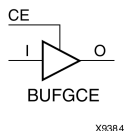
// End of BUFG_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

BUFGCE

Primitive: Global Clock Buffer with Clock Enable



Introduction

This design element is a global clock buffer with a single gated input. Its O output is "0" when clock enable (CE) is Low (inactive). When clock enable (CE) is High, the I input is transferred to the O output.

Logic Table

| Inputs | | Outputs |
|--------|----|---------|
| I | CE | O |
| X | 0 | 0 |
| I | 1 | I |

Port Descriptions

| Port | Type | Width | Function |
|------|--------|-------|---------------------|
| I | Input | 1 | Clock buffer input |
| CE | Input | 1 | Clock enable input |
| O | Output | 1 | Clock buffer output |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGCE: Global Clock Buffer with Clock Enable (active high)
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// BUFGCE: Global Clock Buffer with Clock Enable (active high)
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

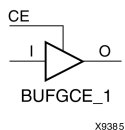
// End of BUFGCE_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

BUFGCE_1

Primitive: Global Clock Buffer with Clock Enable and Output State 1



Introduction

This design element is a multiplexed global clock buffer with a single gated input. Its O output is High (1) when clock enable (CE) is Low (inactive). When clock enable (CE) is High, the I input is transferred to the O output.

Logic Table

| Inputs | | Outputs |
|--------|----|---------|
| I | CE | O |
| X | 0 | 1 |
| I | 1 | I |

Port Descriptions

| Port | Type | Width | Function |
|------|--------|-------|---------------------|
| I | Input | 1 | Clock buffer input |
| CE | Input | 1 | Clock enable input |
| O | Output | 1 | Clock buffer output |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- BUFGCE_1: Global Clock Buffer with Clock Enable (active low)
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// BUFGCE_1: Global Clock Buffer with Clock Enable (active low)
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

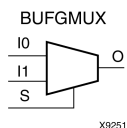
// End of BUFGCE_1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

BUFGMUX

Primitive: Global Clock MUX Buffer



Introduction

BUFGMUX is a multiplexed global clock buffer that can select between two input clocks: I0 and I1. When the select input (S) is Low, the signal on I0 is selected for output (O). When the select input (S) is High, the signal on I1 is selected for output.

BUFGMUX and BUFGMUX_1 are distinguished by the state the output assumes when that output switches between clocks in response to a change in its select input. BUFGMUX assumes output state 0 and BUFGMUX_1 assumes output state 1.

Note BUFGMUX guarantees that when S is toggled, the state of the output remains in the inactive state until the next active clock edge (either I0 or I1) occurs.

Logic Table

| Inputs | | | Outputs |
|--------|----|---|---------|
| I0 | I1 | S | O |
| I0 | X | 0 | I0 |
| X | I1 | 1 | I1 |
| X | X | ↑ | 0 |
| X | X | ↓ | 0 |

Port Descriptions

| Port | Type | Width | Function |
|------|--------|-------|--------------------|
| I0 | Input | 1 | Clock0 input |
| I1 | Input | 1 | Clock1 input |
| O | Output | 1 | Clock MUX output |
| S | Input | 1 | Clock select input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Recommended |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- BUFGMUX: Global Clock Buffer 2-to-1 MUX
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// BUFGMUX: Global Clock Buffer 2-to-1 MUX
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

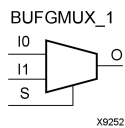
// End of BUFGMUX_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

BUFGMUX_1

Primitive: Global Clock MUX Buffer with Output State 1



Introduction

This design element is a multiplexed global clock buffer that can select between two input clocks: I0 and I1. When the select input (S) is Low, the signal on I0 is selected for output (O). When the select input (S) is High, the signal on I1 is selected for output.

This design element is distinguished from BUFGMUX by the state the output assumes when that output switches between clocks in response to a change in its select input. BUFGMUX assumes output state 0 and BUFGMUX_1 assumes output state 1.

Logic Table

| Inputs | | | Outputs |
|--------|----|---|---------|
| I0 | I1 | S | O |
| I0 | X | 0 | I0 |
| X | I1 | 1 | I1 |
| X | X | ↑ | 1 |
| X | X | ↓ | 1 |

Port Descriptions

| Port | Type | Width | Function |
|------|--------|-------|--------------------|
| I0 | Input | 1 | Clock0 input |
| I1 | Input | 1 | Clock1 input |
| O | Output | 1 | Clock MUX output |
| S | Input | 1 | Clock select input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- BUFGMUX_1: Global Clock Buffer 2-to-1 MUX (inverted select)
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// BUFGMUX_1: Global Clock Buffer 2-to-1 MUX (inverted select)
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

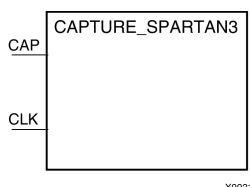
// End of BUFGMUX_1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

CAPTURE_SPARTAN3

Primitive: Spartan®-3 Register State Capture for Bitstream Readback



Introduction

This element provides user control and synchronization over when and how the capture register (flip-flop and latch) information task is requested. The readback function is provided through dedicated configuration port instructions. However, without this element, the readback data is synchronized to the configuration clock. Only register (flip-flop and latch) states can be captured. Although LUT RAM, SRL, and block RAM states are readback, they cannot be captured.

An asserted high CAP signal indicates that the registers in the device are to be captured at the next Low-to-High clock transition. By default, data is captured after every trigger when transition on CLK while CAP is asserted. To limit the readback operation to a single data capture, add the ONESHOT=TRUE attribute to this element.

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--------------------------|
| CAP | Input | 1 | Readback capture trigger |
| CLK | Input | 1 | Readback capture clock |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Recommended |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

Connect all inputs and outputs to the design in order to ensure proper operation.

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|---------|----------------|---------|---|
| ONESHOT | Boolean | TRUE, FALSE | TRUE | Specifies the procedure for performing single readback per CAP trigger. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- CAPTURE_SPARTAN3: Register State Capture for Bitstream Readback
--                               Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of CAPTURE_SPARTAN3_inst instantiation
```

Verilog Instantiation Template

```
// CAPTURE_SPARTAN3: Register State Capture for Bitstream Readback
//                               Spartan-3E/3E
// Xilinx HDL Libraries Guide, version 13.2

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

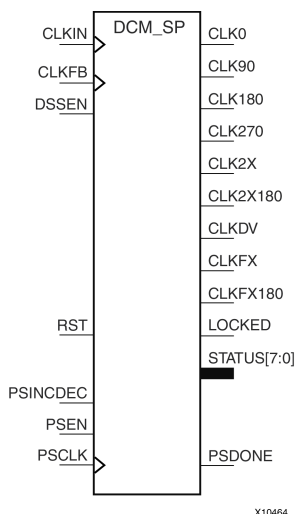
// End of CAPTURE_SPARTAN3_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

DCM_SP

Primitive: Digital Clock Manager



Introduction

This design element is a digital clock manager that provides multiple functions. It can implement a clock delay locked loop (DLL), a digital frequency synthesizer (DFS), and a digital phase shifter (DPS). DCM_SPs are useful for eliminating the clock delay coming on and off the chip, shifting the clock phase to improve data capture, deriving different frequency clocks, as well as other useful clocking functions.

Port Descriptions

| Port | Type | Width | Function |
|----------|--------|-------|--|
| CLKDV | Output | 1 | Divided clock output, controlled by the CLKDV_DIVIDE attribute. The CLKDV output has a 50% duty cycle unless the CLKDV_DIVIDE attribute is a non-integer value. |
| CLKFB | Input | 1 | Clock feedback input to DCM. The feedback input is required unless the DFS outputs, CLKFX or CLKFX180, are used standalone. The source of the CLKFB input must be the CLK0 or CLK2X output from the DCM and the CLK_FEEDBACK must be set to 1X or 2X accordingly. When set to NONE, CLKFB is unused and should be tied low. Ideally, the feedback point includes the delay added by the clock distribution network, either internally or externally. |
| CLKFX | Output | 1 | Synthesized clock output, controlled by the CLKFX_MULTIPLY and CLKFX_DIVIDE attributes. Always has a 50% duty cycle. If no phase relationship is necessary, then no clock feedback is required. |
| CLKFX180 | Output | 1 | Synthesized clock output CLKFX, 180 phase shift (appears to be an inverted version of CLKFX). Always has a 50% duty cycle. If no phase relationship is necessary, then no feedback loop is required. |
| CLKIN | Input | 1 | Clock input to DCM. Always required. The CLKIN frequency and jitter must fall within the limits specified in the data sheet. |
| CLK0 | Output | 1 | Same frequency as CLKIN, 0 phase shift (i.e., not phase shifted). |
| CLK2X | Output | 1 | Double-frequency clock output, 0 phase shift. When available, the CLK2X output always has a 50% duty cycle. Either CLK0 or CLK2X is required as a feedback source for DLL functions. |

| Port | Type | Width | Function |
|-------------|--------|-------|---|
| CLK2X180 | Output | 1 | Double-frequency clock output, 180 phase shift. When available, the CLK2X180 output always has a 50% duty cycle. |
| CLK90 | Output | 1 | Same frequency as CLKIN, 90 phase shift (quarter period). |
| CLK180 | Output | 1 | Same frequency as CLKIN, 180 phase shift (half period). |
| CLK270 | Output | 1 | Same frequency as CLKIN, 270 phase shift (three-quarters period). |
| LOCKED | Output | 1 | All DCM features have locked onto the CLKIN frequency. Clock outputs are now valid, assuming CLKIN is within specified limits. <ul style="list-style-type: none"> 0 - DCM is attempting to lock onto CLKIN frequency. DCM clock outputs are not valid. 1 - DCM is locked onto CLKIN frequency. DCM clock outputs are valid. 1-to-0 - DCM lost lock. Reset DCM. |
| PSCLK | Input | 1 | Clock input to variable phase shifter, clocked on rising edge. When using a global clock buffer, only the upper eight BUFGMUXs can drive PSCLK: BUFGMUX_X2Y1, BUFGMUX_X2Y2, BUFGMUX_X2Y3, BUFGMUX_X2Y4, BUFGMUX_X3Y5, BUFGMUX_X3Y6, BUFGMUX_X3Y7 and BUFGMUX_X3Y8. |
| PSDONE | Output | 1 | Variable phase shift operation complete. <ul style="list-style-type: none"> 0 - No phase shift operation is active or phase shift operation is in progress. 1 - Requested phase shift operation is complete. Output High for one PSCLK cycle. Next variable phase shift operation can commence. |
| PSEN | Input | 1 | Variable phase-shift enable. Can be inverted within a DCM block. Non-inverted behavior shown below. <ul style="list-style-type: none"> 0 - Disable variable phase shift. Ignore inputs to phase shifter. 1 - Enable variable phase shift operations on next rising PSCLK clock edge. Note Tie to 0 when not in use. |
| PSINCDEC | Input | 1 | Increment/decrement variable phase shift. Can be inverted within a DCM block. Non-inverted behavior shown below. <ul style="list-style-type: none"> 0 - Decrement phase shift value on next enabled, rising PSCLK clock edge. 1 - Increment phase shift value on next enabled, rising PSCLK clock edge. |
| RST | Input | 1 | Asynchronous reset input. Resets the DCM logic to its postconfiguration state. Causes DCM to reacquire and relock to the CLKIN input. Invertible within DCM block. Non-inverted behavior shown below. <ul style="list-style-type: none"> 0 - No effect. 1 - Reset DCM block. Hold RST pulse High for at least three valid CLKIN cycles. |
| STATUS[7:0] | Output | 8 | The status output bus provides DCM status. <ul style="list-style-type: none"> STATUS[0] - Variable phase shift overflow. Control output for variable fine phase shifting. The variable phase shifter has reached a minimum or maximum limit value. The limit value is either +/-255 or a lesser value if the phase shift has reached the end of the delay line. |

| Port | Type | Width | Function |
|------|------|-------|--|
| | | | <ul style="list-style-type: none"> – 0 - The phase shift has not yet reached its limit value. – 1 - The phase shift has reached its limited value. • STATUS[1] - CLKIN Input Stopped Indicator. Available only when the CLKFB feedback input is connected. Held in reset until the LOCKED output is asserted. Requires at least one CLKIN cycle to become active. Never asserted if CLKIN never toggles. <ul style="list-style-type: none"> – 0 - CLKIN input is toggling. – 1 - CLKIN input is not toggling even though the locked output can still be High. • STATUS[2] - CLKFX or CLKFX180 output stopped indicator. <ul style="list-style-type: none"> – 0 - CLKFX and CLKFX180 outputs are toggling. – 1 - CLKFX and CLKFX180 outputs are not toggling, even though the LOCKED output can still be High. • STATUS[4:3] - Reserved. |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | No |
| CORE Generator™ and wizards | Recommended |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed_Values | Default | Description |
|-------------------|---------------------------|---|---------|--|
| CLK_FEEDBACK | String | "1X", "2X", "NONE" | "1X" | Defines the DCM feedback mode. <ul style="list-style-type: none"> • 1X: CLK0 as feedback. • 2X: CLK2X as feedback. |
| CLKDV_DIVIDE | 1 significant digit Float | 2.0, 1.5, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0 | 2.0 | Specifies the extent to which the CLKDLL, CLKDLLE, CLKDLLHF, or DCM_SP clock divider (CLKDV output) is to be frequency divided. |
| CLKFX_DIVIDE | Integer | 1 to 32 | 1 | Specifies the frequency divider value for the CLKFX output. |
| CLKFX_MULTIPLY | Integer | 2 to 32 | 4 | Specifies the frequency multiplier value for the CLKFX output. |
| CLKIN_DIVIDE_BY_2 | Boolean | FALSE, TRUE | FALSE | Enables CLKIN divide by two features. |
| CLKIN_PERIOD | Float | 2.000 to 1000.000 | None | Specifies the input period to the DCM_SP CLKIN input in ns. |

| Attribute | Type | Allowed_Values | Default | Description |
|-----------------------|-------------|--|----------------------|--|
| CLKOUT_PHASE_SHIFT | String | "NONE", "FIXED", "VARIABLE" | "NONE" | This attribute specifies the phase shift mode. <ul style="list-style-type: none"> NONE: No phase shift capability. Any set value has no effect. FIXED: DCM outputs are a fixed phase shift from CLKIN. Value is specified by PHASE_SHIFT attribute. VARIABLE: Allows the DCM outputs to be shifted in a positive and negative range relative to CLKIN. Starting value is specified by PHASE_SHIFT. |
| DESKEW_ADJUST | String | "SYSTEM_SYNCHRONOUS", "SOURCE_SYNCHRONOUS" | "SYSTEM_SYNCHRONOUS" | Sets configuration bits affecting the clock delay alignment between the DCM_SP output clocks and an FPGA clock input pin. |
| DFS_FREQUENCY_MODE | String | "LOW", "HIGH" | "LOW" | This is a legacy attribute. The DCM is always in the automatic frequency search mode. Setting High or Low makes no effect. |
| DLL_FREQUENCY_MODE | String | "LOW", "HIGH" | "LOW" | This is a legacy attribute. The DCM is always in the automatic frequency search mode. Setting High or Low makes no effect. |
| DSS_MODE | String | "NONE", "SPREAD_2", "SPREAD_4", "SPREAD_6", "SPREAD_8" | "NONE" | Specifies a frequency spread for output clocks. <ul style="list-style-type: none"> NONE - The default, specifies no spread factors. The digital spread spectrum function is disabled. SPREAD_2 - Creates a new clock period that is +/- 50 ps of the current clock period SPREAD_4 - Creates a new clock period that is +/- 100 ps of the current clock period. SPREAD_6 - Creates a new clock period that is +/- 150 ps of the current clock period. SPREAD_8 - Creates a new clock period that is +/- 200 ps of the current clock period. <p>The spreading is cumulative as the SPREAD_# is increased. For example, SPREAD_2 creates two additional clock frequencies at +/-50 ps relative to the input clock frequency; SPREAD_4 does the same as SPREAD_2, plus it creates two additional clock frequencies at +/-100 ps.</p> |
| DUTY_CYCLE_CORRECTION | Boolean | TRUE, FALSE | TRUE | Unsupported |
| FACTORY_JF | Hexadecimal | 16'h8080 to 16'hffff | 16'hc080 | Unsupported |

| Attribute | Type | Allowed_Values | Default | Description |
|--------------|---------|----------------|---------|---|
| PHASE_SHIFT | Integer | -255 to 255 | 0 | The PHASE_SHIFT attribute is applicable only if the CLKOUT_PHASE_SHIFT attribute is set to FIXED or VARIABLE. Defines the rising-edge skew between CLKIN and all the DCM clock outputs at configuration and consequently phase shifts the DCM clock outputs. The skew or phase shift value is specified as an integer that represents a fraction of the clock period as expressed in the equations in Fine Phase Shifting. Actual allowable values depends on input clock frequency. The actual range is less when TCLKIN > FINE_SHIFT_RANGE. The FINE_SHIFT_RANGE specification represents the total delay of all taps in the delay line. |
| STARTUP_WAIT | Boolean | FALSE, TRUE | FALSE | <p>Controls whether the FPGA configuration signal DONE waits for the DCM to assert its LOCKED signal before going High.</p> <ul style="list-style-type: none"> FALSE - Default. DONE is asserted at the end of configuration without waiting for the DCM to assert LOCKED. TRUE - The DONE signal does not transition High until the LOCKED signal transitions High on the associated DCM. <p>STARTUP_WAIT does not prevent LOCKED from transitioning High. The FPGA startup sequence must also be modified to insert a LCK (lock) cycle before the postponed cycle. The DONE cycle or the GWE cycle are typical choices. When more than one DCM is configured, the FPGA waits until all DCMs are LOCKED.</p> |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

```
-- DCM_SP: Digital Clock Manager Circuit
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2
```

```
DCM_SP_inst : DCM_SP
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
```



```

DLL_FREQUENCY_MODE => "LOW",      -- "HIGH" or "LOW" frequency mode for DLL
DUTY_CYCLE_CORRECTION => TRUE, -- Duty cycle correction, TRUE or FALSE
PHASE_SHIFT => 0,      -- Amount of fixed phase shift from -255 to 255
STARTUP_WAIT => FALSE) -- Delay configuration DONE until DCM_SP LOCK, TRUE/FALSE
port map (
  CLK0 => CLK0,      -- 0 degree DCM CLK output
  CLK180 => CLK180, -- 180 degree DCM CLK output
  CLK270 => CLK270, -- 270 degree DCM CLK output
  CLK2X => CLK2X,    -- 2X DCM CLK output
  CLK2X180 => CLK2X180, -- 2X, 180 degree DCM CLK out
  CLK90 => CLK90,    -- 90 degree DCM CLK output
  CLKDV => CLKDV,    -- Divided DCM CLK out (CLKDV_DIVIDE)
  CLKFX => CLKFX,    -- DCM CLK synthesis out (M/D)
  CLKFX180 => CLKFX180, -- 180 degree CLK synthesis out
  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_SP_inst instantiation

```

Verilog Instantiation Template

```

// DCM_SP: Digital Clock Manager Circuit
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

DCM_SP #(
  .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
  .DLL_FREQUENCY_MODE("LOW"), // HIGH or LOW frequency mode for DLL
  .DUTY_CYCLE_CORRECTION("TRUE"), // Duty cycle correction, TRUE or FALSE
  .PHASE_SHIFT(0), // Amount of fixed phase shift from -255 to 255
  .STARTUP_WAIT("FALSE") // Delay configuration DONE until DCM LOCK, TRUE/FALSE
) DCM_SP_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_SP_inst instantiation

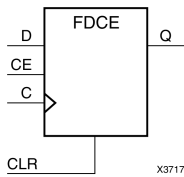
```

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

FDCE

Primitive: D Flip-Flop with Clock Enable and Asynchronous Clear



Introduction

This design element is a single D-type flip-flop with clock enable and asynchronous clear. When clock enable (CE) is High and asynchronous clear (CLR) is Low, the data on the data input (D) of this design element is transferred to the corresponding data output (Q) during the Low-to-High clock (C) transition. When CLR is High, it overrides all other inputs and resets the data output (Q) Low. When CE is Low, clock transitions are ignored.

This flip-flop is asynchronously cleared, outputs Low, when power is applied. For FPGA devices, power-on conditions are simulated when global set/reset (GSR) is active. GSR defaults to active-High but can be inverted by adding an inverter in front of the GSR input of the appropriate `STARTUP_architecture` symbol.

Logic Table

| Inputs | | | | Outputs |
|--------|----|---|---|-----------|
| CLR | CE | D | C | Q |
| 1 | X | X | X | 0 |
| 0 | 0 | X | X | No Change |
| 0 | 1 | D | ↑ | D |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|--------|----------------|---------|--|
| INIT | Binary | 0, 1 | 0 | Sets the initial value of Q output after configuration |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
--       Clock Enable (posedge clk).
--       Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of FDCE_inst instantiation
```

Verilog Instantiation Template

```
// FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
//       Clock Enable (posedge clk).
//       Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

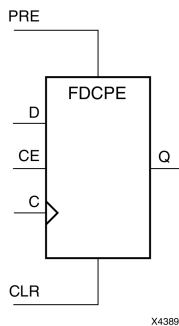
// End of FDCE_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

FDCPE

Primitive: D Flip-Flop with Clock Enable and Asynchronous Preset and Clear



Introduction

This design element is a single D-type flip-flop with data (D), clock enable (CE), asynchronous preset (PRE), and asynchronous clear (CLR) inputs. The asynchronous active high PRE sets the Q output High; that active high CLR resets the output Low and has precedence over the PRE input. Data on the D input is loaded into the flip-flop when PRE and CLR are Low and CE is High on the Low-to-High clock (C) transition. When CE is Low, the clock transitions are ignored and the previous value is retained. The FDCPE is generally implemented as a slice or IOB register within the device.

For FPGA devices, upon power-up, the initial value of this component is specified by the INIT attribute. If a subsequent GSR (Global Set/Reset) is asserted, the flop is asynchronously set to the INIT value.

Note While this device supports the use of asynchronous set and reset, it is not generally recommended to be used for in most cases. Use of asynchronous signals pose timing issues within the design that are difficult to detect and control and also have an adverse affect on logic optimization causing a larger design that can consume more power than if a synchronous set or reset is used.

Logic Table

| Inputs | | | | | Outputs |
|--------|-----|----|---|---|-----------|
| CLR | PRE | CE | D | C | Q |
| 1 | X | X | X | X | 0 |
| 0 | 1 | X | X | X | 1 |
| 0 | 0 | 0 | X | X | No Change |
| 0 | 0 | 1 | D | ↑ | D |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--------------------------|
| Q | Output | 1 | Data output |
| C | Input | 1 | Clock input |
| CE | Input | 1 | Clock enable input |
| CLR | Input | 1 | Asynchronous clear input |
| D | Input | 1 | Data input |
| PRE | Input | 1 | Asynchronous set input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|--------|----------------|---------|--|
| INIT | Binary | 0,1 | 0 | Sets the initial value of Q output after configuration and on GSR. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
--       Clock Enable (posedge clk).
--       Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```

// FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
//       Clock Enable (posedge clk).
//       Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

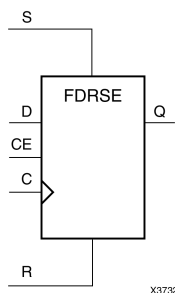
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

FDRSE

Primitive: D Flip-Flop with Synchronous Reset and Set and Clock Enable



Introduction

FDRSE is a single D-type flip-flop with synchronous reset (R), synchronous set (S), clock enable (CE) inputs. The reset (R) input, when High, overrides all other inputs and resets the Q output Low during the Low-to-High clock transition. (Reset has precedence over Set.) When the set (S) input is High and R is Low, the flip-flop is set, output High, during the Low-to-High clock (C) transition. Data on the D input is loaded into the flip-flop when R and S are Low and CE is High during the Low-to-High clock transition.

Upon power-up, the initial value of this component is specified by the INIT attribute. If a subsequent GSR (Global Set/Reset) is asserted, the flop is asynchronously set to the INIT value.

