Virtex and Virtex-E Libraries Guide for HDL Designs

ISE 10.1

Xilinx Trademarks and Copyright Information

Table of Contents

About this Guide

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.
- Individual 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.

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. In addition to a comprehensive Unified Library containing all design elements, beginning in 2003, Xilinx developed a separate library for each architecture. This guide is one in a series of architecture-specific libraries.

Design Entry Methods

For each design element in this guide, Xilinx evaluates the four options 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 Coregen or 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 complex to instantiate by just using the primitives. The synthesis tools will automatically expand the 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.

Clock Components

Design Element	Description
BUFG	Primitive: Global Clock Buffer
CLKDLL	Primitive: Clock Delay Locked Loop
CLKDLLE	Primitive: Clock Delay Locked Loop with Expanded Output
CLKDLLHF	Primitive: High Frequency Clock Delay Locked Loop
IBUFG	Primitive: Dedicated Input Clock Buffer

Config/BSCAN Components

Design Element	Description
BSCAN_VIRTEX	Primitive: Virtex Boundary Scan Logic Control Circuit
CAPTURE_VIRTEX	Primitive: Virtex Register State Capture for Bitstream Readback
STARTUP_VIRTEX	Primitive: Virtex User Interface to Global Clock, Reset, and 3-State Controls

I/O Components

Design Element	Description
IBUF	Primitive: Input Buffer
IBUFG	Primitive: Dedicated Input Clock Buffer
IOBUF	Primitive: Bi-Directional Buffer
KEEPER	Primitive: KEEPER Symbol
OBUF	Primitive: Output Buffer
OBUFT	Primitive: 3-State Output Buffer with 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	No: 16-Deep by 2-Wide Static Synchronous RAM
RAM16X4S	No: 16-Deep by 4-Wide Static Synchronous RAM
RAM16X8S	No: 16-Deep by 8-Wide Static Synchronous RAM
RAM32X1S	Primitive: 32-Deep by 1-Wide Static Synchronous RAM
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
RAM32X4S	Primitive: 32-Deep by 4-Wide Static Synchronous RAM
RAM32X8S	Primitive: 32-Deep by 8-Wide Static Synchronous RAM
RAMB4_S1	Primitive: 4K-bit Single-Port Synchronous Block RAM with Port Width Configured to 1 Bit
RAMB4_S1_S1	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 1-bit
RAMB4_S1_S16	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 1-bit and 16-bits
RAMB4_S1_S2	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 1-bit and 2-bits
RAMB4_S1_S4	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 1-bit and 4-bits
RAMB4_S1_S8	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 1-bit and 8-bits
RAMB4_S16	Primitive: 4096-Bit Single-Port Synchronous Block RAM with Port Width Configured to 16 Bits
RAMB4_S16_S16	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 16-bits
RAMB4_S2	Primitive: 4K-bit Single-Port Synchronous Block RAM with Port Width Configured to 2-bits
RAMB4_S2_S16	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 2-bits and 16-bits
RAMB4_S2_S2	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 2-bits and 2-bits
RAMB4_S2_S4	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 2-bits and 4-bits
RAMB4_S2_S8	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 2-bits and 8-bits
RAMB4_S4	Primitive: 4k-bit Single-Port Synchronous Block RAM with Port Width Configured to 4-bits
RAMB4_S4_S16	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 4-bits and 16-bits

Design Element	Description
RAMB4_S4_S4	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 4-bits and 4-bits
RAMB4_S4_S8	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 4-bits and 8-bits
RAMB4_S8	Primitive: 4k-bit Single-Port Synchronous Block RAM with Port Width Configured to 8-bits
RAMB4_S8_S16	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 8-bits and 16-bits
RAMB4_S8_S8	Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 8-bits
ROM16X1	Primitive: 16-Deep by 1-Wide ROM
ROM32X1	Primitive: 32-Deep by 1-Wide ROM

Registers & Latches

Design Element	Description
FDCE	Primitive: D Flip-Flop with Clock Enable and Asynchronous Clear
FDCE_1	Primitive: D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous Clear
FDCPE	Primitive: D Flip-Flop with Clock Enable and Asynchronous Preset and Clear
FDCPE_1	Primitive: D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous Preset and Clear
FDRSE	Primitive: D Flip-Flop with Synchronous Reset and Set and Clock Enable
FDRSE_1	Primitive: D Flip-Flop with Negative-Clock Edge, Synchronous Reset and Set, and Clock Enable
LDCPE	Primitive: Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable

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

Slice/CLB Primitives

Design Element	Description
BUFCF	Primitive: Fast Connect Buffer
LUT1	Primitive: 1-Bit Look-Up-Table with General Output

Design Element	Description
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
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
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

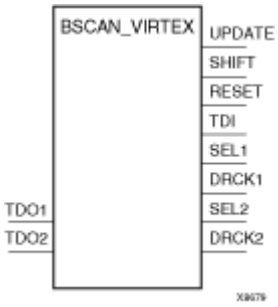
This section describes the design elements that can be used with this architecture. The design elements are organized alphabetically.

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_VIRTEX

Primitive: Virtex Boundary Scan Logic Control Circuit



Introduction

This design element is used to create internal boundary scan chains. The 4-pin JTAG interface (TDI, TDO, TCK, and TMS) are dedicated pins. To use normal JTAG for boundary scan purposes, just hook up the JTAG pins to the port and go. The pins on this element do not need to be connected, unless those special functions are needed to drive an internal scan chain.

A signal on the TDO1 input is passed to the external TDO output when the USER1 instruction is executed; the SEL1 output goes High to indicate that the USER1 instruction is active. The DRCK1 output provides USER1 access to the data register clock (generated by the TAP controller). The TDO2 and SEL2 pins perform a similar function for the USER2 instruction and the DRCK2 output provides USER2 access to the data register clock (generated by the TAP controller). The RESET, UPDATE, and SHIFT pins represent the decoding of the corresponding state of the boundary scan internal state machine. The TDI pin provides access to the TDI signal of the JTAG port in order to shift data into an internal scan chain.

Note For specific information on boundary scan for an architecture, see *The Programmable Logic Data Sheets*

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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_VIRTEX: Boundary Scan primitive for connecting internal logic to
--               JTAG interface. Virtex/E, Spartan-IIE
-- Xilinx HDL Libraries Guide, version 10.1.2

BSCAN_VIRTEX_inst : BSCAN_VIRTEX
port map (
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_VIRTEX_inst instantiation
```

Verilog Instantiation Template

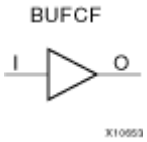
```
// BSCAN_VIRTEX: Boundary Scan primitive for connecting internal logic to
//               JTAG interface. Virtex/E, Spartan-IIE
// Xilinx HDL Libraries Guide, version 10.1.2

BSCAN_VIRTEX BSCAN_VIRTEX_inst (
.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_VIRTEX_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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.
--        For use with all FPGAs.
-- Xilinx HDL Libraries Guide, version 10.1.2

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.
//        For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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.

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen 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)
--       All Devices
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of BUFG_inst instantiation
```

Verilog Instantiation Template

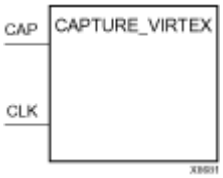
```
// BUFG: Global Clock Buffer (source by an internal signal)
//       All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of BUFG_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

CAPTURE_VIRTEX

Primitive: Virtex 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.

The readback function is provided through dedicated configuration port instructions.

Note Virtex and Virtex-E allow for capturing register (flip-flop and latch) states only. Although LUT RAM, SRL, and block RAM states are read back, 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 (transition on CLK while CAP is asserted). To limit the readback operation to a single data capture, add the ONESHOT attribute to CAPTURE_VIRTEX.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

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

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_VIRTEX: Register State Capture for Bitstream Readback
--                 Virtex/E
-- Xilinx HDL Libraries Guide, version 10.1.2

CAPTURE_VIRTEX_inst : CAPTURE_VIRTEX
port map (
CAP => CAP,    -- Capture input
CLK => CLK     -- Clock input
);

-- End of CAPTURE_VIRTEX_inst instantiation
```

Verilog Instantiation Template

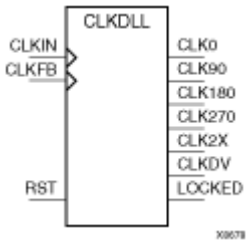
```
// CAPTURE_VIRTEX: Register State Capture for Bitstream Readback
//                 Virtex/E
// Xilinx HDL Libraries Guide, version 10.1.2

CAPTURE_VIRTEX CAPTURE_VIRTEX_inst (
.CAP(CAP),   // Capture input
.CLK(CLK)    // Clock input
);

// End of CAPTURE_VIRTEX_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

CLKDLL

Primitive: Clock Delay Locked Loop



Introduction

This design element is a clock delay locked loop used to minimize clock skew. It synchronizes the clock signal at the feedback clock input (CLKFB) to the clock signal at the input clock (CLKIN). The locked output (LOCKED) is high when the two signals are in phase. The signals are considered to be in phase when their rising edges are within a specific range of each other (see The Programmable Logic Data Sheets for the most current value).

The frequency of the clock signal at the CLKIN input must be in a specific range depending on speed grade (see The Programmable Logic Data Sheets for the most current values). The CLKIN pin must be driven by an IBUFG or a BUFG. If phase alignment is not required, CLKIN can also be driven by IBUF.

On-chip synchronization is achieved by connecting the CLKFB input to a point on the global clock network driven by a BUFG, a global clock buffer. The BUFG connected to the CLKFB input of the CLKDLL must be sourced from either the CLK0 or CLK2X outputs of the same CLKDLL. The CLKIN input should be connected to the output of an IBUFG, with the IBUFG input connected to a pad driven by the system clock.

Off-chip synchronization is achieved by connecting the CLKFB input to the output of an IBUFG, with the IBUFG input connected to a pad. Either the CLK0 or CLK2X output can be used but not both. The CLK0 or CLK2X must be connected to the input of OBUF, an output buffer.

The duty cycle of the CLK0 output is 50-50 unless the DUTY_CYCLE_CORRECTION attribute is set to FALSE, in which case the duty cycle is the same as that of the CLKIN input. The duty cycle of the phase shifted outputs (CLK90, CLK180, and CLK270) is the same as that of the CLK0 output. The duty cycle of the CLK2X and CLKDV outputs is always 50-50. The frequency of the CLKDV output is determined by the value assigned to the CLKDV_DIVIDE attribute.

The master reset input (RST) resets CLKDLL to its initial (power-on) state. The signal at the RST input is asynchronous and must be held High for just 2ns.

Port Descriptions

Port	Function
CLK0	Clock at 1x CLKIN frequency
CLK180	Clock at 1x CLKIN frequency, shifted 180 º with regards to CLK0
CLK270	Clock at 1x CLKIN frequency, shifted 270 º with regards to CLK0
CLK2X	Clock at 2x CLKIN frequency, in phase with CLK0
CLK90	Clock at 1x CLKIN frequency, shifted 90 º with regards to CLK0
CLKDV	Clock at (1/n)x CLKIN frequency, n=CLKDV_DIVIDE value. CLKDV is in phase with CLK0.
LOCKED	CLKDLL locked

Note See the "PERIOD Specifications on CLKDLLs and DCM" in the *Constraints Guide* for additional information on using the TNM, TNM_NET, and PERIOD attributes with CLKDLL components.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- CLKDLL: Delay Locked Loop Circuit for Virtex and Spartan-II (Low frequency)
-- Xilinx HDL Libraries Guide, version 10.1.2

CLKDLL_inst : CLKDLL
generic map (
CLKDV_DIVIDE => 2.0, --  Divide by: 1.5,2.0,2.5,3.0,4.0,5.0,8.0 or 16.0
DUTY_CYCLE_CORRECTION => TRUE, --  Duty cycle correction, TRUE or FALSE
FACTORY_JF => X"C080",  --  FACTORY JF Values
STARTUP_WAIT => FALSE)  --  Delay config DONE until DLL LOCK, TRUE/FALSE
port map (
CLK0 => CLK0,     -- 0 degree DLL CLK ouptput
CLK180 => CLK180, -- 180 degree DLL CLK output
CLK270 => CLK270, -- 270 degree DLL CLK output
CLK2X => CLK2X,   -- 2X DLL CLK output
CLK90 => CLK90,   -- 90 degree DLL CLK output
CLKDV => CLKDV,   -- Divided DLL CLK out (CLKDV_DIVIDE)
LOCKED => LOCKED, -- DLL LOCK status output
CLKFB => CLKFB,   -- DLL clock feedback
CLKIN => CLKIN,   -- Clock input (from IBUFG, BUFG or DLL)
RST => RST        -- DLL asynchronous reset input
);

-- End of CLKDLL_inst instantiation
```

Verilog Instantiation Template

```
// CLKDLL: Delay Locked Loop Circuit for Virtex and Spartan-II (Low frequency)
// Xilinx HDL Libraries Guide, version 10.1.2

CLKDLL #(
.CLKDV_DIVIDE(2.0),     // Divide by: 1.5,2.0,2.5,3.0,4.0,5.0,8.0 or 16.0
.DUTY_CYCLE_CORRECTION("TRUE"),  // Duty cycle correction, TRUE or FALSE
.FACTORY_JF(16'hC080),  // FACTORY JF Values
.STARTUP_WAIT("FALSE") // Delay config DONE until DLL LOCK, TRUE/FALSE
) CLKDLL_inst (
.CLK0(CLK0),     // 0 degree DLL CLK output
.CLK180(CLK180), // 180 degree DLL CLK output
.CLK270(CLK270), // 270 degree DLL CLK output
.CLK2X(CLK2X),   // 2X DLL CLK output
.CLK90(CLK90),   // 90 degree DLL CLK output
.CLKDV(CLKDV),   // Divided DLL CLK out (CLKDV_DIVIDE)
.LOCKED(LOCKED), // DLL LOCK status output
.CLKFB(CLKFB),   // DLL clock feedback
.CLKIN(CLKIN),   // Clock input (from IBUFG, BUFG or DLL)
.RST(RST)        // DLL asynchronous reset input
```

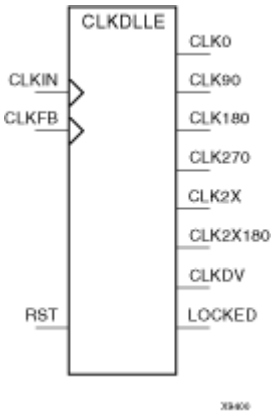
```
);
// End of CLKDLL_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

CLKDLLE

Primitive: Clock Delay Locked Loop with Expanded Output



Introduction

This design element is a clock delay locked loop used to minimize clock skew. It synchronizes the clock signal at the feedback clock input (CLKFB) to the clock signal at the input clock (CLKIN). The locked output (LOCKED) is high when the two signals are in phase. The signals are considered to be in phase when their rising edges are within a specific range of each other (see The Programmable Logic Data Sheets for the most current value).

