Spartan-3A and Spartan-3A DSP Libraries Guide for HDL Designs

ISE 10.1

Xilinx Trademarks and Copyright Information

Table of Contents

About this Guide 5
Design Element Retargeting 7
Functional Categories 11
About Design Elements 15
- BSCAN_SPARTAN3A 16
- BUFCF 18
- BUFG 20
- BUFGCE 22
- BUFGMUX 24
- CAPTURE_SPARTAN3A 26
- DCM_SP 28
- DNA_PORT 31
- DSP48A 33
- FDCPE 39
- FDCPE_1 42
- FDRSE 44
- FDRSE_1 46
- IBUF 48
- IBUF_DLY_ADJ 51
- IBUFDS 53
- IBUFDS_DLY_ADJ 56
- IBUFG 58
- IBUFGDS 60
- ICAP_SPARTAN3A 63
- IDDR2 65
- IOBUF 68
- IOBUFDS 70
- KEEPER 72
- LDCPE 74
- LUT1 77
- LUT1_D 79
- LUT1_L 82
- LUT2 85
- LUT2_D 87
- LUT2_L 90
- LUT3 92
- LUT3_D 94
- LUT3_L 96
- LUT4 98
- LUT4_D 101
- LUT4_L 104
- MULT_AND 107
- MULT18X18SIO 109
- MUXCY 111
- MUXCY_D 113
- MUXCY_L 115
- MUXF5 117
- MUXF5_D 119
- MUXF5_L 121
- MUXF6 123
- MUXF6_D 125
- MUXF6_L 127
- MUXF7 129
- MUXF7_D 131
- MUXF7_L 133
- MUXF8 135

MUXF8_D 137
MUXF8_L 139
OBUF 141
OBUFDS 143
OBUFT 145
OBUFTDS 147
ODDR2 149
PULLDOWN 152
PULLUP 154
RAM16X1D 156
RAM16X1S 159
RAM32X1S 161
RAM64X1S 163
RAMB16BWE 166
RAMB16BWE_S18 173
RAMB16BWE_S18_S18 179
RAMB16BWE_S18_S9 186
RAMB16BWE_S36 193
RAMB16BWE_S36_S18 199
RAMB16BWE_S36_S36 206
RAMB16BWE_S36_S9 212
RAMB16BWER 219
ROM128X1 223
ROM16X1 225
ROM256X1 227
ROM32X1 230
ROM64X1 232
SPI_ACCESS 234
SRL16 237
SRL16_1 239
SRL16E 241
SRL16E_1 243
SRLC16 245
SRLC16_1 247
SRLC16E 249
SRLC16E_1 251
STARTUP_SPARTAN3A 254
XORCY 256
XORCY_D 258
XORCY_L 260

About this Guide

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

This guide contains the following:

- A general introduction to the design elements, including descriptions of the three types of elements encompassed within this architecture.
- A list of *retargeted elements*, the pre-existing design elements that are automatically changed by the ISE software tools when they are used in this architecture. Retargeting ensures that you are always able to take full advantage of the latest circuit design advances.
- A list of the design elements that are supported in this architecture, organized by functional categories. Click on the element of your choice to immediately access its profile.
- Individual profiles describing each of the primitives.

About This Architecture

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.

Design Element Retargeting

To ensure that Xilinx customers are able to take full advantage of the latest circuit design advances, certain design elements are automatically changed by the ISE software tools when they are used in this architecture.

The following table lists these elements and the more advanced elements into which they are transformed.

Original Element	Modern Equivalent
BUFGCE_1	BUFGCE + INV
BUFGDLL	DCM_SP + BUFG
BUFGMUX_1	BUFGMUX + INV
BUFGP	BUFG
CAPTURE_SPARTAN3	CAPTURE_SPARTAN3a
CLKDLL	DCM_SP
CLKDLLE	DCM_SP
CLKDLLHF	DCM_SP
FD	FDCPE
FD_1	FDCPE + INV
FDC	FDCPE
FDC_1	FDCPE + INV
FDCE	FDCPE
FDCE_1	FDCPE + INV
FDCP	FDCPE
FDCP_1	FDCPE + INV
FDE	FDCPE
FDE_1	FDCPE + INV
FDPE	FDCPE
FDPE_1	FDCPE + INV
FDR	FDRSE
FDR_1	FDRSE + INV
FDRE	FDRSE
FDRE_1	FDRSE + INV
FDRS	FDRSE
FDRS_1	FDRSE + INV
FDS	FDRSE
FDS_1	FDRSE + INV
FDSE	FDRSE
FDSE_1	FDRSE + INV
LD	LDCPE
LD_1	LDCPE + INV
LDC	LDCPE

Original Element	Modern Equivalent
LDC_1	LDCPE + INV
LDCE	LDCPE
LDCE_1	LDCPE + INV
LDE	LDCPE
LDE_1	LDCPE + INV
LDP	LDCPE
LDP_1	LDCPE + INV
LDPE	LDCPE
LDPE_1	LDCPE + INV
RAM128X1S_1	RAM128x1s + INV on clock
RAM16X1D_1	RAM16X1D + INV on clock
RAM16X1S_1	RAM16X1S + INV on clock
RAM16X2S	2=RAM16x1s
RAM16X4S	4=RAM16x1s
RAM16X8S	8=RAM16x1s
RAM32X1D_1	RAM32x1d + INV on clock
RAM32X1S_1	RAM32x1s + INV on clock
RAM32X2S	2 RAM32x1s
RAM32X4S	4 RAM32x1s
RAM32X8S	8 RAM32x1s
RAM64X1S_1	RAM64x1s + INV on clock
RAM64X2S	2 RAM64x1s
RAMB16_S1_S1	RAMB16BWE
RAMB16_S1_S18	RAMB16BWE
RAMB16_S1_S2	RAMB16BWE
RAMB16_S1_S36	RAMB16BWE
RAMB16_S1_S4	RAMB16BWE
RAMB16_S1_S9	RAMB16BWE
RAMB16_S1	RAMB16BWE
RAMB16_S18_S18	RAMB16BWE
RAMB16_S18_S36	RAMB16BWE
RAMB16_S18	RAMB16BWE
RAMB16_S2_S18	RAMB16BWE
RAMB16_S2_S2	RAMB16BWE
RAMB16_S2_S36	RAMB16BWE
RAMB16_S2_S4	RAMB16BWE
RAMB16_S2_S9	RAMB16BWE
RAMB16_S2	RAMB16BWE

Original Element	Modern Equivalent
RAMB16_S36_S36	RAMB16BWE
RAMB16_S36	RAMB16BWE
RAMB16_S4_S18	RAMB16BWE
RAMB16_S4_S36	RAMB16BWE
RAMB16_S4_S4	RAMB16BWE
RAMB16_S4_S9	RAMB16BWE
RAMB16_S4	RAMB16BWE
RAMB16_S9_S18	RAMB16BWE
RAMB16_S9_S36	RAMB16BWE
RAMB16_S9_S9	RAMB16BWE
RAMB16_S9	RAMB16BWE
RAMB4_S1_S1	RAMB16BWE
RAMB4_S1_S16	RAMB16BWE
RAMB4_S1_S2	RAMB16BWE
RAMB4_S1_S4	RAMB16BWE
RAMB4_S1_S8	RAMB16BWE
RAMB4_S1	RAMB16BWE
RAMB4_S16_S16	RAMB16BWE
RAMB4_S16	RAMB16BWE
RAMB4_S2_S16	RAMB16BWE
RAMB4_S2_S2	RAMB16BWE
RAMB4_S2_S4	RAMB16BWE
RAMB4_S2_S8	RAMB16BWE
RAMB4_S2	RAMB16BWE
RAMB4_S4_S16	RAMB16BWE
RAMB4_S4_S4	RAMB16BWE
RAMB4_S4_S8	RAMB16BWE
RAMB4_S4	RAMB16BWE
RAMB4_S8_S16	RAMB16BWE
RAMB4_S8_S8	RAMB16BWE
STARTUP_SPARTAN3	STARTUP_SPARTAN3a

Functional Categories

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

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

Arithmetic Functions

Design Element	Description
DSP48A	Primitive: Multi-Functional, Cascadable, 48-bit Output, Arithmetic Block
MULT18X18SIO	Primitive: 18 x 18 Cascadable Signed Multiplier with Optional Input and Output Registers, Clock Enable, and Synchronous Reset

Clock Components

Design Element	Description
BUFG	Primitive: Global Clock Buffer
BUFGCE	Primitive: Global Clock Buffer with Clock Enable
BUFGMUX	Primitive: Global Clock MUX Buffer
DCM_SP	Primitive: Digital Clock Manager
IBUFG	Primitive: Dedicated Input Clock Buffer
IBUFGDS	Primitive: Differential Signaling Dedicated Input Clock Buffer and Optional Delay

Config/BSCAN Components

Design Element	Description
BSCAN_SPARTAN3A	Primitive: Spartan-3A Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface
CAPTURE_SPARTAN3A	Primitive: Spartan-3A Register State Capture for Bitstream Readback
DNA_PORT	Primitive: Device DNA Data Access Port
ICAP_SPARTAN3A	Primitive: Internal Configuration Access Port
SPI_ACCESS	Primitive: Internal Logic Access to the Serial Peripheral Interface (SPI) PROM Data
STARTUP_SPARTAN3A	Primitive: Spartan-3A Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface

I/O Components

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

RAM/ROM

Design Element	Description
RAM16X1D	Primitive: 16-Deep by 1-Wide Static Dual Port Synchronous RAM
RAM16X1S	Primitive: 16-Deep by 1-Wide Static Synchronous RAM
RAM32X1S	Primitive: 32-Deep by 1-Wide Static Synchronous RAM
RAM64X1S	Primitive: 64-Deep by 1-Wide Static Synchronous RAM
RAMB16BWE	Primitive: 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM
RAMB16BWE_S18	Primitive: 16K-bit Data and 2K-bit Parity Synchronous Single Port Block RAM with 18-bit Port
RAMB16BWE_S18_S18	Primitive: 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 18-bit Ports
RAMB16BWE_S18_S9	Primitive: 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 18-bit and 9-bit Ports
RAMB16BWE_S36	Primitive: 16K-bit Data and 2K-bit Parity Synchronous Single Port Block RAM with 36-Bit Port
RAMB16BWE_S36_S18	Primitive: 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit and 18-bit Ports
RAMB16BWE_S36_S36	Primitive: 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit Ports
RAMB16BWE_S36_S9	Primitive: 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit and 9-bit Ports

Design Element	Description
RAMB16BWER	Primitive: 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers
ROM128X1	Primitive: 128-Deep by 1-Wide ROM
ROM16X1	Primitive: 16-Deep by 1-Wide ROM
ROM256X1	Primitive: 256-Deep by 1-Wide ROM
ROM32X1	Primitive: 32-Deep by 1-Wide ROM
ROM64X1	Primitive: 64-Deep by 1-Wide ROM

Registers & Latches

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

Shift Register LUT

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

Slice/CLB Primitives

Design Element	Description
BUFCF	Primitive: Fast Connect Buffer

Design Element	Description
LUT1	Primitive: 1-Bit Look-Up-Table with General Output
LUT1_D	Primitive: 1-Bit Look-Up-Table with Dual Output
LUT1_L	Primitive: 1-Bit Look-Up-Table with Local Output
LUT2	Primitive: 2-Bit Look-Up-Table with General Output
LUT2_D	Primitive: 2-Bit Look-Up-Table with Dual Output
LUT2_L	Primitive: 2-Bit Look-Up-Table with Local Output
LUT3	Primitive: 3-Bit Look-Up-Table with General Output
LUT3_D	Primitive: 3-Bit Look-Up-Table with Dual Output
LUT3_L	Primitive: 3-Bit Look-Up-Table with Local Output
LUT4	Primitive: 4-Bit Look-Up-Table with General Output
LUT4_D	Primitive: 4-Bit Look-Up-Table with Dual Output
LUT4_L	Primitive: 4-Bit Look-Up-Table with Local Output
MULT_AND	Primitive: Fast Multiplier AND
MUXCY	Primitive: 2-to-1 Multiplexer for Carry Logic with General Output
MUXCY_D	Primitive: 2-to-1 Multiplexer for Carry Logic with Dual Output
MUXCY_L	Primitive: 2-to-1 Multiplexer for Carry Logic with Local Output
MUXF5	Primitive: 2-to-1 Look-Up Table Multiplexer with General Output
MUXF5_D	Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF5_L	Primitive: 2-to-1 Look-Up Table Multiplexer with Local Output
MUXF6	Primitive: 2-to-1 Look-Up Table Multiplexer with General Output
MUXF6_D	Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF6_L	Primitive: 2-to-1 Look-Up Table Multiplexer with Local Output
MUXF7	Primitive: 2-to-1 Look-Up Table Multiplexer with General Output
MUXF7_D	Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF7_L	Primitive: 2-to-1 Look-Up Table Multiplexer with Local Output
MUXF8	Primitive: 2-to-1 Look-Up Table Multiplexer with General Output
MUXF8_D	Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF8_L	Primitive: 2-to-1 Look-Up Table Multiplexer with Local Output
XORCY	Primitive: XOR for Carry Logic with General Output
XORCY_D	Primitive: XOR for Carry Logic with Dual Output
XORCY_L	Primitive: XOR for Carry Logic with Local Output

About Design Elements

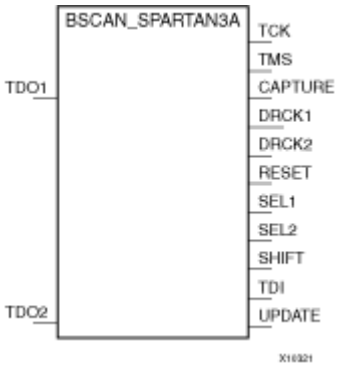
This section describes the design elements that can be used with this architecture. The design elements are organized alphabetically.

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

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

BSCAN_SPARTAN3A

Primitive: Spartan-3A Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface



Introduction

This design element allows access to and from the JTAG Boundary Scan logic controller from internal logic, thus accommodating communications between the internal running design and the dedicated JTAG pins of the FPGA.