Logic Table

| Inputs | | | | | Outputs |
|--------|---|----|---|---|-----------|
| R | S | CE | D | C | Q |
| 1 | X | X | X | ↑ | 0 |
| 0 | 1 | X | X | ↑ | 1 |
| 0 | 0 | 0 | X | X | No Change |
| 0 | 0 | 1 | 1 | ↑ | 1 |
| 0 | 0 | 1 | 0 | ↑ | 0 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|--------|----------------|---------|--|
| INIT | Binary | 0,1 | 0 | Sets the initial value of Q output after configuration and on GSR. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
--       Clock Enable (posedge clk).
--       Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//       Clock Enable (posedge clk).
//       Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

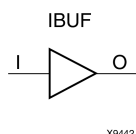
// End of FDRSE_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

IBUF

Primitive: Input Buffer



Introduction

This design element is automatically inserted (inferred) by the synthesis tool to any signal directly connected to a top-level input or in-out port of the design. You should generally let the synthesis tool infer this buffer. However, it can be instantiated into the design if required. In order to do so, connect the input port (I) directly to the associated top-level input or in-out port, and connect the output port (O) to the logic sourced by that port. Modify any necessary generic maps (VHDL) or named parameter value assignment (Verilog) in order to change the default behavior of the component.

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|---------------|
| O | Output | 1 | Buffer output |
| I | Input | 1 | Buffer input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

This element is usually inferred by the synthesis tool for any specified top-level input port to the design, and therefore it is generally not necessary to specify the element in source code. However, if desired, this element may be manually instantiated by copying the instantiation code from below and pasting it into the top-level entity/module of your code. Xilinx recommends that you put all I/O components on the top-level of the design to help facilitate hierarchical design methods. Connect the I port directly to the top-level input port of the design and the O port to the logic in which this input is to source. Specify the desired generic/default values in order to configure the proper behavior of the buffer.

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|------------------|--------|-------------------------|-----------|--|
| IBUF_DELAY_VALUE | String | "0" through "12" | "0" | Specifies the amount of additional delay to add to the non-registered path out of the IOB. |
| IFD_DELAY_VALUE | String | "AUTO", "0" through "6" | "AUTO" | Specifies the amount of additional delay to add to the registered path within the IOB. |
| IOSTANDARD | String | See Data Sheet. | "DEFAULT" | Assigns an I/O standard to the element. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- IBUF: Single-ended Input Buffer
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

IBUF_inst : IBUF
generic map (
    IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer,
                           -- "0"-12" (Spartan-3E)
    IFD_DELAY_VALUE => "AUTO", -- Specify the amount of added delay for input register,
                           -- "AUTO", "0"-6"
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Buffer output
    I => I       -- Buffer input (connect directly to top-level port)
);

-- End of IBUF_inst instantiation
```

Verilog Instantiation Template

```
// IBUF: Single-ended Input Buffer
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

IBUF #(
    .IBUF_DELAY_VALUE("0"), // Specify the amount of added input delay for
                           // the buffer: "0"-12" (Spartan-3E)
    .IFD_DELAY_VALUE("AUTO"), // Specify the amount of added delay for input
                           // register: "AUTO", "0"-6" (Spartan-3E)
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
)IBUF_inst (
    .O(O), // Buffer output
    .I(I) // Buffer input (connect directly to top-level port)
);

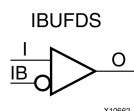
// End of IBUF_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

IBUFDS

Primitive: Differential Signaling Input Buffer



Introduction

This design element is an input buffer that supports low-voltage, differential signaling. In IBUFDS, a design level interface signal is represented as two distinct ports (I and IB), one deemed the "master" and the other the "slave." The master and the slave are opposite phases of the same logical signal (for example, MYNET_P and MYNET_N). Optionally, a programmable differential termination feature is available to help improve signal integrity and reduce external components.

Logic Table

| Inputs | | Outputs |
|--------|----|-----------|
| I | IB | O |
| 0 | 0 | No Change |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | No Change |

Port Descriptions

| Port | Type | Width | Function |
|------|--------|-------|---------------------|
| I | Input | 1 | Diff_p Buffer Input |
| IB | Input | 1 | Diff_n Buffer Input |
| O | Output | 1 | Buffer Output |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Recommended |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

Put all I/O components on the top-level of the design to help facilitate hierarchical design methods. Connect the I port directly to the top-level "master" input port of the design, the IB port to the top-level "slave" input port, and the O port to the logic in which this input is to source. Specify the desired generic/defparam values in order to configure the proper behavior of the buffer.

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|------------------|---------|-------------------------|-----------|--|
| DIFF_TERM | Boolean | TRUE or FALSE | FALSE | Enables the built-in differential termination resistor. |
| IBUF_DELAY_VALUE | String | "0" through "12" | "0" | Specifies the amount of additional delay to add to the non-registered path out of the IOB. |
| IFD_DELAY_VALUE | String | "AUTO", "0" thru "6" | "AUTO" | Specifies the amount of additional delay to add to the registered path within the IOB. |
| IOSTANDARD | String | See Data Sheet. | "DEFAULT" | Assigns an I/O standard to the element. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFDS: Differential Input Buffer
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

IBUFDS_inst : IBUFDS
generic map (
    DIFF_TERM => FALSE, -- Differential Termination
    IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer,
                        -- "0"-"12"
    IFD_DELAY_VALUE => "AUTO", -- Specify the amount of added delay for input register,
                        -- "AUTO", "0"-"6"
    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 Instantiation Template

```

// IBUFDS: Differential Input Buffer
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

IBUFDS #(
    .IBUF_DELAY_VALUE("0"), // Specify the amount of added input delay for
                        // the buffer: "0"-"12" (Spartan-3E)
    .IFD_DELAY_VALUE("AUTO"), // Specify the amount of added delay for input
                        // register: "AUTO", "0"-"6" (Spartan-3E)
    .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

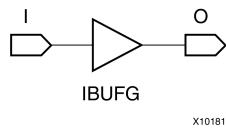
```

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

IBUFG

Primitive: Dedicated Input Clock Buffer



Introduction

The IBUFG is a dedicated input to the device which should be used to connect incoming clocks to the FPGA's global clock routing resources. The IBUFG provides dedicated connections to the DCM_SP and BUFG providing the minimum amount of clock delay and jitter to the device. The IBUFG input can only be driven by the global clock pins. The IBUFG output can drive CLKIN of a DCM_SP, BUFG, or your choice of logic.

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|---------------------|
| O | Output | 1 | Clock Buffer output |
| I | Input | 1 | Clock Buffer input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|------------------|--------|------------------|-----------|---|
| IBUF_DELAY_VALUE | String | "0" through "12" | "0" | Specifies the amount of additional delay to add to the non-registered path out of the IOB |
| IOSTANDARD | String | See Data Sheet | "DEFAULT" | Assigns an I/O standard to the element. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- IBUFG: Single-ended global clock input buffer
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

IBUFG_inst : IBUFG
generic map (
    IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer,
                             -- "0"-"12"
    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 Instantiation Template

```
// IBUFG: Single-ended global clock input buffer
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

IBUFG #(
    .IBUF_DELAY_VALUE("0"), // Specify the amount of added input delay for
                             // the buffer: "0"-"12" (Spartan-3E)
    .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)
);

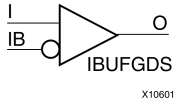
// End of IBUFG_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

IBUFGDS

Primitive: Differential Signaling Dedicated Input Clock Buffer and Optional Delay



Introduction

This design element is a dedicated differential signaling input buffer for connection to the clock buffer (BUFG) or . In IBUFGDS, a design-level interface signal is represented as two distinct ports (I and IB), one deemed the "master" and the other the "slave." The master and the slave are opposite phases of the same logical signal (for example, MYNET_P and MYNET_N). Optionally, a programmable differential termination feature is available to help improve signal integrity and reduce external components. Also available is a programmable delay is to assist in the capturing of incoming data to the device.

Logic Table

| Inputs | | Outputs |
|--------|----|-----------|
| I | IB | O |
| 0 | 0 | No Change |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | No Change |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|---------------------------|
| O | Output | 1 | Clock Buffer output |
| IB | Input | 1 | Diff_n Clock Buffer Input |
| I | Input | 1 | Diff_p Clock Buffer Input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Recommended |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

Put all I/O components on the top-level of the design to help facilitate hierarchical design methods. Connect the I port directly to the top-level "master" input port of the design, the IB port to the top-level "slave" input port and the O port to , BUFG or logic in which this input is to source. Some synthesis tools infer the BUFG automatically if necessary, when connecting an IBUFG to the clock resources of the FPGA. Specify the desired generic/defparam values in order to configure the proper behavior of the buffer.

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|------------------|---------|------------------|-----------|---|
| DIFF_TERM | Boolean | TRUE or FALSE | FALSE | Enables the built-in differential termination resistor. |
| IBUF_DELAY_VALUE | String | "0" through "12" | "0" | Specifies the amount of additional delay to add to the non-registered path out of the IOB |
| IOSTANDARD | String | See Data Sheet | "DEFAULT" | Assigns an I/O standard to the element. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- IBUFGDS: Differential Global Clock Input Buffer
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

IBUFGDS_inst : IBUFGDS
generic map (
    DIFF_TERM => FALSE, -- Differential Termination (Spartan-3E/5 only)
    IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer,
                           -- "0"-"12"
    IOSTANDARD => "DEFAULT")
port map (
    O => O, -- Clock buffer output
    I => I, -- Diff_p clock buffer input (connect directly to top-level port)
    IB => IB -- Diff_n clock buffer input (connect directly to top-level port)
);

-- End of IBUFGDS_inst instantiation
```

Verilog Instantiation Template

```
// IBUFGDS: Differential Global Clock Input Buffer
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

IBUFGDS #(
    .IBUF_DELAY_VALUE("0"), // Specify the amount of added input delay for
                           // the buffer, "0"-"12" (Spartan-3E)
    .DIFF_TERM("FALSE"), // Differential Termination
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFGDS_inst (
    .O(O), // Clock buffer output
    .I(I), // Diff_p clock buffer input (connect directly to top-level port)
    .IB(IB) // Diff_n clock buffer input (connect directly to top-level port)
);

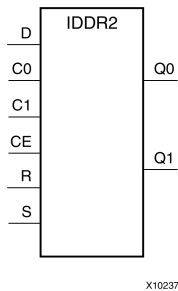
// End of IBUFGDS_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

IDDR2

Primitive: Double Data Rate Input D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset



Introduction

This design element is a dedicated input register designed to receive external dual data rate (DDR) signals into Xilinx® FPGAs. The IDDR2 requires two clocks to be connected to the component, C0 and C1, so that data is captured at the positive edge of both C0 and C1 clocks. The IDDR2 features an active high clock enable port, CE, which be used to suspend the operation of the registers, and both set and reset ports that be configured to be synchronous or asynchronous to the respective clocks. The IDDR2 has an optional alignment feature that allows both output data ports to the component to be aligned to a single clock.

Logic Table

| Input | | | | | | Output | |
|-------|---|----|---|----|----|-------------|-------------|
| S | R | CE | D | C0 | C1 | Q0 | Q1 |
| 1 | x | x | x | x | x | INIT_Q0 | INIT_Q1 |
| 0 | 1 | x | x | x | x | not INIT_Q0 | not INIT_Q1 |
| 0 | 0 | 0 | x | x | x | No Change | No Change |
| 0 | 0 | 1 | D | ↑ | x | D | No Change |
| 0 | 0 | 1 | D | x | ↑ | No Change | D |

Set/Reset can be synchronous via SRTYPE value

Design Entry Method

| Instantiation | Recommended |
|-----------------------------|-------------|
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

To change the default behavior of the IDDR2, modify attributes via the generic map (VHDL) or named parameter value assignment (Verilog) as a part of the instantiated component. The IDDR2 can be connected directly to a top-level input port in the design, where an appropriate input buffer can be inferred, or directly to an instantiated IBUF, IOBUF, IBUFDS or IOBUFDS. All inputs and outputs of this component should either be connected or properly tied off.

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|---------------|---------|------------------|---------|--|
| DDR_ALIGNMENT | String | NONE, "C0", "C1" | "NONE" | Sets the output alignment more for the DDR register <ul style="list-style-type: none"> NONE - Makes the data available on the Q0 and Q1 outputs shortly after the corresponding C0 or C1 positive clock edge. C0 – Makes the data on both Q0 and Q1 align to the positive edge of the C0 clock. C1 - Makes the data on both Q0 and Q1 align to the positive edge of the C1 clock. |
| INIT_Q0 | Integer | 0, 1 | 0 | Sets initial state of the Q0 output to 0 or 1. |
| INIT_Q1 | Integer | 0, 1 | 0 | Sets initial state of the Q1 output to 0 or 1. |
| SRTYPE | String | "SYNC", "ASYN" | "SYNC" | Specifies SYNC" or "ASYN" set/reset. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IDDR2: Input Double Data Rate Input Register with Set, Reset
--      and Clock Enable.
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

IDDR2_inst : IDDR2
generic map(
  DDR_ALIGNMENT => "NONE", -- Sets output alignment to "NONE", "C0", "C1"
  INIT_Q0 => '0', -- Sets initial state of the Q0 output to '0' or '1'
  INIT_Q1 => '0', -- Sets initial state of the Q1 output to '0' or '1'
  SRTYPE => "SYNC") -- Specifies "SYNC" or "ASYN" set/reset
port map (
  Q0 => Q0, -- 1-bit output captured with C0 clock
  Q1 => Q1, -- 1-bit output captured with C1 clock
  C0 => C0, -- 1-bit clock input
  C1 => C1, -- 1-bit clock input
  CE => CE, -- 1-bit clock enable input
  D => D,   -- 1-bit data input
  R => R,   -- 1-bit reset input
  S => S    -- 1-bit set input
);

-- End of IDDR2_inst instantiation

```

Verilog Instantiation Template

```

// IDDR2: Input Double Data Rate Input Register with Set, Reset
//      and Clock Enable.
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

IDDR2 #(
  .DDR_ALIGNMENT("NONE"), // Sets output alignment to "NONE", "C0" or "C1"
  .INIT_Q0(1'b0), // Sets initial state of the Q0 output to 1'b0 or 1'b1
  .INIT_Q1(1'b0), // Sets initial state of the Q1 output to 1'b0 or 1'b1
  .SRTYPE("SYNC") // Specifies "SYNC" or "ASYN" set/reset
) IDDR2_inst (
  .Q0(Q0), // 1-bit output captured with C0 clock
  .Q1(Q1), // 1-bit output captured with C1 clock
  .C0(C0), // 1-bit clock input

```

```
.Cl(C1), // 1-bit clock input
.CE(CE), // 1-bit clock enable input
.D(D),   // 1-bit DDR data input
.R(R),   // 1-bit reset input
.S(S)    // 1-bit set input
);

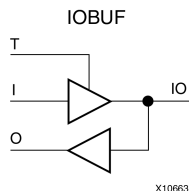
// End of IDDR2_inst instantiation
```

For More Information

See the [Spartan-3 Generation FPGA User Guide](#).

IOBUF

Primitive: Bi-Directional Buffer



Introduction

The design element is a bidirectional single-ended I/O Buffer used to connect internal logic to an external bidirectional pin.

Logic Table

| Inputs | | Bidirectional | Outputs |
|--------|---|---------------|---------|
| T | I | IO | O |
| 1 | X | Z | IO |
| 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|----------------------|
| O | Output | 1 | Buffer output |
| IO | Inout | 1 | Buffer inout |
| I | Input | 1 | Buffer input |
| T | Input | 1 | 3-State enable input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|------------------|---------|------------------------------|-----------|--|
| DRIVE | Integer | 2, 4, 6, 8, 12, 16, 24 | 12 | Selects output drive strength (mA) for the SelectIO™ buffers that use the LVTTTL, LVC MOS12, LVC MOS15, LVC MOS18, LVC MOS25, or LVC MOS33 interface I/O standard. |
| IBUF_DELAY_VALUE | String | "0" through "12" | "0" | Specifies the amount of additional delay to add to the non-registered path out of the IOB |
| IFD_DELAY_VALUE | String | "AUTO", "0" through "6" | "AUTO" | Specifies the amount of additional delay to add to the registered path within the IOB |
| IOSTANDARD | String | See Data Sheet | "DEFAULT" | Assigns an I/O standard to the element. |
| SLEW | String | "SLOW", "FAST", "QUIETIO" | "SLOW" | Sets the output rise and fall time. See the Data Sheet for recommendations of the best setting for this attribute. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IOBUF: Single-ended Bi-directional Buffer
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

IOBUF_inst : IOBUF
generic map (
  DRIVE => 12,
  IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer,
                        -- "0"-"12"
  IFD_DELAY_VALUE => "AUTO", -- Specify the amount of added delay for input register,
                        -- "AUTO", "0"-"6"
  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 Instantiation Template

```
// IOBUF: Single-ended Bi-directional Buffer
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

IOBUF #(
    .DRIVE(12), // Specify the output drive strength
    .IBUF_DELAY_VALUE("0"), // Specify the amount of added input delay for the buffer,
                          // "0"-12" (Spartan-3E only)
    .IFD_DELAY_VALUE("AUTO"), // Specify the amount of added delay for input register,
                          // "AUTO", "0"-6" (Spartan-3E only)
    .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
);

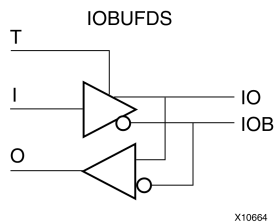
// End of IOBUF_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

IOBUFDS

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



Introduction

The design element is a bidirectional buffer that supports low-voltage, differential signaling. For the IOBUFDS, a design level interface signal is represented as two distinct ports (IO and IOB), one deemed the "master" and the other the "slave." The master and the slave are opposite phases of the same logical signal (for example, MYNET_P and MYNET_N). Optionally, a programmable differential termination feature is available to help improve signal integrity and reduce external components. Also available is a programmable delay to assist in the capturing of incoming data to the device.

Logic Table

| Inputs | | Bidirectional | | Outputs |
|--------|---|---------------|-----|-----------|
| I | T | IO | IOB | O |
| X | 1 | Z | Z | No Change |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|----------------------|
| O | Output | 1 | Buffer output |
| IO | Inout | 1 | Diff_p inout |
| IOB | Inout | 1 | Diff_n inout |
| I | Input | 1 | Buffer input |
| T | Input | 1 | 3-state enable input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Recommended |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|------------------|--------|-------------------------|-----------|---|
| IBUF_DELAY_VALUE | String | "0" through "12" | "0" | Specifies the amount of additional delay to add to the non-registered path out of the IOB |
| IFD_DELAY_VALUE | String | "AUTO", "0" through "6" | "AUTO" | Specifies the amount of additional delay to add to the registered path within the IOB |
| IOSTANDARD | String | See Data Sheet | "DEFAULT" | Assigns an I/O standard to the element. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IOBUFDS: Differential Bi-directional Buffer
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

IOBUFDS_inst : IOBUFDS
generic map (
    IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer,
                           -- "0"-"12"
    IFD_DELAY_VALUE => "AUTO", -- Specify the amount of added delay for input register,
                           -- "AUTO", "0"-"6"
    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 Instantiation Template

```

// IOBUFDS: Differential Bi-directional Buffer
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

IOBUFDS #(
    .IBUF_DELAY_VALUE("0"), // Specify the amount of added input delay for the buffer,
                           // "0"-"12" (Spartan-3E only)
    .IFD_DELAY_VALUE("AUTO"), // Specify the amount of added delay for input register,
                           // "AUTO", "0"-"6" (Spartan-3E only)
    .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

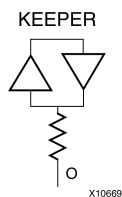
```

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

KEEPER

Primitive: KEEPER Symbol



Introduction

The design element is a weak keeper element that retains the value of the net connected to its bidirectional O pin. For example, if a logic 1 is being driven onto the net, KEEPER drives a weak/resistive 1 onto the net. If the net driver is then 3-stated, KEEPER continues to drive a weak/resistive 1 onto the net.

Port Descriptions

| Name | Direction | Width | Function |
|------|-----------|-------|---------------|
| O | Output | 1-Bit | Keeper output |

Design Entry Method

| | |
|-----------------------------|-----|
| Instantiation | Yes |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- KEEPER: I/O Buffer Weak Keeper
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of KEEPER_inst instantiation
```

Verilog Instantiation Template

```
// KEEPER: I/O Buffer Weak Keeper
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

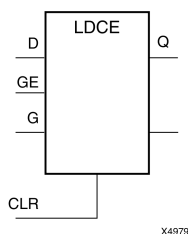
// End of KEEPER_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LDCE

Primitive: Transparent Data Latch with Asynchronous Clear and Gate Enable



Introduction

This design element is a transparent data latch with asynchronous clear and gate enable. When the asynchronous clear input (CLR) is High, it overrides the other inputs and resets the data (Q) output Low. Q reflects the data (D) input while the gate (G) input and gate enable (GE) are High and CLR is Low. If (GE) is Low, data on (D) cannot be latched. The data on the (D) input during the High-to-Low gate transition is stored in the latch. The data on the (Q) output remains unchanged as long as (G) or (GE) remains low.

This latch is asynchronously cleared, outputs Low, when power is applied. For FPGA devices, power-on conditions are simulated when global set/reset (GSR) is active. GSR defaults to active-High but can be inverted by adding an inverter in front of the GSR input of the appropriate *STARTUP_architecture* symbol.

Logic Table

| Inputs | | | | Outputs |
|--------|----|---|---|-----------|
| CLR | GE | G | D | Q |
| 1 | X | X | X | 0 |
| 0 | 0 | X | X | No Change |
| 0 | 1 | 1 | D | D |
| 0 | 1 | 0 | X | No Change |
| 0 | 1 | ↓ | D | D |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|--------|----------------|---------|---|
| INIT | Binary | 0, 1 | 0 | Sets the initial value of Q output after configuration. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- LDCE: Transparent latch with Asynchronous Reset and
--       Gate Enable.
--       Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LDCE: Transparent latch with Asynchronous Reset and Gate Enable.
//       Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

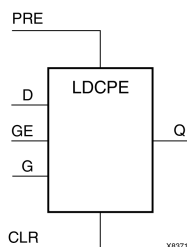
// End of LDCE_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LDCPE

Primitive: Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable



Introduction

This design element is a transparent data latch with data (D), asynchronous clear (CLR), asynchronous preset (PRE), and gate enable (GE). When (CLR) is High, it overrides the other inputs and resets the data (Q) output Low. When (PRE) is High and (CLR) is Low, it presets the data (Q) output High. Q reflects the data (D) input while the gate (G) input and gate enable (GE) are High and (CLR) and PRE are Low. The data on the (D) input during the High-to-Low gate transition is stored in the latch. The data on the Q output remains unchanged as long as (G) or (GE) remains Low.

This latch is asynchronously cleared, outputs Low, when power is applied. For FPGA devices, power-on conditions are simulated when global set/reset (GSR) is active. GSR defaults to active-High but can be inverted by adding an inverter in front of the GSR input of the appropriate `STARTUP_architecture` symbol.

Logic Table

| Inputs | | | | | Outputs |
|--------|-----|----|---|---|-----------|
| CLR | PRE | GE | G | D | Q |
| 1 | X | X | X | X | 0 |
| 0 | 1 | X | X | X | 1 |
| 0 | 0 | 0 | X | X | No Change |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | X | No Change |
| 0 | 0 | 1 | ↓ | D | D |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--------------------------------|
| Q | Output | 1 | Data Output |
| CLR | Input | 1 | Asynchronous clear/reset input |
| D | Input | 1 | Data Input |
| G | Input | 1 | Gate Input |
| GE | Input | 1 | Gate Enable Input |
| PRE | Input | 1 | Asynchronous preset/set input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|---------|----------------|---------|---|
| INIT | Integer | 0, 1 | 0 | Sets the initial value of Q output after configuration. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- LDCPE: Transparent latch with Asynchronous Reset, Preset and
--      Gate Enable.
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LDCPE: Transparent latch with Asynchronous Reset, Preset and
//      Gate Enable.
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

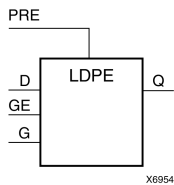
// End of LDCPE_inst instantiation
```

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

LDPE

Primitive: Transparent Data Latch with Asynchronous Preset and Gate Enable



Introduction

This design element is a transparent data latch with asynchronous preset and gate enable. When the asynchronous preset (PRE) is High, it overrides the other input and presets the data (Q) output High. Q reflects the data (D) input while the gate (G) input and gate enable (GE) are High. The data on the (D) input during the High-to-Low gate transition is stored in the latch. The data on the (Q) output remains unchanged as long as (G) or (GE) remains Low.