The frequency of the clock signal at the CLKIN input must be in a specific range depending on speed grade (see The Programmable Logic Data Sheets for the most current values). The CLKIN pin must be driven by an IBUFG or a BUFG.

On-chip synchronization is achieved by connecting the CLKFB input to a point on the global clock network driven by a BUFG, a global clock buffer. The BUFG input can only be connected to the CLK0 or CLK2X output of CLKDLLE. The BUFG connected to the CLKFB input of the CLKDLLE must be sourced from either the CLK0 or CLK2X outputs of the same CLKDLLE. The CLKIN input should be connected to the output of an IBUFG, with the IBUFG input connected to a pad driven by the system clock.

Off-chip synchronization is achieved by connecting the CLKFB input to the output of an IBUFG, with the IBUFG input connected to a pad. Either the CLK0 or CLK2X output can be used but not both. The CLK0 or CLK2X must be connected to the input of OBUF, an output buffer.

The duty cycle of the CLK0 output is 50-50 unless the DUTY_CYCLE_CORRECTION attribute is set to FALSE, in which case the duty cycle is the same as that of the CLKIN input. The duty cycle of the phase shifted outputs (CLK90, CLK180, and CLK270) is the same as that of the CLK0 output. The duty cycle of the CLK2X and CLKDV outputs is always 50-50. The frequency of the CLKDV output is determined by the value assigned to the CLKDV_DIVIDE attribute.

The master reset input (RST) resets CLKDLL to its initial (power-on) state. The signal at the RST input is asynchronous and must be held High for just 2ns.

Port Descriptions

Port	Function
CLK0	Clock at 1x CLKIN frequency
CLK180	Clock at 1x CLK0 frequency, shifted 180 º with regards to CLK0
CLK270	Clock at 1x CLK0 frequency, shifted 270 º with regards to CLK0
CLK2X	Clock at 2x CLK0 frequency, in phase with CLK0

Port	Function
CLK2X180	Clock at 1x CLK2X frequency shifted 180 o with regards to CLK2X
CLK90	Clock at 1x CLK0 frequency, shifted 90 o with regards to CLK0
CLKDV	Clock at (1/n) x CLK0 frequency, where n=CLKDV_DIVIDE value. CLKDV is in phase with CLK0.
LOCKED	CLKDLLE locked. CLKIN and CLKFB synchronized.

Note See the "PERIOD Specifications on CLKDLLs and DCM" in the Constraints Guide for additional information on using the TNM, TNM_NET, and PERIOD attributes with CLKDLLE components.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- CLKDLLE: Delay Locked Loop Circuit for VirtexE and Spartan-IIE (Low frequency)
-- Xilinx HDL Libraries Guide, version 10.1.2

CLKDLLE_inst : CLKDLLE
generic map (
CLKDV_DIVIDE => 2.0, --   Divide by: 1.5,2.0,2.5,3.0,4.0,5.0,8.0 or 16.0
DUTY_CYCLE_CORRECTION => TRUE, --  Duty cycle correction, TRUE or FALSE
FACTORY_JF => X"C080",  --  FACTORY JF Values
STARTUP_WAIT => FALSE)   --  Delay config DONE until DLL LOCK, TRUE/FALSE
port map (
CLK0 => CLK0,     -- 0 degree DLL CLK ouptput
CLK180 => CLK180, -- 180 degree DLL CLK output
CLK270 => CLK270, -- 270 degree DLL CLK output
CLK2X => CLK2X,   -- 2X DLL CLK output
CLK90 => CLK90,   -- 90 degree DLL CLK output
CLKDV => CLKDV,   -- Divided DLL CLK out (CLKDV_DIVIDE)
LOCKED => LOCKED, -- DLL LOCK status output
CLKFB => CLKFB,   -- DLL clock feedback
CLKIN => CLKIN,   -- Clock input (from IBUFG, BUFG or DLL)
RST => RST        -- DLL asynchronous reset input
);

-- End of CLKDLLE_inst instantiation
```

Verilog Instantiation Template

```
// CLKDLLE: Delay Locked Loop Circuit for VirtexE and Spartan-IIE (Low frequency)
// Xilinx HDL Libraries Guide, version 10.1.2

CLKDLLE #(
.CLKDV_DIVIDE(2.0),     //  Divide by: 1.5,2.0,2.5,3.0,4.0,5.0,8.0 or 16.0
.DUTY_CYCLE_CORRECTION("TRUE"),  // Duty cycle correction, TRUE or FALSE
.FACTORY_JF(16'hC080),  // FACTORY JF Values
```

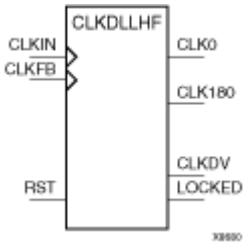
```
.STARTUP_WAIT("FALSE") // Delay config DONE until DLL LOCK, TRUE/FALSE
) CLKDLLE_inst (
.CLK0(CLK0),     // 0 degree DLL CLK output
.CLK180(CLK180), // 180 degree DLL CLK output
.CLK270(CLK270), // 270 degree DLL CLK output
.CLK2X(CLK2X),   // 2X DLL CLK output
.CLK90(CLK90),   // 90 degree DLL CLK output
.CLKDV(CLKDV),   // Divided DLL CLK out (CLKDV_DIVIDE)
.LOCKED(LOCKED), // DLL LOCK status output
.CLKFB(CLKFB),   // DLL clock feedback
.CLKIN(CLKIN),   // Clock input (from IBUFG, BUFG or DLL)
.RST(RST)        // DLL asynchronous reset input
);

// End of CLKDLLE_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

CLKDLLHF

Primitive: High Frequency Clock Delay Locked Loop



Introduction

This design element is a high frequency clock delay locked loop used to minimize clock skew. It synchronizes the clock signal at the feedback clock input (CLKFB) to the clock signal at the input clock (CLKIN). The locked output (LOCKED) is high when the two signals are in phase. The signals are considered to be in phase when their rising edges are within a specific range of each other (see The Programmable Logic Data Sheets for the most current value).

The frequency of the clock signal at the CLKIN input must be in a specific range depending on speed grade (see The Programmable Logic Data Sheets for the most current values). The CLKIN pin must be driven by an IBUFG or a BUFG.

On-chip synchronization is achieved by connecting the CLKFB input to a point on the global clock network driven by a BUFG, a global clock buffer. The BUFG input can only be connected to the CLK0 output of CLKDLLHF. The BUFG connected to the CLKFB input of the CLKDLLHF must be sourced from the CLK0 output of the same CLKDLLHF. The CLKIN input should be connected to the output of an IBUFG, with the IBUFG input connected to a pad driven by the system clock.

Off-chip synchronization is achieved by connecting the CLKFB input to the output of an IBUFG, with the IBUFG input connected to a pad. Only the CLK0 output can be used. CLK0 must be connected to the input of OBUF, an output buffer.

The duty cycle of the CLK0 output is 50-50 unless the DUTY_CYCLE_CORRECTION attribute is set to FALSE, in which case the duty cycle is the same as that of the CLKIN input. The duty cycle of the phase shifted output (CLK180) is the same as that of the CLK0 output. The frequency of the CLKDV output is determined by the value assigned to the CLKDV_DIVIDE attribute.

The master reset input (RST) resets CLKDLL to its initial (power-on) state. The signal at the RST input is asynchronous and must be held High for just 2ns.

Port Descriptions

Output	Function
CLK0	Clock at 1x CLKIN frequency
CLK180	Clock at 1x CLKIN frequency, shifted 180o with regards to CLK0
CLKDV	Clock at (1/n)x CLKIN frequency, n=CLKDV_DIVIDE value. CLKDV is in phase with CLK0.
LOCKED	CLKDLLHF locked

Note See the "PERIOD Specifications on CLKDLLs and DCM" section of the "Xilinx Constraints P" chapter in the Constraints Guide for additional information on using the TNM, TNM_NET, and PERIOD attributes with CLKDLLHF components.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- CLKDLLHF: Delay Locked Loop Circuit for Virtex/E and Spartan-II/IIE (High frequency)
-- Xilinx HDL Libraries Guide, version 10.1.2

CLKDLLHF_inst : CLKDLLHF
generic map (
CLKDV_DIVIDE => 2.0, --  Divide by: 1.5,2.0,2.5,3.0,4.0,5.0,8.0 or 16.0
DUTY_CYCLE_CORRECTION => TRUE, --  Duty cycle correct, TRUE or FALSE
FACTORY_JF => X"C080", --  FACTORY JF Values
STARTUP_WAIT => FALSE) --  Delay config DONE until DLL LOCK, TRUE/FALSE
port map (
CLK0 => CLK0,     -- 0 degree DLL CLK ouptput
CLK180 => CLK180, -- 180 degree DLL CLK output
CLKDV => CLKDV,   -- Divided DLL CLK out (CLKDV_DIVIDE)
LOCKED => LOCKED, -- DLL LOCK status output
CLKFB => CLKFB,   -- DLL clock feedback
CLKIN => CLKIN,   -- Clock input (from IBUFG, BUFG or DLL)
RST => RST        -- DLL asynchronous reset input
);

-- End of CLKDLLHF_inst instantiation
```

Verilog Instantiation Template

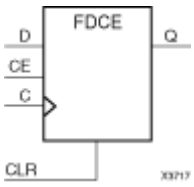
```
// CLKDLLHF: Delay Locked Loop Circuit for Virtex/E and Spartan-II/IIE (High frequency)
// Xilinx HDL Libraries Guide, version 10.1.2

CLKDLLHF #(
.CLKDV_DIVIDE(2.0), // Divide by: 1.5,2.0,2.5,3.0,4.0,5.0,8.0 or 16.0
.DUTY_CYCLE_CORRECTION("TRUE"), // Duty cycle correct, TRUE or FALSE
.FACTORY_JF(16'hC080),  // FACTORY JF Values
.STARTUP_WAIT("FALSE")  // Delay config DONE until DLL LOCK, TRUE/FALSE
) CLKDLLHF_inst (
.CLK0(CLK0),     // 0 degree DLL CLK output
.CLK180(CLK180), // 180 degree DLL CLK output
.CLKDV(CLKDV),   // Divided DLL CLK out (CLKDV_DIVIDE)
.LOCKED(LOCKED), // DLL LOCK status output
.CLKFB(CLKFB),   // DLL clock feedback
.CLKIN(CLKIN),   // Clock input (from IBUFG, BUFG or DLL)
.RST(RST)        // DLL asynchronous reset input
);

// End of CLKDLLHF_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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.

For XC9500XL and XC9500XV devices, logic connected to the clock enable (CE) input may be implemented using the clock enable product term (p-term) in the macrocell, provided the logic can be completely implemented using the single p-term available for clock enable without requiring feedback from another macrocell. Only FDCE and FDPE flip-flops may take advantage of the clock-enable p-term.