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

Port Descriptions

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

Design Entry Method

Instantiation	Recommended
Inference	No

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_SPARTAN3A: Boundary Scan primitive for connecting internal logic to
--                  JTAG interface.
--                  Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

BSCAN_SPARTAN3A_inst : BSCAN_SPARTAN3A
port map (
CAPTURE => CAPTURE, -- CAPTURE output from TAP controller
DRCK1 => DRCK1,     -- Data register output for USER1 functions
DRCK2 => DRCK2,     -- Data register output for USER2 functions
RESET => RESET,     -- Reset output from TAP controller
SEL1 => SEL1,       -- USER1 active output
SEL2 => SEL2,       -- USER2 active output
SHIFT => SHIFT,     -- SHIFT output from TAP controller
TCK => TCK,         -- TCK output from TAP controller
TDI => TDI,         -- TDI output from TAP controller
TMS => TMS,         -- TMS 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_SPARTAN3A_inst instantiation
```

Verilog Instantiation Template

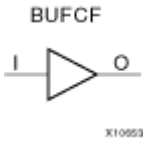
```
// BSCAN_SPARTAN3A: Boundary Scan primitive for connecting internal logic to
//                  JTAG interface.
//                  Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

BSCAN_SPARTAN3A BSCAN_SPARTAN3A_inst (
.CAPTURE(CAPTURE), // CAPTURE output from TAP controller
.DRCK1(DRCK1),     // Data register output for USER1 functions
.DRCK2(DRCK2),     // Data register output for USER2 functions
.RESET(RESET),     // Reset output from TAP controller
.SEL1(SEL1),       // USER1 active output
.SEL2(SEL2),       // USER2 active output
.SHIFT(SHIFT),     // SHIFT output from TAP controller
.TCK(TCK),         // TCK output from TAP controller
.TDI(TDI),         // TDI output from TAP controller
.TMS(TMS),         // TMS 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_SPARTAN3A_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

BUFCF

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

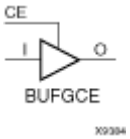
```
// BUFG: Global Clock Buffer (source by an internal signal)
//       All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of BUFG_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

BUFGCE

Primitive: Global Clock Buffer with Clock Enable



Introduction

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

Logic Table

Inputs		Outputs
I	**CE**	**O**
X	0	0
I	1	I

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;
```

```
-- BUFGCE: Global Clock Buffer with Clock Enable (active high)
--         Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of BUFGCE_inst instantiation
```

Verilog Instantiation Template

```
// BUFGCE: Global Clock Buffer with Clock Enable (active high)
//         Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

BUFGCE BUFGCE_inst (
.O(O),   // Clock buffer output
```

```
.CE(CE), // Clock enable input
.I(I)    // Clock buffer input
);

// End of BUFGCE_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

BUFGMUX

Primitive: Global Clock MUX Buffer

Introduction

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

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

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

Logic Table

Inputs			Outputs
I0	**I1**	**S**	**O**
I0	X	0	I0
X	I1	1	I1
X	X	↑	0
X	X	↓	0

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;
```

```
-- BUFGMUX: Global Clock Buffer 2-to-1 MUX
--          Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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

```
-- End of BUFGMUX_inst instantiation
```

Verilog Instantiation Template

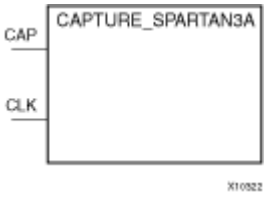
```
// BUFGMUX: Global Clock Buffer 2-to-1 MUX
//          Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of BUFGMUX_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

CAPTURE_SPARTAN3A

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



Introduction

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

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

Port Descriptions

Name	Direction	Width	Function
CAP	Input	1-bit	Readback capture trigger
CLK	Input	1-bit	Readback capture clock

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.

Available Attributes

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

VHDL Instantiation Template

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

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

```
-- CAPTURE_SPARTAN3A: Register State Capture for Bitstream Readback
--                    Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

CAPTURE_SPARTAN3A_inst : CAPTURE_SPARTAN3A
generic map (
ONESHOT => TRUE)  -- TRUE or FALSE
port map (
CAP => CAP,    -- Capture input
CLK => CLK     -- Clock input
);

-- End of CAPTURE_SPARTAN3A_inst instantiation
```

Verilog Instantiation Template

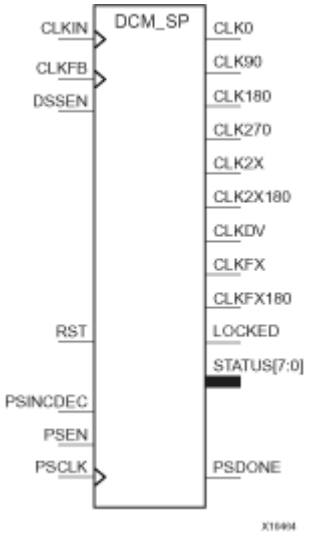
```
// CAPTURE_SPARTAN3A: Register State Capture for Bitstream Readback
//                    Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

CAPTURE_SPARTAN3A  #(
.ONESHOT("TRUE") // "TRUE" or "FALSE"
) CAPTURE_SPARTAN3A_inst (
.CAP(CAP),    // Capture input
.CLK(CLK)     // Clock input
);

// End of CAPTURE_SPARTAN3A_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

DCM_SP

Primitive: Digital Clock Manager



Introduction

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

Design Entry Method

Instantiation	Yes
Inference	No
Coregen and wizards	Recommended
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default
CLK_FEEDBACK	String	"NONE", “2X”, or "1X"	"1X”
CLKDV_DIVIDE	REAL	1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0 or 16.0	2.0
CLKFX_DIVIDE	Integer	1 to 32	1
CLKFX_MULTIPLY	Integer	2 to 32	4
CLKIN_DIVIDE_BY_2	Boolean	FALSE, TRUE	FALSE
CLKIN_PERIOD	REAL	0.0001 to 1000	0

Attribute	Type	Allowed Values	Default
CLKOUT_PHASE_ SHIFT	String	"NONE", "FIXED" or "VARIABLE"	"NONE"
DESKEW_ADJUST	String	"SOURCE_SYNCHRONOUS", "SYSTEM_SYNCHRONOUS" or "0" to "15"	""SYSTEM_SYNCHRONOUS"
FACTORY_JF	16-Bit Hexadecimal	Any 16-Bit Hexadecimal value	C080
PHASE_SHIFT	Integer	-255 to 255	0
DFS_FREQUENCY_MODE	String	"LOW," "HIGH"	"LOW"
DLL_FREQUENCY_MODE	String	"LOW", "HIGH"	"LOW"
DSS_MODE	String	--	"NONE"
DUTY_CYCLE_CORREC-TION	Boolean	TRUE, FALSE	TRUE
STARTUP_WAIT	Boolean	TRUE, FALSE	TRUE

VHDL Instantiation Template

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

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

```
-- DCM_SP: Digital Clock Manager Circuit
--         Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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

```
STATUS => STATUS, -- 8-bit DCM status bits output
CLKFB => CLKFB,   -- DCM clock feedback
CLKIN => CLKIN,   -- Clock input (from IBUFG, BUFG or DCM)
PSCLK => PSCLK,   -- Dynamic phase adjust clock input
PSEN => PSEN,     -- Dynamic phase adjust enable input
PSINCDEC => PSINCDEC, -- Dynamic phase adjust increment/decrement
RST => RST        -- DCM asynchronous reset input
);

-- End of DCM_SP_inst instantiation
```