The latch is asynchronously preset, output High, when power is applied. For FPGA devices, power-on conditions are simulated when global set/reset (GSR) is active. GSR defaults to active-High but can be inverted by adding an inverter in front of the GSR input of the appropriate *STARTUP_architecture* symbol.

Logic Table

| Inputs | | | | Outputs |
|--------|----|---|---|-----------|
| PRE | GE | G | D | Q |
| 1 | X | X | X | 1 |
| 0 | 0 | X | X | No Change |
| 0 | 1 | 1 | D | D |
| 0 | 1 | 0 | X | No Change |
| 0 | 1 | ↓ | D | D |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|--------|----------------|---------|---|
| INIT | Binary | 0, 1 | 1 | Specifies the initial value upon power-up or the assertion of GSR for the (Q) port. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- LDPE: Transparent latch with Asynchronous Set and
--       Gate Enable.
--       Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LDPE: Transparent latch with Asynchronous Preset and Gate Enable.
//       Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

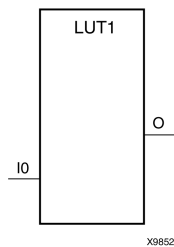
// End of LDPE_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT1

Primitive: 1-Bit Look-Up Table with General Output



Introduction

This design element is a 1-bit look-up table (LUT) with general output (O).

An INIT attribute with an appropriate number of hexadecimal digits for the number of inputs must be attached to the LUT to specify its function. This element provides a look-up table version of a buffer or inverter. These elements are the basic building blocks. Two LUTs are available in each CLB slice; four LUTs are available in each CLB. Multiple variants of LUTs accommodate additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | Outputs |
|---|----------|
| I0 | O |
| 0 | INIT[0] |
| 1 | INIT[1] |
| INIT = Binary number assigned to the INIT attribute | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-----------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 2-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- LUT1: 1-input Look-Up Table with general output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of LUT1_inst instantiation
```

Verilog Instantiation Template

```
// LUT1: 1-input Look-Up Table with general output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

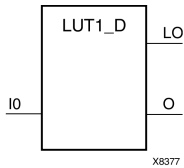
// End of LUT1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT1_D

Primitive: 1-Bit Look-Up Table with Dual Output



Introduction

This design element is a 1-bit look-up table (LUT) with two functionally identical outputs, O and LO. It provides a look-up table version of a buffer or inverter.

The O output is a general interconnect. The LO output is used to connect to another output within the same CLB slice and to the fast connect buffer. A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | Outputs | |
|---|---------|---------|
| I0 | O | LO |
| 0 | INIT[0] | INIT[0] |
| 1 | INIT[1] | INIT[1] |
| INIT = Binary number assigned to the INIT attribute | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-----------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 2-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- LUT1_D: 1-input Look-Up Table with general and local outputs
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LUT1_D: 1-input Look-Up Table with general and local outputs
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

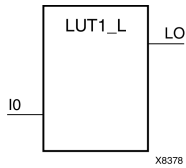
// End of LUT1_D_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT1_L

Primitive: 1-Bit Look-Up Table with Local Output



Introduction

This design element is a 1-bit look-up table (LUT) with a local output (LO) that is used to connect to another output within the same CLB slice and to the fast connect buffer. It provides a look-up table version of a buffer or inverter.

A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | Outputs |
|---|---------|
| IO | LO |
| 0 | INIT[0] |
| 1 | INIT[1] |
| INIT = Binary number assigned to the INIT attribute | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-----------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 2-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- LUT1_L: 1-input Look-Up Table with local output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LUT1_L: 1-input Look-Up Table with local output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

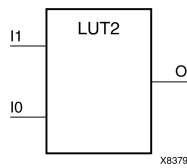
// End of LUT1_L_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT2

Primitive: 2-Bit Look-Up Table with General Output



Introduction

This design element is a 2-bit look-up table (LUT) with general output (O).

An INIT attribute with an appropriate number of hexadecimal digits for the number of inputs must be attached to the LUT to specify its function. This element provides a look-up table version of a buffer or inverter. These elements are the basic building blocks. Two LUTs are available in each CLB slice; four LUTs are available in each CLB. Multiple variants of LUTs accommodate additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | | Outputs |
|---|----|---------|
| I1 | I0 | O |
| 0 | 0 | INIT[0] |
| 0 | 1 | INIT[1] |
| 1 | 0 | INIT[2] |
| 1 | 1 | INIT[3] |
| INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-----------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 4-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT2: 2-input Look-Up Table with general output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```

// LUT2: 2-input Look-Up Table with general output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

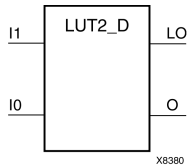
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT2_D

Primitive: 2-Bit Look-Up Table with Dual Output



Introduction

This design element is a 2-bit look-up table (LUT) with two functionally identical outputs, O and LO.

The O output is a general interconnect. The LO output is used to connect to another output within the same CLB slice and to the fast connect buffer. A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The LogicTable Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | | Outputs | |
|---|----|---------|---------|
| 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 = Binary equivalent of the hexadecimal number assigned to the INIT attribute | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-----------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 4-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- LUT2_D: 2-input Look-Up Table with general and local outputs
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LUT2_D: 2-input Look-Up Table with general and local outputs
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

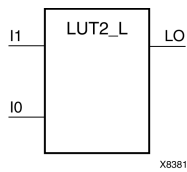
// End of LUT2_D_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT2_L

Primitive: 2-Bit Look-Up Table with Local Output



Introduction

This design element is a 2-bit look-up table (LUT) with a local output (LO) that is used to connect to another output within the same CLB slice and to the fast connect buffer. It provides a look-up table version of a buffer or inverter.

A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | | Outputs |
|---|----|---------|
| I1 | I0 | LO |
| 0 | 0 | INIT[0] |
| 0 | 1 | INIT[1] |
| 1 | 0 | INIT[2] |
| 1 | 1 | INIT[3] |
| INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-----------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 4-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT2_L: 2-input Look-Up Table with local output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

LUT2_L_inst : LUT2_L
generic map (
    INIT => X"0"
)
port map (
    LO => LO, -- LUT local output
    I0 => I0, -- LUT input
    I1 => I1 -- LUT input
);

-- End of LUT2_L_inst instantiation

```

Verilog Instantiation Template

```

// LUT2_L: 2-input Look-Up Table with local output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

LUT2_L #(
    .INIT(4'h0) // Specify LUT Contents
) LUT2_L_inst (
    .LO(LO), // LUT local output
    .I0(I0), // LUT input
    .I1(I1) // LUT input
);

// End of LUT2_L_inst instantiation

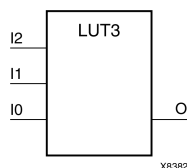
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT3

Primitive: 3-Bit Look-Up Table with General Output



Introduction

This design element is a 3-bit look-up table (LUT) with general output (O). A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

An INIT attribute with an appropriate number of hexadecimal digits for the number of inputs must be attached to the LUT to specify its function. This element provides a look-up table version of a buffer or inverter. These elements are the basic building blocks. Two LUTs are available in each CLB slice; four LUTs are available in each CLB. Multiple variants of LUTs accommodate additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | | | Outputs |
|---|----|----|---------|
| 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 = Binary equivalent of the hexadecimal number assigned to the INIT attribute | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-----------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 8-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- LUT3: 3-input Look-Up Table with general output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LUT3: 3-input Look-Up Table with general output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

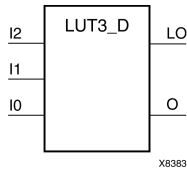
// End of LUT3_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT3_D

Primitive: 3-Bit Look-Up Table with Dual Output



Introduction

This design element is a 3-bit look-up table (LUT) with two functionally identical outputs, O and LO.

The O output is a general interconnect. The LO output is used to connect to another output within the same CLB slice and to the fast connect buffer. A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | | | Outputs | |
|---|----|----|---------|---------|
| 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 = Binary equivalent of the hexadecimal number assigned to the INIT attribute | | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-----------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 8-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- LUT3_D: 3-input Look-Up Table with general and local outputs
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LUT3_D: 3-input Look-Up Table with general and local outputs
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

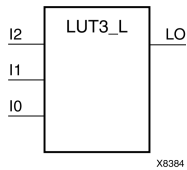
// End of LUT3_D_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT3_L

Primitive: 3-Bit Look-Up Table with Local Output



Introduction

This design element is a 3-bit look-up table (LUT) with a local output (LO) that is used to connect to another output within the same CLB slice and to the fast connect buffer. It provides a look-up table version of a buffer or inverter.

A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | | | Outputs |
|---|----|----|---------|
| 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 = Binary equivalent of the hexadecimal number assigned to the INIT attribute | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-----------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 8-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- LUT3_L: 3-input Look-Up Table with local output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of LUT3_L_inst instantiation
```

Verilog Instantiation Template

```
// LUT3_L: 3-input Look-Up Table with local output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

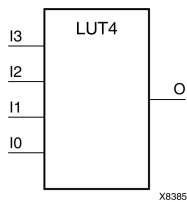
// End of LUT3_L_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT4

Primitive: 4-Bit Look-Up-Table with General Output



Introduction

This design element is a 4-bit look-up table (LUT) with general output (O).

An INIT attribute with an appropriate number of hexadecimal digits for the number of inputs must be attached to the LUT to specify its function. This element provides a look-up table version of a buffer or inverter. These elements are the basic building blocks. Two LUTs are available in each CLB slice; four LUTs are available in each CLB. Multiple variants of LUTs accommodate additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | | | | Outputs |
|--------|----|----|----|----------|
| 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] |

| Inputs | | | | Outputs |
|---|----|----|----|----------|
| I3 | I2 | I1 | I0 | O |
| 1 | 1 | 0 | 0 | INIT[12] |
| 1 | 1 | 0 | 1 | INIT[13] |
| 1 | 1 | 1 | 0 | INIT[14] |
| 1 | 1 | 1 | 1 | INIT[15] |
| INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute | | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4: 4-input Look-Up Table with general output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LUT4: 4-input Look-Up Table with general output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

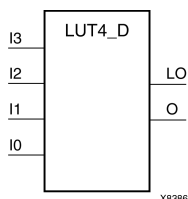
// End of LUT4_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT4_D

Primitive: 4-Bit Look-Up Table with Dual Output



Introduction

This design element is a 4-bit look-up table (LUT) with two functionally identical outputs, O and LO

The O output is a general interconnect. The LO output is used to connect to another output within the same CLB slice and to the fast connect buffer. A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | | | | Outputs | |
|--------|----|----|----|----------|----------|
| 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] |

| Inputs | | | | Outputs | |
|---|----|----|----|----------|----------|
| 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 = Binary equivalent of the hexadecimal number assigned to the INIT attribute | | | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_D: 4-input Look-Up Table with general and local outputs
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LUT4_D: 4-input Look-Up Table with general and local outputs
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

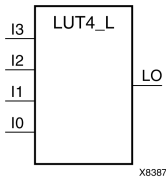
// End of LUT4_D_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

LUT4_L

Primitive: 4-Bit Look-Up Table with Local Output



Introduction

This design element is a 4-bit look-up table (LUT) with a local output (LO) that is used to connect to another output within the same CLB slice and to the fast connect buffer. It provides a look-up table version of a buffer or inverter.

A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

The Equation Method -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

| Inputs | | | | Outputs |
|--------|----|----|----|----------|
| 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] |

| Inputs | | | | Outputs |
|---|----|----|----|----------|
| I3 | I2 | I1 | I0 | LO |
| 1 | 1 | 0 | 1 | INIT[13] |
| 1 | 1 | 1 | 0 | INIT[14] |
| 1 | 1 | 1 | 1 | INIT[15] |
| INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute | | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|-----------------------------|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Initializes look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_L: 4-input Look-Up Table with local output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// LUT4_L: 4-input Look-Up Table with local output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

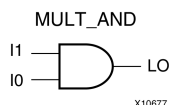
// End of LUT4_L_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MULT_AND

Primitive: Fast Multiplier AND



Introduction

The design element is an AND component located within the slice where the two inputs are shared with the 4-input LUT and the output drives into the carry logic. This added logic is especially useful for building fast and smaller multipliers. However, it can be used for other purposes as well. The I1 and I0 inputs must be connected to the I1 and I0 inputs of the associated LUT. The LO output must be connected to the DI input of the associated MUXCY, MUXCY_D, or MUXCY_L.

Logic Table

| Inputs | | Outputs |
|--------|----|---------|
| I1 | I0 | LO |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MULT_AND: 2-input AND gate connected to Carry chain
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// MULT_AND: 2-input AND gate connected to Carry chain
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

MULT_AND MULT_AND_inst (
    .LO(LO),    // MULT_AND output (connect to MUXCY DI)
    .IO(I0),    // MULT_AND data[0] input
    .I1(I1)    // MULT_AND data[1] input
);

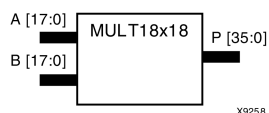
// End of MULT_AND_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MULT18X18

Primitive: 18 x 18 Signed Multiplier



Introduction

MULT18X18 is a combinational signed 18-bit by 18-bit multiplier. The value represented in the 18-bit input A is multiplied by the value represented in the 18-bit input B. Output P is the 36-bit product of A and B.

Logic Table

| Inputs | | Output |
|-----------------------------------|---|--------|
| A | B | P |
| A | B | A x B |
| A, B, and P are two's complement. | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- MULT18X18: 18 x 18 signed asynchronous multiplier
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// MULT18X18: 18 x 18 signed asynchronous multiplier
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

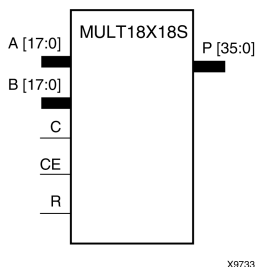
// End of MULT18X18_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MULT18X18S

Primitive: 18 x 18 Signed Multiplier Registered Version



Introduction

MULT18X18S is the registered version of the 18 x 18 signed multiplier with output P and inputs A, B, C, CE, and R. The registers are initialized to 0 after the GSR pulse.

The value represented in the 18-bit input A is multiplied by the value represented in the 18-bit input B. Output P is the 36-bit product of A and B.

Logic Table

| Inputs | | | | | Output |
|--------|----|----|----|---|-----------|
| C | CE | Am | Bn | R | P |
| ↑ | X | X | X | 1 | 0 |
| ↑ | 1 | Am | Bn | 0 | A x B |
| X | 0 | X | X | 0 | No Change |

A, B, and P are two's complement.

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- MULT18X18S: 18 x 18 signed synchronous multiplier
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// MULT18X18S: 18 x 18 signed synchronous multiplier
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

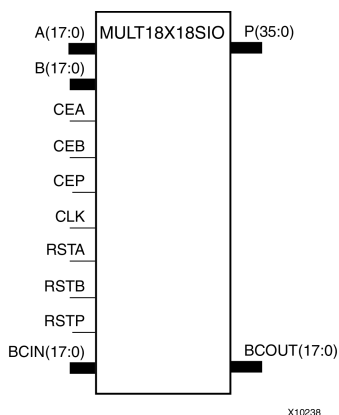
// End of MULT18X18S_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MULT18X18SIO

Primitive: 18 x 18 Cascadable Signed Multiplier with Optional Input and Output Registers, Clock Enable, and Synchronous Reset



Introduction

This design element is a 36-bit output, 18x18-bit input dedicated signed multiplier. This component can perform asynchronous multiplication operations when the attributes AREG, BREG and PREG are all set to 0. Alternatively, synchronous multiplication operations of different latency and performance characteristics can be performed when any combination of those attributes is set to 1. When using the multiplier in synchronous operation, the MULT18X18SIO features active high clock enables for each set of register banks in the multiplier, CEA, CEB and CEP, as well as synchronous resets, RSTA, RSTB, and RSTP. Multiple MULT18X18SIOs can be cascaded to create larger multiplication functions using the BCIN and BCOUT ports in combination with the B_INPUT attribute.

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Descriptions |
|-----------|---------|-----------------------|----------|---|
| AREG | Integer | 0, 1 | 1 | Specifies the use of the input registers on the A port. A zero disables the use of the register; a one enables the register. |
| BREG | Integer | 0, 1 | 1 | Specifies the use of the input registers on the B port. A zero disables the use of the register; a one enables the register. |
| B_INPUT | String | "DIRECT" or "CASCADE" | "DIRECT" | Specifies whether the B port is connected to the general FPGA fabric, "DIRECT" or is connected to the BCOUT port of another MULT18X18SIO. |
| PREG | Integer | 0, 1 | 1 | Specifies the use of the output registers of the multiplier. A zero disables the use of the register; a one enables the register. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MULT18X18SIO: 18 x 18 cascadable, signed synchronous/asynchronous multiplier
--                Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

MULT18X18SIO_inst : MULT18X18SIO
generic map (
    AREG => 1, -- Enable the input registers on the A port (1=on, 0=off)
    BREG => 1, -- Enable the input registers on the B port (1=on, 0=off)
    B_INPUT => "DIRECT", -- B cascade input "DIRECT" or "CASCADE"
    PREG => 1) -- Enable the input registers on the P port (1=on, 0=off)
port map (
    BCOUT => BCOUT, -- 18-bit cascade output
    P => P,         -- 36-bit multiplier output
    A => A,         -- 18-bit multiplier input
    B => B,         -- 18-bit multiplier input
    BCIN => BCIN, -- 18-bit cascade input
    CEA => CEA, -- Clock enable input for the A port
    CEB => CEB, -- Clock enable input for the B port
    CEP => CEP, -- Clock enable input for the P port
    CLK => CLK,  -- Clock input
    RSTA => RSTA, -- Synchronous reset input for the A port
    RSTB => RSTB, -- Synchronous reset input for the B port
    RSTP => RSTP, -- Synchronous reset input for the P port
);

-- End of MULT18X18SIO_inst instantiation
```

Verilog Instantiation Template

```
// MULT18X18SIO: 18 x 18 cascadable, signed synchronous/asynchronous multiplier
//                Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

MULT18X18SIO #(
    .AREG(1), // Enable the input registers on the A port (1=on, 0=off)
    .BREG(1), // Enable the input registers on the B port (1=on, 0=off)
    .B_INPUT("DIRECT"), // B cascade input "DIRECT" or "CASCADE"
    .PREG(1) // Enable the input registers on the P port (1=on, 0=off)
) MULT18X18SIO_inst (
    .BCOUT(BCOUT), // 18-bit cascade output
    .P(P),         // 36-bit multiplier output
    .A(A),         // 18-bit multiplier input
    .B(B),         // 18-bit multiplier input
    .BCIN(BCIN), // 18-bit cascade input
    .CEA(CEA), // Clock enable input for the A port
    .CEB(CEB), // Clock enable input for the B port
    .CEP(CEP), // Clock enable input for the P port
    .CLK(CLK), // Clock input
    .RSTA(RSTA), // Synchronous reset input for the A port
    .RSTB(RSTB), // Synchronous reset input for the B port
    .RSTP(RSTP) // Synchronous reset input for the P port
);

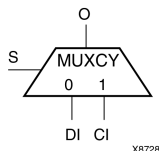
// End of MULT18X18SIO_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXCY

Primitive: 2-to-1 Multiplexer for Carry Logic with General Output



Introduction

The direct input (DI) of a slice is connected to the (DI) input of the MUXCY. The carry in (CI) input of an LC is connected to the CI input of the MUXCY. The select input (S) of the MUXCY is driven by the output of the look-up table (LUT) and configured as a MUX function. The carry out (O) of the MUXCY reflects the state of the selected input and implements the carry out function of each LC. When Low, S selects DI; when High, S selects CI.

The variants “MUXCY_D” and “MUXCY_L” provide additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

Logic Table

| Inputs | | | Outputs |
|--------|----|----|---------|
| S | DI | CI | O |
| 0 | 1 | X | 1 |
| 0 | 0 | X | 0 |
| 1 | X | 1 | 1 |
| 1 | X | 0 | 0 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MUXCY: Carry-Chain MUX with general output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// MUXCY: Carry-Chain MUX with general output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

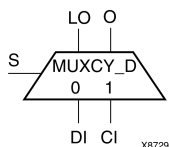
// End of MUXCY_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXCY_D

Primitive: 2-to-1 Multiplexer for Carry Logic with Dual Output



Introduction

This design element implements a 1-bit, high-speed carry propagate function. One such function can be implemented per logic cell (LC), for a total of 4-bits per configurable logic block (CLB). The direct input (DI) of an LC is connected to the DI input of the MUXCY_D. The carry in (CI) input of an LC is connected to the CI input of the MUXCY_D. The select input (S) of the MUX is driven by the output of the look-up table (LUT) and configured as an XOR function. The carry out (O and LO) of the MUXCY_D reflects the state of the selected input and implements the carry out function of each LC. When Low, S selects DI; when High, S selects CI.

Outputs O and LO are functionally identical. The O output is a general interconnect. See also “MUXCY” and “MUXCY_L”.

Logic Table

| Inputs | | | Outputs | |
|--------|----|----|---------|----|
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXCY_D: Carry-Chain MUX with general and local outputs
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// MUXCY_D: Carry-Chain MUX with general and local outputs
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

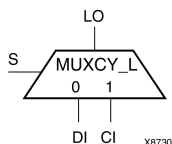
// End of MUXCY_D_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXCY_L

Primitive: 2-to-1 Multiplexer for Carry Logic with Local Output



Introduction

This design element implements a 1-bit high-speed carry propagate function. One such function is implemented per logic cell (LC), for a total of 4-bits per configurable logic block (CLB). The direct input (DI) of an LC is connected to the DI input of the MUXCY_L. The carry in (CI) input of an LC is connected to the CI input of the MUXCY_L. The select input (S) of the MUXCY_L is driven by the output of the look-up table (LUT) and configured as an XOR function. The carry out (LO) of the MUXCY_L reflects the state of the selected input and implements the carry out function of each (LC). When Low, (S) selects DI; when High, (S) selects (CI).

See also “MUXCY” and “MUXCY_D.”

Logic Table

| Inputs | | | Outputs |
|--------|----|----|---------|
| S | DI | CI | LO |
| 0 | 1 | X | 1 |
| 0 | 0 | X | 0 |
| 1 | X | 1 | 1 |
| 1 | X | 0 | 0 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

```
-- MUXCY_L: Carry-Chain MUX with local output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2
```

```
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 Instantiation Template

```
// MUXCY_L: Carry-Chain MUX with local output
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

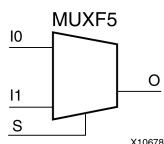
// End of MUXCY_L_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF5

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



Introduction

This design element provides a multiplexer function in a CLB slice for creating a function-of-5 look-up table or a 4-to-1 multiplexer in combination with the associated look-up tables. The local outputs (LO) from the two look-up tables are connected to the I0 and I1 inputs of the MUXF5. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The variants, “MUXF5_D” and “MUXF5_L”, provide additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

Logic Table

| Inputs | | | Outputs |
|--------|----|----|---------|
| S | I0 | I1 | O |
| 0 | 1 | X | 1 |
| 0 | 0 | X | 0 |
| 1 | X | 1 | 1 |
| 1 | X | 0 | 0 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF5: Slice MUX to tie two LUT4's together with general output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// MUXF5: Slice MUX to tie two LUT4's together with general output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

MUXF5 MUXF5_inst (
    .O(0),    // 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
);

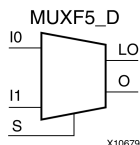
// End of MUXF5_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF5_D

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



Introduction

This design element provides a multiplexer function in a CLB slice for creating a function-of-5 look-up table or a 4-to-1 multiplexer in combination with the associated look-up tables. The local outputs (LO) from the two look-up tables are connected to the I0 and I1 inputs of the MUXF5. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

Outputs O and LO are functionally identical. The O output is a general interconnect. The LO output connects to other inputs in the same CLB slice. See also “MUXF5” and “MUXF5_L”.

Logic Table

| Inputs | | | Outputs | |
|--------|----|----|---------|----|
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

```
-- MUXF5_D: Slice MUX to tie two LUT4's together with general and local outputs
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2
```

```
MUXF5_D_inst : MUXF5_D
port map (
  LO => LO, -- Ouput of MUX to local routing
  O  => O,  -- Output of MUX to general routing
  I0 => I0, -- 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 Instantiation Template

```
// MUXF5_D: Slice MUX to tie two LUT4's together with general and local outputs
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

MUXF5_D MUXF5_D_inst (
    .LO(LO), // Output of MUX to local routing
    .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
);

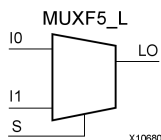
// End of MUXF5_D_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF5_L

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



Introduction

This design element provides a multiplexer function in a CLB slice for creating a function-of-5 look-up table or a 4-to-1 multiplexer in combination with the associated look-up tables. The local outputs (LO) from the two look-up tables are connected to the I0 and I1 inputs of the MUXF5. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The LO output connects to other inputs in the same CLB slice.