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
Coregen 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).  All families.
-- Xilinx HDL Libraries Guide, version 10.1.2

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

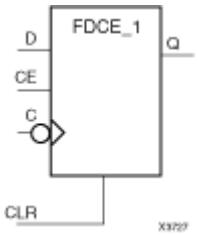
```
// FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
//       Clock Enable (posedge clk).
//       All families.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

FDCE_1

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



Introduction

This design element is a single D-type flip-flop with data (D), clock enable (CE), asynchronous clear (CLR) inputs, and data output (Q). The asynchronous CLR input, when High, overrides all other inputs and sets the Q output Low. The data on the (D) input is loaded into the flip-flop when CLR is Low and CE is High on the High-to-Low clock (C) transition. When CE is Low, the 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	?	No Change
0	1	1	?	1
0	1	0	?	0

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen 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_1: Single Data Rate D Flip-Flop with Asynchronous Clear and
--         Clock Enable (negedge clock).  All families.
-- Xilinx HDL Libraries Guide, version 10.1.2

FDCE_1_inst : FDCE_1
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_1_inst instantiation
```

Verilog Instantiation Template

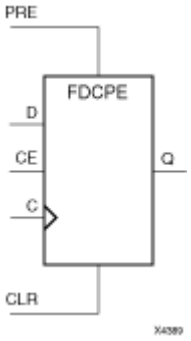
```
// FDCE_1: Single Data Rate D Flip-Flop with Asynchronous Clear and
//         Clock Enable (negedge clock).
//         All families.
// Xilinx HDL Libraries Guide, version 10.1.2

FDCE_1 #(
.INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDCE_1_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_1_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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

Port	Direction	Width	Function
D	Input	1	Data input
PRE	Input	1	Asynchronous set input

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen 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).  All families.
-- Xilinx HDL Libraries Guide, version 10.1.2

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

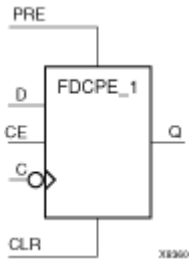
```
// FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
//        Clock Enable (posedge clk).
//        All families.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

FDCPE_1

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



Introduction

FDCPE_1 is a single D-type flip-flop with data (D), clock enable (CE), asynchronous preset (PRE), and asynchronous clear (CLR) inputs and data output (Q). The asynchronous PRE, when High, sets the (Q) output High; CLR, when High, resets the output Low. Data on the (D) input is loaded into the flip-flop when PRE and CLR are Low and CE is High on the High-to-Low clock (C) transition. When CE is Low, the 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	**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
Coregen 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;
```

```
-- FDCPE_1: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
--          Clock Enable (negedge clock).  All families.
-- Xilinx HDL Libraries Guide, version 10.1.2

FDCPE_1_inst : FDCPE_1
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_1_inst instantiation
```

Verilog Instantiation Template

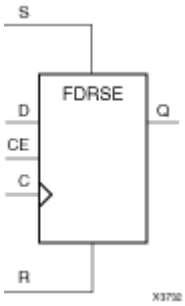
```
// FDCPE_1: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
//          Clock Enable (negedge clock).
//          All families.
// Xilinx HDL Libraries Guide, version 10.1.2

FDCPE_1 #(
.INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDCPE_1_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_1_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary	0 or 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).  All families.
-- Xilinx HDL Libraries Guide, version 10.1.2

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

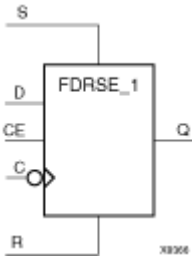
```
// FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//        Clock Enable (posedge clk).
//        All families.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

FDRSE_1

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



Introduction

FDRSE_1 is a single D-type flip-flop with synchronous reset (R), synchronous set (S), and clock enable (CE) inputs and data output (Q). The reset (R) input, when High, overrides all other inputs and resets the (Q) output Low during the High-to-Low 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 High-to-Low 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 High-to-Low clock transition.

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**
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	D	↓	D

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	**Type**	**Allowed Values**	**Default**	**Description**
INIT	Binary	0 or 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_1: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
--          Clock Enable (negedge clock).  All families.
-- Xilinx HDL Libraries Guide, version 10.1.2

FDRSE_1_inst : FDRSE_1
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_1_inst instantiation
```

Verilog Instantiation Template

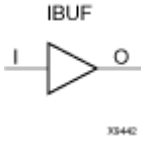
```
// FDRSE_1: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//          Clock Enable (negedge clock).
//          All families.
// Xilinx HDL Libraries Guide, version 10.1.2

FDRSE_1 #(
.INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDRSE_1_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_1_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 input
I	Input	1	Buffer output

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	No
Macro support	No

In general, this element is inferred by the synthesis tool for any specified top-level input port to the design. It is generally not necessary to specify them in the source code however if desired, they be manually instantiated by either copying the instantiation code from the ISE Libraries Guide HDL Template and paste it into the top-level entity/module of your code. It is recommended to always 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/defparam values in order to configure the proper behavior of the buffer.

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;
```

```
-- IBUF: Single-ended Input Buffer
--       All devices
-- Xilinx HDL Libraries Guide, version 10.1.2

IBUF_inst : IBUF
generic map (
IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer, "0"-"16" (Spartan-3E/3A only)
IFD_DELAY_VALUE => "AUTO", -- Specify the amount of added delay for input register, "AUTO", "0"-"8" (Spartan-3E/3A only)
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

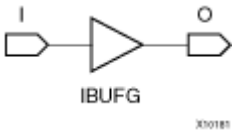
```
// IBUF: Single-ended Input Buffer
//       All devices
// Xilinx HDL Libraries Guide, version 10.1.2

IBUF #(
.IBUF_DELAY_VALUE("0"),   // Specify the amount of added input delay for
//   the buffer, "0"-"16" (Spartan-3E/3A only)
.IFD_DELAY_VALUE("AUTO"), // Specify the amount of added delay for input
//   register, "AUTO", "0"-"8" (Spartan-3E/3A only)
.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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 to the 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. The IBUFG can be routed to your choice of logic to allow the use of the dedicated clock pins for general logic.

Port Descriptions

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

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen 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;
```

```
-- IBUFG: Global Clock Buffer (sourced by an external pin)
-- Xilinx HDL Libraries Guide, version 10.1.2

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

```
);

-- End of IBUFG_inst instantiation
```

Verilog Instantiation Template

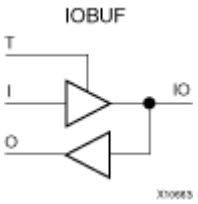
```
// IBUFG: Global Clock Buffer (sourced by an external pin)
//        All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

IBUFG #(
.IOSTANDARD("DEFAULT")
) 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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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	X
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
Coregen 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 LVTTL, LVCMOS12, LVCMOS15, LVCMOS18, LVCMOS25, or LVCMOS33 interface I/O standard.
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.
SLEW	String	"SLOW", "FAST"	"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
--        All devices
-- Xilinx HDL Libraries Guide, version 10.1.2

IOBUF_inst : IOBUF
generic map (
DRIVE => 12,
IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer, "0"-"16" (Spartan-3E/3A only)
IFD_DELAY_VALUE => "AUTO", -- Specify the amount of added delay for input register, "AUTO", "0"-"8" (Spartan-3E/3A only)
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
);

-- End of IOBUF_inst instantiation
```

Verilog Instantiation Template

```
// IOBUF: Single-ended Bi-directional Buffer
//        All devices
// Xilinx HDL Libraries Guide, version 10.1.2

IOBUF #(
.DRIVE(12), // Specify the output drive strength
.IBUF_DELAY_VALUE("0"),   // Specify the amount of added input delay for the buffer, "0"-"16" (Spartan-3E only)
.IFD_DELAY_VALUE("AUTO"), // Specify the amount of added delay for input register, "AUTO", "0"-"8" (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
);

// End of IOBUF_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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	Recommended
Coregen 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
--         All FPGA, CoolRunner-II
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of KEEPER_inst instantiation
```

Verilog Instantiation Template

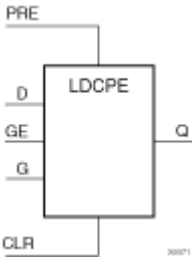
```
// KEEPER: I/O Buffer Weak Keeper
//         All FPGA, CoolRunner-II
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of KEEPER_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Integer	0 or 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.
--        All families.
-- Xilinx HDL Libraries Guide, version 10.1.2

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

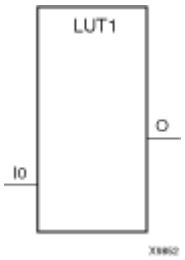
```
// LDCPE: Transparent latch with Asynchronous Reset, Preset and
//        Gate Enable.
//        All families.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

LUT1

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



Introduction

This design element is a 1-bit look-up-tables (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 Truth Table Method -A common method to determine the desired INIT value for a LUT is using a truth table. To do so, simply create a binary truth 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 that the above method however 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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

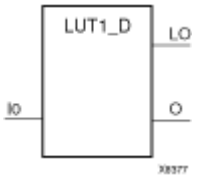
```
// LUT1: 1-input Look-Up Table with general output
//       For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 Truth Table Method -A common method to determine the desired INIT value for a LUT is using a truth table. To do so, simply create a binary truth 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 that the above method however 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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

LUT1_D_inst : LUT1_D
generic map (
INIT => "00")
port map (
LO => LO, -- LUT local output
O => O,   -- LUT general output
I0 => I0  -- LUT input
);

-- End of LUT1_D_inst instantiation
```

Verilog Instantiation Template

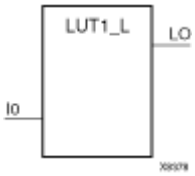
```
// LUT1_D: 1-input Look-Up Table with general and local outputs
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

LUT1_D #(
.INIT(2'b00)  // Specify LUT Contents
) LUT1_D_inst (
.LO(LO), // LUT local output
.O(O),   // LUT general output
.I0(I0)  // LUT input
);

// End of LUT1_D_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

LUT1_L

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



Introduction

This design element is a 1- bit look-up-tables (LUTs) 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 Truth Table Method -A common method to determine the desired INIT value for a LUT is using a truth table. To do so, simply create a binary truth 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 that the above method however does require the code to first specify the appropriate parameters.

Logic Table

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

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

LUT1_L_inst : LUT1_L
generic map (
INIT => "00")
port map (
LO => LO, -- LUT local output
I0 => I0  -- LUT input
);

-- End of LUT1_L_inst instantiation
```

Verilog Instantiation Template

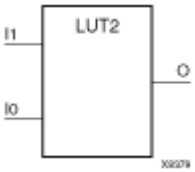
```
// LUT1_L: 1-input Look-Up Table with local output
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

LUT1_L #(
.INIT(2'b00)  // Specify LUT Contents
) LUT1_L_inst (
.LO(LO), // LUT local output
.I0(I0)  // LUT input
);

// End of LUT1_L_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 Truth Table Method -A common method to determine the desired INIT value for a LUT is using a truth table. To do so, simply create a binary truth 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 that the above method however 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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

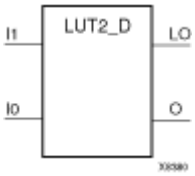
```
// LUT2: 2-input Look-Up Table with general output
//       For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

LUT2_D

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



Introduction

This design element is a 2-bit look-up-tables (LUTs) 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 Truth Table Method -A common method to determine the desired INIT value for a LUT is using a truth table. To do so, simply create a binary truth 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 that the above method however 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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

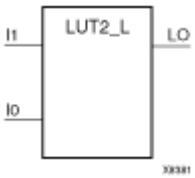
```
// LUT2_D: 2-input Look-Up Table with general and local outputs
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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_L_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

LUT2_L

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



Introduction

This design element is a 2- bit look-up-tables (LUTs) 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 Truth Table Method -A common method to determine the desired INIT value for a LUT is using a truth table. To do so, simply create a binary truth 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 that the above method however 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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

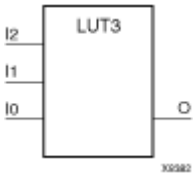
```
// LUT2_L: 2-input Look-Up Table with local output
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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.

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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

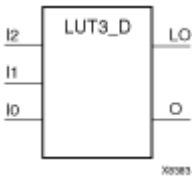
```
// LUT3: 3-input Look-Up Table with general output
//       For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

LUT3_D

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



Introduction

This design element is a 3-bit look-up-tables (LUTs) 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 truth 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 that the above method however 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

Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

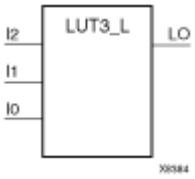
```
// LUT3_D: 3-input Look-Up Table with general and local outputs
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

LUT3_L

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



Introduction

This design element is a 3- bit look-up-tables (LUTs) 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 Truth Table Method -A common method to determine the desired INIT value for a LUT is using a truth table. To do so, simply create a binary truth 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 that the above method however 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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of LUT3_L_inst instantiation
```

Verilog Instantiation Template

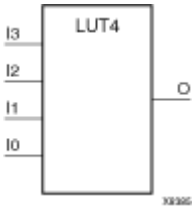
```
// LUT3_L: 3-input Look-Up Table with local output
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of LUT3_L_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

LUT4

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



Introduction

This design element is a 4-bit look-up-tables (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 Truth Table Method -A common method to determine the desired INIT value for a LUT is using a truth table. To do so, simply create a binary truth 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 that the above method however 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]

Inputs				Outputs
I3	**I2**	**I1**	**I0**	**O**
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]
1	1	0	1	INIT[13]
1	1	1	0	INIT14]
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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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
//       For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

LUT4 #(
```

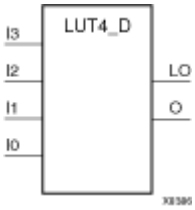
```
.INIT(16'h0000)  // Specify LUT Contents
) LUT4_inst (
.O(O),   // LUT general output
.I0(I0), // LUT input
.I1(I1), // LUT input
.I2(I2), // LUT input
.I3(I3)  // LUT input
);

// End of LUT4_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

LUT4_D

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



Introduction

This design element is a 4-bit look-up-tables (LUTs) 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 Truth Table Method -A common method to determine the desired INIT value for a LUT is using a truth table. To do so, simply create a binary truth 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 that the above method however 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	INIT14]	INIT14]
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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

LUT4_D #(
.INIT(16'h0000)  // Specify LUT Contents
) LUT4_D_inst (
```

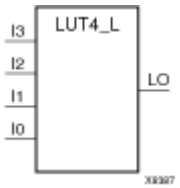
```
.LO(LO), // LUT local output
.O(O),   // LUT general output
.I0(I0), // LUT input
.I1(I1), // LUT input
.I2(I2), // LUT input
.I3(I3)  // LUT input
);

// End of LUT4_D_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

LUT4_L

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



Introduction

This design element is a 4- bit look-up-tables (LUTs) 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 Truth Table Method -A common method to determine the desired INIT value for a LUT is using a truth table. To do so, simply create a binary truth 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 that the above method however 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	INIT14]
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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

LUT4_L #(
.INIT(16'h0000)  // Specify LUT Contents
) LUT4_L_inst (
.LO(LO), // LUT local output
```

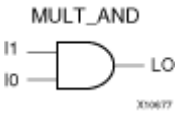
```
.I0(I0), // LUT input
.I1(I1), // LUT input
.I2(I2), // LUT input
.I3(I3)  // LUT input
);

// End of LUT4_L_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 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
Coregen 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
--           All FPGA devices except Virtex-5
-- Xilinx HDL Libraries Guide, version 10.1.2

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
//           For use with all FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

MULT_AND MULT_AND_inst (
.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
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2
```

```
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

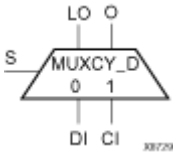
```
// MUXCY: Carry-Chain MUX with general output
//        For use with All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

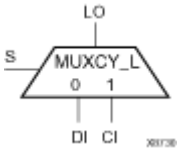
```
// MUXCY_D: Carry-Chain MUX with general and local outputs
//          For use with All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2
```