Verilog Instantiation Template

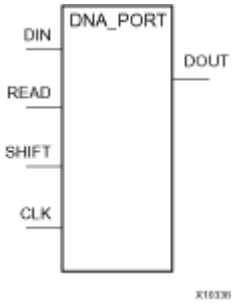
```
// DCM_SP: Digital Clock Manager Circuit
//         Spartan-3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of DCM_SP_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

DNA_PORT

Primitive: Device DNA Data Access Port



Introduction

The DNA_PORT allows access to a dedicated shift register that can be loaded with the Device DNA data bits (unique ID) for a given Spartan-3A device. In addition to shifting out the DNA data bits, this component allows for the inclusion of supplemental data bits for additional data, or allows for the DNA data to rollover (repeat DNA data after initial data has been shifted out). This component is primarily used in conjunction with other circuitry to build added copy protection for the FPGA bitstream from possible theft.

Port Descriptions

Name	Direction	Width	Function
DOUT	Output	1-bit	Serial shifted output data
DIN	Input	1-bit	Your data input to the shift register
READ	Input	1-bit	Synchronous load of the shift register with the Device DNA data
SHIFT	Input	1-bit	Active high shift enable input
CLK	Input	1-bit	Clock Input

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	No
Macro support	No

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

To access the Device DNA data, you must first load the shift register by setting the active high READ signal for one clock cycle. After the shift register is loaded, the data can be synchronously shifted out by enabling the active high SHIFT input and capturing the data out the DOUT output port. Additional data can be appended to the end of the 57-bit shift register by connecting the appropriate logic to the DIN port. If DNA data rollover is desired, connect the DOUT port directly to the DIN port to allow for the same data to be sifted out after completing the 57-bit shift operation. If no additional data is necessary, the DIN port can be tied to a logic zero. The attribute SIM_DNA_VALUE can be optionally set to allow for simulation of a possible DNA data sequence. By default, the Device DNA data bits are all zeros in the simulation model.

Available Attributes

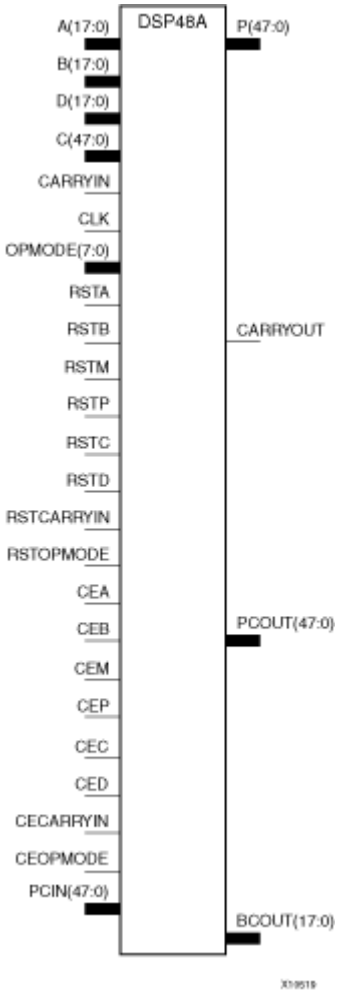
Attribute	Type	Allowed Values	Default	Description
SIM_DNA_VALUE	57-bit vector	Any 57-bit value	All zeros	Specifies a DNA value for simulat device used)

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

DSP48A

Primitive: Multi-Functional, Cascadable, 48-bit Output, Arithmetic Block



Introduction

Note This element is available only for Spartan-3A DSP parts.

The DSP48A is a versatile, scalable, hard IP block that allows for the creation of compact, high-speed, arithmetic-intensive operations, such as those seen for many DSP algorithms. The block consists of a configurable, 18-bit, pre-add/sub, followed by an 18x18 signed multiplier, followed by a 48-bit post-add/sub/accum. Several configurable pipeline registers exist within the block, allowing for higher clock speeds with the trade-off of added latency. Opmode pins allow the block operation to change from one clock-cycle to the next, thus allowing a single block to serve several arithmetic functions within a design. Furthermore, multiple MSP1 blocks can be cascaded to efficiently form larger multiplication and addition functions.

Port Descriptions

Name	Direction	Width (Bits)	Function
Data Ports			
A	Input	18	18-bit data input to multiplier or post add/sub depending on the value of OPMODE[1:0

Name	Direction	Width (Bits)	Function
B	Input	18	18-bit data input to multiplier, pre-add/sub and, perhaps, a post-add/sub depending on the value of OPMODE[3:0].
C	Input	48	48-bit data input to post-add/sub.
D	Input	18	18-bit data input to pre-add/sub.
CARRYIN	Input	1	External carry input to the post-add/sub. Should only be connected to the CARRYOUT pin of another DSP48A block.
P	Output	48	Primary data output.
CARRYOUT	Output	1	Carry out signal for post-add/sub. Should only be connected to the CARRYIN pin of another DSP48A.
Control Inputs			
CLK	Input	1	DSP48A clock
OPMODE	Input	8	Control input to select the arithmetic operations of the DSP48A.
OPMODE[1:0]			Specifies the source of the X input to the post-add/sub 0 - Specifies to place all zeroes (disable the post-add/sub). 1 - Use the POUT output signal. 2 - Use the concatenated D, B, A input signals. 3 - Use the multiplier product.
OPMODE[3:2]			Specifies the source of the Y input to the post-add/sub 0 - Specifies to place all zeroes (disable the post-add/sub and propagate the multiplier product to POUT). 1 - Use the PCIN. 2 - Use the POUT port (accumulator). 3 - Use the C port.
OPMODE[4]			Specifies the use of the pre-add/sub 0 - Selects to use the pre-adder adding or subtracting the values on the B and D ports prior to the multiplier. 1- Bypass the pre-adder, supplying the data on Port B directly to the multiplier.
OPMODE[5]			Force a value on carry-in to the post-adder. Only applicable when CARRYINSEL = "OPMODE5".
OPMODE[6]			Specifies whether the pre-add/sub is an adder or subtracter 0 - Specifies pre-add/sub to perform an addition operation. 1 - Specifies pre-add/sub to perform a subtract operation.
OPMODE[7]			Specifies whether the post-add/sub is an adder or subtracter 0 - Specifies post-add/sub to perform an addition operation. 1 - Specifies post-add/sub to perform a subtract operation.
Reset/Clock Enable Inputs			

Name	Direction	Width (Bits)	Function
RSTA	Input	1	Active high, reset for the A port registers (AREG=1 or 2). Tie to logic zero if not used. This reset is configurable to be synchronous or asynchronous, depending on the value the RSTTYPE attribute.
RSTB	Input	1	Active high, reset for the B port registers (BREG!="NONE"). Tie to logic zero if not use This reset is configurable to be synchronous or asynchronous depending on the value the RSTTYPE attribute.
RSTC	Input	1	Active high, reset for the C input registers (CREG=1). Tie to logic zero if not used. This reset is configurable to be synchronous or asynchronous depending on the value of the RSTTYPE attribute.
RSTD	Input	1	Active high, reset for the D port registers (DREG=1). Tie to logic zero if not used. This reset is configurable to be synchronous or asynchronous depending on the value of the RSTTYPE attribute.
RSTM	Input	1	Active high, reset for the multiplier registers (MREG=1). Tie to logic zero if not used. This reset is configurable to be synchronous or asynchronous depending on the value the RSTTYPE attribute.
RSTP	Input	1	Active high, reset for the P output registers (PREG=1). Tie to logic zero if not used. This reset is configurable to be synchronous or asynchronous depending on the value of the RSTTYPE attribute.
RSTCARRYIN	Input	1	Active high, reset for the carry-in register (CARRYINREG =1). Tie to logic zero if not used. This reset is configurable to be synchronous or asynchronous depending on the value of the RSTTYPE attribute.
RSTOPMODE	Input	1	Active high, reset for the OPMODE registers (OPMODEREG=1). Tie to logic zero if no used. This reset is configurable to be synchronous or asynchronous depending on the value of the RSTTYPE attribute.
CEA	Input	1	Active high, clock enable for the A port registers (AREG=1 or 2). Tie to logic one if not used and AREG=1 or 2. Tie to logic zero if AREG=0.
CEB	Input	1	Active high, clock enable for the B port registers (BREG¼"NONE"). Tie to logic one if used and BREG!="NONE". Tie to logic zero if BREG="NONE".
CEC	Input	1	Active high, clock enable for the C port registers (CREG=1). Tie to logic one if not used and CREG=1. Tie to a logic zero if CREG=0..
CED	Input	1	Active high, clock enable for the D port registers (DREG=1). Tie to logic one if not used and DREG=1. Tie to a logic zero if DREG=0.
CEM	Input	1	Active high, clock enable for the multiplier registers (MREG=1). Tie to logic one if not used and MREG=1. Tie to a logic zero if MREG=0.
CEP	Input	1	Active high, clock enable for the output port registers (PREG=1). Tie to logic one if not used and PREG=1. Tie to a logic zero if PREG=0.
CECARRYIN	Input	1	Active high, clock enable for the carry-in registers (CARRYINREG=1). Tie to logic one not used and CARRYINREG=1. Tie to a logic zero if CARRINREG=0.
CEOPMODE	Input	1	Clock enable for the OPMODE input registers (OPMODEREG=1). Tie to logic one if no used and OPMODEREG=1. Tie to a logic zero if OPMODEREG=0.
Cascade Ports			
PCIN	Input	48	Cascade input for Port P. If used, connect to PCOUT of upstream cascaded DSP48A. If not used, tie port to all zeros.
PCOUT	Output	48	Cascade output for Port P. If used, connect to PCIN of downstream cascaded DSP48A. If not used, leave unconnected.
BCOUT	Output	18	Cascade output for Port B. If used, connect to the B port of downstream cascaded DSP48A. If not used, leave unconnected.

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
A0REG	Integer	0 or 1	0	Specifies to enable/disable the first pipeline stage on t
A1REG	Integer	0 or 1	1	Specifies to enable/disable the second pipeline stage o
B0REG	Integer	0 or 1	0	Specifies to enable/disable the first pipeline stage on t
B1REG	Integer	0 or 1	1	Specifies to enable/disable the second pipeline stage o
CARRYINREG	Integer	0 or 1	1	Selects whether to register the CARRYIN input to the should only be used when CARRYINSEL is set to "CA CARRYIN pin is used.
CARRYINSEL	String	"CARRYIN", "OPMODE5"	"CARRYIN"	Selects whether the post add/sub carry-in signal shou the CARRYIN pin (connected to the CARRYOUT of a dynamically controlled from the FPGA fabric by the C
CREG	Integer	0 or 1	1	Selects whether to register the C input to the DSP48A
DREG	Integer	0 or 1	1	Selects whether to register the D input to the DSP48A
MREG	Integer	0 or 1	1	Selects whether to register the multiplier stage of the
OPMODEREG	Integer	0 or 1	1	Selects whether to register the OPMODE inputs to the
PREG	Integer	0 or 1	1	Selects whether to register the P output of the DSP48A
RSTTYPE	String	ASYNC", "SYNC"	"SYNC"	Selects whether all resets for the DSP48A should have or asynchronous reset capability. Due to improved tim stability, it is recommended to always have this set to asynchronous reset is absolutely necessary.

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;
```

```
-- DSP48A: DSP Function Block
--         Spartan-3A DSP
-- Xilinx HDL Libraries Guide, version 10.1.2

DSP48A_inst : DSP48A
generic map (
A0REG => 1,          -- Enable=1/disable=0 first stage A input pipeline register
A1REG => 1,          -- Enable=1/disable=0 second stage A input pipeline register
B0REG => 1,          -- Enable=1/disable=0 first stage B input pipeline register
B1REG => 1,          -- Enable=1/disable=0 second stage A input pipeline register
CARRYINREG => 1,     -- Enable=1/disable=0 first stage A input pipeline register
CARRYINSEL => "CARRYIN",  -- Specify carry-in source, "CARRYIN" or "OPMODE5"
CREG => 1,           -- Enable=1/disable=0 C input pipeline register
DREG => 1,           -- Enable=1/disable=0 D pre-adder input pipeline register
MREG => 1,           -- Enable=1/disable=0 M pipeline register
OPMODEREG => 1,      -- Enable=1/disable=0 OPMODE input pipeline register
PREG => 1,           -- Enable=1/disable=0 P output pipeline register
```

```
RSTTYPE => "SYNC") -- Specify reset type, "SYNC" or "ASYNC"
port map (
BCOUT => BCOUT,  -- 18-bit B port cascade output
CARRYOUT => CARRYOUT, -- 1-bit carry output
P => P,          -- 48-bit output
PCOUT => PCOUT,  -- 48-bit cascade output
A => A,          -- 18-bit A data input
B => B,          -- 18-bit B data input (can be connected to fabric or BCOUT of adjacent DSP48A)
C => C,          -- 48-bit C data input
CARRYIN => CARRYIN, -- 1-bit carry input signal
CEA => CEA,      -- 1-bit active high clock enable input for A input registers
CEB => CEB,      -- 1-bit active high clock enable input for B input registers
CEC => CEC,      -- 1-bit active high clock enable input for C input registers
CECARRYIN => CECARRYIN, -- 1-bit active high clock enable input for CARRYIN registers
CED => CED,      -- 1-bit active high clock enable input for D input registers
CEM => CEM,      -- 1-bit active high clock enable input for multiplier registers
CEM => CEM,       -- 1-bit active high clock enable input for multiplier registers
CEOPMODE => CEOPMODE, -- 1-bit active high clock enable input for OPMODE registers
CEP => CEP,       -- 1-bit active high clock enable input for P output registers
CLK => CLK,       -- Clock input
D => D,           -- 18-bit B pre-adder data input
OPMODE => OPMODE, -- 8-bit operation mode input
PCIN => PCIN,     -- 48-bit P cascade input
RSTA => RSTA,     -- 1-bit reset input for A input pipeline registers
RSTB => RSTB,     -- 1-bit reset input for B input pipeline registers
RSTC => RSTC,     -- 1-bit reset input for C input pipeline registers
RSTCARRYIN => RSTCARRYIN, -- 1-bit reset input for CARRYIN input pipeline registers
RSTD => RSTD,     -- 1-bit reset input for D input pipeline registers
RSTM => RSTM, -- 1-bit reset input for M pipeline registers
RSTOPMODE => RSTOPMODE, -- 1-bit reset input for OPMODE input pipeline registers
RSTP => RSTP  -- 1-bit reset input for P pipeline registers
);

-- End of DSP48A_inst instantiation
```

Verilog Instantiation Template

```
// DSP48A: DSP Function Block
//         Spartan-3A DSP
// Xilinx HDL Libraries Guide, version 10.1.2

DSP48A #(
.A0REG(0), // Enable=1/disable=0 first stage A input pipeline register
.A1REG(1), // Enable=1/disable=0 second stage A input pipeline register
.B0REG(0), // Enable=1/disable=0 first stage B input pipeline register
.B1REG(1), // Enable=1/disable=0 second stage B input pipeline register
.CARRYINREG(1), // Enable=1/disable=0 CARRYIN input pipeline register
.CARRYINSEL("CARRYIN"), // Specify carry-in source, "CARRYIN" or "OPMODE5"
.CREG(1),  // Enable=1/disable=0 C input pipeline register
.DREG(1),  // Enable=1/disable=0 D pre-adder input pipeline register
.MREG(1),  // Enable=1/disable=0 M pipeline register
.OPMODEREG(1), // Enable=1/disable=0 OPMODE input pipeline register
.PREG(1),  // Enable=1/disable=0 P output pipeline register
.RSTTYPE("SYNC") // Specify reset type, "SYNC" or "ASYNC"
) DSP48A_inst (
.BCOUT(BCOUT), // 18-bit B port cascade output
.CARRYOUT(CARRYOUT), // 1-bit carry output
.P(P), // 48-bit output
.PCOUT(PCOUT), // 48-bit cascade output
.A(A), // 18-bit A data input
.B(B), // 18-bit B data input (can be connected to fabric or BCOUT of adjacent DSP48A)
.C(C), // 48-bit C data input
.CARRYIN(CARRYIN), // 1-bit carry input signal
.CEA(CEA), // 1-bit active high clock enable input for A input registers
.CEB(CEB), // 1-bit active high clock enable input for B input registers
.CEC(CEC), // 1-bit active high clock enable input for C input registers
.CECARRYIN(CECARRYIN), // 1-bit active high clock enable input for CARRYIN registers
.CED(CED), // 1-bit active high clock enable input for D input registers
.CEM(CEM), // 1-bit active high clock enable input for multiplier registers
```

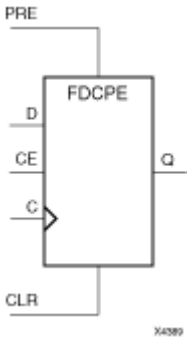
```
.CEOPMODE(CEOPMODE), // 1-bit active high clock enable input for OPMODE registers
.CEP(CEP), // 1-bit active high clock enable input for P output registers
.CLK(CLK), // Clock input
.D(D), // 18-bit B pre-adder data input
.OPMODE(OPMODE), // 8-bit operation mode input
.PCIN(PCIN), // 48-bit P cascade input
.RSTA(RSTA), // 1-bit reset input for A input pipeline registers
.RSTB(RSTB), // 1-bit reset input for B input pipeline registers
.RSTC(RSTC), // 1-bit reset input for C input pipeline registers
.RSTCARRYIN(RSTCARRYIN), // 1-bit reset input for CARRYIN input pipeline registers
.RSTD(RSTD), // 1-bit reset input for D input pipeline registers
.RSTM(RSTM), // 1-bit reset input for M pipeline registers
.RSTOPMODE(RSTOPMODE), // 1-bit reset input for OPMODE input pipeline registers
.RSTP(RSTP) // 1-bit reset input for P output pipeline registers
);

// End of DSP48A_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

Name	Direction	Width	Function
Q	Output	1-Bit	Data output
C	Input	1-Bit	Clock input
CE	Input	1-Bit	Clock enable input
CLR	Input	1-Bit	Asynchronous clear input
D	Input	1-Bit	Data input
PRE	Input	1-Bit	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	1-Bit 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;
```

```
-- 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

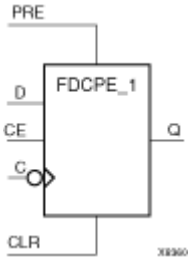
```
// FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
//        Clock Enable (posedge clk).
//        All families.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

Name	Direction	Width	Function
Q	Output	1-Bit	Data output
C	Input	1-Bit	Clock input
CE	Input	1-Bit	Clock enable input
CLR	Input	1-Bit	Asynchronous clear input
D	Input	1-Bit	Data input
PRE	Input	1-Bit	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	1-Bit Binary	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;
```

```
-- 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

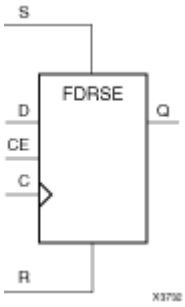
```
// FDCPE_1: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
//          Clock Enable (negedge clock).
//          All families.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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	1-Bit 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

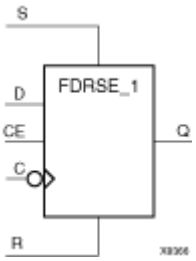
```
// FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//        Clock Enable (posedge clk).
//        All families.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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	1-Bit 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

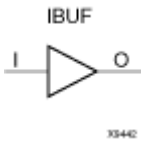
```
// FDRSE_1: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//          Clock Enable (negedge clock).
//          All families.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

Name	Direction	Width	Function
O	Output	1-Bit	Buffer input
I	Input	1-Bit	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 Libaries 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 Note Below	DEFAULT	Sets the programmable I/O standard for the input.
IBUF_DELAY _VALUE	Binary	0 thru 12	0	Specifies the amount of additional delay to add to the non-registered path out of the IOB
IFD_DELAY _VALUE	Binary	AUTO, 0 thru 6	AUTO	Specifies the amount of additional delay to add to the registered path within the IOB

Note Consult the device user guide or databook for the allowed values and the default value.

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

IBUF_DLY_ADJ

Primitive: Dynamically Adjustable Input Delay Buffer

Introduction

This design element is an input buffer with an adjustable delay element allowing dynamic delay adjustment (delay tuning) of an input signal into the FPGA. This is particularly useful for data aligning and capturing of high-speed input signals into the FPGA over process, voltage, and temperature variations. This component consists of a 3-bit select bus, which allows 8 unique values of delay to be added to the incoming signal. Additionally, the component can be programmed with a delay offset to delay adjustment within either the lower 8 or upper 8 of the 16 contiguous delay values.

See "For More Information" for details on the amount of delay and further details about usage of this component.

Port Descriptions

Name	Direction	Width	Function
O	Output	1-bit	Delayed output from the buffer
I	Input	1-bit	Differential input data (positive)
IB	Input	1-bit	Differential input data (negative)
S	Input	3-bit bus	Dynamic delay adjustment select lines

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
IOSTANDARD	String	Consult Data Sheet	"DEFAULT"	Specifies the I/O Standard to be used for this input.
DELAY_OFFSET	String	"OFF" or "ON"	"OFF"	When set to "OFF", the IBUFDS_DLY_ADJ operates at the lower range of delay values. This should be used when a smaller amount of additional delay is needed. When set to "ON", the component operates at the upper (longer) range of delay values. This should be used when a larger amount of additional delay is needed.
DIFF_TERM	Boolean	TRUE or FALSE	FALSE	Specifies the procedure for enabling or disabling (default) the internal differential termination capability.

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_DLY_ADJ: Single-ended Input Buffer
--               Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

IBUF_DLY_ADJ_inst : IBUF_DLY_ADJ
generic map (
DELAY_OFFSET => "OFF",  -- Enable initial Delay Offset, "OFF" or "ON"
IOSTANDARD => "DEFAULT")   -- Specify the input I/O standard
port map (
O => O,  -- Buffer output
I => I,  -- Buffer input (connect directly to top-level port)
S => S   -- 3-bit buffer delay select input
);

-- End of IBUF_DLY_ADJ_inst instantiation
```