See also “MUXF5” and “MUXF5_D”.

Logic Table

| Inputs | | | Output |
|--------|----|----|--------|
| S | I0 | I1 | LO |
| 0 | 1 | X | 1 |
| 0 | 0 | X | 0 |
| 1 | X | 1 | 1 |
| 1 | X | 0 | 0 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MUXF5_L: Slice MUX to tie two LUT4's together with local output
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// MUXF5_L: Slice MUX to tie two LUT4's together with local output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

MUXF5_L MUXF5_L_inst (
    .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
);

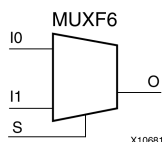
// End of MUXF5_L_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF6

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



Introduction

This design element provides a multiplexer function in two slices for creating a function-of-6 look-up table or an 8-to-1 multiplexer in combination with the associated four look-up tables and two MUXF5s. The local outputs (LO) from the two MUXF5s in the CLB are connected to the I0 and I1 inputs of the MUXF6. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The variants, “MUXF6_D” and “MUXF6_L”, provide additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

Logic Table

| Inputs | | | Outputs |
|--------|----|----|---------|
| S | I0 | I1 | O |
| 0 | 1 | X | 1 |
| 0 | 0 | X | 0 |
| 1 | X | 1 | 1 |
| 1 | X | 0 | 0 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MUXF6: CLB MUX to tie two MUXF5's together with general output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

MUXF6_inst : MUXF6
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- 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 Instantiation Template

```
// MUXF6: CLB MUX to tie two MUXF5's together with general output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

MUXF6 MUXF6_inst (
    .O(O),    // Output of MUX to general routing
    .I0(I0),  // Input (tie to MUXF5 LO out)
    .I1(I1),  // Input (tie to MUXF5 LO out)
    .S(S)     // Input select to MUX
);

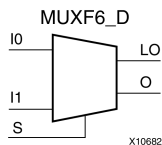
// End of MUXF6_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF6_D

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



Introduction

This design element provides a multiplexer function in a two slices for creating a function-of-6 look-up table or an 8-to-1 multiplexer in combination with the associated four look-up tables and two MUXF5s. The local outputs (LO) from the two MUXF5s in the CLB are connected to the I0 and I1 inputs of the MUXF6. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

Outputs O and LO are functionally identical. The O output is a general interconnect. The LO output connects to other inputs in the same CLB slice.

Logic Table

| Inputs | | | Outputs | |
|--------|----|----|---------|----|
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

```
-- MUXF6_D: CLB MUX to tie two MUXF5's together with general and local outputs
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2
```

```
MUXF6_D_inst : MUXF6_D
port map (
  LO => LO, -- Ouput of MUX to local routing
  O  => O,  -- Output of MUX to general routing
  I0 => I0, -- 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 Instantiation Template

```
// MUXF6_D: CLB MUX to tie two MUXF5's together with general and local outputs
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

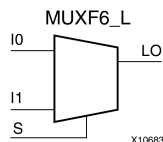
// End of MUXF6_D_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF6_L

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



Introduction

This design element provides a multiplexer function for use in creating a function-of-6 look-up table or an 8-to-1 multiplexer in combination with the associated four look-up tables and two MUXF5s. The local outputs (LO) from the two MUXF5s in the CLB are connected to the I0 and I1 inputs of the MUXF6. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The LO output connects to other inputs in the same CLB slice.

Logic Table

| Inputs | | | Output |
|--------|----|----|--------|
| S | I0 | I1 | LO |
| 0 | 1 | X | 1 |
| 0 | 0 | X | 0 |
| 1 | X | 1 | 1 |
| 1 | X | 0 | 0 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MUXF6_L: CLB MUX to tie two MUXF5's together with local output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

MUXF6_L_inst : MUXF6_L
port map (
    LO => LO,  -- Output of MUX to local routing
    I0 => I0,  -- 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 Instantiation Template

```
// MUXF6_L: CLB MUX to tie two MUXF5's together with local output
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

MUXF6_L MUXF6_L_inst (
    .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
);

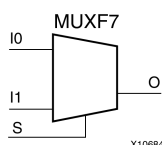
// End of MUXF6_L_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF7

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



Introduction

This design element provides a multiplexer function for use in creating a function-of-7 look-up table or an 8-to-1 multiplexer in combination with the associated look-up tables. Local outputs (LO) of MUXF6 are connected to the I0 and I1 inputs of the MUXF7. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The variants, “MUXF7_D” and “MUXF7_L”, provide additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

Logic Table

| Inputs | | | Outputs |
|--------|----|----|---------|
| S | I0 | I1 | O |
| 0 | I0 | X | I0 |
| 1 | X | I1 | I1 |
| X | 0 | 0 | 0 |
| X | 1 | 1 | 1 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|----------------------------------|
| O | Output | 1 | Output of MUX to general routing |
| I0 | Input | 1 | Input (tie to MUXF6 LO out) |
| I1 | Input | 1 | Input (tie to MUXF6 LO out) |
| S | Input | 1 | Input select to MUX |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MUXF7: CLB MUX to tie two MUXF6's together with general output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

MUXF7_inst : MUXF7
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,    -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S      -- Input select to MUX
);

-- End of MUXF7_inst instantiation
```

Verilog Instantiation Template

```
// MUXF7: CLB MUX to tie two LUT6's or MUXF6's together with general output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

MUXF7 MUXF7_inst (
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .I1(I1),    // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .S(S)      // Input select to MUX
);

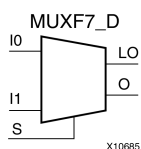
// End of MUXF7_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF7_D

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



Introduction

This design element provides a multiplexer function for use in creating a function-of-7 look-up table or a 16-to-1 multiplexer in combination with the associated look-up tables. Local outputs (LO) of MUXF6 are connected to the I0 and I1 inputs of the MUXF7. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

Outputs O and LO are functionally identical. The O output is a general interconnect. The LO output connects to other inputs in the same CLB slice.

Logic Table

| Inputs | | | Outputs | |
|--------|----|----|---------|----|
| 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 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|----------------------------------|
| O | Output | 1 | Output of MUX to general routing |
| LO | Output | 1 | Output of MUX to local routing |
| I0 | Input | 1 | Input (tie to MUXF6 LO out) |
| I1 | Input | 1 | Input (tie to MUXF6 LO out) |
| S | Input | 1 | Input select to MUX |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MUXF7_D: CLB MUX to tie two MUXF6's together with general and local outputs
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// MUXF7_D: CLB MUX to tie two LUT6's or MUXF6's together with general and local outputs
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

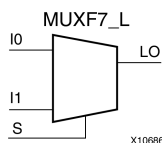
// End of MUXF7_D_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF7_L

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



Introduction

This design element provides a multiplexer function for use in creating a function-of-7 look-up table or a 16-to-1 multiplexer in combination with the associated look-up tables. Local outputs (LO) of MUXF6 are connected to the I0 and I1 inputs of the MUXF7. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The LO output connects to other inputs in the same CLB slice.

Logic Table

| Inputs | | | Output |
|--------|----|----|--------|
| S | I0 | I1 | LO |
| 0 | I0 | X | I0 |
| 1 | X | I1 | I1 |
| X | 0 | 0 | 0 |
| X | 1 | 1 | 1 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--------------------------------|
| LO | Output | 1 | Output of MUX to local routing |
| I0 | Input | 1 | Input |
| I1 | Input | 1 | Input |
| S | Input | 1 | Input select to MUX |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MUXF7_L: CLB MUX to tie two MUXF6's together with local output
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of MUXF7_L_inst instantiation
```

Verilog Instantiation Template

```
// MUXF7_L: CLB MUX to tie two LUT6's or MUXF6's together with local output
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

MUXF7_L MUXF7_L_inst (
    .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
);

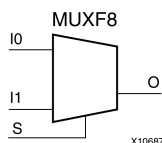
// End of MUXF7_L_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF8

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



Introduction

This design element provides a multiplexer function in eight slices for creating a function-of-8 look-up table or a 16-to-1 multiplexer in combination with the associated look-up tables, MUXF5s, MUXF6s, and MUXF7s. Local outputs (LO) of MUXF7 are connected to the I0 and I1 inputs of the MUXF8. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

Logic Table

| Inputs | | | Outputs |
|--------|----|----|---------|
| S | I0 | I1 | O |
| 0 | I0 | X | I0 |
| 1 | X | I1 | I1 |
| X | 0 | 0 | 0 |
| X | 1 | 1 | 1 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|----------------------------------|
| O | Output | 1 | Output of MUX to general routing |
| I0 | Input | 1 | Input (tie to MUXF7 LO out) |
| I1 | Input | 1 | Input (tie to MUXF7 LO out) |
| S | Input | 1 | Input select to MUX |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MUXF8: CLB MUX to tie two MUXF7's together with general output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of MUXF8_inst instantiation
```

Verilog Instantiation Template

```
// MUXF8: CLB MUX to tie two MUXF7's together with general output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

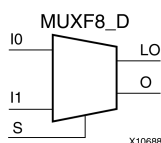
// End of MUXF8_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF8_D

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



Introduction

This design element provides a multiplexer function in eight slices for creating a function-of-8 look-up table or a 32-to-1 multiplexer in combination with the associated four look-up tables and two MUXF8s. Local outputs (LO) of MUXF7 are connected to the I0 and I1 inputs of the MUXF8. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

Outputs O and LO are functionally identical. The O output is a general interconnect. The LO output connects to other inputs in the same CLB slice.

Logic Table

| Inputs | | | Outputs | |
|--------|----|----|---------|----|
| 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 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|----------------------------------|
| O | Output | 1 | Output of MUX to general routing |
| LO | Output | 1 | Output of MUX to local routing |
| I0 | Input | 1 | Input (tie to MUXF7 LO out) |
| I1 | Input | 1 | Input (tie to MUXF7 LO out) |
| S | Input | 1 | Input select to MUX |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of MUXF8_D_inst instantiation
```

Verilog Instantiation Template

```
// MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

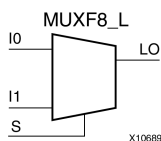
// End of MUXF8_D_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

MUXF8_L

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



Introduction

This design element provides a multiplexer function in eight slices for creating a function-of-8 look-up table or a 32-to-1 multiplexer in combination with the associated four look-up tables and two MUXF8s. Local outputs (LO) of MUXF7 are connected to the I0 and I1 inputs of the MUXF8. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The LO output connects to other inputs in the same CLB slice.

Logic Table

| Inputs | | | Output |
|--------|----|----|--------|
| S | I0 | I1 | LO |
| 0 | I0 | X | I0 |
| 1 | X | I1 | I1 |
| X | 0 | 0 | 0 |
| X | 1 | 1 | 1 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--------------------------------|
| LO | Output | 1 | Output of MUX to local routing |
| I0 | Input | 1 | Input (tie to MUXF7 LO out) |
| I1 | Input | 1 | Input (tie to MUXF7 LO out) |
| S | Input | 1 | Input select to MUX |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- MUXF8_L: CLB MUX to tie two MUXF7's together with local output
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// MUXF8_L: CLB MUX to tie two MUXF7's together with local output
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

MUXF8_L MUXF8_L_inst (
    .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
);

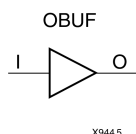
// End of MUXF8_L_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

OBUF

Primitive: Output Buffer



Introduction

This design element is a simple output buffer used to drive output signals to the FPGA device pins that do not need to be 3-stated (constantly driven). Either an OBUF, OBUFT, OBUFDS, or OBUFTDS must be connected to every output port in the design.

This element isolates the internal circuit and provides drive current for signals leaving a chip. It exists in input/output blocks (IOB). Its output (O) is connected to an OPAD or an IOPAD. The interface standard used by this element is LVTTTL. Also, this element has selectable drive and slew rates using the DRIVE and SLOW or FAST constraints. The defaults are DRIVE=12 mA and SLOW slew.

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|---|
| O | Output | 1 | Output of OBUF to be connected directly to top-level output port. |
| I | Input | 1 | Input of OBUF. Connect to the logic driving the output port. |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|------------|---------|------------------------|-----------|---|
| DRIVE | Integer | 2, 4, 6, 8, 12, 16, 24 | 12 | Specifies the output current drive strength of the I/O. It is suggested that you set this to the lowest setting tolerable for the design drive and timing requirements. |
| IOSTANDARD | String | See Data Sheet | "DEFAULT" | Assigns an I/O standard to the element. |
| SLEW | String | "SLOW" or "FAST" | "SLOW" | Specifies the slew rate of the output driver. Consult the product Data Sheet for recommendations of the best setting for this attribute. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- OBUF: Single-ended Output Buffer
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of OBUF_inst instantiation
```

Verilog Instantiation Template

```
// OBUF: Single-ended Output Buffer
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

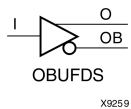
// End of OBUF_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

OBUFDS

Primitive: Differential Signaling Output Buffer



Introduction

This design element is a single output buffer that supports low-voltage, differential signaling (1.8 v CMOS). OBUFDS isolates the internal circuit and provides drive current for signals leaving the chip. Its output is represented as two distinct ports (O and OB), one deemed the "master" and the other the "slave." The master and the slave are opposite phases of the same logical signal (for example, MYNET and MYNETB).

Logic Table

| Inputs | Outputs | |
|--------|---------|----|
| I | O | OB |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--|
| O | Output | 1 | Diff_p output (connect directly to top level port) |
| OB | Output | 1 | Diff_n output (connect directly to top level port) |
| I | Input | 1 | Buffer input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Recommended |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|------------|--------|----------------|-----------|---|
| IOSTANDARD | String | See Data Sheet | "DEFAULT" | Assigns an I/O standard to the element. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- OBUFDS: Differential Output Buffer
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of OBUFDS_inst instantiation
```

Verilog Instantiation Template

```
// OBUFDS: Differential Output Buffer
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

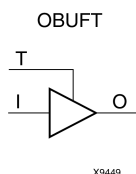
// End of OBUFDS_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

OBUFT

Primitive: 3-State Output Buffer with Active Low Output Enable



Introduction

This design element is a single, 3-state output buffer with input I, output O, and active-Low output enables (T). This element uses the LVTTL standard and has selectable drive and slew rates using the DRIVE and SLOW or FAST constraints. The defaults are DRIVE=12 mA and SLOW slew.

When T is Low, data on the inputs of the buffers is transferred to the corresponding outputs. When T is High, the output is high impedance (off or Z state). OBUFTs are generally used when a single-ended output is needed with a 3-state capability, such as the case when building bidirectional I/O.

Logic Table

| Inputs | | Outputs |
|--------|---|---------|
| T | I | O |
| 1 | X | Z |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--|
| O | Output | 1 | Buffer output (connect directly to top-level port) |
| I | Input | 1 | Buffer input |
| T | Input | 1 | 3-state enable input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|------------|---------|------------------------|-----------|---|
| DRIVE | Integer | 2, 4, 6, 8, 12, 16, 24 | 12 | Specifies the output current drive strength of the I/O. It is suggested that you set this to the lowest setting tolerable for the design drive and timing requirements. |
| IOSTANDARD | String | See Data Sheet | "DEFAULT" | Assigns an I/O standard to the element. |
| SLEW | String | "SLOW" or "FAST" | "SLOW" | Specifies the slew rate of the output driver. See the Data Sheet for recommendations of the best setting for this attribute. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- OBUFT: Single-ended 3-state Output Buffer
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// OBUFT: Single-ended 3-state Output Buffer
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

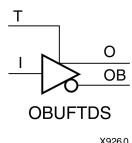
// End of OBUFT_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

OBUFTDS

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



Introduction

This design element is an output buffer that supports low-voltage, differential signaling. For the OBUFTDS, a design level interface signal is represented as two distinct ports (O and OB), one deemed the "master" and the other the "slave." The master and the slave are opposite phases of the same logical signal (for example, MYNET_P and MYNET_N).

Logic Table

| Inputs | | Outputs | |
|--------|---|---------|----|
| I | T | O | OB |
| X | 1 | Z | Z |
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--|
| O | Output | 1 | Diff_p output (connect directly to top level port) |
| OB | Output | 1 | Diff_n output (connect directly to top level port) |
| I | Input | 1 | Buffer input |
| T | Input | 1 | 3-state enable input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Recommended |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|------------|--------|----------------|-----------|---|
| IOSTANDARD | String | See Data Sheet | "DEFAULT" | Assigns an I/O standard to the element. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- OBUFTDS: Differential 3-state Output Buffer
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of OBUFTDS_inst instantiation
```

Verilog Instantiation Template

```
// OBUFTDS: Differential 3-state Output Buffer
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

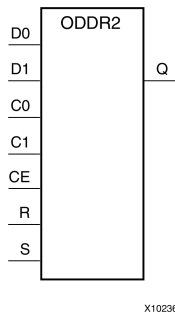
// End of OBUFTDS_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

ODDR2

Primitive: Dual Data Rate Output D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset



Introduction

The design element is an output double data rate (DDR) register useful in producing double data rate signals exiting the FPGA. The ODDR2 requires two clocks (C0 and C1) to be connected to the component so that data is provided at the positive edge of both clocks. The ODDR2 features an active high clock enable port, CE, which can be used to suspend the operation of the registers and both set and reset ports that can be configured to be synchronous or asynchronous to the respective clocks. The ODDR2 has an optional alignment feature, which allows data to be captured by a single clock and clocked out by two clocks.

Logic Table

| Inputs | | | | | | | Outputs |
|---|---|----|----|----|----|----|-----------|
| S | R | CE | D0 | D1 | C0 | C1 | O |
| 1 | X | X | X | X | X | X | 1 |
| 0 | 1 | X | X | X | X | X | not INIT |
| 0 | 0 | 0 | X | X | X | X | No Change |
| 0 | 0 | 1 | D0 | X | ↑ | X | D0 |
| 0 | 0 | 1 | X | D1 | X | ↑ | D1 |
| Set/Reset can be synchronous via SRTYPE value | | | | | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Recommended |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Descriptions |
|---------------|---------|--------------------|---------|---|
| DDR_ALIGNMENT | String | "NONE", "C0", "C1" | "NONE" | Sets the input capture behavior for the DDR register. "NONE" clocks in data to the D0 input on the positive transition of the C0 clock and D1 on the positive transition of the C1 clock. "C0" allows the input clocking of both D0 and D1 align to the positive edge of the C0 clock. "C1" allows the input clocking of both D0 and D1 align to the positive edge of the C1 clock. |
| INIT | Integer | 0, 1 | 0 | Sets initial state of the Q0 output to 0 or 1. |
| SRTYPE | String | "SYNC", "ASYNC" | "SYNC" | Specifies "SYNC" or "ASYNC" set/reset. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- ODDR2: Output Double Data Rate Output Register with Set, Reset
--      and Clock Enable.
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

ODDR2_inst : ODDR2
generic map(
    DDR_ALIGNMENT => "NONE", -- Sets output alignment to "NONE", "C0", "C1"
    INIT => '0', -- Sets initial state of the Q output to '0' or '1'
    SRTYPE => "SYNC") -- Specifies "SYNC" or "ASYNC" set/reset
port map (
    Q => Q, -- 1-bit output data
    C0 => C0, -- 1-bit clock input
    C1 => C1, -- 1-bit clock input
    CE => CE, -- 1-bit clock enable input
    D0 => D0, -- 1-bit data input (associated with C0)
    D1 => D1, -- 1-bit data input (associated with C1)
    R => R, -- 1-bit reset input
    S => S -- 1-bit set input
);

-- End of ODDR2_inst instantiation
```

Verilog Instantiation Template

```
// ODDR2: Output Double Data Rate Output Register with Set, Reset
//      and Clock Enable.
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

ODDR2 #(
    .DDR_ALIGNMENT("NONE"), // Sets output alignment to "NONE", "C0" or "C1"
    .INIT(1'b0), // Sets initial state of the Q output to 1'b0 or 1'b1
    .SRTYPE("SYNC")) // Specifies "SYNC" or "ASYNC" set/reset
) ODDR2_inst (
    .Q(Q), // 1-bit DDR output data
    .C0(C0), // 1-bit clock input
    .C1(C1), // 1-bit clock input
    .CE(CE), // 1-bit clock enable input
    .D0(D0), // 1-bit data input (associated with C0)
    .D1(D1), // 1-bit data input (associated with C1)
```

```
.R(R),    // 1-bit reset input
.S(S)    // 1-bit set input
);

// End of ODDR2_inst instantiation
```

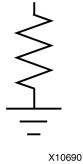
For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

PULLDOWN

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

PULLDOWN



Introduction

This resistor element is connected to input, output, or bidirectional pads to guarantee a logic Low level for nodes that might float.

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--|
| O | Output | 1 | Pulldown output (connect directly to top level port) |

Design Entry Method

| | |
|-----------------------------|-----|
| Instantiation | Yes |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- PULLDOWN: I/O Buffer Weak Pull-down
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of PULLDOWN_inst instantiation

```

Verilog Instantiation Template

```
// PULLDOWN: I/O Buffer Weak Pull-down
//          Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

PULLDOWN PULLDOWN_inst (
    .O(0)      // Pulldown output (connect directly to top-level port)
);

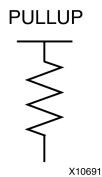
// End of PULLDOWN_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

PULLUP

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



Introduction

This design element allows for an input, 3-state output or bi-directional port to be driven to a weak high value when not being driven by an internal or external source. This element establishes a High logic level for open-drain elements and macros when all the drivers are off.

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--|
| O | Output | 1 | Pullup output (connect directly to top level port) |

Design Entry Method

| | |
|-----------------------------|-----|
| Instantiation | Yes |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- PULLUP: I/O Buffer Weak Pull-up
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

-- End of PULLUP_inst instantiation
```