```
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

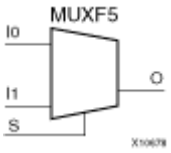
```
// MUXCY_L: Carry-Chain MUX with local output
//        For use with All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 lookup table or a 4-to-1 multiplexer in combination with the associated lookup tables. The local outputs (LO) from the two lookup 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
Coregen 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
--        All FPGA Devices except Virtex-5
-- Xilinx HDL Libraries Guide, version 10.1.2

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 directoy to the output of LUT4)
S => S      -- Input select to MUX
```

```
);

-- End of MUXF5_inst instantiation
```

Verilog Instantiation Template

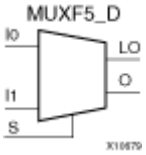
```
// MUXF5: Slice MUX to tie two LUT4's together with general output
//        For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of MUXF5_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 lookup table or a 4-to-1 multiplexer in combination with the associated lookup tables. The local outputs (LO) from the two lookup 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
Coregen 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
--          All FPGA Devices except Virtex-5
-- Xilinx HDL Libraries Guide, version 10.1.2
```

```
MUXF5_D_inst : MUXF5_D
port map (
LO => LO,  -- Ouptut 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

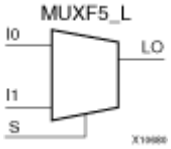
```
// MUXF5_D: Slice MUX to tie two LUT4's together with general and local outputs
//          For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

MUXF5_D MUXF5_D_inst (
.LO(LO),  // Ouptut 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
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 lookup table or a 4-to-1 multiplexer in combination with the associated lookup tables. The local outputs (LO) from the two lookup 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
Coregen 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
--          All FPGA Devices except Virtex-5
-- Xilinx HDL Libraries Guide, version 10.1.2

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 directoy to the output of LUT4)
```

```
S => S     -- Input select to MUX
);

-- End of MUXF5_L_inst instantiation
```

Verilog Instantiation Template

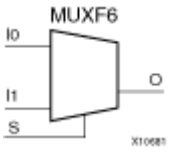
```
// MUXF5_L: Slice MUX to tie two LUT4's together with local output
//          For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

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 directoy to the output of LUT4)
.S(S)     // Input select to MUX
);

// End of MUXF5_L_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 lookup table or an 8-to-1 multiplexer in combination with the associated four lookup 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
Coregen 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
--        All FPGA Devices except Virtex-5
-- Xilinx HDL Libraries Guide, version 10.1.2

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

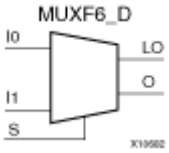
```
// MUXF6: CLB MUX to tie two MUXF5's together with general output
//        For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 lookup table or an 8-to-1 multiplexer in combination with the associated four lookup 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
Coregen 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
--          All FPGA Devices except Virtex-5
-- Xilinx HDL Libraries Guide, version 10.1.2

MUXF6_D_inst : MUXF6_D
port map (
LO => LO,  -- Ouptut 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

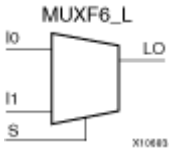
```
// MUXF6_D: CLB MUX to tie two MUXF5's together with general and local outputs
//          For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

MUXF6_D MUXF6_D_inst (
.LO(LO),  // Ouptut 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
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 lookup table or an 8-to-1 multiplexer in combination with the associated four lookup 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
Coregen 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
--          All FPGA Devices except Virtex-5
-- Xilinx HDL Libraries Guide, version 10.1.2

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

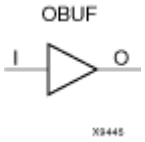
```
// MUXF6_L: CLB MUX to tie two MUXF5's together with local output
//          For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

MUXF6_L MUXF6_L_inst (
.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
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 LVTTL. 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
Coregen 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;
```

```
-- OBUF: Single-ended Output Buffer
--       All devices
-- Xilinx HDL Libraries Guide, version 10.1.2

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

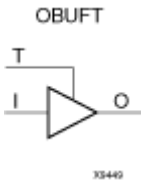
```
// OBUF: Single-ended Output Buffer
//       All devices
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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	I	F

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
Coregen 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;
```

```
-- OBUFT: Single-ended 3-state Output Buffer
--        All devices
-- Xilinx HDL Libraries Guide, version 10.1.2

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

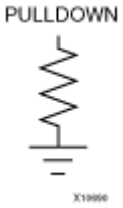
```
// OBUFT: Single-ended 3-state Output Buffer
//        All devices
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

PULLDOWN

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



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	Recommended
Coregen 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
--           All FPGA
-- Xilinx HDL Libraries Guide, version 10.1.2

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
//           All FPGA
// Xilinx HDL Libraries Guide, version 10.1.2
```

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

// End of PULLDOWN_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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	Recommended
Coregen 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
--         All FPGA, CoolRunner-II
-- Xilinx HDL Libraries Guide, version 10.1.2

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
//         All FPGA, CoolRunner-II
// Xilinx HDL Libraries Guide, version 10.1.2
```

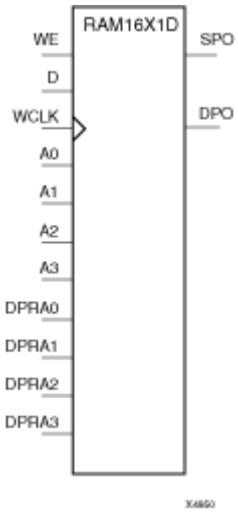
```
PULLUP PULLUP_inst (
.O(O)     // Pullup output (connect directly to top-level port)
);

// End of PULLUP_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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

Inputs			Outputs	
WE (mode)	**WCLK**	**D**	**SPO**	**DPO**
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
Coregen 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
--           All FPGAs
-- Xilinx HDL Libraries Guide, version 10.1.2

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 ddress[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

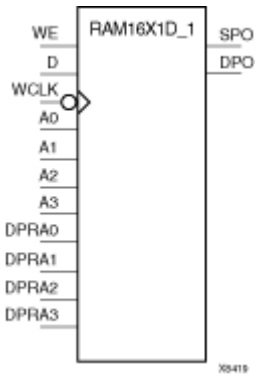
```
// RAM16X1D: 16 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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-High (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
Coregen 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
--             All FPGA
-- Xilinx HDL Libraries Guide, version 10.1.2

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 ddress[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

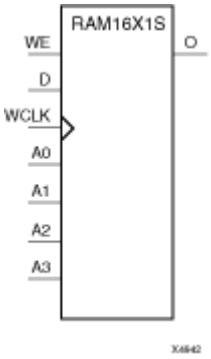
```
// RAM16X1D_1: 16 x 1 negative edge write, asynchronous read dual-port distributed RAM
//             Virtex/E/-II/-II-Pro, Spartan-II/IIE/3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAM16X1D_1 #(
.INIT(16'h0000) // Initial contents of RAM
) RAM16X1D_1_inst (
.DPO(DPO),      // Read-only 1-bit data output
.SPO(SPO),      // R/W 1-bit data output
.A0(A0),        // R/W address[0] input bit
.A1(A1),        // R/W address[1] input bit
.A2(A2),        // R/W address[2] input bit
.A3(A3),        // R/W address[3] input bit
.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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
--           All FPGA
-- Xilinx HDL Libraries Guide, version 10.1.2

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

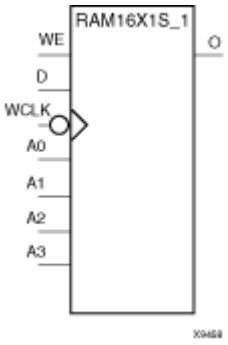
```
// RAM16X1S: 16 x 1 posedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
--             All FPGA
-- Xilinx HDL Libraries Guide, version 10.1.2

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

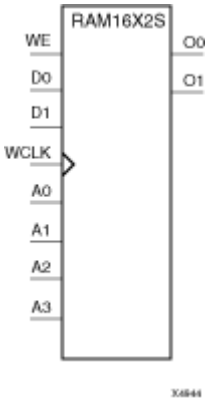
```
// RAM16X1S_1: 16 x 1 negedge write distributed (LUT) RAM
//             All FPGA
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAM16X2S

Macro: 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 Virtex-4 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
Coregen 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
--           Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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
//           Virtex-II/II-Pro, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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
```

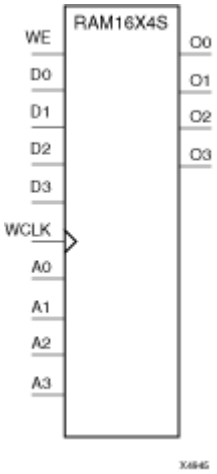
```
.D0(D0),     // RAM data[0] input
.D1(D1),     // RAM data[1] input
.WCLK(WCLK), // Write clock input
.WE(WE)      // Write enable input
);

// End of RAM16X2S_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAM16X4S

Macro: 16-Deep by 4-Wide Static Synchronous RAM



Introduction

This element is a 16-word by 4-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 (D3 – 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 (O3 – O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

Logic Table

Inputs			**Outputs**
WE (mode)	**WCLK**	**D3 – D0**	**O3 – O0**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D3-D0	D3-D0
1 (read)	↓	X	Data
Data = word addressed by bits A3 – A0.			

Design Entry Method

Instantiation	Yes
Inference	Recommended

Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_03	Hexadecimal	Any 16-Bit Value	All zeros	INIT for bit 0 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;
```

```
-- RAM16X4S: 16 x 4 posedge write distributed  => LUT RAM
--           Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

RAM16X4S_inst : RAM16X4S
generic map (
INIT_00 => X"0000", --  INIT for bit 0 of RAM
INIT_01 => X"0000", --  INIT for bit 1 of RAM
INIT_02 => X"0000", --  INIT for bit 2 of RAM
INIT_03 => X"0000") --  INIT for bit 3 of RAM
port map (
O0 => O0,     -- RAM data[0] output
O1 => O1,     -- RAM data[1] output
O2 => O2,     -- RAM data[2] output
O3 => O3,     -- RAM data[3] 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
D2 => D2,     -- RAM data[2] input
D3 => D3,     -- RAM data[3] input
WCLK => WCLK, -- Write clock input
WE => WE      -- Write enable input
);

-- End of RAM16X4S_inst instantiation
```

Verilog Instantiation Template

```
// RAM16X4S: 16 x 4 posedge write distributed (LUT) RAM
//           Virtex-II/II-Pro, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAM16X4S #(
.INIT_00(16'h0000), // INIT for bit 0 of RAM
.INIT_01(16'h0000), // INIT for bit 1 of RAM
.INIT_02(16'h0000), // INIT for bit 2 of RAM
.INIT_03(16'h0000)  // INIT for bit 3 of RAM
) RAM16X4S_inst (
.O0(O0),     // RAM data[0] output
.O1(O1),     // RAM data[1] output
.O2(O2),     // RAM data[2] output
.O3(O3),     // RAM data[3] output
.A0(A0),     // RAM address[0] input
```

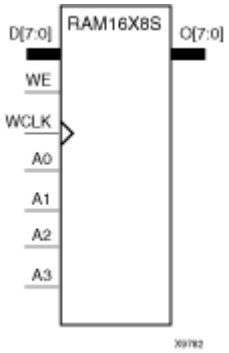
```
.A1(A1),     // RAM address[1] input
.A2(A2),     // RAM address[2] input
.A3(A3),     // RAM address[3] input
.D0(D0),     // RAM data[0] input
.D1(D1),     // RAM data[1] input
.D2(D2),     // RAM data[2] input
.D3(D3),     // RAM data[3] input
.WCLK(WCLK), // Write clock input
.WE(WE)      // Write enable input
);

// End of RAM16X4S_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAM16X8S

Macro: 16-Deep by 8-Wide Static Synchronous RAM



Introduction

This element is a 16-word by 8-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 data inputs (D7–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 (O7–O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

Logic Table

Inputs			**Outputs**
WE (mode)	**WCLK**	**D7-D0**	**O7-O0**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D7-D0	D7-D0
1 (read)	↓	X	Data
Data = word addressed by bits A3–A0			

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 To INIT_07	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;
```

```
-- RAM16X8S: 16 x 8 posedge write distributed  => LUT RAM
--           Virtex-II/II-Pro
-- Xilinx HDL Libraries Guide, version 10.1.2

RAM16X8S_inst : RAM16X8S
generic map (
INIT_00 => X"0000", --  INIT for bit 0 of RAM
INIT_01 => X"0000", --  INIT for bit 1 of RAM
INIT_02 => X"0000", --  INIT for bit 2 of RAM
INIT_03 => X"0000", --  INIT for bit 3 of RAM
INIT_04 => X"0000", --  INIT for bit 4 of RAM
INIT_05 => X"0000", --  INIT for bit 5 of RAM
INIT_06 => X"0000", --  INIT for bit 6 of RAM
INIT_07 => X"0000") --  INIT for bit 7 of RAM
port map (
O => O,       -- 8-bit RAM data 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,       -- 8-bit RAM data input
WCLK => WCLK, -- Write clock input
WE => WE      -- Write enable input
);

-- End of RAM16X8S_inst instantiation
```

Verilog Instantiation Template

```
// RAM16X8S: 16 x 8 posedge write distributed (LUT) RAM
//           Virtex-II/II-Pro
// Xilinx HDL Libraries Guide, version 10.1.2