Verilog Instantiation Template

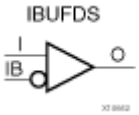
```
// IBUF_DLY_ADJ: Dynamically Adjustable Delay, Single-ended Input Buffer
//               Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

IBUF_DLY_ADJ #(
.DELAY_OFFSET("OFF"),   // Enable Initial Delay Offset, "OFF" or "ON"
.IOSTANDARD("DEFAULT")  // Specify the input I/O standard
)IBUF_DLY_ADJ_inst (
.O(O),   // Buffer output
.I(I),   // Buffer input (connect directly to top-level port)
.S(S)    // 3-bit buffer delay select input
);

// End of IBUF_DLY_ADJ_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

IBUFDS

Primitive: Differential Signaling Input Buffer with Optional Delay



Introduction

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

Logic Table

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

Port Descriptions

Name	Direction	Width	Function
O	Output	1-Bit	Diff_p Buffer input
IB	Input	1-Bit	Diff_n buffer input
I	Input	1-Bit	Buffer output

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

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

Available Attributes

Attribute	Type	Allowed Values	Default	Description
IOSTANDARD	String	See Note Below	DEFAULT	Sets the programmable I/O standard fo
IBUF_DELAY_VALUE	Binary	0 thru 12	0	Specifies the amount of additional dela the IOB
IFD_DELAY_VALUE	Binary	AUTO, 0 thru 6	AUTO	Specifies the amount of additional dela IOB
DIFF_TERM	Boolean	TRUE or FALSE	FALSE	Enables the built-in differential termin

Note Consult the device user guide or databook for the allowed values and the default value.

VHDL Instantiation Template

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

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

-- IBUFDS: Differential Input Buffer
--         Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

IBUFDS_inst : IBUFDS
generic map (
CAPACITANCE => "DONT_CARE", -- "LOW", "NORMAL", "DONT_CARE" (Virtex-4 only)
DIFF_TERM => FALSE, -- Differential Termination (Virtex-4/5, Spartan-3E/3A)
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,  -- Clock buffer output
I => I,  -- Diff_p clock buffer input (connect directly to top-level port)
IB => IB -- Diff_n clock buffer input (connect directly to top-level port)
);

-- End of IBUFDS_inst instantiation
```

Verilog Instantiation Template

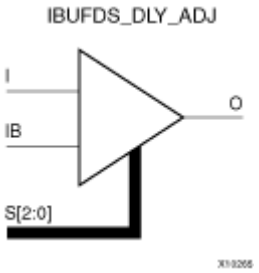
```
// IBUFDS: Differential Input Buffer
//         Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

IBUFDS #(
.CAPACITANCE("DONT_CARE"), // "LOW", "NORMAL", "DONT_CARE" (Virtex-4 only)
.DIFF_TERM("FALSE"),       // Differential Termination (Virtex-4/5, Spartan-3E/3A)
.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/3A only)
.IOSTANDARD("DEFAULT")     // Specify the input I/O standard
) IBUFDS_inst (
.O(O),  // Buffer output
.I(I),  // Diff_p buffer input (connect directly to top-level port)
.IB(IB) // Diff_n buffer input (connect directly to top-level port)
);

// End of IBUFDS_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

IBUFDS_DLY_ADJ

Primitive: Dynamically Adjustable Differential Input Delay Buffer



Introduction

This design element is a differential input buffer with an adjustable delay element allowing dynamic delay adjustment (delay tuning) of an input signal into the FPGA. This is particularly useful for data aligning and capturing of high-speed input signals into the FPGA over process, voltage, and temperature variations. This component consists of a 3-bit select bus, which allows 8 unique values of delay to be added to the incoming signal. Additionally, the component can be programmed with a delay offset to delay adjustment within either the lower 8 or upper 8 of the 16 contiguous delay values.

Port Descriptions

Name	Direction	Width	Function
O	Output	1-bit	Delayed output from the buffer
I	Input	1-bit	Differential input data (positive)
IB	Input	1-bit	Differential input data (negative)
S	Input	3-bit bus	Dynamic delay adjustment select lines

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
IOSTANDARD	String	Consult Data Sheet	"DEFAULT"	Specifies the I/O Standard to be used for this in
DELAY_OFFSET	String	"OFF" or "ON"	"OFF"	When set to OFF", the IBUFDS_DLY_ADJ opera range of delay values. This should be used wh amount of additional delay is needed. When s the component operates at the upper (longer) r values. This should be used when a larger amou delay is needed.
DIFF_TERM	Boolean	TRUE or FALSE	FALSE	Specifies the procedure for enabling or disablin internal differential termination capability.

VHDL Instantiation Template

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

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

```
-- IBUFDS_DLY_ADJ: Differential Input Buffer
--                 Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

IBUFDS_DLY_ADJ_inst : IBUFDS_DLY_ADJ
generic map (
DIFF_TERM => FALSE, -- Differential Termination
DELAY_OFFSET => "OFF",  -- Enable initial Delay Offset, "OFF" or "ON"
IOSTANDARD => "DEFAULT")   -- Specify the input I/O standard
port map (
O => O,  -- Buffer output
I => I,  -- Diff_p buffer input (connect directly to top-level port)
IB => IB, -- Diff_n buffer input (connect directly to top-level port)
S => S,  -- 3-bit buffer delay select input
);

-- End of IBUFDS_DLY_ADJ_inst instantiation
```

Verilog Instantiation Template

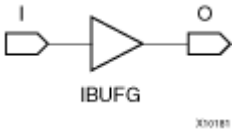
```
// IBUFDS_DLY_ADJ: Dynamically Adjustable Delay, Differential Input Buffer
//                 Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

IBUFDS_DLY_ADJ #(
.DELAY_OFFSET("OFF"),   // Enable Initial Delay Offset, "OFF" or "ON"
.DIFF_TERM("FALSE"),    // Differential Termination
.IOSTANDARD("DEFAULT")  // Specify the input I/O standard
) IBUFDS_DLY_ADJ_inst (
.O(O),   // Buffer output
.I(I),   // Diff_p buffer input (connect directly to top-level port)
.IB(IB), // Diff_n buffer input (connect directly to top-level port)
.S(S)    // 3-bit buffer delay select input
);

// End of IBUFDS_DLY_ADJ_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

Name	Direction	Width	Function
O	Output	1-Bit	Clock Buffer input
I	Input	1-Bit	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 Note Below	"DEFAULT"	Sets the programmable I/O standard for the i
IFD_DELAY_VALUE	Binary	AUTO, 0 thru 8	AUTO	Specifies the amount of additional delay to a within the IOB

Note Consult the device user guide or databook for the allowed values and the default value.

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

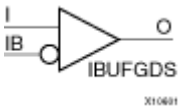
```
// IBUFG: Global Clock Buffer (sourced by an external pin)
//        All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

IBUFGDS

Primitive: Differential Signaling Dedicated Input Clock Buffer and Optional Delay



Introduction

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

Logic Table

Inputs		Outputs
I	**IB**	**O**
0	0	No Change
0	1	0
1	0	1
1	1	No Change

Port Descriptions

Name	Direction	Width	Function
O	Output	1-Bit	Diff_p Clock Buffer Input
IB	Input	1-Bit	Diff_n Clock Buffer Input
I	Input	1-Bit	Clock Buffer output

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

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

Available Attributes

Attribute	Type	Allowed Values	Default	Description
IOSTANDARD	String	See Note Below	"DEFAULT"	Sets the programmable I/O standard for the in
IFD_DELAY_VALUE	Binary	AUTO, 0 thru 6	AUTO	Specifies the amount of additional delay to ad registered path within the IOB.
DIFF_TERM	Boolean	TRUE or FALSE	FALSE	Enables the built-in differential termination re

Note Consult the device user guide or databook for the allowed values and the default value.

VHDL Instantiation Template

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

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

```
-- IBUFGDS: Differential Global Clock Buffer (sourced by an external pin)
--          Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

IBUFGDS_inst : IBUFGDS
generic map (
IOSTANDARD => "DEFAULT")
port map (
O => O,  -- Clock buffer output
I => I,  -- Diff_p clock buffer input
IB => IB -- Diff_n clock buffer input
);

-- End of IBUFGDS_inst instantiation
```

Verilog Instantiation Template

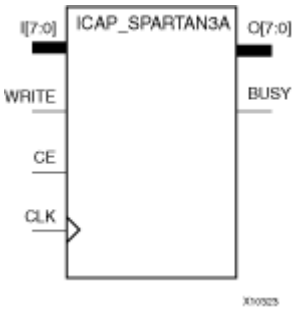
```
// IBUFGDS: Differential Global Clock Buffer (sourced by an external pin)
//          Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

IBUFGDS #(
.DIFF_TERM("FALSE"), // Differential Termination (Virtex-4/5, Spartan-3E/3A)
.IOSTANDARD("DEFAULT") // Specifies the I/O standard for this buffer
) IBUFGDS_inst (
.O(O),  // Clock buffer output
.I(I),  // Diff_p clock buffer input
.IB(IB) // Diff_n clock buffer input
);

// End of IBUFGDS_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

ICAP_SPARTAN3A

Primitive: Internal Configuration Access Port



Introduction

This design element allows users access to the configuration functions of the FPGA from the FPGA fabric. This component's primary usage is to control Multiboot operations in Spartan-3A FPGAs. Using this component, commands and data can be written to and read from the configuration logic of the FPGA array. Because the improper use of this function can have a negative effect on the functionality and reliability of the FPGA, you are encouraged to gain a thorough understanding of this component before incorporating it into your designs.

Port Descriptions

Name	Direction	Width	Function
O	Output	8-bits	Configuration data output bus
Busy	Output	8-bits	Busy output
I	Input	8-bits	Configuration data input bus
WRITE	Input	8-bits	Active Low Write input
CE	Input8-bits	8-bits	Active Low Clock Enable Input
CLK	Input8-bits8-bits	8-bits	Clock Input

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;
```

```
-- ICAP_SPARTAN3A: Internal Configuration Access Port
--                 Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

ICAP_SPARTAN3A_inst : ICAP_SPARTAN3A
port map (
BUSY => BUSY,    -- Busy output
O => O,          -- 8-bit data output
```

```
CE => CE,       -- Clock enable input
CLK => CLK,     -- Clock input
I => I,         -- 8-bit data input
WRITE => WRITE  -- Write input
);

-- End of ICAP_SPARTAN3A_inst instantiation
```

Verilog Instantiation Template

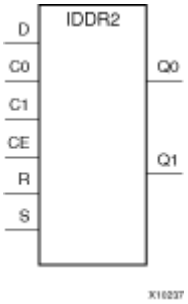
```
// ICAP_SPARTAN3A: Internal Configuration Access Port
//                 Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

ICAP_SPARTAN3A ICAP_SPARTAN3A_inst (
.BUSY(BUSY),   // Busy output
.O(O),         // 8-bit data output
.CE(CE),       // Clock enable input
.CLK(CLK),     // Clock input
.I(I),         // 8-bit data input
.WRITE(WRITE)  // Write input
);

// End of ICAP_SPARTAN3A_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

IDDR2

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



Introduction

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

Logic Table

Input						**Output**	
S	R	CE	D	C0	C1	Q0	Q1
1	x	x	x	x	x	INIT_Q0	INIT_Q1
0	1	x	x	x	x	not INIT_Q0	not INIT_Q1
0	0	0	x	x	x	No Change	No Change
0	0	1	D	Rising	x	D	No Change
0	0	1	D	x	Rising	No Change	D
Set/Reset can be synchronous via SRTYPE value							

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

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

Available Attributes

Attribute	Type	Allowed Values	Default	Description
DDR_ALIGNMENT	String	NONE, "C0" or "C1″	NONE″	Sets the output alignment more for the DDR reg available on the Q0 and Q1 outputs shortly afte C1 positive clock edge. "C0" makes the data on positive edge of the C0 clock. "C1" makes the d to the positive edge of the C1 clock.
INIT_Q0	Integer	0 or 1	0	Sets initial state of the Q0 output to 0 or 1.
INIT_Q1	Integer	0 or 1	0	Sets initial state of the Q1 output to 0 or 1.
SRTYPE	String	"SYNC" or "ASYNC″	"SYNC″	Specifies SYNC" or "ASYNC" set/reset.

VHDL Instantiation Template

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

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

```
-- IDDR2: Input Double Data Rate Input Register with Set, Reset
--       and Clock Enable. Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of IDDR2_inst instantiation
```

Verilog Instantiation Template

```
// IDDR2: Input Double Data Rate Input Register with Set, Reset
//        and Clock Enable.
//        Spartan-3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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

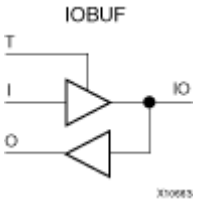
```
);
// End of IDDR2_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

Name	Direction	Width	Function
O	Output	1-Bit	Buffer output
IO	Inout	1-Bit	Buffer inout
I	Input	1-Bit	Buffer input
T	Input	1-Bit	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	Descriptions
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	"DEFAULT"	"DEFAULT"	Use to assign an I/O standard to an I/O primitive.

Attribute	Type	Allowed Values	Default	Descriptions
IBUF_DELAY _VALUE	Binary	0 thru 12	0	Specifies the amount of additional delay to add to the non-registered path out of the IOB
IFD_DELAY _VALUE	Binary	AUTO, 0 thru 6	AUTO	Specifies the amount of additional delay to add to the registered path within the IOB
SLEW	Integer	"SLOW","FAST"	"SLOW"	Sets the output rise and fall time.