Verilog Instantiation Template

```
// PULLUP: I/O Buffer Weak Pull-up
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

PULLUP PULLUP_inst (
    .O(0)      // Pullup output (connect directly to top-level port)
);

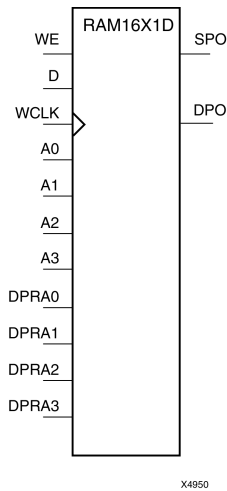
// End of PULLUP_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAM16X1D

Primitive: 16-Deep by 1-Wide Static Dual Port Synchronous RAM



Introduction

This element is a 16-word by 1-bit static dual port random access memory with synchronous write capability. The device has two address ports: the read address (DPRA3:DPRA0) and the write address (A3:A0). These two address ports are asynchronous. The read address controls the location of the data driven out of the output pin (DPO), and the write address controls the destination of a valid write transaction. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected.

When WE is High, any positive transition on (WCLK) loads the data on the data input (D) into the word selected by the 4-bit write address. For predictable performance, write address and data inputs must be stable before a Low-to-High (WCLK) transition. This RAM block assumes an active-High (WCLK). (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The SPO output reflects the data in the memory cell addressed by A3:A0. The DPO output reflects the data in the memory cell addressed by DPRA3:DPRA0.

Note The write process is not affected by the address on the read address port.

You can use the INIT attribute to directly specify an initial value. The value must be a hexadecimal number, for example, INIT=ABAC. If the INIT attribute is not specified, the RAM is initialized with all zeros.

Logic Table

Mode selection is shown in the following logic table:

| Inputs | | | Outputs | |
|---|------|---|---------|--------|
| WE (mode) | WCLK | D | SPO | DPO |
| 0 (read) | X | X | data_a | data_d |
| 1 (read) | 0 | X | data_a | data_d |
| 1 (read) | 1 | X | data_a | data_d |
| 1 (write) | ↑ | D | D | data_d |
| 1 (read) | ↓ | X | data_a | data_d |
| data_a = word addressed by bits A3-A0 | | | | |
| data_d = word addressed by bits DPRA3-DPRA0 | | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|------------|--|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros. | Initializes RAMs, registers, and look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM16X1D: 16 x 1 positive edge write, asynchronous read dual-port distributed RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```

// RAM16X1D: 16 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

RAM16X1D #(
    .INIT(16'h0000) // Initial contents of RAM
) RAM16X1D_inst (
    .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 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
);

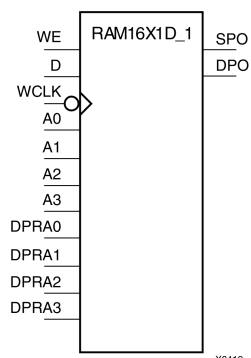
// End of RAM16X1D_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAM16X1D_1

Primitive: 16-Deep by 1-Wide Static Dual Port Synchronous RAM with Negative-Edge Clock



Introduction

This is a 16-word by 1-bit static dual port random access memory with synchronous write capability and negative-edge clock. The device has two separate address ports: the read address (DPRA3:DPRA0) and the write address (A3:A0). These two address ports are asynchronous. The read address controls the location of the data driven out of the output pin (DPO), and the write address controls the destination of a valid write transaction.

When the write enable (WE) is set to Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the word selected by the 4-bit write address. For predictable performance, write address and data inputs must be stable before a High-to-Low WCLK transition. This RAM block assumes an active-Low (WCLK). (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

You can initialize RAM16X1D_1 during configuration using the INIT attribute.

The SPO output reflects the data in the memory cell addressed by A3:A0. The DPO output reflects the data in the memory cell addressed by DPRA3:DPRA0.

Note The write process is not affected by the address on the read address port.

Logic Table

Mode selection is shown in the following logic table:

| Inputs | | | Outputs | |
|---|------|---|---------|--------|
| WE (mode) | WCLK | D | SPO | DPO |
| 0 (read) | X | X | data_a | data_d |
| 1 (read) | 0 | X | data_a | data_d |
| 1 (read) | 1 | X | data_a | data_d |
| 1 (write) | ↓ | D | D | data_d |
| 1 (read) | ↑ | X | data_a | data_d |
| data_a = word addressed by bits A3:A0 | | | | |
| data_d = word addressed by bits DPRA3:DPRA0 | | | | |

Port Descriptions

| Port | Direction | Width | Function |
|-------|-----------|-------|-----------------------------|
| DPO | Output | 1 | Read-only 1-Bit data output |
| SPO | Output | 1 | R/W 1-Bit data output |
| A0 | Input | 1 | R/W address[0] input |
| A1 | Input | 1 | R/W address[1] input |
| A2 | Input | 1 | R/W address[2] input |
| A3 | Input | 1 | R/W address[3] input |
| D | Input | 1 | Write 1-Bit data input |
| DPRA0 | Input | 1 | Read-only address[0] input |
| DPRA1 | Input | 1 | Read-only address[1] input |
| DPRA2 | Input | 1 | Read-only address[2] input |
| DPRA3 | Input | 1 | Read-only address[3] input |
| WCLK | Input | 1 | Write clock input |
| WE | Input | 1 | Write enable input |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|--|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Initializes RAMs, registers, and look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

```
-- RAM16X1D_1: 16 x 1 negative edge write, asynchronous read dual-port distributed RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2
```

```
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 Instantiation Template

```

// RAM16X1D_1: 16 x 1 negative edge write, asynchronous read dual-port distributed RAM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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

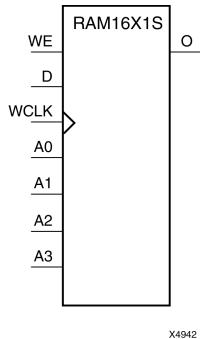
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAM16X1S

Primitive: 16-Deep by 1-Wide Static Synchronous RAM



Introduction

This element is a 16-word by 1-bit static random access memory with synchronous write capability. When the write enable (WE) is set Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is set High, any positive transition on WCLK loads the data on the data input (D) into the word selected by the 4-bit address (A3:A0). This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins. You can initialize RAM16X1S during configuration using the INIT attribute.

Logic Table

| Inputs | | | Outputs |
|-------------------------------------|------|---|---------|
| WE(mode) | WCLK | D | O |
| 0 (read) | X | X | Data |
| 1 (read) | 0 | X | Data |
| 1 (read) | 1 | X | Data |
| 1 (write) | ↑ | D | D |
| 1 (read) | ↓ | X | Data |
| Data = word addressed by bits A3:A0 | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|--|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Specifies initial contents of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAM16X1S: 16 x 1 posedge write distributed (LUT) RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// RAM16X1S: 16 x 1 posedge write distributed (LUT) RAM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

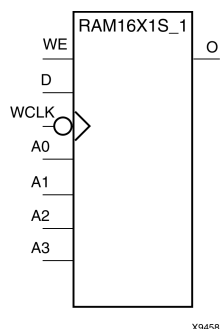
// End of RAM16X1S_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAM16X1S_1

Primitive: 16-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



Introduction

This element is a 16-word by 1-bit static random access memory with synchronous write capability and negative-edge clock. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the word selected by the 4-bit address (A3:A0). For predictable performance, address and data inputs must be stable before a High-to-Low WCLK transition. This RAM block assumes an active-Low (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can initialize this element during configuration using the INIT attribute.

Logic Table

| Inputs | | | Outputs |
|-------------------------------------|------|---|---------|
| WE(mode) | WCLK | D | O |
| 0 (read) | X | X | Data |
| 1 (read) | 0 | X | Data |
| 1 (read) | 1 | X | Data |
| 1 (write) | ↓ | D | D |
| 1 (read) | ↑ | X | Data |
| Data = word addressed by bits A3:A0 | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|--|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Specifies initial contents of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAM16X1S_1: 16 x 1 negedge write distributed (LUT) RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// RAM16X1S_1: 16 x 1 negedge write distributed (LUT) RAM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

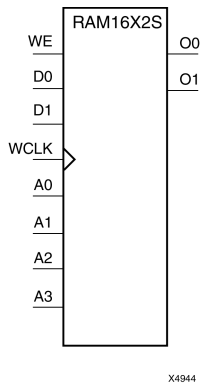
// End of RAM16X1S_1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAM16X2S

Primitive: 16-Deep by 2-Wide Static Synchronous RAM



Introduction

This element is a 16-word by 2-bit static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is High, any positive transition on WCLK loads the data on the data input (D1:D0) into the word selected by the 4-bit address (A3:A0). For predictable performance, address and data inputs must be stable before a Low-to-High WCLK transition. This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pins (O1:O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can use the INIT_xx properties to specify the initial contents of a wide RAM. INIT_00 initializes the RAM cells corresponding to the O0 output, INIT_01 initializes the cells corresponding to the O1 output, etc. For example, a RAM16X2S instance is initialized by INIT_00 and INIT_01 containing 4 hex characters each. A RAM16X8S instance is initialized by eight properties INIT_00 through INIT_07 containing 4 hex characters each. A RAM64x2S instance is completely initialized by two properties INIT_00 and INIT_01 containing 16 hex characters each.

Except for Virtex-4 devices, the initial contents of this element cannot be specified directly.

Logic Table

| Inputs | | | Outputs |
|-------------------------------------|------|-------|---------|
| WE (mode) | WCLK | D1:D0 | O1:O0 |
| 0 (read) | X | X | Data |
| 1(read) | 0 | X | Data |
| 1(read) | 1 | X | Data |
| 1(write) | ↑ | D1:D0 | D1:D0 |
| 1(read) | ↓ | X | Data |
| Data = word addressed by bits A3:A0 | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|--------------------|-------------|------------------|-----------|--|
| INIT_00 to INIT_01 | Hexadecimal | Any 16-Bit Value | All zeros | Initializes RAMs, registers, and look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAM16X2S: 16 x 2 posedge write distributed (LUT) RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// RAM16X2S: 16 x 2 posedge write distributed (LUT) RAM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
```



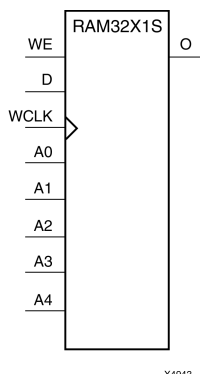
```
.WE(WE)      // Write enable input  
);  
  
// End of RAM16X2S_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAM32X1S

Primitive: 32-Deep by 1-Wide Static Synchronous RAM



Introduction

The design element is a 32-word by 1-bit static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any positive transition on (WCLK) loads the data on the data input (D) into the word selected by the 5-bit address (A4-A0). For predictable performance, address and data inputs must be stable before a Low-to-High (WCLK) transition. This RAM block assumes an active-High (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins. You can initialize RAM32X1S during configuration using the INIT attribute.

Logic Table

| Inputs | | | Outputs |
|-----------|------|---|---------|
| WE (Mode) | WCLK | D | O |
| 0 (read) | X | X | Data |
| 1 (read) | 0 | X | Data |
| 1 (read) | 1 | X | Data |
| 1 (write) | ↓ | D | D |
| 1 (read) | ↑ | X | Data |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Descriptions |
|-----------|-------------|------------------|-----------|--|
| INIT | Hexadecimal | Any 32-Bit Value | All zeros | Specifies initial contents of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

RAM32X1S_inst : RAM32X1S
generic map (
    INIT => X"00000000")
port map (
    O => O,          -- RAM output
    A0 => A0,         -- RAM address[0] input
    A1 => A1,         -- RAM address[1] input
    A2 => A2,         -- RAM address[2] input
    A3 => A3,         -- RAM address[3] input
    A4 => A4,         -- RAM address[4] input
    D => D,           -- RAM data input
    WCLK => WCLK,     -- Write clock input
    WE => WE          -- Write enable input
);

-- End of RAM32X1S_inst instantiation
```

Verilog Instantiation Template

```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

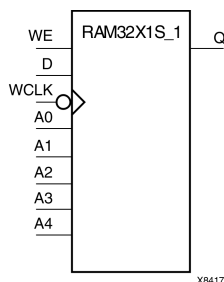
// End of RAM32X1S_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAM32X1S_1

Primitive: 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



Introduction

The design element is a 32-word by 1-bit static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the word selected by the 5-bit address (A4:A0). For predictable performance, address and data inputs must be stable before a High-to-Low (WCLK) transition. This RAM block assumes an active-Low (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins. You can initialize RAM32X1S_1 during configuration using the INIT attribute.

Logic Table

| Inputs | | | Outputs |
|-------------------------------------|------|---|---------|
| WE (Mode) | WCLK | D | O |
| 0 (read) | X | X | Data |
| 1 (read) | 0 | X | Data |
| 1 (read) | 1 | X | Data |
| 1 (write) | ↓ | D | D |
| 1 (read) | ↑ | X | Data |
| Data = word addressed by bits A4:A0 | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Descriptions |
|-----------|-------------|------------------|---------|--|
| INIT | Hexadecimal | Any 32-Bit Value | 0 | Initializes RAMs, registers, and look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

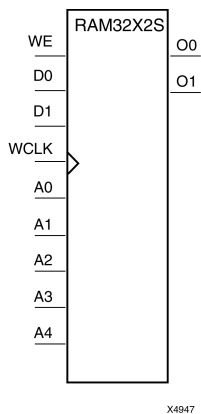
// End of RAM32X1S_1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAM32X2S

Primitive: 32-Deep by 2-Wide Static Synchronous RAM



Introduction

The design element is a 32-word by 2-bit static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any positive transition on (WCLK) loads the data on the data input (D1-D0) into the word selected by the 5-bit address (A4-A0). For predictable performance, address and data inputs must be stable before a Low-to-High (WCLK) transition. This RAM block assumes an active-High (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block. The signal output on the data output pins (O1-O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can use the INIT_00 and INIT_01 properties to specify the initial contents of RAM32X2S.

Logic Table

| Inputs | | | Outputs |
|-------------------------------------|------|-------|---------|
| WE (Mode) | WCLK | D | O0-O1 |
| 0 (read) | X | X | Data |
| 1 (read) | 0 | X | Data |
| 1 (read) | 1 | X | Data |
| 1 (write) | ↑ | D1:D0 | D1:D0 |
| 1 (read) | ↓ | X | Data |
| Data = word addressed by bits A4:A0 | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Descriptions |
|-----------|-------------|------------------|-----------|------------------------|
| INIT_00 | Hexadecimal | Any 32-Bit Value | All zeros | INIT for bit 0 of RAM. |
| INIT_01 | Hexadecimal | Any 32-Bit Value | All zeros | INIT for bit 1 of RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

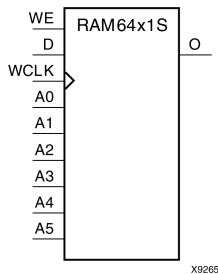
// End of RAM32X2S_inst instantiation
```

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

RAM64X1S

Primitive: 64-Deep by 1-Wide Static Synchronous RAM



Introduction

This design element is a 64-word by 1-bit static random access memory (RAM) with synchronous write capability. When the write enable is set Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is set High, any positive transition on WCLK loads the data on the data input (D) into the word selected by the 6-bit address (A5:A0). This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can initialize this element during configuration using the INIT attribute.

Logic Table

Mode selection is shown in the following logic table

| Inputs | | | Outputs |
|-------------------------------------|------|---|---------|
| WE (mode) | WCLK | D | O |
| 0 (read) | X | X | Data |
| 1 (read) | 0 | X | Data |
| 1 (read) | 1 | X | Data |
| 1 (write) | ↑ | D | D |
| 1 (read) | ↓ | X | Data |
| Data = word addressed by bits A5:A0 | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|--|
| INIT | Hexadecimal | Any 64-Bit Value | All zeros | Initializes ROMs, RAMs, registers, and look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

RAM64X1S_inst : RAM64X1S
generic map (
  INIT => X"0000000000000000")
port map (
  O => O,           -- 1-bit data output
  A0 => A0,          -- Address[0] input bit
  A1 => A1,          -- Address[1] input bit
  A2 => A2,          -- Address[2] input bit
  A3 => A3,          -- Address[3] input bit
  A4 => A4,          -- Address[4] input bit
  A5 => A5,          -- Address[5] input bit
  D => D,           -- 1-bit data input
  WCLK => WCLK,      -- Write clock input
  WE => WE           -- Write enable input
);

-- End of RAM64X1S_inst instantiation
```

Verilog Instantiation Template

```
// RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

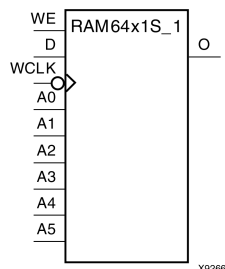
// End of RAM64X1S_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAM64X1S_1

Primitive: 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



Introduction

This design element is a 64-word by 1-bit static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the word selected by the 6-bit address (A5:A0). For predictable performance, address and data inputs must be stable before a High-to-Low (WCLK) transition. This RAM block assumes an active-Low (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can initialize this element during configuration using the INIT attribute.

Logic Table

| Inputs | | | Outputs |
|-------------------------------------|------|---|---------|
| WE (mode) | WCLK | D | O |
| 0 (read) | X | X | Data |
| 1 (read) | 0 | X | Data |
| 1 (read) | 1 | X | Data |
| 1 (write) | ↓ | D | D |
| 1 (read) | ↑ | X | Data |
| Data = word addressed by bits A5:A0 | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|--|
| INIT | Hexadecimal | Any 64-Bit Value | All zeros | Initializes ROMs, RAMs, registers, and look-up tables. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

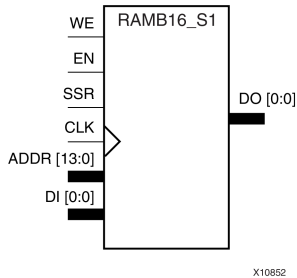
// End of RAM64X1S_1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAMB16_S1

Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 1-bit Port



Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

| Data Cells | | Parity Cells | | | | |
|------------|-------|--------------|-------|-------------|----------|------------|
| Depth | Width | Depth | Width | Address Bus | Data Bus | Parity Bus |
| 16384 | 1 | - | - | (13:0) | (0:0) | - |

The enable EN pin controls read, write, and reset. When EN is Low, no data is written and the outputs (DO and DOP) retain the last state. When EN is High and reset (SSR) is High, DO and DOP are set to SRVAL during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI and DIP. When EN is High and WE is Low, the data stored in the RAM address (ADDR) is read during the Low-to-High clock transition. By default, WRITE_MODE=WRITE_FIRST, when EN and WE are High, the data on the data inputs (DI and DIP) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data outputs (DO and DOP) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Logic Table

| Inputs | | | | | | | | Outputs | | | |
|--------|----|-----|----|-----|------|------|-------|-------------------------------|-------------------------------|------------------|-------------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT | INIT | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL | SRVAL | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ RAM | No Change ¹ RAM | RAM(addr) =>data | RAM(addr) =>pdata |

| Inputs | | | | | | | | Outputs | | | |
|--------|----|-----|----|-----|------|----|-----|--|---|--------------|------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| | | | | | | | | (addr) ² data ³ | (addr) ² pdata ³ | | |

GSR=Global Set Reset signal
 INIT=Value specified by the INIT attribute for data memory. Default is all zeros.
 SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE=NO_CHANGE
²WRITE_MODE=READ_FIRST
³WRITE_MODE=WRITE_FIRST

Initialization

Initializing Memory Contents

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

Any INIT_xx or INITP_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

Selecting Write Mode

The WRITE_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE_MODE is set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE_MODE to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE_MODE to NO_CHANGE to have the input written to memory without changing the output.

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-------------------------|------------------------|---|---------------|---|
| INIT | Binary/ Hexadecimal | Any Hex Value | All zeros | Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM. |
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SRVAL | Binary/ Hexadecimal | Any Hex Value | All zeros | Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM. |
| WRITE_MODE | String | "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAMB16_S1: Single-Port RAM
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

RAMB16_S1_inst : RAMB16_S1
generic map (
  INIT => X"0", -- Value of output RAM registers at startup
  SRVAL => X"0", -- Ouput value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
```

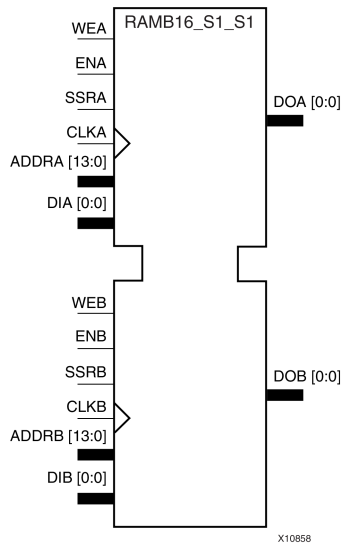
176 www.xilinx.com Spartan-3E Libraries Guide for HDL Designs UG617 (v 13.2) July 6, 2011

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

RAMB16_S1_S1

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------------------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR _A | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

¹WRITE_MODE_A=NO_CHANGE.

²WRITE_MODE_A=READ_FIRST.

³WRITE_MODE_A=WRITE_FIRST.

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S1_S1 | 16384 x 1 | - | (13:0) | (0:0) | - | 16384 x 1 | - | (13:0) | (0:0) | - |
| ¹ Depth x Width | | | | | | | | | | |

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Width port})) - 1$$

$$\text{End} = (\text{ADDR port}) * (\text{Width port})$$

The following tables show address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|--|---|--|--|--|--|---|--|--|---|
| 1 | 2048 | <----- | 3 | | | | | 2 | | | | | 1 | | | 0 |
| 2 | 1024 | <----- | 1 | | | | | | | | | | 0 | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_A | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

```
-- RAMB16_S1_S1: 16k x 1 Dual-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2
```

```
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
```

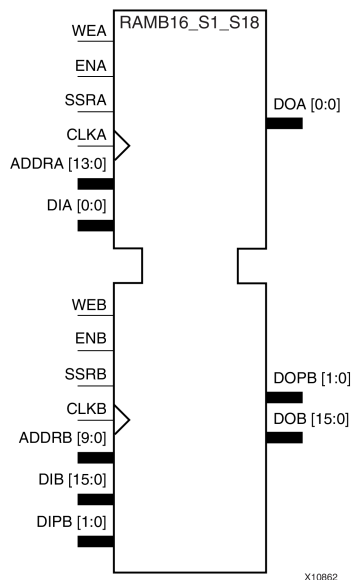
187

Verilog Instantiation Template

Spartan-3E Libraries Guide for HDL Designs

RAMB16_S1_S18

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 18-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------------------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR _A | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

¹WRITE_MODE_A=NO_CHANGE.

²WRITE_MODE_A=READ_FIRST.

³WRITE_MODE_A=WRITE_FIRST.

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
 SRVAL_B=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S1_S18 | 16384 x 1 | - | (13:0) | (0:0) | - | 1024 x 16 | 1024 x 2 | (9:0) | (15:0) | (1:0) |

¹ Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables show address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|--|---|--|--|--|--|---|--|--|---|
| 1 | 2048 | <----- | 3 | | | | | 2 | | | | | 1 | | | 0 |
| 2 | 1024 | <----- | 1 | | | | | | | | | | 0 | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIA | DIPA |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|--------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/Hexidecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/Hexidecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/Hexidecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/Hexidecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL, WARNING_ONLY, "GENERATE_X_ONLY, NONE | "ALL | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> ALL - Warning produced and affected outputs/memory location go unknown (X). WARNING_ONLY - Warning produced and affected outputs/memory retain last value. GENERATE_X_ONLY - No warning. However, affected outputs/memory go unknown (X). NONE - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/Hexidecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_B | Binary/ Hexidecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

[illegible]

UG617 (v 13.2) July 6, 2011

```

INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 1-bit Data Output
  DOB => DOB,      -- Port B 16-bit Data Output
  DOPB => DOPB,    -- Port B 2-bit Parity Output
  ADDRA => ADDRA,  -- Port A 14-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 1-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_S1_S18_inst instantiation

```

Verilog Instantiation Template

```

// RAMB16_S1_S18: 16k/1k x 1/16 + 0/2 Parity bits Dual-Port RAM
//                      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

RAMB16_S1_S18 #(
  .INIT_A(1'b0),      // 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(1'b0),     // 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
  // Port A Address 0 to 4095, Port B Address 0 to 255
  .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),
  // Port A Address 4096 to 8191, Port B Address 256 to 511
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

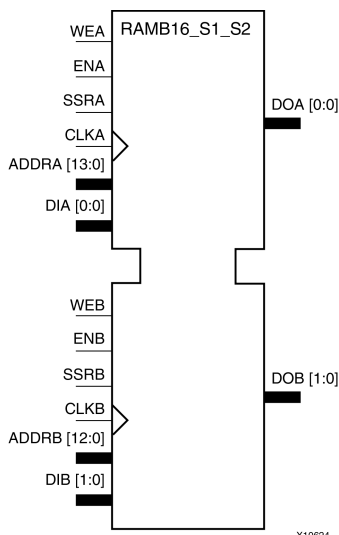
200 www.xilinx.com Spartan-3E Libraries Guide for HDL Designs UG617 (v 13.2) July 6, 2011

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

RAMB16_S1_S2

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 2-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

| Inputs | | | | | | | | Outputs | | |
|--|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| <p>RAM(addr)=RAM contents at address ADDR.</p> <p>data=RAM input data.</p> <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE_A=NO_CHANGE.</p> <p>²WRITE_MODE_A=READ_FIRST.</p> <p>³WRITE_MODE_A=WRITE_FIRST.</p> | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
 SRVAL_B=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S1_S2 | 16384 x 1 | - | (13:0) | (0:0) | - | 8192 x 2 | - | (12:0) | (1:0) | - |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables show address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|--|---|--|--|--|---|--|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 | |
| 2 | 1024 | <----- | 1 | | | | | | | | 0 | | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan-3A, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|--------|--|---------------|---|
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S2: 16k/8k x 1/2 Dual-Port RAM
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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
  ADDR8 => ADDR8,   -- 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,       -- 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_S2_inst instantiation
```

Verilog Instantiation Template

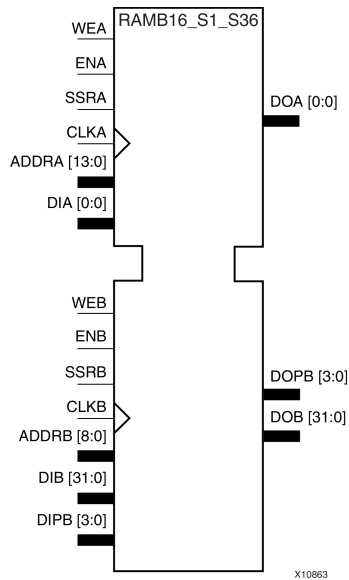
[illegible]

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

RAMB16_S1_S36

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 36-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------------------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR _A | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

¹WRITE_MODE_A=NO_CHANGE.

²WRITE_MODE_A=READ_FIRST.

³WRITE_MODE_A=WRITE_FIRST.

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S1_S36 | 16384 x 1 | - | (13:0) | (0:0) | - | 512 x 32 | 512 x 4 | (8:0) | (31:0) | (3:0) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables show address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|-------|---|--|--|--|--|---|--|--|--|--|---|--|--|---|
| 1 | 2048 | <---- | 3 | | | | | 2 | | | | | 1 | | | 0 |
| 2 | 1024 | <---- | 1 | | | | | | | | | | 0 | | | |
| 4 | 512 | <---- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIA | DIPA |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|--------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S36: 16k/512 x 1/32 + 0/4 Parity bits Dual-Port RAM
--
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```

221

```
-- 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
  ADDR_B => ADDR_B, -- 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 Instantiation Template

```
// RAMB16_S1_S36: 16k/512 x 1/32 + 0/4 Parity bits Dual-Port RAM
//
// Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

RAMB16_S1_S36 #(
  .INIT_A(1'b0),      // 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(1'b0),     // 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"

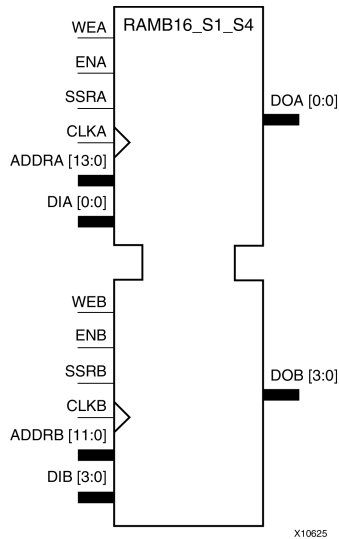
  // The following INIT_xx declarations specify the initial contents of the RAM
  // Port A Address 0 to 4095, Port B Address 0 to 127
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .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),
  // Port A Address 4096 to 8191, Port B Address 128 to 255
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
```


For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

RAMB16_S1_S4

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 4-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------------------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR _A | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

¹WRITE_MODE_A=NO_CHANGE.