RAM16X8S #(
.INIT_00(16'h0000), // INIT for bit 0 of RAM
.INIT_01(16'h0000), // INIT for bit 1 of RAM
.INIT_02(16'h0000), // INIT for bit 2 of RAM
.INIT_03(16'h0000), // INIT for bit 3 of RAM
.INIT_04(16'h0000), // INIT for bit 4 of RAM
.INIT_05(16'h0000), // INIT for bit 5 of RAM
.INIT_06(16'h0000), // INIT for bit 6 of RAM
.INIT_07(16'h0000)  // INIT for bit 7 of RAM
) RAM16X8S_inst (
.O(O),       // 8-bit RAM data 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),       // 8-bit RAM data input
.WCLK(WCLK), // Write clock input
.WE(WE)      // Write enable input
```

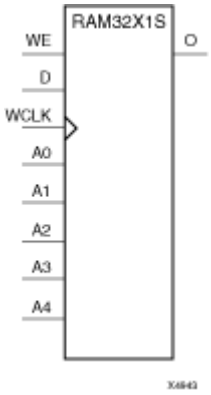
```
);
// End of RAM16X8S_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
--           All FPGA
-- Xilinx HDL Libraries Guide, version 10.1.2

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

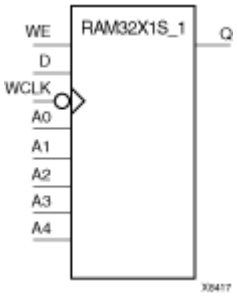
```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
--             All FPGA
-- Xilinx HDL Libraries Guide, version 10.1.2

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

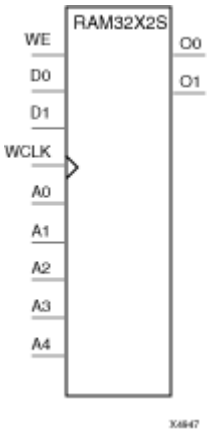
```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
//             Virtex/E/-II/-II-Pro, Spartan-II/IIE/3/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAM32X2S

Macro: 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

Coregen 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
--           Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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
//           Virtex-II/II-Pro, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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
```

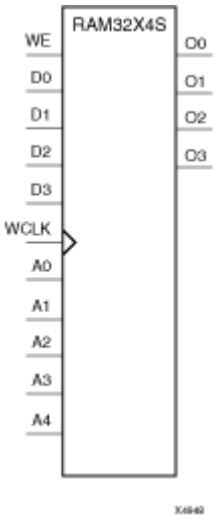
```
.WE(WE)      // Write enable input
);

// End of RAM32X2S_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAM32X4S

Macro: 32-Deep by 4-Wide Static Synchronous RAM



Introduction

This design element is a 32-word by 4-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 inputs (D3-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 (O3-O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

Logic Table

Inputs			**Outputs**
WE	**WCLK**	**D3-D0**	**O3-O0**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D3-D0	D3-D0
1 (read)	↓	X	Data
Data = word addressed by bits A4-A0			

Design Entry Method

Instantiation	Yes
Inference	Recommended

Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
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.
INIT_02	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 2 of RAM.
INIT_03	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 3 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;
```

```
-- RAM32X4S: 32 x 4 posedge write distributed  => LUT RAM
--           Virtex-II/II-Pro
-- Xilinx HDL Libraries Guide, version 10.1.2

RAM32X4S_inst : RAM32X4S
generic map (
INIT_00 => X"00000000", --  INIT for bit 0 of RAM
INIT_01 => X"00000000", --  INIT for bit 1 of RAM
INIT_02 => X"00000000", --  INIT for bit 2 of RAM
INIT_03 => X"00000000") --  INIT for bit 3 of RAM
port map (
O0 => O0,     -- RAM data[0] output
O1 => O1,     -- RAM data[1] output
O2 => O2,     -- RAM data[2] output
O3 => O3,     -- RAM data[3] 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
D2 => D2,     -- RAM data[2] input
D3 => D3,     -- RAM data[3] input
WCLK => WCLK, -- Write clock input
WE => WE      -- Write enable input
);

-- End of RAM32X4S_inst instantiation
```

Verilog Instantiation Template

```
// RAM32X4S: 32 x 4 posedge write distributed (LUT) RAM
//           Virtex-II/II-Pro
// Xilinx HDL Libraries Guide, version 10.1.2

RAM32X4S #(
.INIT_00(32'h00000000), // INIT for bit 0 of RAM
.INIT_01(32'h00000000), // INIT for bit 1 of RAM
.INIT_02(32'h00000000), // INIT for bit 2 of RAM
.INIT_03(32'h00000000)  // INIT for bit 3 of RAM
```

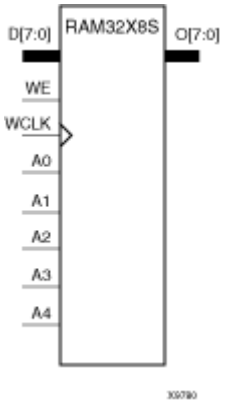
```
) RAM32X4S_inst (
.O0(O0),     // RAM data[0] output
.O1(O1),     // RAM data[1] output
.O2(O2),     // RAM data[2] output
.O3(O3),     // RAM data[3] output
.A0(A0),     // RAM address[0] input
.A1(A1),     // RAM address[1] input
.A2(A2),     // RAM address[2] input
.A3(A3),     // RAM address[3] input
.A4(A4),     // RAM address[4] input
.D0(D0),     // RAM data[0] input
.D1(D1),     // RAM data[1] input
.D2(D2),     // RAM data[2] input
.D3(D3),     // RAM data[3] input
.WCLK(WCLK), // Write clock input
.WE(WE)      // Write enable input
);

// End of RAM32X4S_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAM32X8S

Macro: 32-Deep by 8-Wide Static Synchronous RAM



Introduction

This design element is a 32-word by 8-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 inputs (D7 – 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 (O7 – O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

Logic Table

Inputs			Outputs
WE (mode)	**WCLK**	**D7-D0**	**O7-O0**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D7-D0	D7-D0
1 (read)	↓	X	Data
Data = word addressed by bits A4 – A0			

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
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.
INIT_02	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 2 of RAM.
INIT_03	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 3 of RAM.
INIT_04	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 4 of RAM.
INIT_05	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 5 of RAM.
INIT_06	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 6 of RAM.
INIT_07	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 7 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;

-- RAM32X8S: 32 x 8 posedge write distributed  => LUT RAM
--           Virtex-II/II-Pro
-- Xilinx HDL Libraries Guide, version 10.1.2

RAM32X8S_inst : RAM32X8S
generic map (
INIT_00 => X"00000000", --  INIT for bit 0 of RAM
INIT_01 => X"00000000", --  INIT for bit 1 of RAM
INIT_02 => X"00000000", --  INIT for bit 2 of RAM
INIT_03 => X"00000000", --  INIT for bit 3 of RAM
INIT_04 => X"00000000", --  INIT for bit 4 of RAM
INIT_05 => X"00000000", --  INIT for bit 5 of RAM
INIT_06 => X"00000000", --  INIT for bit 6 of RAM
INIT_07 => X"00000000") --  INIT for bit 7 of RAM
port map (
O => O,       -- 8-bit RAM data 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,       -- 8-bit RAM data input
WCLK => WCLK, -- Write clock input
WE => WE      -- Write enable input
);

-- End of RAM32X8S_inst instantiation
```

Verilog Instantiation Template

```
// RAM32X8S: 32 x 8 posedge write distributed (LUT) RAM
//           Virtex-II/II-Pro
// Xilinx HDL Libraries Guide, version 10.1.2

RAM32X8S #(
.INIT_00(32'h00000000), // INIT for bit 0 of RAM
.INIT_01(32'h00000000), // INIT for bit 1 of RAM
.INIT_02(32'h00000000), // INIT for bit 2 of RAM
.INIT_03(32'h00000000), // INIT for bit 3 of RAM
```

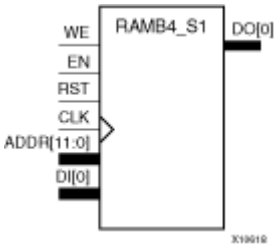
```
.INIT_04(32'h00000000), // INIT for bit 4 of RAM
.INIT_05(32'h00000000), // INIT for bit 5 of RAM
.INIT_06(32'h00000000), // INIT for bit 6 of RAM
.INIT_07(32'h00000000)  // INIT for bit 7 of RAM
) RAM32X8S_inst (
.O(O),       // 8-bit RAM data output
.A0(A0),     // RAM address[0] input
.A1(A1),     // RAM address[1] input
.A2(A2),     // RAM address[2] input
.A3(A3),     // RAM address[3] input
.A4(A4),     // RAM address[4] input
.D(D),       // 8-bit RAM data input
.WCLK(WCLK), // Write clock input
.WE(WE)      // Write enable input
);

// End of RAM32X8S_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S1

Primitive: 4K-Bit Single-Port Synchronous Block RAM with Port Width Configured to 1 Bit



Introduction

This design element is a dedicated, random access memory block with synchronous write capability. It provides the capability for fast, discrete, large blocks of RAM in each device. This element is configured as indicated in the following table:

Design Element	Depth	Width	Address Bus	Data Bus
RAMB4_S1	4096	1	(11:0)	(0:0)

The enable (EN) pin controls read, write, and reset. When EN is Low, no data is written and the output (DO) retains the last state. When EN is High and reset (RST) is High, DO is cleared during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI. 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. When EN and WE are High, the data on the data input (DI) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data output (DO) reflects the selected (addressed) word. The above description assumes an active High EN, WE, RST, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

This element can be initialized during configuration. Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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	
EN	**RST**	**WE**	**CLK**	**ADDR**	**DI**	**DO**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address. RAM(addr)=RAM contents at address ADDR. data=RAM input data.							

Specifying Initial Contents of a Block RAM -

You can use the INIT_xx attributes to specify an initial value during device configuration. The initialization of each of these elements is set by 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- RAMB4_S1: Virtex/E, Spartan-II/IIE 4k x 1 Single-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_inst : RAMB4_S1
generic map (
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 map (
DO => DO,     -- 1-bit data output
ADDR => ADDR, -- 12-bit address input
CLK => CLK,   -- Clock input
DI => DI,     -- 1-bit data input
EN => EN,     -- RAM enable input
RST => RST,   -- Synchronous reset input
WE => WE      -- RAM write enable input
);

-- End of RAMB4_S1_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S1: Virtex/E, Spartan-II/IIE 4k x 1 Single-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1 #(
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

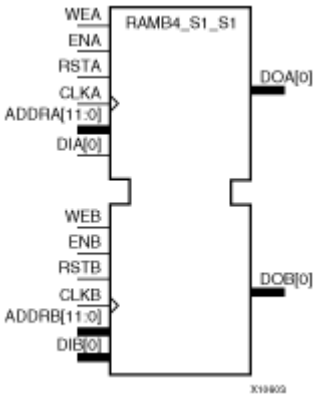
```
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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)
) RAMB4_S1_inst (
.DO(DO),      // 1-bit data output
.ADDR(ADDR), // 12-bit address input
.CLK(CLK),   // Clock input
.DI(DI),      // 1-bit data input
.EN(EN),      // RAM enable input
.RST(RST),   // Synchronous reset input
.WE(WE)      // RAM write enable input
);

// End of RAMB4_S1_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S1_S1

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 1-bit



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S1_S1	4096	1	(11:0)	(0:0)	4096	1	(11:0)	(0:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																	
1	4096	<-----	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;

-- RAMB4_S1_S1: Virtex/E, Spartan-II/IIE 4k x 1 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_S1_inst : RAMB4_S1_S1
generic map (
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 map (
DOA => DOA,     -- Port A 1-bit data output
DOB => DOB,     -- Port B 1-bit data output
ADDRA => ADDRA, -- Port A 12-bit address input
ADDRB => ADDRB, -- Port B 12-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 1-bit data input
DIB => DIB,     -- Port B 1-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S1_S1_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S1_S1: Virtex/E, Spartan-II/IIE 4k x 1 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_S1 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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),
```

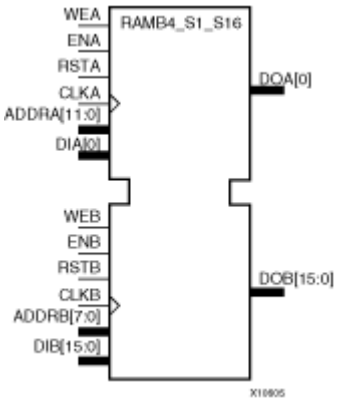
```
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S1_S1_inst (
.DOA(DOA),     // Port A 1-bit data output
.DOB(DOB),     // Port B 1-bit data output
.ADDRA(ADDRA), // Port A 12-bit address input
.ADDRB(ADDRB), // Port B 12-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 1-bit data input
.DIB(DIB),     // Port B 1-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S1_S1_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S1_S16

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 1-bit and 16-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S1_S16	4096	1	(11:0)	(0:0)	256	16	(7:0)	(15:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																	
1	4096	<-----	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16	256	<-----	0															

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- RAMB4_S1_S16: Virtex/E, Spartan-II/IIE 4k/256 x 1/16 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_S16_inst : RAMB4_S1_S16
generic map (
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 map (
DOA => DOA,     -- Port A 1-bit data output
DOB => DOB,     -- Port B 16-bit data output
ADDRA => ADDRA, -- Port A 12-bit address input
ADDRB => ADDRB, -- Port B 8-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 1-bit data input
DIB => DIB,     -- Port B 16-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S1_S16_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S1_S16: Virtex/E, Spartan-II/IIE 4k/256 x 1/16 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_S16 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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),
```