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

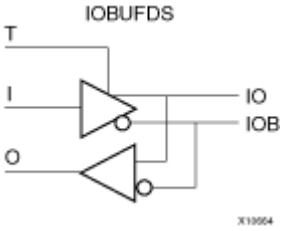
```
// IOBUF: Single-ended Bi-directional Buffer
//        All devices
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

IOBUFDS

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



Introduction

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

Logic Table

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

Port Descriptions

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

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Descriptions
IFD_DELAY_VALUE	String	"AUTO" or 0 to 6	"AUTO"	Specifies the amount of additional delay to add to the registered path within the IOB.
IOSTANDARD	String	"DEFAULT"	"DEFAULT"	Use to assign an I/O standard to an I/O primitive.

VHDL Instantiation Template

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

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

-- IOBUFDS: Differential Bi-directional Buffer
--          Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

IOBUFDS_inst : IOBUFDS
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
IO => IO,   -- Diff_p inout (connect directly to top-level port)
IOB => IOB, -- Diff_n inout (connect directly to top-level port)
I => I,     -- Buffer input
T => T      -- 3-state enable input
);

-- End of IOBUFDS_inst instantiation
```

Verilog Instantiation Template

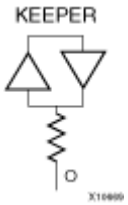
```
// IOBUFDS: Differential Bi-directional Buffer
//          Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

IOBUFDS #(
.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
) IOBUFDS_inst (
.O(O),     // Buffer output
.IO(IO),   // Diff_p inout (connect directly to top-level port)
.IOB(IOB), // Diff_n inout (connect directly to top-level port)
.I(I),     // Buffer input
.T(T)      // 3-state enable input
);

// End of IOBUFDS_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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)
);
```

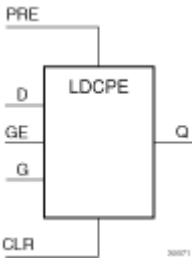
```
// End of KEEPER_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

Name	Direction	Width	Function
Q	Output	1-Bit	Data Output
CLR	Input	1-Bit	Asynchronous clear/reset input
D	Input	1-Bit	Data Input
G	Input	1-Bit	Gate Input
GE	Input	1-Bit	Gate Enable Input
PRE	Input	1-Bit	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

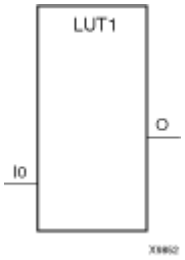
```
// LDCPE: Transparent latch with Asynchronous Reset, Preset and
//        Gate Enable.
//        All families.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

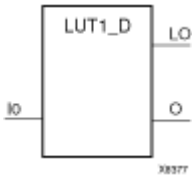
```
// LUT1: 1-input Look-Up Table with general output
//       For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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. *LUTD_1* 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

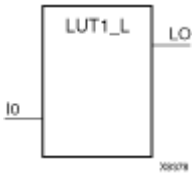
```
// LUT1_D: 1-input Look-Up Table with general and local outputs
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

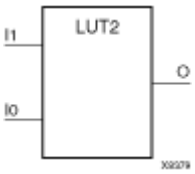
```
// LUT1_L: 1-input Look-Up Table with local output
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

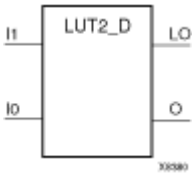
```
// LUT2: 2-input Look-Up Table with general output
//       For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

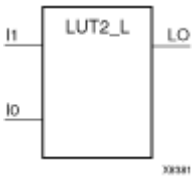
```
// LUT2_D: 2-input Look-Up Table with general and local outputs
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

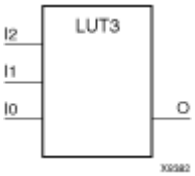
```
// LUT2_L: 2-input Look-Up Table with local output
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

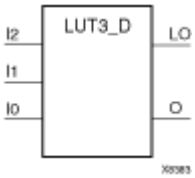
```
// LUT3: 3-input Look-Up Table with general output
//       For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

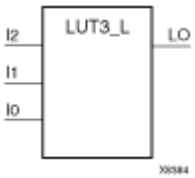
```
// LUT3_D: 3-input Look-Up Table with general and local outputs
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

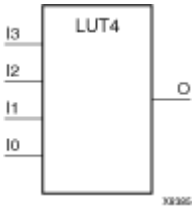
```
// LUT3_L: 3-input Look-Up Table with local output
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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]
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]

Inputs				Outputs
I3	**I2**	**I1**	**I0**	**O**
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
```

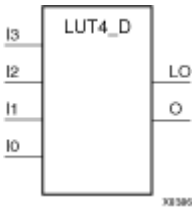
```
);
// End of LUT4_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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]
1	1	0	1	INIT[13]	INIT[13]

Inputs				Outputs	
I3	**I2**	**I1**	**I0**	**O**	**LO**
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
```

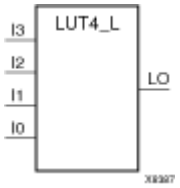
```
);
// End of LUT4_D_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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]
1	1	0	1	INIT[13]

Inputs				Outputs
I3	**I2**	**I1**	**I0**	**LO**
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
);
```

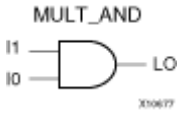
```
// End of LUT4_L_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

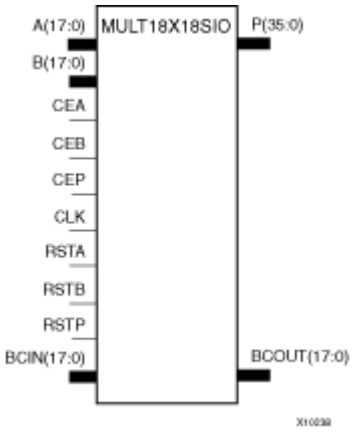
```
// MULT_AND: 2-input AND gate connected to Carry chain
//           For use with all FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

MULT18X18SIO

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



Introduction

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

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	No
Macro support	No

Available Attributes

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

VHDL Instantiation Template

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

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

```
-- MULT18X18SIO: 18 x 18 cascadable, signed synchronous/asynchronous multiplier
--               Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of MULT18X18SIO_inst instantiation
```

Verilog Instantiation Template

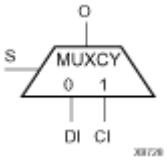
```
// MULT18X18SIO: 18 x 18 cascadable, signed synchronous/asynchronous multiplier
//               Spartan-3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of MULT18X18SIO_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

MUXCY

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



Introduction

This design element is used to implement a 4-bit high-speed carry propagate function. One such function can be implemented per slice, for a total of 4 bits per configurable logic block (CLB) for Spartan-3A.

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

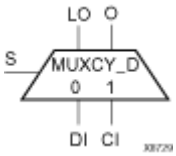
```
// MUXCY: Carry-Chain MUX with general output
//        For use with All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

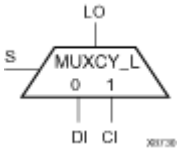
```
// MUXCY_D: Carry-Chain MUX with general and local outputs
//          For use with All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

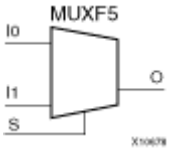
```
// MUXCY_L: Carry-Chain MUX with local output
//        For use with All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

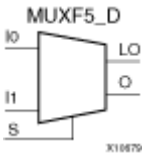
```
// MUXF5: Slice MUX to tie two LUT4's together with general output
//        For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

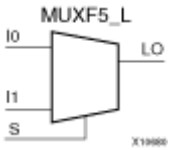
```
// MUXF5_D: Slice MUX to tie two LUT4's together with general and local outputs
//          For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

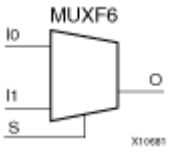
```
// MUXF5_L: Slice MUX to tie two LUT4's together with local output
//          For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

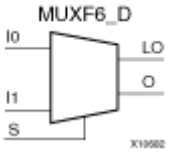
```
// MUXF6: CLB MUX to tie two MUXF5's together with general output
//        For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

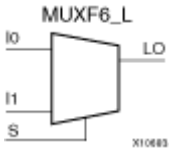
```
// MUXF6_D: CLB MUX to tie two MUXF5's together with general and local outputs
//          For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

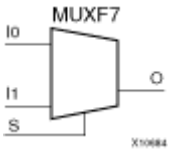
```
// MUXF6_L: CLB MUX to tie two MUXF5's together with local output
//          For use with All FPGAs except Virtex-5
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

MUXF7

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



Introduction

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

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

Logic Table

Inputs			Outputs
S	**I0**	**I1**	**O**
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

Port Descriptions

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

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;

-- MUXF7: CLB MUX to tie two MUXF6's together with general output
```

```
-- Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of MUXF7_inst instantiation
```

Verilog Instantiation Template

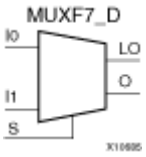
```
// MUXF7: CLB MUX to tie two LUT6's or MUXF6's together with general output
//        For use with Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of MUXF7_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

MUXF7_D

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



Introduction

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

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

Logic Table

Inputs			Outputs	
S	**I0**	**I1**	**O**	**LO**
0	I0	X	I0	I0
1	X	I1	I1	I1
X	0	0	0	0
X	1	1	1	1

Port Descriptions

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

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;
```

```
-- MUXF7_D: CLB MUX to tie two MUXF6's together with general and local outputs
-- Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

MUXF7_D_inst : MUXF7_D
port map (
LO => LO,  -- Ouptut of MUX to local routing
O => O,    -- Output of MUX to general routing
I0 => I0,  -- Input (tie to MUXF6 LO out)
I1 => I1,  -- Input (tie to MUXF6 LO out)
S => S     -- Input select to MUX
);

-- End of MUXF7_D_inst instantiation
```

Verilog Instantiation Template

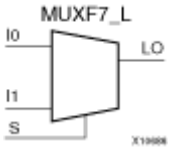
```
// MUXF7_D: CLB MUX to tie two LUT6's or MUXF6's together with general and local outputs
//          For use with Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

MUXF7_D MUXF7_D_inst (
.LO(LO),  // Ouptut of MUX to local routing
.O(O),    // Output of MUX to general routing
.I0(I0),  // Input (tie to MUXF6 LO out)
.I1(I1),  // Input (tie to MUXF6 LO out)
.S(S)     // Input select to MUX
);

// End of MUXF7_D_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

MUXF7_L

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



Introduction

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

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

Logic Table

Inputs			Output
S	**I0**	**I1**	**LO**
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

Port Descriptions

Name	Direction	Width	Function
LO	Output	1-Bit	Output of MUX to local routing
I0	Input	1-Bit	Input (tie to MUXF6 LO out)
I1	Input	1-Bit	Input (tie to MUXF6 LO out)
S	Input	1-Bit	Input select to MUX

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;

-- MUXF7_L: CLB MUX to tie two MUXF6's together with local output
-- Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
```

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

MUXF7_L_inst : MUXF7_L
port map (
LO => LO,  -- Output of MUX to local routing
I0 => I0,  -- Input (tie to MUXF6 LO out)
I1 => I1,  -- Input (tie to MUXF6 LO out)
S => S     -- Input select to MUX
);

-- End of MUXF7_L_inst instantiation
```

Verilog Instantiation Template

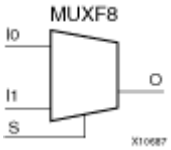
```
// MUXF7_L: CLB MUX to tie two LUT6's or MUXF6's together with local output
//          For use with Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

MUXF7_L MUXF7_L_inst (
.LO(LO),  // Output of MUX to local routing
.I0(I0),  // Input (tie to MUXF6 LO out)
.I1(I1),  // Input (tie to MUXF6 LO out)
.S(S)     // Input select to MUX
);

// End of MUXF7_L_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

MUXF8

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



Introduction

This design element provides a multiplexer function in eight slices for creating a function-of-8 Look-Up Table or a 32-to-1 multiplexer in combination with the associated Look-Up Tables, MUXF5s, MUXF6s, and MUXF7s. Local outputs (LO) of MUXF7 are connected to the I0 and I1 inputs of the MUXF8. The S input is driven from any internal net. When Low, (S) selects I0. When High, (S) selects I1.

Logic Table

Inputs			Outputs
S	**I0**	**I1**	**O**
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

Port Descriptions

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

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;
```

```
-- MUXF8: CLB MUX to tie two MUXF7's together with general output
-- Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2
```

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

-- End of MUXF8_inst instantiation
```

Verilog Instantiation Template

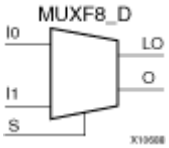
```
// MUXF8: CLB MUX to tie two MUXF7's together with general output
//        For use with Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of MUXF8_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

MUXF8_D

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



Introduction

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

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

Logic Table

Inputs			Outputs	
S	**I0**	**I1**	**O**	**LO**
0	I0	X	I0	I0
1	X	I1	I1	I1
X	0	0	0	0
X	1	1	1	1

Port Descriptions

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

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;
```

```
-- MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
-- Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of MUXF8_D_inst instantiation
```

Verilog Instantiation Template

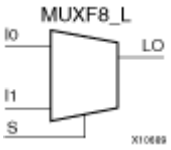
```
// MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
//          For use with Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

MUXF8_D MUXF8_D_inst (
.LO(LO),  // Ouptut of MUX to local routing
.O(O),    // Output of MUX to general routing
.I0(I0),  // Input (tie to MUXF7 LO out)
.I1(I1),  // Input (tie to MUXF7 LO out)
.S(S)     // Input select to MUX
);

// End of MUXF8_D_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

MUXF8_L

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



Introduction

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

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

Logic Table

Inputs			Output
S	**I0**	**I1**	**LO**
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

Port Descriptions

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

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;

-- MUXF8_L: CLB MUX to tie two MUXF7's together with local output
```

```
-- Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of MUXF8_L_inst instantiation
```

Verilog Instantiation Template

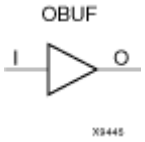
```
// MUXF8_L: CLB MUX to tie two MUXF7's together with local output
//          For use with Virtex-II/II-Pro/4/5 and Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

MUXF8_L MUXF8_L_inst (
.LO(LO),  // Output of MUX to local routing
.I0(I0),  // Input (tie to MUXF7 LO out)
.I1(I1),  // Input (tie to MUXF7 LO out)
.S(S)     // Input select to MUX
);

// End of MUXF8_L_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

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

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	Specifies the output current drive strength of the I/O. It is suggested that you set this to the lowest setting tolerable for the design drive and timing requirements.
IOSTANDARD	String	Consult the product Data Sheet.	"DEFAULT"	Specifies the I/O standard to be used for this output.
SLEW	String	"SLOW" or "FAST"	"SLOW"	Specifies the slew rate of the output driver. Consult the product Data Sheet for recommendations of the best setting for this attribute.