²WRITE_MODE_A=READ_FIRST.

³WRITE_MODE_A=WRITE_FIRST.

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S1_S4 | 16384 x 1 | - | (13:0) | (0:0) | - | 4096 x 4 | - | (11:0) | (3:0) | - |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|--|---|--|--|--|---|--|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 | |
| 2 | 1024 | <----- | 1 | | | | | | | | 0 | | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIA | DIPA |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", NONE | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_A | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> • WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. • READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. • NO_CHANGE - keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> • WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. • READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. • NO_CHANGE - keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S4: 16k/4k x 1/4 Dual-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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
ADDRB => 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 Instantiation Template

```

// RAMB16_S1_S4: 16k/4k x 1/4 Dual-Port RAM
//          Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

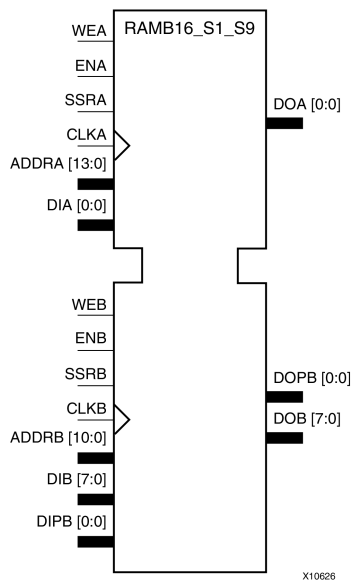
RAMB16_S1_S4 #(
    .INIT_A(1'b0), // Value of output RAM registers on Port A at startup
    .INIT_B(4'h0), // Value of output RAM registers on Port B at startup
    .SRVAL_A(1'b0), // Port A output value upon SSR assertion
    .SRVAL_B(4'h0), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // 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),
    .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),
    // Port A Address 4096 to 8191, Port B Address 1024 to 2047
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),

```


RAMB16_S1_S9

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 9-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------------------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR _A | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.
 SRVAL_A=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_A=NO_CHANGE.
²WRITE_MODE_A=READ_FIRST.
³WRITE_MODE_A=WRITE_FIRST.

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
 SRVAL_B=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S1_S9 | 16384 x 1 | - | (13:0) | (0:0) | - | 2048 x 8 | 2048 x 1 | (10:0) | (7:0) | (0:0) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|--|---|--|--|--|--|---|--|--|---|
| 1 | 2048 | <----- | 3 | | | | | 2 | | | | | 1 | | | 0 |
| 2 | 1024 | <----- | 1 | | | | | | | | | | 0 | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|--------|--|---------------|---|
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S9: 16k/2k x 1/8 + 0/1 Parity bit Dual-Port RAM
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```

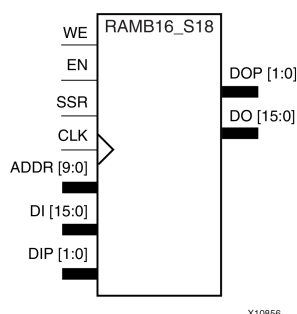
244 www.xilinx.com Spartan-3E Libraries Guide for HDL Designs UG617 (v 13.2) July 6, 2011

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAMB16_S18

Primitive: 16K-bit Data + 2K-bit Parity Memory, Single-Port Synchronous Block RAM with 18-bit Port



Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

| Data Cells | | Parity Cells | | | | |
|------------|-------|--------------|-------|-------------|----------|------------|
| Depth | Width | Depth | Width | Address Bus | Data Bus | Parity Bus |
| 1024 | 16 | 1024 | 2 | (9:0) | (15:0) | (1:0) |

The enable EN pin controls read, write, and reset. When EN is Low, no data is written and the outputs (DO and DOP) retain the last state. When EN is High and reset (SSR) is High, DO and DOP are set to SRVAL during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI and DIP. When EN is High and WE is Low, the data stored in the RAM address (ADDR) is read during the Low-to-High clock transition. By default, WRITE_MODE=WRITE_FIRST, when EN and WE are High, the data on the data inputs (DI and DIP) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data outputs (DO and DOP) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Logic Table

| Inputs | | | | | | | | Outputs | | | |
|--------|----|-----|----|-----|------|------|-------|-------------------------------|-------------------------------|------------------|-------------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT | INIT | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL | SRVAL | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ RAM | No Change ¹ RAM | RAM(addr) =>data | RAM(addr) =>pdata |

| Inputs | | | | | | | | Outputs | | | |
|---|----|-----|----|-----|------|----|-----|--|---|--------------|------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| | | | | | | | | (addr) ² data ³ | (addr) ² pdata ³ | | |
| <p>GSR=Global Set Reset signal</p> <p>INIT=Value specified by the INIT attribute for data memory. Default is all zeros.</p> <p>SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.</p> <p>addr=RAM address.</p> <p>RAM(addr)=RAM contents at address ADDR.</p> <p>data=RAM input data.</p> <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE=NO_CHANGE</p> <p>²WRITE_MODE=READ_FIRST</p> <p>³WRITE_MODE=WRITE_FIRST</p> | | | | | | | | | | | |

Initialization

Initializing Memory Contents

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

Any INIT_xx or INITP_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

Selecting Write Mode

The WRITE_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE_MODE is set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE_MODE to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE_MODE to NO_CHANGE to have the input written to memory without changing the output.

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-------------------------|------------------------|---|---------------|---|
| INIT | Binary/ Hexadecimal | Any Hex Value | All zeros | Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM. |
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SRVAL | Binary/ Hexadecimal | Any Hex Value | All zeros | Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM. |
| WRITE_MODE | String | "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S18: 1k x 16 + 2 Parity bits Single-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

RAMB16_S18_inst : RAMB16_S18
generic map (
  INIT => X"00000", -- Value of output RAM registers at startup
  SRVAL => X"00000", -- Ouput value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE

```



```

-- 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 Instantiation Template

```

// RAMB16_S18: 1k x 16 + 2 Parity bits Single-Port RAM
//          Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

RAMB16_S18 #(
    .INIT(18'h000000), // 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),
    .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),

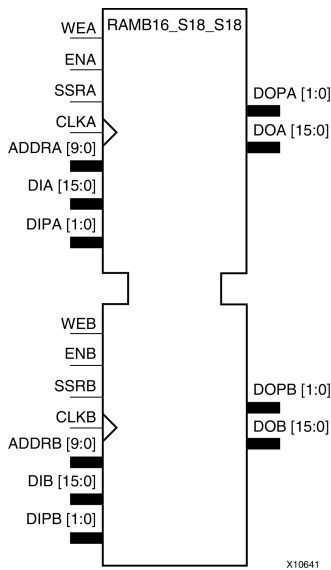
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAMB16_S18_S18

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

| Inputs | | | | | | | | Outputs | | |
|--|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| SRVAL_A=register value. addr=RAM address. RAM(addr)=RAM contents at address ADDR. data=RAM input data. pdata=RAM parity data. ¹ WRITE_MODE_A=NO_CHANGE. ² WRITE_MODE_A=READ_FIRST. ³ WRITE_MODE_A=WRITE_FIRST. | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| 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) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables show address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|--|---|--|--|--|--|---|--|--|---|
| 1 | 2048 | <----- | 3 | | | | | 2 | | | | | 1 | | | 0 |
| 2 | 1024 | <----- | 1 | | | | | | | | | | 0 | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A or above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIA | DIPA |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|--------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S18_S18: 1k x 16 + 2 Parity bits Dual-Port RAM
--                      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```

261

```
-- Address 768 to 1023
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  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 => ADDRA,  -- Port A 10-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 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
```

Verilog Instantiation Template

```
// RAMB16_S18_S18: 1k x 16 + 2 Parity bits Dual-Port RAM
//                      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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),
```



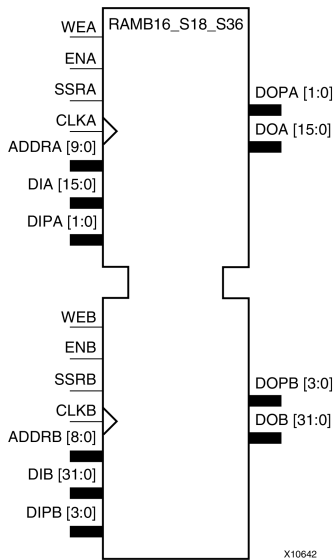
```
// End of RAMB16_S18_S18_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAMB16_S18_S36

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit and 36-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

| Inputs | | | | | | | | Outputs | | |
|---|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| <p>addr=RAM address.</p> <p>RAM(addr)=RAM contents at address ADDR.</p> <p>data=RAM input data.</p> <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE_A=NO_CHANGE.</p> <p>²WRITE_MODE_A=READ_FIRST.</p> <p>³WRITE_MODE_A=WRITE_FIRST.</p> | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| 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) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|-----|---|--|--|--|--|---|--|--|--|--|---|--|--|---|
| 1 | 2048 | <-- | 3 | | | | | 2 | | | | | 1 | | | 0 |
| 2 | 1024 | <-- | 1 | | | | | | | | | | 0 | | | |
| 4 | 512 | <-- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIA | DIPA |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|--------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S18_S36: 1k/512 x 16/32 + 2/4 Parity bits Dual-Port RAM
--               Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```


273

```
-- 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
  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
```

Verilog Instantiation Template

```
// RAMB16_S18_S36: 1k/512 x 16/32 + 2/4 Parity bits Dual-Port RAM
//                      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

RAMB16_S18_S36 #(
  .INIT_A(18'h00000), // Value of output RAM registers on Port A at startup
  .INIT_B(36'h000000000), // Value of output RAM registers on Port B at startup
  .SRVAL_A(18'h00000), // Port A output value upon SSR assertion
  .SRVAL_B(36'h000000000), // 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 255, Port B Address 0 to 127
  .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),
  // Port A Address 256 to 511, Port B Address 128 to 255
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

275

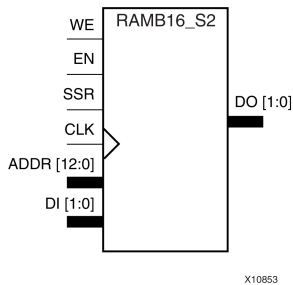
```
// End of RAMB16_S18_S36_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAMB16_S2

Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 2-bit Port



Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

| Data Cells | | Parity Cells | | | | |
|------------|-------|--------------|-------|-------------|----------|------------|
| Depth | Width | Depth | Width | Address Bus | Data Bus | Parity Bus |
| 8192 | 2 | - | - | (12:0) | (1:0) | - |

The enable EN pin controls read, write, and reset. When EN is Low, no data is written and the outputs (DO and DOP) retain the last state. When EN is High and reset (SSR) is High, DO and DOP are set to SRVAL during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI and DIP. When EN is High and WE is Low, the data stored in the RAM address (ADDR) is read during the Low-to-High clock transition. By default, WRITE_MODE=WRITE_FIRST, when EN and WE are High, the data on the data inputs (DI and DIP) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data outputs (DO and DOP) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Logic Table

| Inputs | | | | | | | | Outputs | | | |
|--------|----|-----|----|-----|------|------|-------|-------------------------------|-------------------------------|------------------|-------------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT | INIT | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL | SRVAL | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ RAM | No Change ¹ RAM | RAM(addr) =>data | RAM(addr) =>pdata |

| Inputs | | | | | | | | Outputs | | | |
|---|----|-----|----|-----|------|----|-----|--|---|--------------|------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| | | | | | | | | (addr) ² data ³ | (addr) ² pdata ³ | | |
| <p>GSR=Global Set Reset signal</p> <p>INIT=Value specified by the INIT attribute for data memory. Default is all zeros.</p> <p>SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.</p> <p>addr=RAM address.</p> <p>RAM(addr)=RAM contents at address ADDR.</p> <p>data=RAM input data.</p> <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE=NO_CHANGE</p> <p>²WRITE_MODE=READ_FIRST</p> <p>³WRITE_MODE=WRITE_FIRST</p> | | | | | | | | | | | |

Initialization

Initializing Memory Contents

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

Any INIT_xx or INITP_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

Selecting Write Mode

The WRITE_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE_MODE is set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE_MODE to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE_MODE to NO_CHANGE to have the input written to memory without changing the output.

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-------------------------|------------------------|---|---------------|---|
| INIT | Binary/ Hexadecimal | Any Hex Value | All zeros | Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM. |
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SRVAL | Binary/ Hexadecimal | Any Hex Value | All zeros | Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM. |
| WRITE_MODE | String | "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAMB16_S2: 8k x 2 Single-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

RAMB16_S2_inst : RAMB16_S2
generic map (
  INIT => X"0", -- Value of output RAM registers at startup
  SRVAL => X"0", -- Ouput value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
```

280 www.xilinx.com Spartan-3E Libraries Guide for HDL Designs UG617 (v 13.2) July 6, 2011


```

    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 Instantiation Template

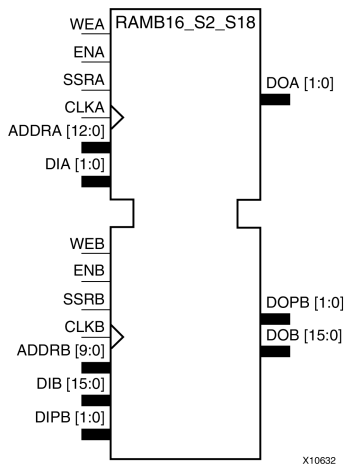
[illegible]

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

RAMB16_S2_S18

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 18-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

| Inputs | | | | | | | | Outputs | | |
|---|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| data=RAM input data. pdata=RAM parity data. ¹ WRITE_MODE_A=NO_CHANGE. ² WRITE_MODE_A=READ_FIRST. ³ WRITE_MODE_A=WRITE_FIRST. | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S2_S18 | 8192 x 2 | - | (12:0) | (1:0) | - | 1024 x 16 | 1024 x 2 | (9:0) | (15:0) | (1:0) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|--|---|--|--|--|---|--|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 | |
| 2 | 1024 | <----- | 1 | | | | | | | | 0 | | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectIO™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|--------|--|---------------|---|
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S18: 8k/1k x 2/16 + 0/2 Parity bit Dual-Port RAM
-- Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```

291

```

DOPB => DOPB,      -- Port B 2-bit Parity Output
ADDRA => ADDRA,    -- Port A 13-bit Address Input
ADDRB => ADDRBR,    -- 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

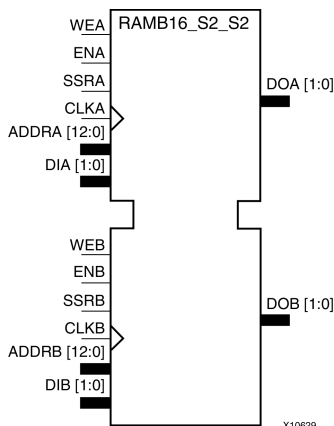
```

Verilog Instantiation Template

[illegible]

RAMB16_S2_S2

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

| Inputs | | | | | | | | Outputs | | |
|---|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| ¹ WRITE_MODE_A=NO_CHANGE. ² WRITE_MODE_A=READ_FIRST. ³ WRITE_MODE_A=WRITE_FIRST. | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
 SRVAL_B=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S2_S2 | 8192 x 2 | - | (12:0) | (1:0) | - | 8192 x 2 | - | (12:0) | (1:0) | - |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|--|---|--|--|--|---|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 |
| 2 | 1024 | <----- | 1 | | | | | | | | 0 | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexidecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexidecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexidecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexidecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/ Hexidecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|--|
| SRVAL_B | Binary/ Hexidecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S2: 8k x 2 Dual-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 ouput value upon SSR assertion
  SRVAL_B => X"0", -- Port B ouput 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",

```

302 www.xilinx.com Spartan-3E Libraries Guide for HDL Designs UG617 (v 13.2) July 6, 2011

```

        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 Instantiation Template

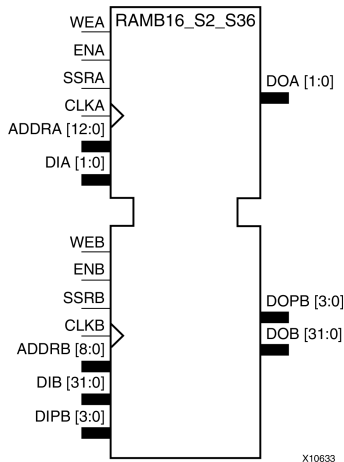
[illegible]

For More Information

- UG617 (v 13.2) July 6, 2011

RAMB16_S2_S36

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 36-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

| Inputs | | | | | | | | Outputs | | |
|---|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| data=RAM input data. pdata=RAM parity data. ¹ WRITE_MODE_A=NO_CHANGE. ² WRITE_MODE_A=READ_FIRST. ³ WRITE_MODE_A=WRITE_FIRST. | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S2_S36 | 8192 x 2 | - | (12:0) | (1:0) | - | 512 x 32 | 512 x 4 | (8:0) | (31:0) | (3:0) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|--|---|--|--|--|---|--|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 | |
| 2 | 1024 | <----- | 1 | | | | | | | | 0 | | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |

313

```

DOPB => DOPB,      -- Port B 4-bit Parity Output
ADDRA => ADDR_A,   -- Port A 13-bit Address Input
ADDRB => ADDR_B,   -- Port B 9-bit Address Input
CLKA => CLK_A,     -- Port A Clock
CLKB => CLK_B,     -- 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

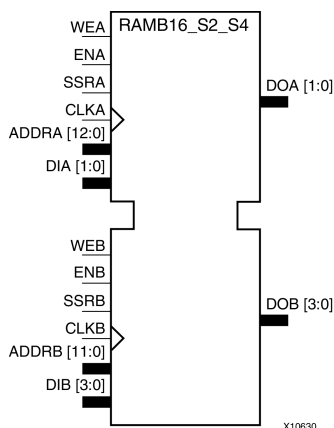
```

Verilog Instantiation Template

[illegible]

RAMB16_S2_S4

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 4-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

| Inputs | | | | | | | | Outputs | | |
|---|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE_A=NO_CHANGE.</p> <p>²WRITE_MODE_A=READ_FIRST.</p> <p>³WRITE_MODE_A=WRITE_FIRST.</p> | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
 SRVAL_B=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S2_S4 | 8192 x 2 | - | (12:0) | (1:0) | - | 4096 x 4 | - | (11:0) | (3:0) | - |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|--|---|--|--|--|---|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 |
| 2 | 1024 | <----- | 1 | | | | | | | | 0 | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|--|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexidecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexidecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexidecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexidecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "NONE", "WARNING", or "GENERATE_X_ONLY" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> • ALL - Warning produced and affected outputs/memory location go unknown (X). • WARNING_ONLY - Warning produced and affected outputs/memory retain last value. • GENERATE_X_ONLY - No warning. However, affected outputs/memory go unknown (X). • NONE - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/ Hexidecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |
| SRVAL_B | Binary/ Hexidecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|--------|--|---------------|---|
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S4: 8k/4k x 2/4 Dual-Port RAM
-- Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 ouput value upon SSR assertion
  SRVAL_B => X"0", -- Port B ouput 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",

```

324 www.xilinx.com Spartan-3E Libraries Guide for HDL Designs UG617 (v 13.2) July 6, 2011

Verilog Instantiation Template

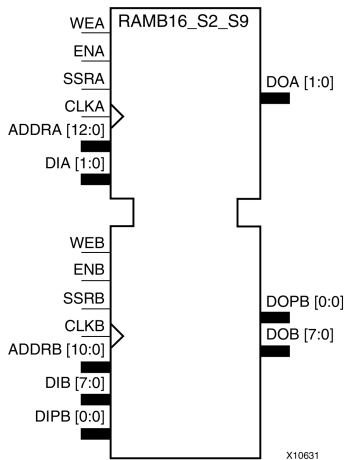
Spartan-3E Libraries Guide for HDL Designs
UG617 (v 13.2) July 6, 2011

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

RAMB16_S2_S9

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 9-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

| Inputs | | | | | | | | Outputs | | |
|---|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| <p>data=RAM input data.</p> <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE_A=NO_CHANGE.</p> <p>²WRITE_MODE_A=READ_FIRST.</p> <p>³WRITE_MODE_A=WRITE_FIRST.</p> | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S2_S9 | 8192 x 2 | - | (12:0) | (1:0) | - | 2048 x 8 | 2048 x 1 | (10:0) | (7:0) | (0:0) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|--|---|--|--|--|---|--|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 | |
| 2 | 1024 | <----- | 1 | | | | | | | | 0 | | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan®-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|--|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexidecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexidecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexidecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexidecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "NONE", "WARNING", or "GENERATE_X_ONLY" | "ALL" | Specifies the behavior during simulation in the event of a data collision (data being read or written to the same address from both ports of the RAM simultaneously). <ul style="list-style-type: none"> ALL - Issues a warning to simulator console and generates an X or all unknown data due to the collision. This is the recommended setting. WARNING - Generates a warning only. GENERATE_X_ONLY - Generates an X for unknown data but does not output the occurrence to the simulation console. NONE - completely ignores the error. It is suggested to only change this attribute if you can ensure the data generated during a collision is discarded. |
| SRVAL_A | Binary/ Hexidecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|---|---------------|---|
| SRVAL_B | Binary/ Hexidecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> • WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. • READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. • NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> • WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. • READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. • NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S9: 8k/2k x 2/8 + 0/1 Parity bit Dual-Port RAM
--               Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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

```

335

```

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
  ADDR_B => ADDR_B, -- 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 Instantiation Template

```

// RAMB16_S2_S9: 8k/2k x 2/8 + 0/1 Parity bit Dual-Port RAM
//          Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

RAMB16_S2_S9 #(
  .INIT_A(2'b00), // Value of output RAM registers on Port A at startup
  .INIT_B(9'h000), // Value of output RAM registers on Port B at startup
  .SRVAL_A(2'b00), // Port A output value upon SSR assertion
  .SRVAL_B(9'h000), // 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 511
  .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),
  // Port A Address 2048 to 4095, Port B Address 512 to 1023
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),

```


337

```
.SSRB(SSRB),      // Port B Synchronous Set/Reset Input
.WEA(WEA),        // Port A Write Enable Input
.WEB(WEB)         // Port B Write Enable Input
);

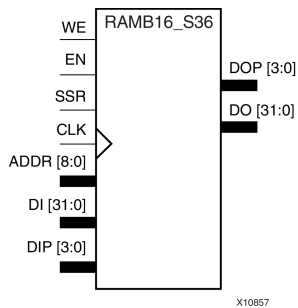
// End of RAMB16_S2_S9_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAMB16_S36

Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 36-bit Port



Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

| Data Cells | | Parity Cells | | | | |
|------------|-------|--------------|-------|-------------|----------|------------|
| Depth | Width | Depth | Width | Address Bus | Data Bus | Parity Bus |
| 512 | 32 | 512 | 4 | (8:0) | (31:0) | (3:0) |

The enable EN pin controls read, write, and reset. When EN is Low, no data is written and the outputs (DO and DOP) retain the last state. When EN is High and reset (SSR) is High, DO and DOP are set to SRVAL during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI and DIP. When EN is High and WE is Low, the data stored in the RAM address (ADDR) is read during the Low-to-High clock transition. By default, WRITE_MODE=WRITE_FIRST, when EN and WE are High, the data on the data inputs (DI and DIP) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data outputs (DO and DOP) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Logic Table

| Inputs | | | | | | | | Outputs | | | |
|--------|----|-----|----|-----|------|------|-------|-------------------------------|-------------------------------|------------------|-------------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT | INIT | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL | SRVAL | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ RAM | No Change ¹ RAM | RAM(addr) =>data | RAM(addr) =>pdata |

| Inputs | | | | | | | | Outputs | | | |
|--------|----|-----|----|-----|------|----|-----|--|---|--------------|------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| | | | | | | | | (addr) ² data ³ | (addr) ² pdata ³ | | |

GSR=Global Set Reset signal
 INIT=Value specified by the INIT attribute for data memory. Default is all zeros.
 SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE=NO_CHANGE
²WRITE_MODE=READ_FIRST
³WRITE_MODE=WRITE_FIRST

Initialization

Initializing Memory Contents

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

Any INIT_xx or INITP_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

Selecting Write Mode

The WRITE_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE_MODE is set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE_MODE to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE_MODE to NO_CHANGE to have the input written to memory without changing the output.