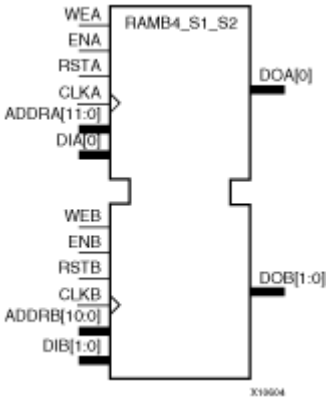
```
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S1_S16_inst (
.DOA(DOA),     // Port A 1-bit data output
.DOB(DOB),     // Port B 16-bit data output
.ADDRA(ADDRA), // Port A 12-bit address input
.ADDRB(ADDRB), // Port B 8-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 1-bit data input
.DIB(DIB),     // Port B 16-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S1_S16_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S1_S2

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 1-bit and 2-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S1_S2	4096	1	(11:0)	(0:0)	2048	2	(10:0)	(1:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																	
1	4096	<-----	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2	2048	<-----	7		6		5		4		3		2		1		0	

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
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. "ALL" issues a warning to simulator console and generate an X or all unknown data due to the collision. This is the recommended setting. "WARNING" generates a warning only and "GENERATE_X_ONLY" generates an X for unknown data but won't 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.
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 "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
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 "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

Attribute	Type	Allowed Values	Default	Description
				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;
```

```
-- RAMB4_S1_S2: Virtex/E, Spartan-II/IIE 4k/2k x 1/2 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_S2_inst : RAMB4_S1_S2
generic map (
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 map (
DOA => DOA,     -- Port A 1-bit data output
DOB => DOB,     -- Port B 2-bit data output
ADDRA => ADDRA, -- Port A 12-bit address input
ADDRB => ADDRB, -- Port B 11-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 1-bit data input
DIB => DIB,     -- Port B 2-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S1_S2_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S1_S2: Virtex/E, Spartan-II/IIE 4k/2k x 1/2 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_S2 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

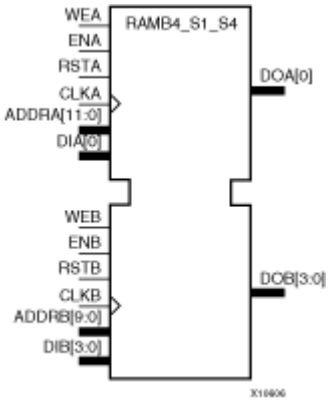
```
.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S1_S2_inst (
.DOA(DOA),     // Port A 1-bit data output
.DOB(DOB),     // Port B 2-bit data output
.ADDRA(ADDRA), // Port A 12-bit address input
.ADDRB(ADDRB), // Port B 11-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 1-bit data input
.DIB(DIB),     // Port B 2-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S1_S2_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S1_S4

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 1-bit and 4-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S1_S4	4096	1	(11:0)	(0:0)	1024	4	(9:0)	(3:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																	
1	4096	<-----	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4	1024	<-----	3				2				1				0			

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;

-- RAMB4_S1_S4: Virtex/E, Spartan-II/IIE 4k/1k x 1/4 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_S4_inst : RAMB4_S1_S4
generic map (
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 map (
DOA => DOA,     -- Port A 1-bit data output
DOB => DOB,     -- Port B 4-bit data output
ADDRA => ADDRA, -- Port A 12-bit address input
ADDRB => ADDRB, -- Port B 10-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
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
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S1_S4_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S1_S4: Virtex/E, Spartan-II/IIE 4k/1k x 1/4 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_S4 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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),
```

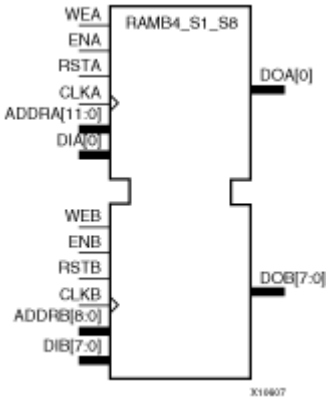
```
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S1_S4_inst (
.DOA(DOA),     // Port A 1-bit data output
.DOB(DOB),     // Port B 4-bit data output
.ADDRA(ADDRA), // Port A 12-bit address input
.ADDRB(ADDRB), // Port B 10-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.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
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S1_S4_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S1_S8

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 1-bit and 8-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S1_S8	4096	1	(11:0)	(0:0)	512	8	(8:0)	(7:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																	
1	4096	<-----	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8	512	<-----	1								0							

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- RAMB4_S1_S8: Virtex/E, Spartan-II/IIE 4k/512 x 1/8 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_S8_inst : RAMB4_S1_S8
generic map (
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 map (
DOA => DOA,     -- Port A 1-bit data output
DOB => DOB,     -- Port B 8-bit data output
ADDRA => ADDRA, -- Port A 12-bit address input
ADDRB => ADDRB, -- Port B 9-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 1-bit data input
DIB => DIB,     -- Port B 8-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S1_S8_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S1_S8: Virtex/E, Spartan-II/IIE 4k/512 x 1/8 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S1_S8 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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),
```

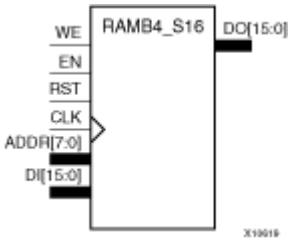
```
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S1_S8_inst (
.DOA(DOA),     // Port A 1-bit data output
.DOB(DOB),     // Port B 8-bit data output
.ADDRA(ADDRA), // Port A 12-bit address input
.ADDRB(ADDRB), // Port B 9-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 1-bit data input
.DIB(DIB),     // Port B 8-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S1_S8_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S16

Primitive: 4096-Bit Single-Port Synchronous Block RAM with Port Width Configured to 16 Bits



Introduction

This design element is a dedicated, random access memory block with synchronous write capability. It provides the capability for fast, discrete, large blocks of RAM in each device. This element is configured as indicated in the following table:

Design Element	Depth	Width	Address Bus	Data Bus
RAMB4_S16	256	16	(7:0)	(15:0)

The enable (EN) pin controls read, write, and reset. When EN is Low, no data is written and the output (DO) retains the last state. When EN is High and reset (RST) is High, DO is cleared during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI. 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. When EN and WE are High, the data on the data input (DI) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data output (DO) reflects the selected (addressed) word. The above description assumes an active High EN, WE, RST, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

This element can be initialized during configuration. Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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	
EN	**RST**	**WE**	**CLK**	**ADDR**	**DI**	**DO**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address. RAM(addr)=RAM contents at address ADDR. data=RAM input data.							

Specifying Initial Contents of a Block RAM -

You can use the INIT_xx attributes to specify an initial value during device configuration. The initialization of each of these elements is set by 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- RAMB4_S16: Virtex/E, Spartan-II/IIE 256 x 16 Single-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S16_inst : RAMB4_S16
generic map (
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 map (
DO => DO,     -- 16-bit data output
ADDR => ADDR, -- 8-bit address input
CLK => CLK,   -- Clock input
DI => DI,     -- 16-bit data input
EN => EN,     -- RAM enable input
RST => RST,   -- Synchronous reset input
WE => WE      -- RAM write enable input
);

-- End of RAMB4_S16_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S16: Virtex/E, Spartan-II/IIE 256 x 16 Single-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S16 #(
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

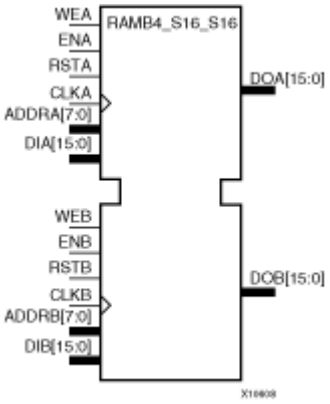
```
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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)
) RAMB4_S16_inst (
.DO(DO),      // 16-bit data output
.ADDR(ADDR), // 8-bit address input
.CLK(CLK),   // Clock input
.DI(DI),      // 16-bit data input
.EN(EN),      // RAM enable input
.RST(RST),   // Synchronous reset input
.WE(WE)      // RAM write enable input
);

// End of RAMB4_S16_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S16_S16

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 16-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S16_S16	256	16	(7:0)	(15:0)	256	16	(7:0)	(15:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word.

The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																
16	256	<-----	0														

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- RAMB4_S16_S16: Virtex/E, Spartan-II/IIE 256 x 16 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S16_S16_inst : RAMB4_S16_S16
generic map (
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 map (
DOA => DOA,     -- Port A 16-bit data output
DOB => DOB,     -- Port B 16-bit data output
ADDRA => ADDRA, -- Port A 8-bit address input
ADDRB => ADDRB, -- Port B 8-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 16-bit data input
DIB => DIB,     -- Port B 16-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S16_S16_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S16_S16: Virtex/E, Spartan-II/IIE 256 x 16 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S16_S16 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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),
```

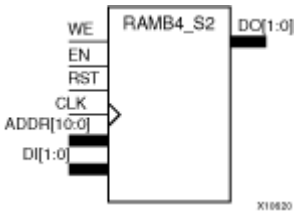
```
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S16_S16_inst (
.DOA(DOA),     // Port A 16-bit data output
.DOB(DOB),     // Port B 16-bit data output
.ADDRA(ADDRA), // Port A 8-bit address input
.ADDRB(ADDRB), // Port B 8-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 16-bit data input
.DIB(DIB),     // Port B 16-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S16_S16_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S2

Primitive: 4K-bit Single-Port Synchronous Block RAM with Port Width Configured to 2-bits



Introduction

This design element is a dedicated, random access memory block with synchronous write capability. It provides the capability for fast, discrete, large blocks of RAM in each device. This element is configured as indicated in the following table:

Design Element	Depth	Width	Address Bus	Data Bus
RAMB4_S2	2048	2	(10:0)	(1:0)

The enable (EN) pin controls read, write, and reset. When EN is Low, no data is written and the output (DO) retains the last state. When EN is High and reset (RST) is High, DO is cleared during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI. 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. When EN and WE are High, the data on the data input (DI) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data output (DO) reflects the selected (addressed) word. The above description assumes an active High EN, WE, RST, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

This element can be initialized during configuration. Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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	
EN	**RST**	**WE**	**CLK**	**ADDR**	**DI**	**DO**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

Specifying Initial Contents of a Block RAM -

You can use the INIT_xx attributes to specify an initial value during device configuration. The initialization of each of these elements is set by 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- RAMB4_S2: Virtex/E, Spartan-II/IIE 2k x 2 Single-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S2_inst : RAMB4_S2
generic map (
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 map (
DO => DO,     -- 2-bit data output
ADDR => ADDR, -- 11-bit address input
CLK => CLK,   -- Clock input
DI => DI,     -- 2-bit data input
EN => EN,     -- RAM enable input
RST => RST,   -- Synchronous reset input
WE => WE      -- RAM write enable input
);

-- End of RAMB4_S2_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S2: Virtex/E, Spartan-II/IIE 2k x 2 Single-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S2 #(
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

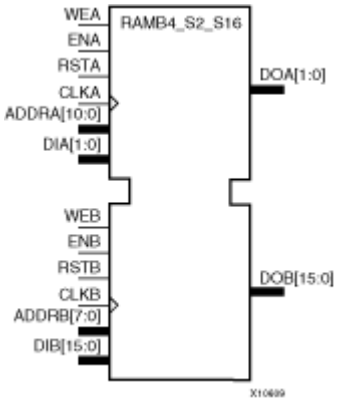
```
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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)
) RAMB4_S2_inst (
.DO(DO),      // 2-bit data output
.ADDR(ADDR), // 11-bit address input
.CLK(CLK),   // Clock input
.DI(DI),      // 2-bit data input
.EN(EN),      // RAM enable input
.RST(RST),   // Synchronous reset input
.WE(WE)      // RAM write enable input
);

// End of RAMB4_S2_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S2_S16

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 2-bits and 16-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S2_S16	2048	2	(10:0)	(1:0)	256	16	(7:0)	(15:0)
ADDR=address bus for the port. DI=data input bus for the port.								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word.

The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																
2	2048	<-----	7		6		5		4		3		2		1		0
16	256	<-----	0														

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No

Coregen 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;

-- RAMB4_S2_S16: Virtex/E, Spartan-II/IIE 2k/256 x 2/16 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S2_S16_inst : RAMB4_S2_S16
generic map (
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 map (
DOA => DOA,     -- Port A 2-bit data output
DOB => DOB,     -- Port B 16-bit data output
ADDRA => ADDRA, -- Port A 11-bit address input
ADDRB => ADDRB, -- Port B 8-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 2-bit data input
DIB => DIB,     -- Port B 16-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S2_S16_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S2_S16: Virtex/E, Spartan-II/IIE 2k/256 x 2/16 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S2_S16 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

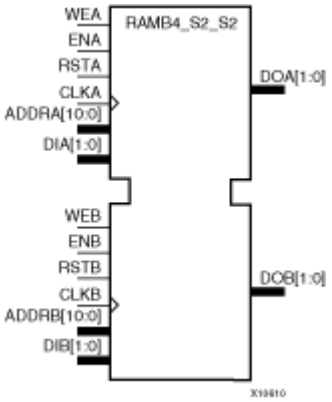
```
.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S2_S16_inst (
.DOA(DOA),     // Port A 2-bit data output
.DOB(DOB),     // Port B 16-bit data output
.ADDRA(ADDRA), // Port A 11-bit address input
.ADDRB(ADDRB), // Port B 8-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 2-bit data input
.DIB(DIB),     // Port B 16-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S2_S16_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S2_S2

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 2-bits and 2-bits



Introduction

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S2_S2	2048	2	(10:0)	(1:0)	2048	2	(10:0)	(1:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word.

The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																	
2	2048	<-----	7		6		5		4		3		2		1		0	

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;

-- RAMB4_S2_S2: Virtex/E, Spartan-II/IIE 2k x 2 Dual-Port RAM
```

```
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S2_S2_inst : RAMB4_S2_S2
generic map (
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 map (
DOA => DOA,     -- Port A 2-bit data output
DOB => DOB,     -- Port B 2-bit data output
ADDRA => ADDRA, -- Port A 11-bit address input
ADDRB => ADDRB, -- Port B 11-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 2-bit data input
DIB => DIB,     -- Port B 2-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S2_S2_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S2_S2: Virtex/E, Spartan-II/IIE 2k x 2 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S2_S2 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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)
) RAMB4_S2_S2_inst (
.DOA(DOA),     // Port A 2-bit data output
.DOB(DOB),     // Port B 2-bit data output
.ADDRA(ADDRA), // Port A 11-bit address input
.ADDRB(ADDRB), // Port B 11-bit address input
.CLKA(CLKA),   // Port A clock input
```