VHDL Instantiation Template

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

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

```
-- OBUF: Single-ended Output Buffer
--       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

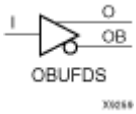
```
// OBUF: Single-ended Output Buffer
//       All devices
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

OBUFDS

Primitive: Differential Signaling Output Buffer



Introduction

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

Logic Table

Inputs		Outputs
I	**O**	**OB**
0	0	1
1	1	0

Port Descriptions

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

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Descriptions
IOSTANDARD	String	"DEFAULT"	"DEFAULT"	Use to assign an I/O standard to an I/O primitive.

VHDL Instantiation Template

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

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

```
-- OBUFDS: Differential Output Buffer
--         Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2
```

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

-- End of OBUFDS_inst instantiation
```

Verilog Instantiation Template

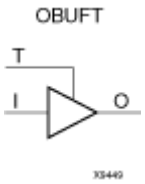
```
// OBUFDS: Differential Output Buffer
//         Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of OBUFDS_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

Name	Direction	Width	Function
O	Output	1-Bit	Buffer output (connect directly to top-level port)
I	Input	1-Bit	Buffer input
T	Input	1-Bit	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	Specifies the output current drive strength of the I/O. It is suggested that you set this to the lowest setting tolerable for the design drive and timing requirements.

Attribute	Type	Allowed Values	Default	Description
IOSTANDARD	String	Consult the product Data Sheet.	"DEFAULT"	Specifies the I/O standard to be used for this output.
SLEW	String	"SLOW" or "FAST"	"SLOW"	Specifies the slew rate of the output driver. Consult the product Data Sheet for recommendations of the best setting for this attribute.

VHDL Instantiation Template

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

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

```
-- 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

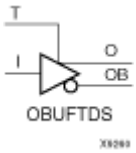
```
// OBUFT: Single-ended 3-state Output Buffer
//        All devices
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

OBUFTDS

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



Introduction

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

Logic Table

Inputs		Outputs	
I	**T**	**O**	**OB**
X	1	Z	Z
0	0	0	1
1	0	1	0

Port Descriptions

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

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Descriptions
IOSTANDARD	String	"DEFAULT"	"DEFAULT"	Use to assign an I/O standard to an I/O primitive.

VHDL Instantiation Template

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

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

```
-- OBUFTDS: Differential 3-state Output Buffer
--          Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of OBUFTDS_inst instantiation
```

Verilog Instantiation Template

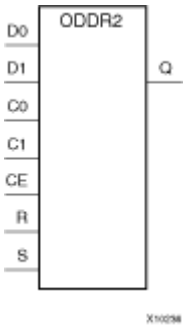
```
// OBUFTDS: Differential 3-state Output Buffer
//          Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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

// End of OBUFTDS_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

ODDR2

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



Introduction

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

Logic Table

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

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

Available Attributes

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

VHDL Instantiation Template

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

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

```
-- ODDR2: Output Double Data Rate Output Register with Set, Reset
--       and Clock Enable. Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

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

-- End of ODDR2_inst instantiation
```

Verilog Instantiation Template

```
// ODDR2: Output Double Data Rate Output Register with Set, Reset
//        and Clock Enable.
//        Spartan-3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

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

```
);
// End of ODDR2_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

Name	Direction	Width	Function
O	Output	1-Bit	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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

Name	Direction	Width	Function
O	Output	1-Bit	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)
);
```

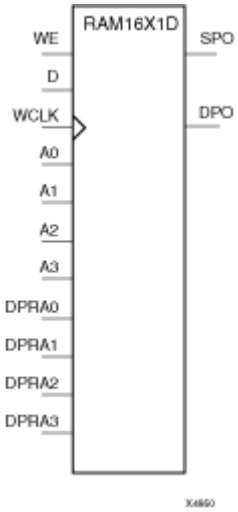
```
// End of PULLUP_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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
```

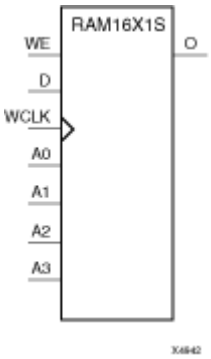
```
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

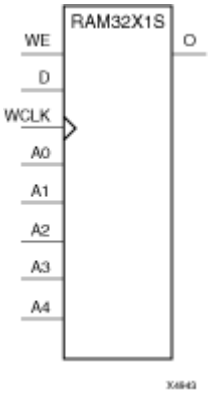
```
// RAM16X1S: 16 x 1 posedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

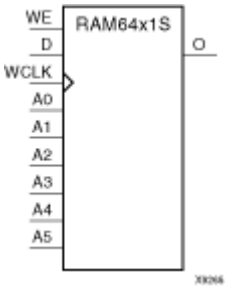
```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

RAM64X1S

Primitive: 64-Deep by 1-Wide Static Synchronous RAM



Introduction

This design element is a 64-word by 1-bit static random access memory (RAM) with synchronous write capability. When the write enable is set Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is set High, any positive transition on WCLK loads the data on the data input (D) into the word selected by the 6-bit address (A5 - A0). This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

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

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

Logic Table

Mode selection is shown in the following logic table

Inputs			**Outputs**
WE (mode)	**WCLK**	**D**	**O**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D	D
1 (read)	↓	X	Data
Data = word addressed by bits A5 – A0			

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 64-Bit Value	All zeros	Initializes ROMs, RAMs, registers, and look-up tables.

VHDL Instantiation Template

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

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

```
-- RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
--           Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

RAM64X1S_inst : RAM64X1S
generic map (
INIT => X"0000000000000000")
port map (
O => O,         -- 1-bit data output
A0 => A0,       -- Address[0] input bit
A1 => A1,       -- Address[1] input bit
A2 => A2,       -- Address[2] input bit
A3 => A3,       -- Address[3] input bit
A4 => A4,       -- Address[4] input bit
A5 => A5,       -- Address[5] input bit
D => D,         -- 1-bit data input
WCLK => WCLK,   -- Write clock input
WE => WE        -- Write enable input
);

-- End of RAM64X1S_inst instantiation
```

Verilog Instantiation Template

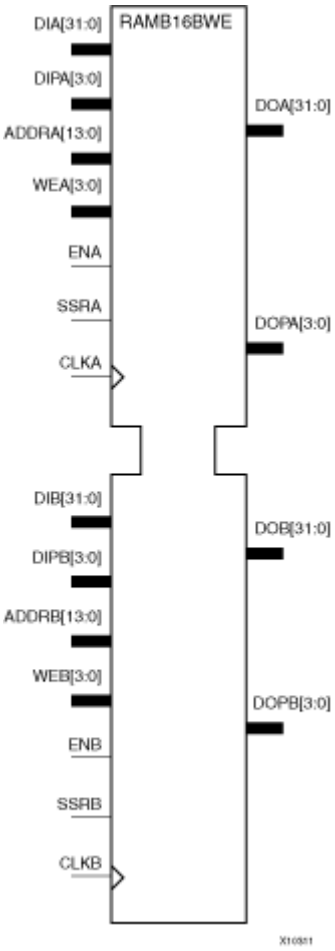
```
// RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
//           Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAM64X1S #(
.INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_inst (
.O(O),         // 1-bit data output
.A0(A0),       // Address[0] input bit
.A1(A1),       // Address[1] input bit
.A2(A2),       // Address[2] input bit
.A3(A3),       // Address[3] input bit
.A4(A4),       // Address[4] input bit
.A5(A5),       // Address[5] input bit
.D(D),         // 1-bit data input
.WCLK(WCLK),   // Write clock input
.WE(WE)        // Write enable input
);

// End of RAM64X1S_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

RAMB16BWE

Primitive: 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM



Introduction

This design element can be configured and used as a 1-bit wide by 16K deep to a 36-bit wide by 512 deep single-port or dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component however Port A and Port B may operate fully independent and asynchronous to each other accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible. This Block RAM memory offers fast and flexible storage of large amounts of on-chip data.

Port Descriptions

Name	Direction	Width (Bits)	Function
DOA, DOB	Output	32-bits	Port A/B data output bus.
DOPA, DOPB	Output	4-bits	Port A/B parity output bus.
DIA, DIB	Input	32-bits	Port A/B data input bus.
DIPA, DIPB	Input	4-bits	Port A/B parity input bus.

Name	Direction	Width (Bits)	Function
ADDRA[13:5], ADDRB[13:5]	Input	14-bits	Port A/B address input bus. LSB always exists on ADDRA/B[0] while the MSB is determined by the settings for DATA_WIDTH_A/B.
WEA[0:0], WEB[0:0]	Input	4-bits	Port A/B byte-wide write enable.
ENA, ENB	Input	1-bit	Port A/B enable
SSRA, SSRB	Input	1-bit	Port A/B output registers synchronous reset.
CLKA, CLKB	Input	1-bit	Port A/B clock input.

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	Yes
Macro support	No

This element can be inferred by most synthesis tools by properly describing the RAM behavior in standard RTL code (consult synthesis tool documentation for details). Alternatively, CORE Generator can also create the desired macro for this RAM. If it is desired to have more control over the implementation or placement of this component, it may also be directly instantiated. In order to instantiate this component, use the ISE HDL Templates or instantiation template below and paste into your code. Connect all necessary inputs to the desired signals in the design. The CLKA/CLKB clock signals must be tied to an active clock for RAM operation and the SSRA/SSRB reset signals must be either tied to a logic zero or to the proper reset signal. ENA/ENB must either be tied to a logic one or a proper RAM port enable signal. Refer to the DATA_WIDTH table below for the necessary data input, data output, write enable and address connection information for each DATA_WIDTH setting as the necessary connections for these signals change based on this attribute. All other output signals can be left unconnected (open) and all unused input signals should be tied to a logic zero.

The following table shows the necessary input and output connections for the variable input ports for each DATA_WIDTH values for either Port A or Port B:

DATA_WIDTH Value	DI, DIP Connections	ADDR Connections	WE Connections	DO, DOP Connections
1	DI[0]	ADDR[13:0]	Connect WE[3:0] to single user WE signal	DO[0]
2	DI[1:0]	ADDR[13:1]	Connect WE[3:0] to single user WE signal	DO[1:0]
4	DI[3:0]	ADDR[13:2]	Connect WE[3:0] to single user WE signal	DO[3:0]
9	DI[7:0], DIP[0]	ADDR[13:3]	Connect WE[3:0] to single user WE signal	DO[7:0], DOP[0]
18	DI[15:0], DIP[1:0]	ADDR[13:4]	Connect WE[0] and WE[2] to user WE[0] and WE[1] and WE[3] to user WE[1]	DO[15:0], DOP[1:0]
36	DI[31:0], DIP[3:0]	ADDR[13:5]	Connect each WE[3:0] signal to the associated byte write enable.	DO[31:0], DOP[3:0]

Alternatively, the prior RAMB16_Sm_Sn design elements can be instantiated if a byte-enable operation is not necessary. Also new convenience macros called RAMB16BWE_Sm_Sn are provided to allow for easier instantiation of this RAM with byte-enable operation. If either of these components is used, the software automatically re-target them to a properly configured RAMB16BWE component.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH_A, DATA_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, or 36	0	Specifies the configurable data width for Ports A and B.
INIT_A, INIT_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the initial value on the Port B output after configuration.
SIM_COLLISION_CHECK	String	ALL, "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"	"ALL"	Allows modification of the simulation behavior so that if a memory collission occurs: "ALL" = warning produced and affected outputs/memory location go unknown (X). WARNING_ONLY = warning produced and affected outputs/memory retain last value. GENERATE_X_ ONLY = no warning, however, affected outputs/memory go unknown (X). NONE = no warning and affected outputs/memory retain last value. *Note:* Setting this to a value other than "ALL" can allow problems in the design to go unnoticed during simulation. Care should be taken when changing the value of this attribute.
SRVAL_A, SRVAL_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the output value of Port B upon the assertion of the synchronous reset (SSRB) signal.
WRITE_MODE_A, WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"	"WRITE_ FIRST"	Specifies output behavior of the port being written to: WRITE_FIRST" = written value appears on output port of the RAM. READ_FIRST = previous RAM contents for that memory location appear on the output port. NO_CHANGE = previous value on the output port remains the same.
INIT_00 to INIT_3F	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 16Kb data memory array.
INITP_00 to INITP_07	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 2Kb parity data memory array.