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-------------------------|------------------------|---|---------------|---|
| INIT | Binary/ Hexadecimal | Any Hex Value | All zeros | Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM. |
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SRVAL | Binary/ Hexadecimal | Any Hex Value | All zeros | Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM. |
| WRITE_MODE | String | "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAMB16_S36: 512 x 32 + 4 Parity bits Single-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

RAMB16_S36_inst : RAMB16_S36
generic map (
  INIT => X"000000000", -- Value of output RAM registers at startup
  SRVAL => X"000000000", -- Ouput value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
```

UG617 (v 13.2) July 6, 2011

```

-- 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 Instantiation Template

```

// RAMB16_S36: 512 x 32 + 4 Parity bits Single-Port RAM
//          Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

RAMB16_S36 #(
    .INIT(36'h000000000), // Value of output RAM registers at startup
    .SRVAL(36'h000000000), // 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),
    .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),

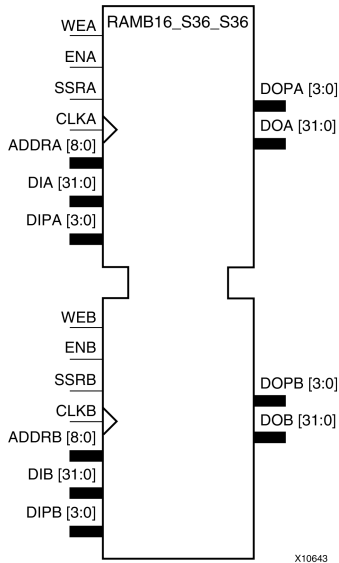
```

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

RAMB16_S36_S36

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with Two 36-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------------------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR _A | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

¹WRITE_MODE_A=NO_CHANGE.

²WRITE_MODE_A=READ_FIRST.

³WRITE_MODE_A=WRITE_FIRST.

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
 SRVAL_B=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| 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) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables show address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|------|---|--|--|--|--|---|--|--|--|--|---|--|--|---|
| 1 | 2048 | <--- | 3 | | | | | 2 | | | | | 1 | | | 0 |
| 2 | 1024 | <--- | 1 | | | | | | | | | | 0 | | | |
| 4 | 512 | <--- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIA | DIPA |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|--------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S36_S36: 512 x 32 + 4 Parity bits Dual-Port RAM
--               Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```


353

```
-- 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
  ADDRb => ADDRb,  -- 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,      -- 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_S36_S36_inst instantiation
```

Verilog Instantiation Template

```
// RAMB16_S36_S36: 512 x 32 + 4 Parity bits Dual-Port RAM
//                      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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),
```



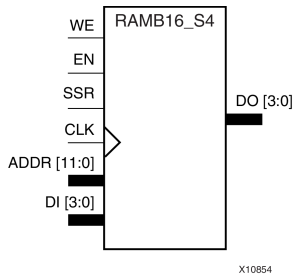
```
// End of RAMB16_S36_S36_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAMB16_S4

Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 4-bit Port



Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

| Data Cells | | Parity Cells | | | | |
|------------|-------|--------------|-------|-------------|----------|------------|
| Depth | Width | Depth | Width | Address Bus | Data Bus | Parity Bus |
| 4096 | 4 | - | - | (11:0) | (3:0) | - |

The enable EN pin controls read, write, and reset. When EN is Low, no data is written and the outputs (DO and DOP) retain the last state. When EN is High and reset (SSR) is High, DO and DOP are set to SRVAL during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI and DIP. When EN is High and WE is Low, the data stored in the RAM address (ADDR) is read during the Low-to-High clock transition. By default, WRITE_MODE=WRITE_FIRST, when EN and WE are High, the data on the data inputs (DI and DIP) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data outputs (DO and DOP) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Logic Table

| Inputs | | | | | | | | Outputs | | | |
|--------|----|-----|----|-----|------|------|-------|-------------------------------|-------------------------------|-------------------|--------------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT | INIT | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL | SRVAL | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ RAM | No Change ¹ RAM | RAM(addr) => data | RAM(addr) => pdata |

| Inputs | | | | | | | | Outputs | | | |
|---|----|-----|----|-----|------|----|-----|--|---|--------------|------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| | | | | | | | | (addr) ² data ³ | (addr) ² pdata ³ | | |
| <p>GSR=Global Set Reset signal</p> <p>INIT=Value specified by the INIT attribute for data memory. Default is all zeros.</p> <p>SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.</p> <p>addr=RAM address.</p> <p>RAM(addr)=RAM contents at address ADDR.</p> <p>data=RAM input data.</p> <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE=NO_CHANGE</p> <p>²WRITE_MODE=READ_FIRST</p> <p>³WRITE_MODE=WRITE_FIRST</p> | | | | | | | | | | | |

Initialization

Initializing Memory Contents

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

Any INIT_xx or INITP_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

Selecting Write Mode

The WRITE_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE_MODE is set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE_MODE to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE_MODE to NO_CHANGE to have the input written to memory without changing the output.

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-------------------------|------------------------|---|---------------|---|
| INIT | Binary/ Hexadecimal | Any Hex Value | All zeros | Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM. |
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SRVAL | Binary/ Hexadecimal | Any Hex Value | All zeros | Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM. |
| WRITE_MODE | String | "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and wont update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- RAMB16_S4: 4k x 4 Single-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

RAMB16_S4_inst : RAMB16_S4
generic map (
  INIT => X"0", -- Value of output RAM registers at startup
  SRVAL => X"0", -- Ouput value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
```

360 www.xilinx.com Spartan-3E Libraries Guide for HDL Designs UG617 (v 13.2) July 6, 2011


```

    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 Instantiation Template

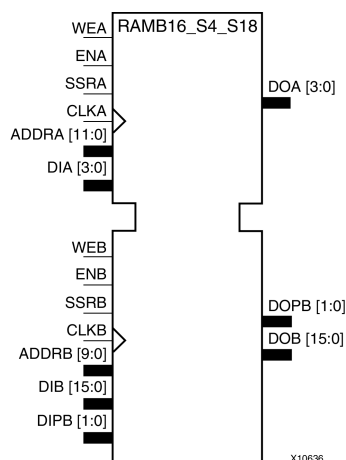
[illegible]

For More Information

- UG617 (v 13.2) July 6, 2011

RAMB16_S4_S18

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 18-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------------------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR _A | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

¹WRITE_MODE_A=NO_CHANGE.

²WRITE_MODE_A=READ_FIRST.

³WRITE_MODE_A=WRITE_FIRST.

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S4_S18 | 4096 x 4 | - | (11:0) | (3:0) | - | 1024 x 16 | 1024 x 2 | (9:0) | (15:0) | (1:0) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|--|---|--|--|--|--|---|--|--|---|
| 1 | 2048 | <----- | 3 | | | | | 2 | | | | | 1 | | | 0 |
| 2 | 1024 | <----- | 1 | | | | | | | | | | 0 | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIA | DIPA |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|--------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S4_S18: 4k/1k x 4/16 + 0/2 Parity bits Dual-Port RAM
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```

371

```
-- 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
  ADDR_B => ADDR_B, -- 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
```

Verilog Instantiation Template

```
// RAMB16_S4_S18: 4k/1k x 4/16 + 0/2 Parity bits Dual-Port RAM
//          Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

RAMB16_S4_S18 #(
  .INIT_A(4'h0),      // 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(4'h0),     // 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
  // Port A Address 0 to 1023, Port B Address 0 to 255
  .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),
  // Port A Address 1024 to 2047, Port B Address 256 to 511
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

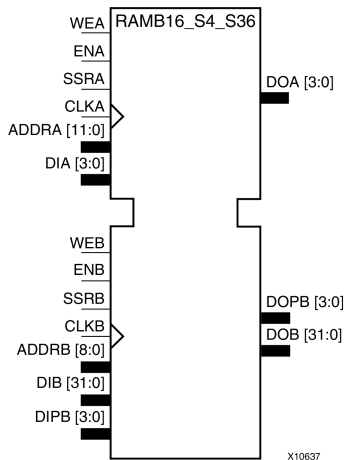
373

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

RAMB16_S4_S36

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 36-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

| Inputs | | | | | | | | Outputs | | |
|---|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| data=RAM input data. pdata=RAM parity data. ¹ WRITE_MODE_A=NO_CHANGE. ² WRITE_MODE_A=READ_FIRST. ³ WRITE_MODE_A=WRITE_FIRST. | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S4_S36 | 4096 x 4 | - | (11:0) | (3:0) | - | 512 x 32 | 512 x 4 | (8:0) | (31:0) | (3:0) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|--|---|--|--|--|---|--|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 | |
| 2 | 1024 | <----- | 1 | | | | | | | | 0 | | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|--------|--|---------------|---|
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S4_S36: 4k/512 x 4/32 + 0/4 Parity bits Dual-Port RAM
-- Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```

383

```

DOPB => DOPB,      -- Port B 4-bit Parity Output
ADDRA => ADDR_A,    -- Port A 12-bit Address Input
ADDRB => ADDR_B,    -- Port B 9-bit Address Input
CLKA => CLK_A,      -- Port A Clock
CLKB => CLK_B,      -- 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 Instantiation Template

[illegible]


```
.INIT_22(256'h0000000000000000000000000000000000000000000000000),  
.INIT_23(256'h0000000000000000000000000000000000000000000000000),  
.INIT_24(256'h0000000000000000000000000000000000000000000000000),  
.INIT_25(256'h0000000000000000000000000000000000000000000000000),  
.INIT_26(256'h0000000000000000000000000000000000000000000000000),  
.INIT_27(256'h0000000000000000000000000000000000000000000000000),  
.INIT_28(256'h0000000000000000000000000000000000000000000000000),  
.INIT_29(256'h0000000000000000000000000000000000000000000000000),  
.INIT_2A(256'h0000000000000000000000000000000000000000000000000),  
.INIT_2B(256'h0000000000000000000000000000000000000000000000000),  
.INIT_2C(256'h0000000000000000000000000000000000000000000000000),  
.INIT_2D(256'h0000000000000000000000000000000000000000000000000),  
.INIT_2E(256'h0000000000000000000000000000000000000000000000000),  
.INIT_2F(256'h0000000000000000000000000000000000000000000000000),  
// Port A Address 3072 to 4095, Port B Address 384 to 511  
.INIT_30(256'h0000000000000000000000000000000000000000000000000),  
.INIT_31(256'h0000000000000000000000000000000000000000000000000),  
.INIT_32(256'h0000000000000000000000000000000000000000000000000),  
.INIT_33(256'h0000000000000000000000000000000000000000000000000),  
.INIT_34(256'h0000000000000000000000000000000000000000000000000),  
.INIT_35(256'h0000000000000000000000000000000000000000000000000),  
.INIT_36(256'h0000000000000000000000000000000000000000000000000),  
.INIT_37(256'h0000000000000000000000000000000000000000000000000),  
.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 A Address 0 to 1023, Port B Address 0 to 127  
.INITP_00(256'h0000000000000000000000000000000000000000000000000),  
.INITP_01(256'h0000000000000000000000000000000000000000000000000),  
// Port A Address 1024 to 2047, Port B Address 128 to 255  
.INITP_02(256'h0000000000000000000000000000000000000000000000000),  
.INITP_03(256'h0000000000000000000000000000000000000000000000000),  
// Port A Address 2048 to 3071, Port B Address 256 to 383  
.INITP_04(256'h0000000000000000000000000000000000000000000000000),  
.INITP_05(256'h0000000000000000000000000000000000000000000000000),  
// Port A Address 3072 to 4095, Port B Address 384 to 511  
.INITP_06(256'h0000000000000000000000000000000000000000000000000),  
.INITP_07(256'h0000000000000000000000000000000000000000000000000),  
)  
RAMBL6_S4_S36_inst (  
    .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(CLK_A),        // Port A Clock  
    .CLKB(CLKB_B),      // Port B Clock  
    .DIA(DI_A),         // Port A 4-bit Data Input  
    .DIB(DI_B),         // Port B 32-bit Data Input  
    .DIPB(DI_PB),       // Port-B 4-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  
    .SRRB(SSR_B),       // Port B Synchronous Set/Reset Input  
    .WEA(WE_A),         // Port A Write Enable Input  
    .WEB(WE_B)          // Port B Write Enable Input  
);
```

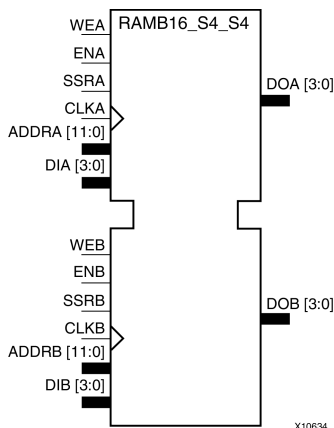
// End of RAMBL6 S4 S36 inst instantiation

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

RAMB16_S4_S4

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

| Inputs | | | | | | | | Outputs | | |
|--|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| ¹ WRITE_MODE_A=NO_CHANGE. | | | | | | | | | | |
| ² WRITE_MODE_A=READ_FIRST. | | | | | | | | | | |
| ³ WRITE_MODE_A=WRITE_FIRST. | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
 SRVAL_B=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S4_S4 | 4096 x 4 | - | (11:0) | (3:0) | - | 4096 x 4 | - | (11:0) | (3:0) | - |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|---|---|--|--|--|---|--|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 | |
| 2 | 1024 | <----- | 1 | | | | | | | 0 | | | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|--------|--|---------------|---|
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S4_S4: 4k x 4 Dual-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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
  ADDRb => ADDRb,   -- 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
);

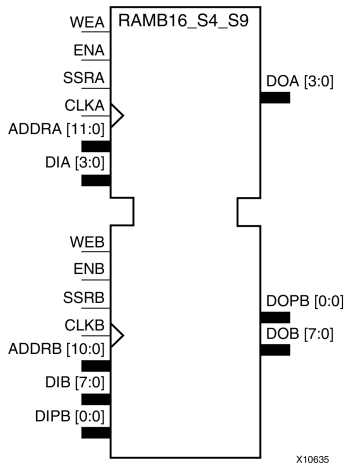
```


For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAMB16_S4_S9

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 9-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.
SRVAL_A=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.

| Inputs | | | | | | | | Outputs | | |
|---|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| data=RAM input data. pdata=RAM parity data. ¹ WRITE_MODE_A=NO_CHANGE. ² WRITE_MODE_A=READ_FIRST. ³ WRITE_MODE_A=WRITE_FIRST. | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
SRVAL_B=register value.
addr=RAM address.
RAM(addr)=RAM contents at address ADDR.
data=RAM input data.
pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| RAMB16_S4_S9 | 4096 x 4 | - | (11:0) | (3:0) | - | 2048 x 8 | 2048 x 1 | (10:0) | (7:0) | (0:0) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|--|---|--|--|--|---|--|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 | |
| 2 | 1024 | <----- | 1 | | | | | | | | 0 | | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|------------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/ Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|--------|--|---------------|---|
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S4_S9: 4k/2k x 4/8 + 0/1 Parity bit Dual-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```

405

```

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

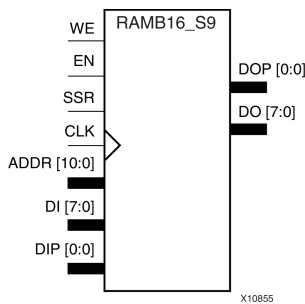
```

Verilog Instantiation Template

[illegible]

RAMB16_S9

Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 9-bit Port



Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

| Data Cells | | Parity Cells | | | | |
|------------|-------|--------------|-------|-------------|----------|------------|
| Depth | Width | Depth | Width | Address Bus | Data Bus | Parity Bus |
| 2048 | 8 | 2048 | 1 | (10:0) | (7:0) | (0:0) |

The enable EN pin controls read, write, and reset. When EN is Low, no data is written and the outputs (DO and DOP) retain the last state. When EN is High and reset (SSR) is High, DO and DOP are set to SRVAL during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI and DIP. When EN is High and WE is Low, the data stored in the RAM address (ADDR) is read during the Low-to-High clock transition. By default, WRITE_MODE=WRITE_FIRST, when EN and WE are High, the data on the data inputs (DI and DIP) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data outputs (DO and DOP) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Logic Table

| Inputs | | | | | | | | Outputs | | | |
|--------|----|-----|----|-----|------|------|-------|-------------------------------|-------------------------------|------------------|-------------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT | INIT | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL | SRVAL | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ RAM | No Change ¹ RAM | RAM(addr) =>data | RAM(addr) =>pdata |

| Inputs | | | | | | | | Outputs | | | |
|---|----|-----|----|-----|------|----|-----|--|---|--------------|------------|
| GSR | EN | SSR | WE | CLK | ADDR | DI | DIP | DO | DOP | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| | | | | | | | | (addr) ² data ³ | (addr) ² pdata ³ | | |
| <p>GSR=Global Set Reset signal</p> <p>INIT=Value specified by the INIT attribute for data memory. Default is all zeros.</p> <p>SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.</p> <p>addr=RAM address.</p> <p>RAM(addr)=RAM contents at address ADDR.</p> <p>data=RAM input data.</p> <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE=NO_CHANGE</p> <p>²WRITE_MODE=READ_FIRST</p> <p>³WRITE_MODE=WRITE_FIRST</p> | | | | | | | | | | | |

Initialization

Initializing Memory Contents

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

Any INIT_xx or INITP_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

Selecting Write Mode

The WRITE_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE_MODE is set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE_MODE to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE_MODE to NO_CHANGE to have the input written to memory without changing the output.

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-------------------------|------------------------|---|---------------|--|
| INIT | Binary/ Hexadecimal | Any Hex Value | All zeros | Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM. |
| INIT_00 to INIT_3F | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INITP_00 to INITP_07 | Binary/ Hexadecimal | Any Hex Value | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SRVAL | Binary/ Hexadecimal | Any Hex Value | All zeros | Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM. |
| WRITE_MODE | String | "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

```
-- RAMB16_S9: 2k x 8 + 1 Parity bit Single-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2
```

```
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
```

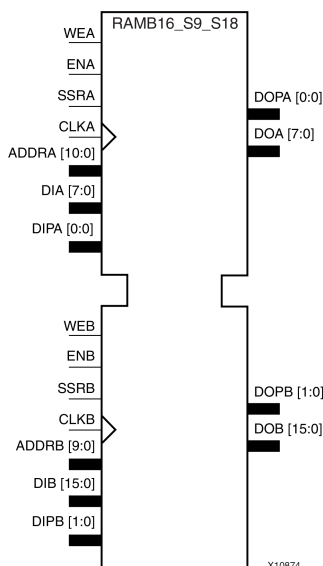
411

Verilog Instantiation Template

412 www.xilinx.com Spartan-3E Libraries Guide for HDL Designs UG617 (v 13.2) July 6, 2011

RAMB16_S9_S18

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 18-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------------------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR _A | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

| Inputs | | | | | | | | Outputs | | |
|--|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| <p>SRVAL_A=register value.</p> <p>addr=RAM address.</p> <p>RAM(addr)=RAM contents at address ADDR.</p> <p>data=RAM input data.</p> <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE_A=NO_CHANGE.</p> <p>²WRITE_MODE_A=READ_FIRST.</p> <p>³WRITE_MODE_A=WRITE_FIRST.</p> | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
 SRVAL_B=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|---------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Component | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| 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) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPB. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables show address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|--|---|--|--|--|--|---|--|--|---|
| 1 | 2048 | <----- | 3 | | | | | 2 | | | | | 1 | | | 0 |
| 2 | 1024 | <----- | 1 | | | | | | | | | | 0 | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIA | DIPA |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|--------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S9_S18: 2k/1k x 8/16 + 1/2 Parity bits Parity bits Dual-Port RAM
--
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```



```

-- 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
  ADDR_B => ADDR_B, -- 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

```

Verilog Instantiation Template

```

// RAMB16_S9_S18: 2k/1k x 8/16 + 1/2 Parity bits Parity bits Dual-Port RAM
//                      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

RAMB16_S9_S18 #(
  .INIT_A(9'h000),      // 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(9'h000),    // 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
  // Port A Address 0 to 511, Port B Address 0 to 255
  .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),
  // Port A Address 512 to 1023, Port B Address 256 to 511
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

424

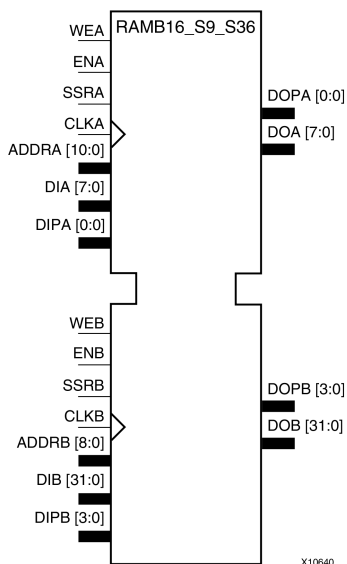

```
// End of RAMB16_S9_S18_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAMB16_S9_S36

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 36-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDR | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

| Inputs | | | | | | | | Outputs | | |
|--|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| <p>SRVAL_A=register value.</p> <p>addr=RAM address.</p> <p>RAM(addr)=RAM contents at address ADDR.</p> <p>data=RAM input data.</p> <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE_A=NO_CHANGE.</p> <p>²WRITE_MODE_A=READ_FIRST.</p> <p>³WRITE_MODE_A=WRITE_FIRST.</p> | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
 SRVAL_B=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| 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) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables shows address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|--|---|--|--|--|--|---|--|--|---|
| 1 | 2048 | <----- | 3 | | | | | 2 | | | | | 1 | | | 0 |
| 2 | 1024 | <----- | 1 | | | | | | | | | | 0 | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIA | DIPA |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|--------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

[illegible]


```

-- 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
  ADDR_B => ADDR_B, -- 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

```

Verilog Instantiation Template

```

// RAMB16_S9_S36: 2k/512 x 8/32 + 1/4 Parity bits Parity bits Dual-Port RAM
//                      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

RAMB16_S9_S36 #(
  .INIT_A(9'h000),      // 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(9'h000),     // 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
  // Port A Address 0 to 511, Port B Address 0 to 127
  .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),
  // Port A Address 512 to 1023, Port B Address 128 to 255
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

436

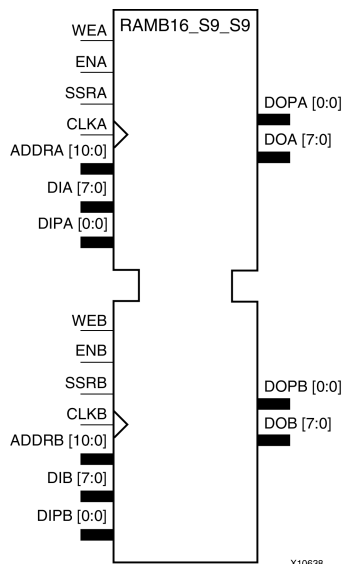
```
// End of RAMB16_S9_S36_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

RAMB16_S9_S9

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit Ports



Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

Logic Table

Truth Table A

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_A | INIT_A | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_A | SRVAL_A | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.