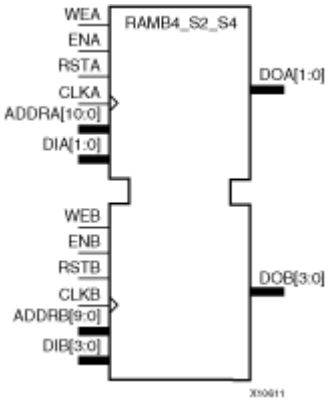
```
.CLKB(CLKB),    // Port B clock input
.DIA(DIA),      // Port A 2-bit data input
.DIB(DIB),      // Port B 2-bit data input
.ENA(ENA),      // Port A RAM enable input
.ENB(ENB),      // Port B RAM enable input
.RSTA(RSTA),    // Port A Synchronous reset input
.RSTB(RSTB),    // Port B Synchronous reset input
.WEA(WEA),      // Port A RAM write enable input
.WEB(WEB)       // Port B RAM write enable input
);

// End of RAMB4_S2_S2_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S2_S4

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 2-bits and 4-bits



Introduction

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S2_S4	2048	2	(10:0)	(1:0)	1024	4	(9:0)	(3:0)
ADDR=address bus for the port. DI=data input bus for the port.								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word.

The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

This component can be initialized during configuration. See the logic table below.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																
2	2048	<-----	7		6		5		4		3		2		1		0
4	1024	<-----	3				2				1				0		

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- RAMB4_S2_S4: Virtex/E, Spartan-II/IIE 2k/1k x 2/4 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S2_S4_inst : RAMB4_S2_S4
generic map (
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 map (
DOA => DOA,     -- Port A 2-bit data output
DOB => DOB,     -- Port B 4-bit data output
ADDRA => ADDRA, -- Port A 11-bit address input
ADDRB => ADDRB, -- Port B 10-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 2-bit data input
DIB => DIB,     -- Port B 4-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S2_S4_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S2_S4: Virtex/E, Spartan-II/IIE 2k/1k x 2/4 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S2_S4 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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),
```

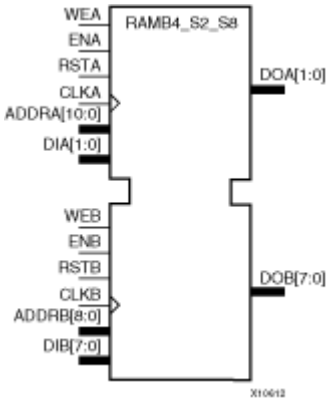
```
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S2_S4_inst (
.DOA(DOA),     // Port A 2-bit data output
.DOB(DOB),     // Port B 4-bit data output
.ADDRA(ADDRA), // Port A 11-bit address input
.ADDRB(ADDRB), // Port B 10-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 2-bit data input
.DIB(DIB),     // Port B 4-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S2_S4_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S2_S8

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 2-bits and 8-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S2_S8	2048	2	(10:0)	(1:0)	512	8	(8:0)	(7:0)
ADDR=address bus for the port. DI=data input bus for the port.								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word.

The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																
2	2048	<-----	7		6		5		4		3		2		1		0
8	512	<-----	1								0						

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- RAMB4_S2_S8: Virtex/E, Spartan-II/IIE 2k/512 x 2/8 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S2_S8_inst : RAMB4_S2_S8
generic map (
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 map (
DOA => DOA,     -- Port A 2-bit data output
DOB => DOB,     -- Port B 8-bit data output
ADDRA => ADDRA, -- Port A 11-bit address input
ADDRB => ADDRB, -- Port B 9-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 2-bit data input
DIB => DIB,     -- Port B 8-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S2_S8_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S2_S8: Virtex/E, Spartan-II/IIE 2k/512 x 2/8 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S2_S8 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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),
```

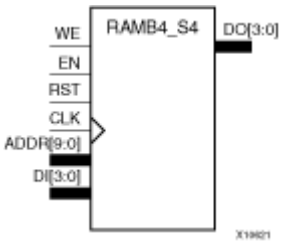
```
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S2_S8_inst (
.DOA(DOA),     // Port A 2-bit data output
.DOB(DOB),     // Port B 8-bit data output
.ADDRA(ADDRA), // Port A 11-bit address input
.ADDRB(ADDRB), // Port B 9-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 2-bit data input
.DIB(DIB),     // Port B 8-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S2_S8_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S4

Primitive: 4k-bit Single-Port Synchronous Block RAM with Port Width Configured to 4-bits



Introduction

This design element is a dedicated, random access memory block with synchronous write capability. It provides the capability for fast, discrete, large blocks of RAM in each device. This element is configured as indicated in the following table:

Design Element	Depth	Width	Address Bus	Data Bus
RAMB4_S4	1024	4	(9:0)	(3:0)

The enable (EN) pin controls read, write, and reset. When EN is Low, no data is written and the output (DO) retains the last state. When EN is High and reset (RST) is High, DO is cleared during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI. 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. When EN and WE are High, the data on the data input (DI) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data output (DO) reflects the selected (addressed) word. The above description assumes an active High EN, WE, RST, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

This element can be initialized during configuration. Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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	
EN	**RST**	**WE**	**CLK**	**ADDR**	**DI**	**DO**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address. RAM(addr)=RAM contents at address ADDR. data=RAM input data.							

Specifying Initial Contents of a Block RAM -

You can use the INIT_xx attributes to specify an initial value during device configuration. The initialization of each of these elements is set by 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

VHDL Instantiation Template

```
-- RAMB4_S4: Virtex/E, Spartan-II/IIE 1k x 4 Single-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S4_inst : RAMB4_S4
generic map (
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 map (
DO => DO,      -- 4-bit data output
ADDR => ADDR,  -- 10-bit address input
CLK => CLK,    -- Clock input
DI => DI,      -- 4-bit data input
EN => EN,      -- RAM enable input
RST => RST,    -- Synchronous reset input
WE => WE       -- RAM write enable input
);

-- End of RAMB4_S4_inst instantiation
```

Verilog Instantiation Template

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

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

```
// RAMB4_S4: Virtex/E, Spartan-II/IIE 1k x 4 Single-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S4 #(
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

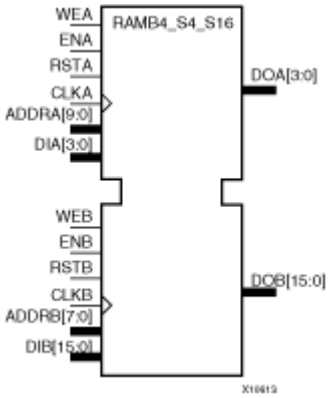
```
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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)
) RAMB4_S4_inst (
.DO(DO),      // 4-bit data output
.ADDR(ADDR), // 10-bit address input
.CLK(CLK),   // Clock input
.DI(DI),      // 4-bit data input
.EN(EN),      // RAM enable input
.RST(RST),   // Synchronous reset input
.WE(WE)      // RAM write enable input
);

// End of RAMB4_S4_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S4_S16

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 4-bits and 16-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S4_S16	1024	4	(9:0)	(3:0)	256	16	(7:0)	(15:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word.

The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																	
4	1024	<-----	3				2				1				0			
16	256	<-----	0															

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No

Coregen 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;

-- RAMB4_S4_S16: Virtex/E, Spartan-II/IIE 1k/256 x 4/16 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S4_S16_inst : RAMB4_S4_S16
generic map (
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 map (
DOA => DOA,     -- Port A 4-bit data output
DOB => DOB,     -- Port B 16-bit data output
ADDRA => ADDRA, -- Port A 10-bit address input
ADDRB => ADDRB, -- Port B 8-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 4-bit data input
DIB => DIB,     -- Port B 16-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S4_S16_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S4_S16: Virtex/E, Spartan-II/IIE 1k/256 x 4/16 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S4_S16 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

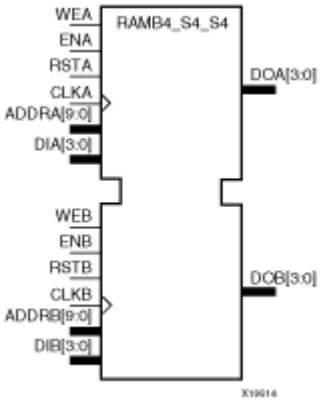
```
.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S4_S16_inst (
.DOA(DOA),     // Port A 4-bit data output
.DOB(DOB),     // Port B 16-bit data output
.ADDRA(ADDRA), // Port A 10-bit address input
.ADDRB(ADDRB), // Port B 8-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 4-bit data input
.DIB(DIB),     // Port B 16-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S4_S16_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S4_S4

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 4-bits and 4-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S4_S4	1024	4	(9:0)	(3:0)	1024	4	(9:0)	(3:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word.

The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																
4	1024	<-----	3				2				1				0			

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No

Coregen 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;

-- RAMB4_S4_S4: Virtex/E, Spartan-II/IIE 1k x 4 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S4_S4_inst : RAMB4_S4_S4
generic map (
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 map (
DOA => DOA,     -- Port A 4-bit data output
DOB => DOB,     -- Port B 4-bit data output
ADDRA => ADDRA, -- Port A 10-bit address input
ADDRB => ADDRB, -- Port B 10-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
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
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S4_S4_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S4_S4: Virtex/E, Spartan-II/IIE 1k x 4 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S4_S4 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

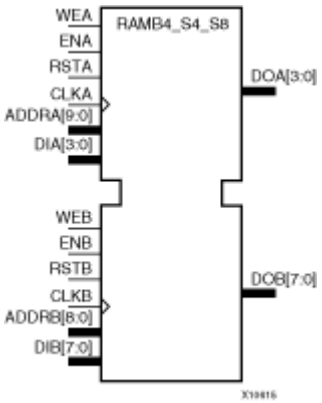
```
.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S4_S4_inst (
.DOA(DOA),     // Port A 4-bit data output
.DOB(DOB),     // Port B 4-bit data output
.ADDRA(ADDRA), // Port A 10-bit address input
.ADDRB(ADDRB), // Port B 10-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.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
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S4_S4_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S4_S8

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 4-bits and 8-bits



Introduction

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S4_S8	1024	4	(9:0)	(3:0)	512	8	(8:0)	(7:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word.

The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																	
4	1024	<-----	3				2				1				0			
8	512	<-----	1								0							

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;

-- RAMB4_S4_S8: Virtex/E, Spartan-II/IIE 1k/512 x 4/8 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S4_S8_inst : RAMB4_S4_S8
generic map (
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 map (
DOA => DOA,     -- Port A 4-bit data output
DOB => DOB,     -- Port B 8-bit data output
ADDRA => ADDRA, -- Port A 10-bit address input
ADDRB => ADDRB, -- Port B 9-bit address input
CLKA => CLKA,   -- Port A clock input
CLKB => CLKB,   -- Port B clock input
DIA => DIA,     -- Port A 4-bit data input
DIB => DIB,     -- Port B 8-bit data input
ENA => ENA,     -- Port A RAM enable input
ENB => ENB,     -- Port B RAM enable input
RSTA => RSTA,   -- Port A Synchronous reset input
RSTB => RSTB,   -- Port B Synchronous reset input
WEA => WEA,     -- Port A RAM write enable input
WEB => WEB      -- Port B RAM write enable input
);

-- End of RAMB4_S4_S8_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S4_S8: Virtex/E, Spartan-II/IIE 1k/512 x 4/8 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S4_S8 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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),
```

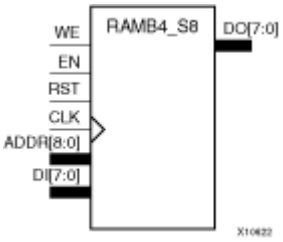
```
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S4_S8_inst (
.DOA(DOA),     // Port A 4-bit data output
.DOB(DOB),     // Port B 8-bit data output
.ADDRA(ADDRA), // Port A 10-bit address input
.ADDRB(ADDRB), // Port B 9-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 4-bit data input
.DIB(DIB),     // Port B 8-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S4_S8_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S8

Primitive: 4k-bit Single-Port Synchronous Block RAM with Port Width Configured to 8-bits



Introduction

This design element is a dedicated, random access memory block with synchronous write capability. It provides the capability for fast, discrete, large blocks of RAM in each device. This element is configured as indicated in the following table:

Design Element	Depth	Width	Address Bus	Data Bus
RAMB4_S8	512	8	(8:0)	(7:0)

The enable (EN) pin controls read, write, and reset. When EN is Low, no data is written and the output (DO) retains the last state. When EN is High and reset (RST) is High, DO is cleared during the Low-to-High clock (CLK) transition; if write enable (WE) is High, the memory contents reflect the data at DI. 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. When EN and WE are High, the data on the data input (DI) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data output (DO) reflects the selected (addressed) word. The above description assumes an active High EN, WE, RST, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

This element can be initialized during configuration. Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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	
EN	**RST**	**WE**	**CLK**	**ADDR**	**DI**	**DO**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address. RAM(addr)=RAM contents at address ADDR. data=RAM input data.							

Specifying Initial Contents of a Block RAM -

You can use the INIT_xx attributes to specify an initial value during device configuration. The initialization of each of these elements is set by 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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;
```

```
-- RAMB4_S8: Virtex/E, Spartan-II/IIE 512 x 8 Single-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S8_inst : RAMB4_S8
generic map (
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 map (
DO => DO,     -- 8-bit data output
ADDR => ADDR, -- 9-bit address input
CLK => CLK,   -- Clock input
DI => DI,     -- 8-bit data input
EN => EN,     -- RAM enable input
RST => RST,   -- Synchronous reset input
WE => WE      -- RAM write enable input
);

-- End of RAMB4_S8_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S8: Virtex/E, Spartan-II/IIE 512 x 8 Single-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S8 #(
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

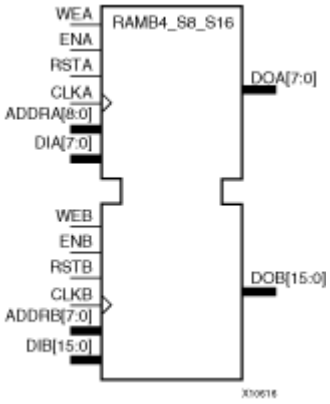
```
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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)
) RAMB4_S8_inst (
.DO(DO),      // 8-bit data output
.ADDR(ADDR), // 9-bit address input
.CLK(CLK),   // Clock input
.DI(DI),      // 8-bit data input
.EN(EN),      // RAM enable input
.RST(RST),   // Synchronous reset input
.WE(WE)      // RAM write enable input
);

// End of RAMB4_S8_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S8_S16

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 8-bits and 16-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S8_S16	512	8	(8:0)	(7:0)	256	16	(7:0)	(15:0)
ADDR=address bus for the port. DI=data input bus for the port.								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word.