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;

-- RAMB16BWE: 16k+2k Parity Paramatizable, byte-wide enable BlockRAM
--            Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_inst : RAMB16BWE
generic map (
DATA_WIDTH_A => 0, -- Valid values are 1, 2, 3, 4, 8, 18, or 36
DATA_WIDTH_B => 0, -- Valid values are 1, 2, 3, 4, 8, 18, or 36
INIT_A => X"000000000", --  Initial values on A output port
INIT_B => X"000000000", --  Initial values on B output port
SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
-- "GENERATE_X_ONLY" or "NONE"
SRVAL_A => X"000000000", --  Port A output value upon SSR assertion
SRVAL_B => X"000000000", --  Port B output value upon SSR assertion
WRITE_MODE_A => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
WRITE_MODE_B => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the initial contents of the RAM
INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DOA => DOA,      -- 32-bit A port Data Output
DOB => DOB,      -- 32-bit B port Data Output
DOPA => DOPA,    -- 4-bit  A port Parity Output
DOPB => DOPB,    -- 4-bit  B port Parity Output
ADDRA => ADDRA,  -- 14-bit A port Address Input
ADDRB => ADDRB,  -- 14-bit B port Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- 32-bit A port Data Input
DIB => DIB,      -- 32-bit B port Data Input
DIPA => DIPA,    -- 4-bit  A port parity Input
DIPB => DIPB,    -- 4-bit  B port parity Input
ENA => ENA,      -- 1-bit  A port Enable Input
ENB => ENB,      -- 1-bit  B port Enable Input
SSRA => SSRA,    -- 1-bit  A port Synchronous Set/Reset Input
SSRB => SSRB,    -- 1-bit  B port Synchronous Set/Reset Input
WEA => WEA,      -- 4-bit  A port Write Enable Input
WEB => WEB       -- 4-bit  B port Write Enable Input
);

-- End of RAMB16BWE_inst instantiation
```

Verilog Instantiation Template

```
// RAMB16BWE: 16k+2k Parity Paramatizable, byte-wide enable BlockRAM
//            Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE #(
.DATA_WIDTH_A(0),  // Valid values are 1, 2, 4, 9, 18, or 36
.DATA_WIDTH_B(0),  // Valid values are 1, 2, 4, 9, 18, or 36
.INIT_A(36'h000000000),  // Initial values on A output port
.INIT_B(36'h000000000),  // Initial values on B output port
.SIM_COLLISION_CHECK("ALL"),  // Collision check enable "ALL", "WARNING_ONLY",
//   "GENERATE_X_ONLY" or "NONE"
.SRVAL_A(36'h000000000), // Set/Reset value for A port output
.SRVAL_B(36'h000000000),  // Set/Reset value for B port output
.WRITE_MODE_A("WRITE_FIRST"),  // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
.WRITE_MODE_B("WRITE_FIRST"),  // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"

// 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),
.INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16BWE_inst (
.DOA(DOA),      // 32-bit A port data output
.DOB(DOB),      // 32-bit B port data output
```

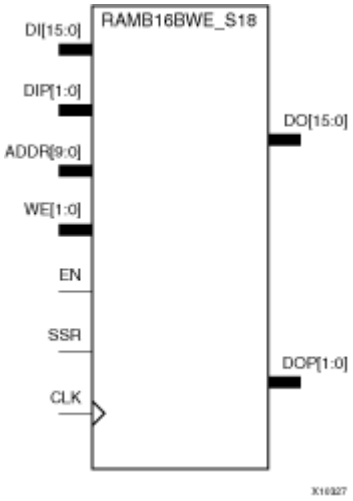
```
.DOPA(DOPA),    // 4-bit A port parity data output
.DOPB(DOPB),    // 4-bit B port parity data output
.ADDRA(ADDRA),  // 14-bit A port address input
.ADDRB(ADDRB),  // 14-bit B port address input
.CLKA(CLKA),    // 1-bit A port clock input
.CLKB(CLKB),    // 1-bit B port clock input
.DIA(DIA),      // 32-bit A port data input
.DIB(DIB),      // 32-bit B port data input
.DIPA(DIPA),    // 4-bit A port parity data input
.DIPB(DIPB),    // 4-bit B port parity data input
.ENA(ENA),      // 1-bit A port enable input
.ENB(ENB),      // 1-bit B port enable input
.SSRA(SSRA),    // 1-bit A port set/reset input
.SSRB(SSRB),    // 1-bit B port set/reset input
.WEA(WEA),      // 4-bit A port write enable input
.WEB(WEB)       // 4-bit B port write enable input
);

// End of RAMB16BWE_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

RAMB16BWE_S18

Primitive: 16K-bit Data and 2K-bit Parity Synchronous Single Port Block RAM with 18-bit Port



Introduction

This design element can be configured and used as a 1-bit wide by 16K deep to a 36-bit wide by 512 deep single-port or dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component however Port A and Port B may operate fully independent and asynchronous to each other accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible. This Block RAM memory offers fast and flexible storage of large amounts of on-chip data.

Port Descriptions

Name	Direction	Width (Bits)	Function
DOA, DOB	Output	32-bits	Port A/B data output bus.
DOPA, DOPB	Output	4-bits	Port A/B parity output bus.
DIA, DIB	Input	32-bits	Port A/B data input bus.
DIPA, DIPB	Input	4-bits	Port A/B parity input bus.
ADDRA[13:5], ADDRB[13:5]	Input	14-bits	Port A/B address input bus. LSB always exists on ADDRA/B[0] while the MSB is determined by the settings for DATA_WIDTH_A/B.
WEA[0:0], WEB[0:0]	Input	4-bits	Port A/B byte-wide write enable.
ENA, ENB	Input	1-bit	Port A/B enable
SSRA, SSRB	Input	1-bit	Port A/B output registers synchronous reset.
CLKA, CLKB	Input	1-bit	Port A/B clock input.

Design Entry Method

Instantiation	Yes
Inference	Recommended

Coregen and wizards	Yes
Macro support	No

This element can be inferred by most synthesis tools by properly describing the RAM behavior in standard RTL code (consult synthesis tool documentation for details). Alternatively, CORE Generator can also create the desired macro for this RAM. If it is desired to have more control over the implementation or placement of this component, it may also be directly instantiated. In order to instantiate this component, use the ISE HDL Templates or instantiation template below and paste into your code. Connect all necessary inputs to the desired signals in the design. The CLKA/CLKB clock signals must be tied to an active clock for RAM operation and the SSRA/SSRB reset signals must be either tied to a logic zero or to the proper reset signal. ENA/ENB must either be tied to a logic one or a proper RAM port enable signal. Refer to the DATA_WIDTH table below for the necessary data input, data output, write enable and address connection information for each DATA_WIDTH setting as the necessary connections for these signals change based on this attribute. All other output signals can be left unconnected (open) and all unused input signals should be tied to a logic zero.

The following table shows the necessary input and output connections for the variable input ports for each DATA_WIDTH values for either Port A or Port B:

DATA_WIDTH Value	DI, DIP Connections	ADDR Connections	WE Connections	DO, DOP Connections
1	DI[0]	ADDR[13:0]	Connect WE[3:0] to single user WE signal	DO[0]
2	DI[1:0]	ADDR[13:1]	Connect WE[3:0] to single user WE signal	DO[1:0]
4	DI[3:0]	ADDR[13:2]	Connect WE[3:0] to single user WE signal	DO[3:0]
9	DI[7:0], DIP[0]	ADDR[13:3]	Connect WE[3:0] to single user WE signal	DO[7:0], DOP[0]
18	DI[15:0], DIP[1:0]	ADDR[13:4]	Connect WE[0] and WE[2] to user WE[0] and WE[1] and WE[3] to user WE[1]	DO[15:0], DOP[1:0]
36	DI[31:0], DIP[3:0]	ADDR[13:5]	Connect each WE[3:0] signal to the associated byte write enable.	DO[31:0], DOP[3:0]

Alternatively, the prior RAMB16_Sm_Sn design elements can be instantiated if a byte-enable operation is not necessary. Also new convenience macros called RAMB16BWE_Sm_Sn are provided to allow for easier instantiation of this RAM with byte-enable operation. If either of these components is used, the software automatically re-target them to a properly configured RAMB16BWE component.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH_A, DATA_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, or 36	0	Specifies the configurable data width for Ports A and B.
INIT_A, INIT_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the initial value on the Port B output after configuration.

Attribute	Type	Allowed Values	Default	Description
SIM_COLLISION_CHECK	String	ALL, "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"	"ALL"	Allows modification of the simulation behavior so that if a memory collission occurs: "ALL" = warning produced and affected outputs/memory location go unknown (X). WARNING_ONLY = warning produced and affected outputs/memory retain last value. GENERATE_X_ ONLY = no warning, however, affected outputs/memory go unknown (X). NONE = no warning and affected outputs/memory retain last value. *Note:* Setting this to a value other than "ALL" can allow problems in the design to go unnoticed during simulation. Care should be taken when changing the value of this attribute.
SRVAL_A, SRVAL_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the output value of Port B upon the assertion of the synchronous reset (SSRB) signal.
WRITE_MODE_A, WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"	"WRITE_ FIRST"	Specifies output behavior of the port being written to: WRITE_FIRST" = written value appears on output port of the RAM. READ_FIRST = previous RAM contents for that memory location appear on the output port. NO_CHANGE = previous value on the output port remains the same.
INIT_00 to INIT_3F	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 16Kb data memory array.
INITP_00 to INITP_07	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 2Kb parity data memory array.

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;
```

```
-- RAMB16BWE_S18: 1k x 16 + 2 Parity bits Single-Port byte-wide write RAM
--                Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S18_inst : RAMB16BWE_S18
generic map (
INIT => X"00000", --  Value of output RAM registers at startup
SRVAL => X"00000", --  Ouput value upon SSR assertion
WRITE_MODE => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the intial contents of the RAM
-- Address 0 to 255
INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 511
INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 512 to 767
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 768 to 1023
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 255
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 511
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 512 to 767
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
-- Address 768 to 1023
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DO => DO,       -- 16-bit Data Output
DOP => DOP,     -- 2-bit parity Output
ADDR => ADDR,   -- 10-bit Address Input
CLK => CLK,     -- 1-bit Clock
DI => DI,       -- 16-bit Data Input
DIP => DIP,     -- 2-bit parity Input
EN => EN,       -- 1-bit RAM Enable Input
SSR => SSR,     -- 1-bit Synchronous Set/Reset Input
WE => WE        -- 2-bit Write Enable Input
);

-- End of RAMB16BWE_S18_inst instantiation
```

Verilog Instantiation Template

```
// RAMB16BWE_S18: 1k x 16 + 2 Parity bits Single-Port byte-wide write RAM
//                Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S18 #(
.INIT(18'h00000),   // Value of output RAM registers at startup
.SRVAL(18'h00000),  // Output value upon SSR assertion
.WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

// The following INIT_xx declarations specify the initial contents of the RAM
// Address 0 to 255
.INIT_00(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_01(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_02(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_03(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_04(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_05(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_06(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_07(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_08(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_09(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
// Address 256 to 511
.INIT_10(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_11(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_12(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_13(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_14(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_15(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_16(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_17(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_18(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_19(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
// Address 512 to 767
.INIT_20(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_21(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_22(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_23(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_24(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
```

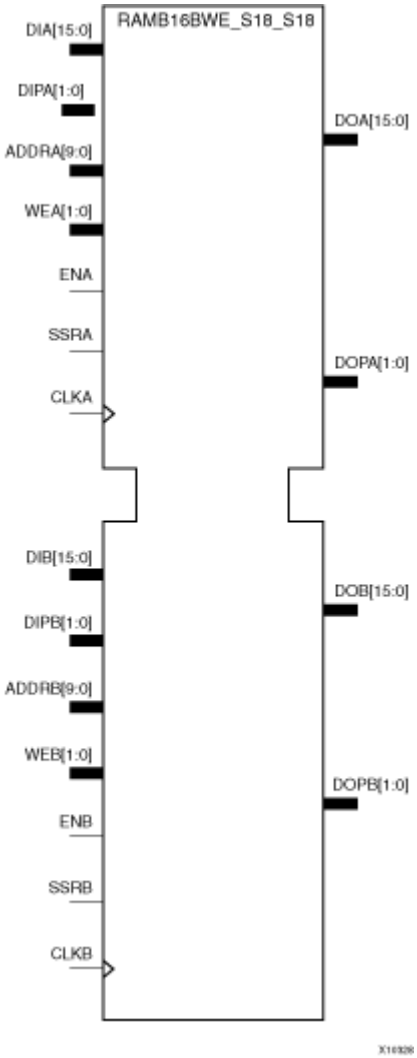
```
.INIT_25(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_26(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_27(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_28(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_29(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
// Address 768 to 1023
.INIT_30(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_31(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_32(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_33(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_34(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_35(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_36(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_37(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_38(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_39(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),

// The next set of INITP_xx are for the parity bits
// Address 0 to 255
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 256 to 511
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 512 to 767
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 768 to 1023
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16BWE_S18_inst (
.DO(DO),      // 16-bit Data Output
.DOP(DOP),    // 2-bit Data Parity Output
.ADDR(ADDR),  // 10-bit Address Input
.CLK(CLK),    // 1-bit Clock
.DI(DI),      // 16-bit Data Input
.DIP(DIP),    // 2-bit parity Input
.EN(EN),      // 1-bit RAM Enable Input
.SSR(SSR),    // 1-bit Synchronous Set/Reset Input
.WE(WE)       // 2-bit Write Enable Input
);

// End of RAMB16BWE_S18_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

RAMB16BWE_S18_S18

Primitive: 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 18-bit Ports



Introduction

This design element can be configured and used as a 1-bit wide by 16K deep to a 36-bit wide by 512 deep single-port or dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component however Port A and Port B may operate fully independent and asynchronous to each other accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible. This Block RAM memory offers fast and flexible storage of large amounts of on-chip data.

Port Descriptions

Name	Direction	Width (Bits)	Function
DOA, DOB	Output	32-bits	Port A/B data output bus.
DOPA, DOPB	Output	4-bits	Port A/B parity output bus.
DIA, DIB	Input	32-bits	Port A/B data input bus.
DIPA, DIPB	Input	4-bits	Port A/B parity input bus.

Name	Direction	Width (Bits)	Function
ADDRA[13:5], ADDRB[13:5]	Input	14-bits	Port A/B address input bus. LSB always exists on ADDRA/B[0] while the MSB is determined by the settings for DATA_WIDTH_A/B.
WEA[0:0], WEB[0:0]	Input	4-bits	Port A/B byte-wide write enable.
ENA, ENB	Input	1-bit	Port A/B enable
SSRA, SSRB	Input	1-bit	Port A/B output registers synchronous reset.
CLKA, CLKB	Input	1-bit	Port A/B clock input.

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	Yes
Macro support	No

This element can be inferred by most synthesis tools by properly describing the RAM behavior in standard RTL code (consult synthesis tool documentation for details). Alternatively, CORE Generator can also create the desired macro for this RAM. If it is desired to have more control over the implementation or placement of this component, it may also be directly instantiated. In order to instantiate this component, use the ISE HDL Templates or instantiation template below and paste into your code. Connect all necessary inputs to the desired signals in the design. The CLKA/CLKB clock signals must be tied to an active clock for RAM operation and the SSRA/SSRB reset signals must be either tied to a logic zero or to the proper reset signal. ENA/ENB must either be tied to a logic one or a proper RAM port enable signal. Refer to the DATA_WIDTH table below for the necessary data input, data output, write enable and address connection information for each DATA_WIDTH setting as the necessary connections for these signals change based on this attribute. All other output signals can be left unconnected (open) and all unused input signals should be tied to a logic zero.