INIT_A=Value specified by the INIT_A attribute for output register. Default is all zeros.

SRVAL_A=register value.

| Inputs | | | | | | | | Outputs | | |
|---|-----|------|-----|------|-------|-----|------|---------|------|------------------------|
| GSR | ENA | SSRA | WEA | CLKA | ADDRA | DIA | DIPA | DOA | DOPA | RAM Contents |
| | | | | | | | | | | Data RAM Parity RAM |
| <p>addr=RAM address.</p> <p>RAM(addr)=RAM contents at address ADDR.</p> <p>data=RAM input data.</p> <p>pdata=RAM parity data.</p> <p>¹WRITE_MODE_A=NO_CHANGE.</p> <p>²WRITE_MODE_A=READ_FIRST.</p> <p>³WRITE_MODE_A=WRITE_FIRST.</p> | | | | | | | | | | |

Truth Table B

| Inputs | | | | | | | | Outputs | | | |
|--------|-----|------|-----|------|-------|------|-------|---|--|---------------------|----------------------|
| GSR | ENB | SSRB | WEB | CLKB | ADDRB | DIB | DIPB | DOB | DOPB | RAM Contents | |
| | | | | | | | | | | Data RAM | Parity RAM |
| 1 | X | X | X | X | X | X | X | INIT_B | INIT_B | No Change | No Change |
| 0 | 0 | X | X | X | X | X | X | No Change | No Change | No Change | No Change |
| 0 | 1 | 1 | 0 | ↑ | X | X | X | SRVAL_B | SRVAL_B | No Change | No Change |
| 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) | No Change | No Change |
| 0 | 1 | 0 | 1 | ↑ | addr | data | pdata | No Change ¹ , RAM(addr) ² , data ³ | No Change ¹ , RAM(addr) ² , pdata ³ | RAM(addr) =>data | RAM(addr) =>pdata |

GSR=Global Set Reset.
 INIT_B=Value specified by the INIT_B attribute for output registers. Default is all zeros.
 SRVAL_B=register value.
 addr=RAM address.
 RAM(addr)=RAM contents at address ADDR.
 data=RAM input data.
 pdata=RAM parity data.
¹WRITE_MODE_B=NO_CHANGE.
²WRITE_MODE_B=READ_FIRST.
³WRITE_MODE_B=WRITE_FIRST.

Port Descriptions

| Port A | | | | | | Port B | | | | |
|----------------|-------------------------|---------------------------|-------------|----------|------------|-------------------------|---------------------------|-------------|----------|------------|
| Design Element | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus | Data Cells ¹ | Parity Cells ¹ | Address Bus | Data Bus | Parity Bus |
| 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) |

¹Depth x Width

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE_MODE_A=WRITE_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE_MODE_B=WRITE_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRB, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

Address Mapping

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)*(Widthport)) -1

End=(ADDRport)*(Widthport)

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

The following tables show address mapping for each port width.

Port Address Mapping for Data

| Data Width | Port Data Addresses | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Port Address Mapping for Parity

| Parity Width | Port Parity Addresses | | | | | | | | | | | | | | | |
|--------------|-----------------------|--------|---|--|--|--|---|--|--|---|---|--|--|--|---|--|
| 1 | 2048 | <----- | 3 | | | | 2 | | | | 1 | | | | 0 | |
| 2 | 1024 | <----- | 1 | | | | | | | 0 | | | | | | |
| 4 | 512 | <----- | 0 | | | | | | | | | | | | | |

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16_Sm_Sn is set by 64 initialization attributes (INIT_00 through INIT_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP_00 through INITP_07) of 64 hex values for a total of 2048 bits.

If any INIT_xx or INITP_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

Initializing the Output Register of a Dual-Port RAMB16

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT_A, INIT_B, SRVAL_A, and SRVAL_B. The INIT_A attribute specifies the output register value at power on for Port A and the INIT_B attribute specifies the value for Port B. You can use the SRVAL_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT_A, INIT_B, SRVAL_A, and SRVAL_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16_S1_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT_A or SRVAL_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT_B or SRVAL_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT_A, INIT_B, SRVAL_A, or SRVAL_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

Write Mode Selection

The WRITE_MODE_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE_MODE_B attribute does the same for Port B. By default, both WRITE_MODE_A and WRITE_MODE_B are set to WRITE_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO_CHANGE to have the input written to memory without changing the output. The Port A and Port B Conflict Resolution section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

Port A and Port B Conflict Resolution

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE_MODE_A and WRITE_MODE_B settings.

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=NO_CHANGE

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | No Change | X | No Change | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | No Change | No Change | No Change | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 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 and WRITE_MODE_B=READ_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIB | DIPB |

WRITE_MODE_A=NO_CHANGE and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----------|-----|-----------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | DIA | DIPA |
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | No Change | X | No Change | X | X | X |

WRITE_MODE_A=READ_FIRST and WRITE_MODE_B=WRITE_FIRST

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----------|------------|
| 0 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | No Change | No Change |
| 1 | 0 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | RAM | RAM | RAM | RAM | DIA | DIPA |

| WEA | WEB | CLKA | CLKB | DIA | DIB | DIPA | DIPB | DOA | DOB | DOPA | DOPB | Data RAM | Parity Ram |
|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|----------|------------|
| 0 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIB | DIPB |
| 1 | 1 | ↑ | ↑ | DIA | DIB | DIPA | DIPB | X | DIB | X | DIPB | DIA | DIPA |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | Yes |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|----------------------|--------------------|---|-----------|---|
| INIT_00 to INIT_3F | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the data portion of the RAM array. |
| INIT_A | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INIT_B | Binary/Hexadecimal | Any | All zeros | Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM. |
| INITP_00 to INITP_07 | Binary/Hexadecimal | Any | All zeros | Specifies the initial contents of the parity portion of the RAM array. |
| SIM_COLLISION_CHECK | String | "ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE" | "ALL" | <p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> "ALL" - Warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X). "NONE" - No warning and affected outputs/memory retain last value. <p>Note Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p> |
| SRVAL_A | Binary/Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM. |

| Attribute | Type | Allowed Values | Default | Description |
|--------------|------------------------|--|---------------|---|
| SRVAL_B | Binary/ Hexadecimal | Any | All zeros | Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM. |
| WRITE_MODE_A | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |
| WRITE_MODE_B | String | "WRITE_FIRST", "READ_FIRST", "NO_CHANGE" | "WRITE_FIRST" | Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to <ul style="list-style-type: none"> WRITE_FIRST - the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. READ_FIRST - displays the prior contents of the RAM to the output port prior to writing the new data. NO_CHANGE - keeps the previous value on the output port and does not update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S9_S9: 2k x 8 + 1 Parity bit Dual-Port RAM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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",

```


448

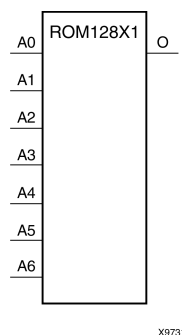

```
// End of RAMB16_S9_S9_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

ROM128X1

Primitive: 128-Deep by 1-Wide ROM



X9731

Introduction

This design element is a 128-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 7-bit address (A6:A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of 32 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H. An error occurs if the INIT=value is not specified.

Logic Table

| Input | | | | Output |
|-------|----|----|----|----------|
| 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) |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-------------------|-----------|------------------------------------|
| INIT | Hexadecimal | Any 128-Bit Value | All zeros | Specifies the contents of the ROM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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]
);

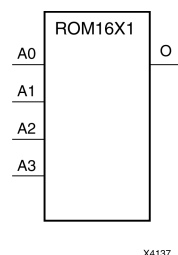
// End of ROM128X1_inst instantiation
```

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

ROM16X1

Primitive: 16-Deep by 1-Wide ROM



Introduction

This design element is a 16-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 4-bit address (A3:A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of four hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H. For example, the INIT=10A7 parameter produces the data stream: 0001 0000 1010 0111. An error occurs if the INIT=value is not specified.

Logic Table

| Input | | | | Output |
|-------|----|----|----|----------|
| 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) |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|------------------------------------|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Specifies the contents of the ROM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- ROM16X1: 16 x 1 Asynchronous Distributed (LUT) ROM
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// ROM16X1: 16 x 1 Asynchronous Distributed (LUT) ROM
//          Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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]
);

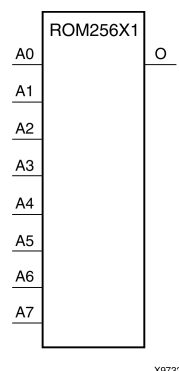
// End of ROM16X1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

ROM256X1

Primitive: 256-Deep by 1-Wide ROM



Introduction

This design element is a 256-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 8-bit address (A7:A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of 64 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H.

An error occurs if the INIT=value is not specified.

Logic Table

| Input | | | | Output |
|-------|----|----|----|----------|
| 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) |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|-------------------|-----------|------------------------------------|
| INIT | Hexadecimal | Any 256-Bit Value | All zeros | Specifies the contents of the ROM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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]
);

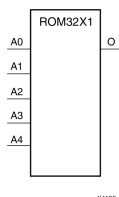
// End of ROM256X1_inst instantiation
```


For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

ROM32X1

Primitive: 32-Deep by 1-Wide ROM



Introduction

This design element is a 32-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 5-bit address (A4:A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of eight hexadecimal digits that are written into the ROM from the most-significant digit A=1FH to the least-significant digit A=00H.

For example, the INIT=10A78F39 parameter produces the data stream: 0001 0000 1010 0111 1000 1111 0011 1001. An error occurs if the INIT=value is not specified.

Logic Table

| Input | | | | Output |
|-------|----|----|----|----------|
| 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) |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|------------------------------------|
| INIT | Hexadecimal | Any 32-Bit Value | All zeros | Specifies the contents of the ROM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
//          Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

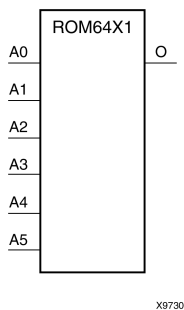
ROM32X1 #(
  .INIT(32'h00000000) // Contents of ROM
) ROM32X1_inst (
  .O(O),    // ROM output
  .A0(A0),  // ROM address[0]
  .A1(A1),  // ROM address[1]
  .A2(A2),  // ROM address[2]
  .A3(A3),  // ROM address[3]
  .A4(A4)   // ROM address[4]
);
// End of ROM32X1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

ROM64X1

Primitive: 64-Deep by 1-Wide ROM



Introduction

This design element is a 64-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 6-bit address (A5:A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of 16 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H. An error occurs if the INIT=value is not specified.

Logic Table

| Input | | | | Output |
|-------|----|----|----|----------|
| 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) |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|------------------------------------|
| INIT | Hexadecimal | Any 64-Bit Value | All zeros | Specifies the contents of the ROM. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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]
);

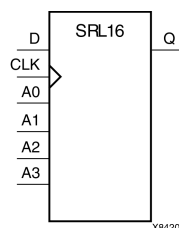
// End of ROM64X1_inst instantiation
```

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

SRL16

Primitive: 16-Bit Shift Register Look-Up Table (LUT)



Introduction

This design element is a shift register look-up table (LUT). The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: $\text{Length} = (8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$. If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

The data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition. During subsequent Low-to-High clock transitions data shifts to the next highest bit position while new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached.

Logic Table

| Inputs | | | Output |
|----------------|-----|---|-----------------------|
| A _m | CLK | D | Q |
| A _m | X | X | Q(A _m) |
| A _m | ↑ | D | Q(A _m - 1) |
| m = 0, 1, 2, 3 | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|---|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Sets the initial value of Q output after configuration. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- SRL16: 16-bit shift register LUT operating on posedge of clock
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// SRL16: 16-bit shift register LUT operating on posedge of clock
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

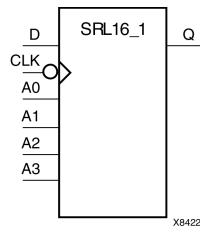
// End of SRL16_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

SRL16_1

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock



Introduction

This design element is a shift register look-up table (LUT). The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: $\text{Length} = (8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$. If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

The data (D) is loaded into the first bit of the shift register during the High-to-Low clock (CLK) transition. During subsequent High-to-Low clock transitions data shifts to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached.

Logic Table

| Inputs | | | Output |
|----------------|-----|---|-----------------------|
| A _m | CLK | D | Q |
| A _m | X | X | Q(A _m) |
| A _m | ↓ | D | Q(A _m - 1) |
| m = 0, 1, 2, 3 | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|--|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Sets the initial value of Q output after configuration |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- SRL16_1: 16-bit shift register LUT operating on negedge of clock
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// SRL16_1: 16-bit shift register LUT operating on negedge of clock
//          Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

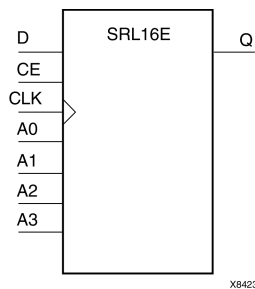
// End of SRL16_1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

SRL16E

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Clock Enable



Introduction

This design element is a shift register look-up table (LUT). The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: Length = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$. If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

When CE is High, the data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition. During subsequent Low-to-High clock transitions, when CE is High, data shifts to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached. When CE is Low, the register ignores clock transitions.

Logic Table

| Inputs | | | | Output |
|----------------|----|-----|---|-----------------------|
| 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 | | | | |

Port Descriptions

| Port | Direction | Width | Function |
|------|-----------|-------|--|
| Q | Output | 1 | Shift register data output |
| D | Input | 1 | Shift register data input |
| CLK | Input | 1 | Clock |
| CE | Input | 1 | Active high clock enable |
| A | Input | 4 | Dynamic depth selection of the SRL <ul style="list-style-type: none"> A=0000 ==> 1-bit shift length A=1111 ==> 16-bit shift length |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|--------------|------------------|-----------|---|
| INIT | Hexa-decimal | Any 16-Bit Value | All zeros | Sets the initial value of content and output of shift register after configuration. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

SRL16E_inst : SRL16E
generic map (
  INIT => X"0000")
port map (
  Q => Q,      -- SRL data output
  A0 => A0,    -- Select[0] input
  A1 => A1,    -- Select[1] input
  A2 => A2,    -- Select[2] input
  A3 => A3,    -- Select[3] input
  CE => CE,    -- Clock enable input
  CLK => CLK,  -- Clock input
  D => D      -- SRL data input
);

-- End of SRL16E_inst instantiation

```

Verilog Instantiation Template

```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

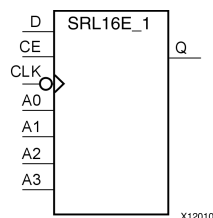
// End of SRL16E_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

SRL16E_1

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock and Clock Enable



Introduction

This design element is a shift register look-up table (LUT) with clock enable (CE). The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: $\text{Length} = (8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$. If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

When CE is High, the data (D) is loaded into the first bit of the shift register during the High-to-Low clock (CLK) transition. During subsequent High-to-Low clock transitions, when CE is High, data is shifted to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached. When CE is Low, the register ignores clock transitions.

Logic Table

| Inputs | | | | Output |
|----------------|----|-----|---|-----------|
| Am | CE | CLK | D | Q |
| Am | 0 | X | X | Q(Am) |
| Am | 1 | ↓ | D | Q(Am - 1) |
| m = 0, 1, 2, 3 | | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|---|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Sets the initial value of content and output of shift register after configuration. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- SRL16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// SRL16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

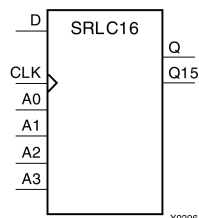
// End of SRL16E_1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

SRLC16

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry



Introduction

This design element is a shift register look-up table (LUT) with Carry. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: $\text{Length} = (8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$. If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

The data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition. During subsequent Low-to-High clock transitions data shifts to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached.

Note The Q15 output is available for you in cascading to multiple shift register LUTs to create larger shift registers.

Logic Table

| Inputs | | | Output |
|----------------|-----|---|-----------------------|
| A _m | CLK | D | Q |
| A _m | X | X | Q(A _m) |
| A _m | ↑ | D | Q(A _m - 1) |
| m = 0, 1, 2, 3 | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|---|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Sets the initial value of content and output of shift register after configuration. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- SRLC16: 16-bit cascadable shift register LUT operating on posedge of clock
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// SRLC16: 16-bit cascadable shift register LUT operating on posedge of clock
//          Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

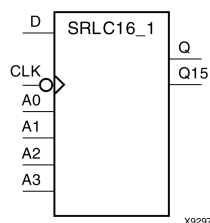
// End of SRLC16_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

SRLC16_1

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry and Negative-Edge Clock



Introduction

This design element is a shift register look-up table (LUT) with carry and a negative-edge clock. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: $\text{Length} = (8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$. If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

Note The Q15 output is available for your use in cascading multiple shift register LUTs to create larger shift registers.

Logic Table

| Inputs | | | Output | |
|----------------|-----|---|-----------|-----------|
| Am | CLK | D | Q | Q15 |
| Am | X | X | Q(Am) | No Change |
| Am | ↓ | D | Q(Am - 1) | Q14 |
| m = 0, 1, 2, 3 | | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|---|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Sets the initial value of content and output of shift register after configuration. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- SRLC16_1: 16-bit cascadable shift register LUT operating on negedge of clock
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// SRLC16_1: 16-bit cascadable shift register LUT operating on negedge of clock
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

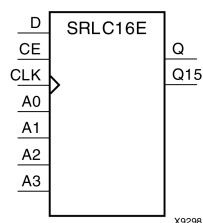
// End of SRLC16_1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

SRLC16E

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry and Clock Enable



Introduction

This design element is a shift register look-up table (LUT) with carry and clock enable. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: $\text{Length} = (8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$. If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

The data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition. When CE is High, during subsequent Low-to-High clock transitions, data shifts to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached.

Note The Q15 output is available for you in cascading to multiple shift register LUTs to create larger shift registers.

Logic Table

| Inputs | | | | Output | |
|---------------|-----|----|---|-----------|-------|
| 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 | | | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|---|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Sets the initial value of content and output of shift register after configuration. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- SRLC16E: 16-bit cascable shift register LUT with clock enable operating on posedge of clock
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// SRLC16E: 16-bit cascable shift register LUT with clock enable operating on posedge of clock
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);

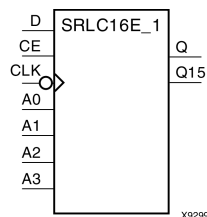
// End of SRLC16E_inst instantiation
```

For More Information

- See the [*Spartan-3 Generation FPGA User Guide*](#).
- See the [*Spartan-3E FPGA Family Complete Data Sheet*](#).

SRLC16E_1

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry, Negative-Edge Clock, and Clock Enable



Introduction

This design element is a shift register look-up table (LUT) with carry, clock enable, and negative-edge clock. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: $\text{Length} = (8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$. If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

When CE is High, the data (D) is loaded into the first bit of the shift register during the High-to-Low clock (CLK) transition. During subsequent High-to-Low clock transitions data shifts to the next highest bit position as new data is loaded when CE is High. The data appears on the Q output when the shift register length determined by the address inputs is reached.

Note The Q15 output is available for your use in cascading multiple shift register LUTs to create larger shift registers.

Logic Table

| Inputs | | | | Output | |
|---------------|----|-----|---|-----------|-----------|
| Am | CE | CLK | D | Q | Q15 |
| Am | 0 | X | X | Q(Am) | No Change |
| Am | 1 | X | X | Q(Am) | No Change |
| Am | 1 | ↓ | D | Q(Am -1) | Q14 |
| m= 0, 1, 2, 3 | | | | | |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

Available Attributes

| Attribute | Type | Allowed Values | Default | Description |
|-----------|-------------|------------------|-----------|---|
| INIT | Hexadecimal | Any 16-Bit Value | All zeros | Sets the initial value of content and output of shift register after configuration. |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- SRLC16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
--           Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// SRLC16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
);
```



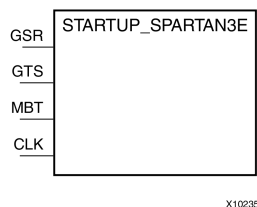
```
// End of SRLC16E_1_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

STARTUP_SPARTAN3E

Primitive: Spartan®-3E User Interface to the GSR, GTS, Configuration Startup Sequence and Multi-Boot Trigger Circuitry



Introduction

This design element allows the connection of ports, or your circuitry, to control certain dedicated circuitry and routes within the FPGA. Signals connected the GSR port of this component can control the global set/reset (referred to as GSR) of the device. The GSR net connects to all registers in the device and places the registers into their initial value state. Connecting a signal to the GTS port connects to the dedicated route controlling the three-state outputs of every pin in the device. Connecting a clock signal to the CLK input allows the startup sequence after configuration to be synchronized to a user-defined clock. The MBT (Multi-Boot Trigger) pin allows the triggering of a new configuration when the device is properly set up for this feature.

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Recommended |
| Inference | No |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- STARTUP_SPARTAN3E: Startup primitive for GSR, GTS, startup sequence
--                   control and Multi-Boot Configuration.
--                   Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

STARTUP_SPARTAN3E_inst : STARTUP_SPARTAN3E
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)
  MBT => MBT -- Multi-Boot Trigger input
);

-- End of STARTUP_SPARTAN3E_inst instantiation
```

Verilog Instantiation Template

```
// STARTUP_SPARTAN3E: Startup primitive for GSR, GTS, startup sequence control
//                               and Multi-Boot Configuration Trigger. Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

STARTUP_SPARTAN3E STARTUP_SPARTAN3E_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)
    .MBT(MBT)            // Multi-Boot Trigger input
);

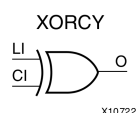
// End of STARTUP_SPARTAN3E_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

XORCY

Primitive: XOR for Carry Logic with General Output



Introduction

This design element is a special XOR with general O output that generates faster and smaller arithmetic functions. The XORCY primitive is a dedicated XOR function within the carry-chain logic of the slice. It allows for fast and efficient creation of arithmetic (add/subtract) or wide logic functions (large AND/OR gate).

Logic Table

| Input | | Output |
|-------|----|--------|
| LI | CI | O |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- XORCY: Carry-Chain XOR-gate with general output
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// XORCY: Carry-Chain XOR-gate with general output
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

XORCY XORCY_inst (
    .O(O),    // XOR output signal
    .CI(CI),  // Carry input signal
    .LI(LI)   // LUT4 input signal
);

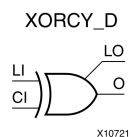
// End of XORCY_inst instantiation
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

XORCY_D

Primitive: XOR for Carry Logic with Dual Output



Introduction

This design element is a special XOR that generates faster and smaller arithmetic functions.

Logic Table

| Input | | Output |
|-------|----|----------|
| LI | CI | O and LO |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

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

-- XORCY_D: Carry-Chain XOR-gate with local and general outputs
--      Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// XORCY_D: Carry-Chain XOR-gate with local and general outputs
//      Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
```

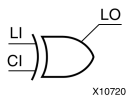
For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

XORCY_L

Primitive: XOR for Carry Logic with Local Output

XORCY_L



Introduction

This design element is a special XOR with local LO output that generates faster and smaller arithmetic functions.

Logic Table

| Input | | Output |
|-------|----|--------|
| LI | CI | LO |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Design Entry Method

| | |
|-----------------------------|-------------|
| Instantiation | Yes |
| Inference | Recommended |
| CORE Generator™ and wizards | No |
| Macro support | No |

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- XORCY_L: Carry-Chain XOR-gate with local  => direct-connect ouput
--          Spartan-3E
-- Xilinx HDL Libraries Guide, version 13.2

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 Instantiation Template

```
// XORCY_L: Carry-Chain XOR-gate with local (direct-connect) output
//           Spartan-3E
// Xilinx HDL Libraries Guide, version 13.2

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
```

For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).