The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																	
8	512	<-----	1								0							
16	256	<-----	0															

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No

Coregen 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;

-- RAMB4_S8_S16: Virtex/E, Spartan-II/IIE 512/256 x 8/16 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S8_S16_inst : RAMB4_S8_S16
generic map (
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 map (
DOA => DOA,      -- Port A 8-bit data output
DOB => DOB,      -- Port B 16-bit data output
ADDRA => ADDRA, -- Port A 9-bit address input
ADDRB => ADDRB, -- Port B 8-bit address input
CLKA => CLKA,    -- Port A clock input
CLKB => CLKB,    -- Port B clock input
DIA => DIA,      -- Port A 8-bit data input
DIB => DIB,      -- Port B 16-bit data input
ENA => ENA,      -- Port A RAM enable input
ENB => ENB,      -- Port B RAM enable input
RSTA => RSTA,    -- Port A Synchronous reset input
RSTB => RSTB,    -- Port B Synchronous reset input
WEA => WEA,      -- Port A RAM write enable input
WEB => WEB       -- Port B RAM write enable input
);


-- End of RAMB4_S8_S16_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S8_S16: Virtex/E, Spartan-II/IIE 512/256 x 8/16 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S8_S16 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

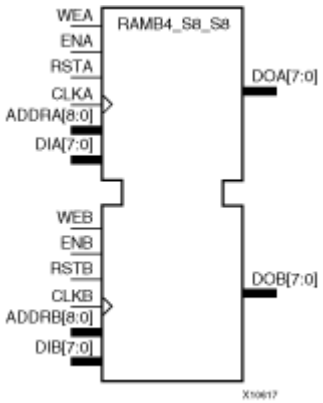
```
.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S8_S16_inst (
.DOA(DOA),     // Port A 8-bit data output
.DOB(DOB),     // Port B 16-bit data output
.ADDRA(ADDRA), // Port A 9-bit address input
.ADDRB(ADDRB), // Port B 8-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 8-bit data input
.DIB(DIB),     // Port B 16-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S8_S16_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

RAMB4_S8_S8

Primitive: 4K-bit Dual-Port Synchronous Block RAM with Port Widths Configured to 8-bits



Introduction

This design element is a 4096-bit dual-ported dedicated random access memory block with synchronous write capability. Each port is independent of the other while accessing the same set of 4096 memory cells. Each port is independently configured to a specific data width, as expressed in the following table:

Design Element	Port A Depth	Port A Width	Port A ADDR	Port A DI	Port B Depth	Port B Width	Port B ADDR	Port B DI
RAMB4_S8_S8	512	8	(8:0)	(7:0)	512	8	(8:0)	(7:0)
ADDR=address bus for the port DI=data input bus for the port								

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 output (DOA) retains the last state. When ENA is High and reset (RSTA) is High, DOA is cleared during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA. 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. When ENA and WEA are High, the data on the data input (DIA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data output (DOA) reflects 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 output (DOB) retains the last state. When ENB is High and reset (RSTB) is High, DOB is cleared during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB. 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. When ENB and WEB are High, the data on the data input (DIB) is loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data output (DOB) reflects the selected (addressed) word.

The above descriptions assume active High control pins (ENA, WEA, RSTA, CLKA, ENB, WEB, RSTB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB4 port is absorbed into the block and does not use a CLB resource.

Block RAM output registers are asynchronously cleared, output Low, when power is applied. The initial contents of the block RAM are not altered. 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.

You can use the INIT_0x attributes to specify an initial value during device configuration. There are 16 initialization attributes (INIT_00 through INIT_0F) of 64 hex values for a total of 4096 bits. If any INIT_0x attribute is not specified, it is configured as zeros. Partial initialization strings are padded with zeros to the left.

Logic Table

Inputs						Outputs	
EN(A/B)	**RST(A/B)**	**WE(A/B)**	**CLK(A/B)**	**ADDR(A/B)**	**DI(A/B)**	**DO(A/B)**	**RAM Contents**
0	X	X	X	X	X	No Change	No Change
1	1	0	↑	X	X	0	No Change
1	1	1	↑	addr	data	0	RAM(addr) =>data
1	0	0	↑	addr	X	RAM(addr)	No Change
1	0	1	↑	addr	data	data	RAM(addr) =>data
addr=RAM address of port A/B RAM(addr)=RAM contents at address ADDRA/ADDRB data=RAM input data at pins DIA/DIB							

Port Descriptions

Address Mapping - Each port accesses the same set of 4096 memory cells using an addressing scheme that is dependent on the width of the port. 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)

Port Width	Port Addresses																
8	512	<-----	1								0						

Port Conflict resolution - This design element is a true dual-ported RAM in that it allows simultaneous reads of the same memory cell. When one port is performing a write to a given memory cell, the other port should not address that memory cell (for a write or a read) within the clock-to-clock setup window.

- If both ports write to the same memory cell simultaneously, violating the clock-to-setup requirement, the data stored will be invalid.
- If one port attempts to read from the same memory cell that the other is simultaneously writing to, violating the clock setup requirement, the write will be successful but the data read will be invalid.

Design Entry Method

Instantiation	Recommended
Inference	No

Coregen 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;

-- RAMB4_S8_S8: Virtex/E, Spartan-II/IIE 512 x 8 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S8_S8_inst : RAMB4_S8_S8
generic map (
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 map (
DOA => DOA,      -- Port A 8-bit data output
DOB => DOB,      -- Port B 8-bit data output
ADDRA => ADDRA,  -- Port A 9-bit address input
ADDRB => ADDRB,  -- Port B 9-bit address input
CLKA => CLKA,    -- Port A clock input
CLKB => CLKB,    -- Port B clock input
DIA => DIA,      -- Port A 8-bit data input
DIB => DIB,      -- Port B 8-bit data input
ENA => ENA,      -- Port A RAM enable input
ENB => ENB,      -- Port B RAM enable input
RSTA => RSTA,    -- Port A Synchronous reset input
RSTB => RSTB,    -- Port B Synchronous reset input
WEA => WEA,      -- Port A RAM write enable input
WEB => WEB       -- Port B RAM write enable input
);

-- End of RAMB4_S8_S8_inst instantiation
```

Verilog Instantiation Template

```
// RAMB4_S8_S8: Virtex/E, Spartan-II/IIE 512 x 8 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB4_S8_S8 #(
.SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
// The following INIT_xx declarations specify the initial contents of the RAM
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

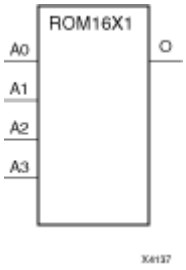
```
.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB4_S8_S8_inst (
.DOA(DOA),     // Port A 8-bit data output
.DOB(DOB),     // Port B 8-bit data output
.ADDRA(ADDRA), // Port A 9-bit address input
.ADDRB(ADDRB), // Port B 9-bit address input
.CLKA(CLKA),   // Port A clock input
.CLKB(CLKB),   // Port B clock input
.DIA(DIA),     // Port A 8-bit data input
.DIB(DIB),     // Port B 8-bit data input
.ENA(ENA),     // Port A RAM enable input
.ENB(ENB),     // Port B RAM enable input
.RSTA(RSTA),   // Port A Synchronous reset input
.RSTB(RSTB),   // Port B Synchronous reset input
.WEA(WEA),     // Port A RAM write enable input
.WEB(WEB)      // Port B RAM write enable input
);

// End of RAMB4_S8_S8_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

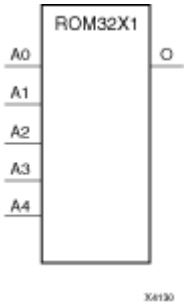
```
// ROM16X1: 16 x 1 Asynchronous Distributed (LUT) ROM
//          All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

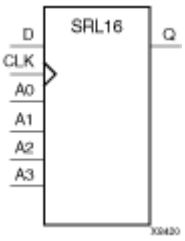
```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
//          All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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: Length = (8 x A3) +(4 x A2) + (2 x 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**
Am	**CLK**	**D**	**Q**
Am	X	X	Q(Am)
Am	↑	D	Q(Am - 1)
m= 0, 1, 2, 3			

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen 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
--        All FPGAs
-- Xilinx HDL Libraries Guide, version 10.1.2

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

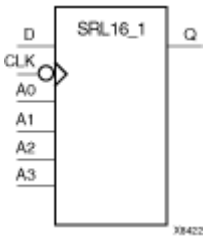
```
// SRL16: 16-bit shift register LUT operating on posedge of clock
//        All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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: Length = (8 x A3) +(4 x A2) + (2 x 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**
Am	**CLK**	**D**	**Q**
Am	X	X	Q(Am)
Am	↓	D	Q(Am - 1)
m= 0, 1, 2, 3			

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen 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
--          All FPGAs
-- Xilinx HDL Libraries Guide, version 10.1.2

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

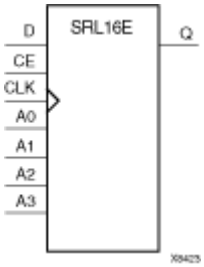
```
// SRL16_1: 16-bit shift register LUT operating on negedge of clock
//          All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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 x A3) +(4 x A2) + (2 x 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**
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

Coregen 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: 16-bit shift register LUT with clock enable operating on posedge of clock
--        All FPGAs
-- Xilinx HDL Libraries Guide, version 10.1.2

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

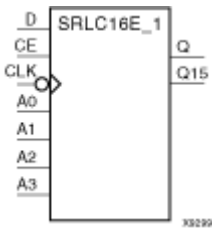
```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//        All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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: Length = (8 x A3) +(4 x A2) + (2 x 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

Coregen 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
--           All FPGAs
-- Xilinx HDL Libraries Guide, version 10.1.2

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

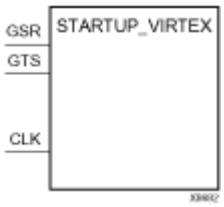
```
// SRL16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
//           All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

STARTUP_VIRTEX

Primitive: Virtex User Interface to Global Clock, Reset, and 3-State Controls



Introduction

This design element is used for Global Set/Reset, global 3-state control, and the user configuration clock. The Global Set/Reset (GSR) input, when High, sets or resets all flip-flops, all latches, and every block RAM (RAMB4) output register in the device, depending on the initialization state (S or R) of the component. For Virtex-II, Virtex-II Pro, and Virtex-II Pro X, see “STARTUP_VIRTEX2”.

Note Block RAMB4 content, LUT RAMs, delay locked loop elements (CLKDLL, CLKDLLHF, BUFGDLL), and shift register LUTs (SRL16, SRL16_1, SRL16E, SRL16E_1) are not set/reset.

Following configuration, the global 3-state control (GTS), when High—and BSCAN is not enabled and executing an EXTEST instruction—forces all the IOB outputs into high impedance mode, which isolates the device outputs from the circuit but leaves the inputs active.

Note GTS= Global 3-State

Including the STARTUP_VIRTEX symbol in a design is optional. You must include the symbol under the following conditions.

- To exert external control over global set/reset, connect the GSR pin to a top level port and an IBUF.
- To exert external control over global 3-state, connect the GTS pin to a top level port and IBUF.
- To synchronize startup to a user clock, connect the user clock signal to the CLK input. Furthermore, “user clock” must be selected in the BitGen program.

You can use location constraints to specify the pin from which GSR or GTS (or both) is accessed.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen 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_VIRTEX: Startup primitive for GSR, GTS or startup sequence
--                 control. Virtex/E, Spartan-IIE
-- Xilinx HDL Libraries Guide, version 10.1.2

STARTUP_VIRTEX_inst : STARTUP_VIRTEX
port map (
```

```
CLK => CLK,       -- Clock input for start-up sequence
GSR => GSR_PORT,  -- Global Set/Reset input (GSR cannot be used for the port name)
GTS => GTS_PORT   -- Global 3-state input(GTS cannot be used for the port name)
);

-- End of STARTUP_VIRTEX_inst instantiation
```

Verilog Instantiation Template

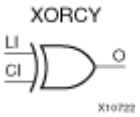
```
// STARTUP_VIRTEX: Startup primitive for GSR, GTS or startup sequence
//                 control. Virtex/E, Spartan-IIE
// Xilinx HDL Libraries Guide, version 10.1.2

STARTUP_VIRTEX STARTUP_VIRTEX_inst (
.CLK(CLK),       // Clock input for start-up sequence
.GSR(GSR_PORT),  // Global Set/Reset input (GSR can not be used as a port name)
.GTS(GTS_PORT)   // Global 3-state input (GTS can not be used as a port name)
);

// End of STARTUP_VIRTEX_inst instantiation
```

For More Information

- See the *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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

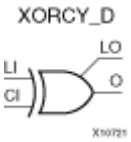
```
// XORCY: Carry-Chain XOR-gate with general output
//        For use with All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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
//          For use with All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.

XORCY_L

Primitive: XOR for Carry Logic with Local Output

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
Coregen 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
-- Xilinx HDL Libraries Guide, version 10.1.2

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) ouput
//          For use with All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Virtex User Guide* and the *Virtex-E User Guide*.
- See the *Virtex Data Sheets* and the *Virtex-E Data Sheets*.