The following table shows the necessary input and output connections for the variable input ports for each DATA_WIDTH values for either Port A or Port B:

DATA_WIDTH Value	DI, DIP Connections	ADDR Connections	WE Connections	DO, DOP Connections
1	DI[0]	ADDR[13:0]	Connect WE[3:0] to single user WE signal	DO[0]
2	DI[1:0]	ADDR[13:1]	Connect WE[3:0] to single user WE signal	DO[1:0]
4	DI[3:0]	ADDR[13:2]	Connect WE[3:0] to single user WE signal	DO[3:0]
9	DI[7:0], DIP[0]	ADDR[13:3]	Connect WE[3:0] to single user WE signal	DO[7:0], DOP[0]
18	DI[15:0], DIP[1:0]	ADDR[13:4]	Connect WE[0] and WE[2] to user WE[0] and WE[1] and WE[3] to user WE[1]	DO[15:0], DOP[1:0]
36	DI[31:0], DIP[3:0]	ADDR[13:5]	Connect each WE[3:0] signal to the associated byte write enable.	DO[31:0], DOP[3:0]

Alternatively, the prior RAMB16_Sm_Sn design elements can be instantiated if a byte-enable operation is not necessary. Also new convenience macros called RAMB16BWE_Sm_Sn are provided to allow for easier instantiation of this RAM with byte-enable operation. If either of these components is used, the software automatically re-target them to a properly configured RAMB16BWE component.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH_A, DATA_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, or 36	0	Specifies the configurable data width for Ports A and B.
INIT_A, INIT_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the initial value on the Port B output after configuration.
SIM_COLLISION_CHECK	String	ALL, "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"	"ALL"	Allows modification of the simulation behavior so that if a memory collission occurs: "ALL" = warning produced and affected outputs/memory location go unknown (X). WARNING_ONLY = warning produced and affected outputs/memory retain last value. GENERATE_X_ ONLY = no warning, however, affected outputs/memory go unknown (X). NONE = no warning and affected outputs/memory retain last value. *Note:* Setting this to a value other than "ALL" can allow problems in the design to go unnoticed during simulation. Care should be taken when changing the value of this attribute.
SRVAL_A, SRVAL_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the output value of Port B upon the assertion of the synchronous reset (SSRB) signal.
WRITE_MODE_A, WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"	"WRITE_ FIRST"	Specifies output behavior of the port being written to: WRITE_FIRST" = written value appears on output port of the RAM. READ_FIRST = previous RAM contents for that memory location appear on the output port. NO_CHANGE = previous value on the output port remains the same.
INIT_00 to INIT_3F	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 16Kb data memory array.
INITP_00 to INITP_07	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 2Kb parity data memory array.

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;

-- RAMB16BWE_S18_S18: 1k x 16 + 2 Parity bits Dual-Port byte-wide write RAM
--                    Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S18_S18_inst : RAMB16BWE_S18_S18
generic map (
INIT_A => X"00000", --  Value of output RAM registers on Port A at startup
INIT_B => X"00000", --  Value of output RAM registers on Port B at startup
SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
SRVAL_A => X"00000", --  Port A ouput value upon SSR assertion
SRVAL_B => X"00000", --  Port B ouput value upon SSR assertion
WRITE_MODE_A => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
WRITE_MODE_B => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the intiial contents of the RAM
-- Address 0 to 255
INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 511
INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 512 to 767
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 768 to 1023
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 255
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 511
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 512 to 767
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 768 to 1023
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DOA => DOA,       -- Port A 16-bit Data Output
DOB => DOB,       -- Port B 16-bit Data Output
DOPA => DOPA,     -- Port A 2-bit Parity Output
DOPB => DOPB,     -- Port B 2-bit Parity Output
ADDRA => ADDRA,   -- Port A 10-bit Address Input
ADDRB => ADDRB,   -- Port B 10-bit Address Input
CLKA => CLKA,     -- Port A 1-bit Clock
CLKB => CLKB,     -- Port B 1-bit Clock
DIA => DIA,       -- Port A 16-bit Data Input
DIB => DIB,       -- Port B 16-bit Data Input
DIPA => DIPA,     -- Port A 2-bit parity Input
DIPB => DIPB,     -- Port-B 2-bit parity Input
ENA => ENA,       -- Port A 1-bit RAM Enable Input
ENB => ENB,       -- Port B 1-bit RAM Enable Input
SSRA => SSRA,     -- Port A 1-bit Synchronous Set/Reset Input
SSRB => SSRB,     -- Port B 1-bit Synchronous Set/Reset Input
WEA => WEA,       -- Port A 2-bit Write Enable Input
WEB => WEB        -- Port B 2-bit Write Enable Input
);

-- End of RAMB16BWE_S18_S18_inst instantiation
```

Verilog Instantiation Template

```
// RAMB16BWE_S18_S18: 1k x 16 + 2 Parity bits Dual-Port byte-wide write RAM
//                    Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S18_S18 #(
.INIT_A(18'h00000),  // Value of output RAM registers on Port A at startup
.INIT_B(18'h00000),  // Value of output RAM registers on Port B at startup
.SIM_COLLISION_CHECK("ALL"),  // Collision check enable "ALL", "WARNING_ONLY",
//   "GENERATE_X_ONLY" or "NONE"
.SRVAL_A(18'h00000), // Port A output value upon SSR assertion
.SRVAL_B(18'h00000), // Port B output value upon SSR assertion
.WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
.WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
.SIM_COLLISION_CHECK("ALL"),  // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

// The following INIT_xx declarations specify the initial contents of the RAM
```

```
// Address 0 to 255
.INIT_00(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_01(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_02(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_03(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_04(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_05(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_06(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_07(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_08(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_09(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_0F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
// Address 256 to 511
.INIT_10(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_11(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_12(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_13(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_14(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_15(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_16(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_17(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_18(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_19(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_1F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
// Address 512 to 767
.INIT_20(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_21(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_22(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_23(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_24(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_25(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_26(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_27(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_28(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_29(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
// Address 768 to 1023
.INIT_30(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_31(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_32(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_33(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_34(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_35(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_36(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_37(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_38(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_39(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),

// The next set of INITP_xx are for the parity bits
// Address 0 to 255
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

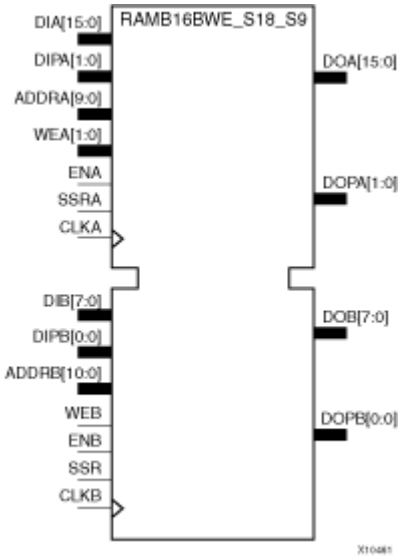
```
// Address 256 to 511
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 512 to 767
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 768 to 1023
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16BWE_S18_S18_inst (
.DOA(DOA),      // Port A 16-bit Data Output
.DOB(DOB),      // Port B 16-bit Data Output
.DOPA(DOPA),    // Port A 2-bit Data Parity Output
.DOPB(DOPB),    // Port B 2-bit Data Parity Output
.ADDRA(ADDRA),  // Port A 10-bit Address Input
.ADDRB(ADDRB),  // Port B 10-bit Address Input
.CLKA(CLKA),    // Port A 1-bit Clock
.CLKB(CLKB),    // Port B 1-bit Clock
.DIA(DIA),      // Port A 16-bit Data Input
.DIB(DIB),      // Port B 16-bit Data Input
.DIPA(DIPA),    // Port A 2-bit parity Input
.DIPB(DIPB),    // Port-B 2-bit parity Input
.ENA(ENA),      // Port A 1-bit RAM Enable Input
.ENB(ENB),      // Port B 1-bit RAM Enable Input
.SSRA(SSRA),    // Port A 1-bit Synchronous Set/Reset Input
.SSRB(SSRB),    // Port B 1-bit Synchronous Set/Reset Input
.WEA(WEA),      // Port A 2-bit Write Enable Input
.WEB(WEB)       // Port B 2-bit Write Enable Input
);

// End of RAMB16BWE_S18_S18_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

RAMB16BWE_S18_S9

Primitive: 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 18-bit and 9-bit Ports



Introduction

This design element can be configured and used as a 1-bit wide by 16K deep to a 36-bit wide by 512 deep single-port or dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component however Port A and Port B may operate fully independent and asynchronous to each other accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible. This Block RAM memory offers fast and flexible storage of large amounts of on-chip data.

Port Descriptions

Name	Direction	Width (Bits)	Function
DOA, DOB	Output	32-bits	Port A/B data output bus.
DOPA, DOPB	Output	4-bits	Port A/B parity output bus.
DIA, DIB	Input	32-bits	Port A/B data input bus.
DIPA, DIPB	Input	4-bits	Port A/B parity input bus.
ADDRA[13:5], ADDRB[13:5]	Input	14-bits	Port A/B address input bus. LSB always exists on ADDRA/B[0] while the MSB is determined by the settings for DATA_WIDTH_A/B.
WEA[0:0], WEB[0:0]	Input	4-bits	Port A/B byte-wide write enable.
ENA, ENB	Input	1-bit	Port A/B enable
SSRA, SSRB	Input	1-bit	Port A/B output registers synchronous reset.
CLKA, CLKB	Input	1-bit	Port A/B clock input.

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	Yes
Macro support	No

This element can be inferred by most synthesis tools by properly describing the RAM behavior in standard RTL code (consult synthesis tool documentation for details). Alternatively, CORE Generator can also create the desired macro for this RAM. If it is desired to have more control over the implementation or placement of this component, it may also be directly instantiated. In order to instantiate this component, use the ISE HDL Templates or instantiation template below and paste into your code. Connect all necessary inputs to the desired signals in the design. The CLKA/CLKB clock signals must be tied to an active clock for RAM operation and the SSRA/SSRB reset signals must be either tied to a logic zero or to the proper reset signal. ENA/ENB must either be tied to a logic one or a proper RAM port enable signal. Refer to the DATA_WIDTH table below for the necessary data input, data output, write enable and address connection information for each DATA_WIDTH setting as the necessary connections for these signals change based on this attribute. All other output signals can be left unconnected (open) and all unused input signals should be tied to a logic zero.

The following table shows the necessary input and output connections for the variable input ports for each DATA_WIDTH values for either Port A or Port B:

DATA_WIDTH Value	DI, DIP Connections	ADDR Connections	WE Connections	DO, DOP Connections
1	DI[0]	ADDR[13:0]	Connect WE[3:0] to single user WE signal	DO[0]
2	DI[1:0]	ADDR[13:1]	Connect WE[3:0] to single user WE signal	DO[1:0]
4	DI[3:0]	ADDR[13:2]	Connect WE[3:0] to single user WE signal	DO[3:0]
9	DI[7:0], DIP[0]	ADDR[13:3]	Connect WE[3:0] to single user WE signal	DO[7:0], DOP[0]
18	DI[15:0], DIP[1:0]	ADDR[13:4]	Connect WE[0] and WE[2] to user WE[0] and WE[1] and WE[3] to user WE[1]	DO[15:0], DOP[1:0]
36	DI[31:0], DIP[3:0]	ADDR[13:5]	Connect each WE[3:0] signal to the associated byte write enable.	DO[31:0], DOP[3:0]

Alternatively, the prior RAMB16_Sm_Sn design elements can be instantiated if a byte-enable operation is not necessary. Also new convenience macros called RAMB16BWE_Sm_Sn are provided to allow for easier instantiation of this RAM with byte-enable operation. If either of these components is used, the software automatically re-target them to a properly configured RAMB16BWE component.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH_A, DATA_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, or 36	0	Specifies the configurable data width for Ports A and B.
INIT_A, INIT_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the initial value on the Port B output after configuration.

Attribute	Type	Allowed Values	Default	Description
SIM_COLLISION_CHECK	String	ALL, "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"	"ALL"	Allows modification of the simulation behavior so that if a memory collission occurs: "ALL" = warning produced and affected outputs/memory location go unknown (X). WARNING_ONLY = warning produced and affected outputs/memory retain last value. GENERATE_X_ ONLY = no warning, however, affected outputs/memory go unknown (X). NONE = no warning and affected outputs/memory retain last value. *Note:* Setting this to a value other than "ALL" can allow problems in the design to go unnoticed during simulation. Care should be taken when changing the value of this attribute.
SRVAL_A, SRVAL_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the output value of Port B upon the assertion of the synchronous reset (SSRB) signal.
WRITE_MODE_A, WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"	"WRITE_ FIRST"	Specifies output behavior of the port being written to: WRITE_FIRST" = written value appears on output port of the RAM. READ_FIRST = previous RAM contents for that memory location appear on the output port. NO_CHANGE = previous value on the output port remains the same.
INIT_00 to INIT_3F	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 16Kb data memory array.
INITP_00 to INITP_07	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 2Kb parity data memory array.

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;
```

```
-- RAMB16BWE_S18_S9: 1k/2k x 16/8 + 2/1 Parity bits Dual-Port byte-wide write RAM
--                   Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S18_S9_inst : RAMB16BWE_S18_S9
generic map (
INIT_A => X"00000", --  Value of output RAM registers on Port A at startup
INIT_B => X"000", --  Value of output RAM registers on Port B at startup
SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
SRVAL_A => X"00000", --  Port A ouput value upon SSR assertion
SRVAL_B => X"000", --  Port B ouput value upon SSR assertion
WRITE_MODE_A => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
WRITE_MODE_B => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the intiial contents of the RAM
-- Port A Address 0 to 255, Port B address 0 to 127
```

```
INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port A Address 0 to 255, Port B Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DOA => DOA,       -- Port A 16-bit Data Output
DOB => DOB,       -- Port B 8-bit Data Output
DOPA => DOPA,     -- Port A 2-bit Parity Output
DOPB => DOPB,     -- Port B 1-bit Parity Output
ADDRA => ADDRA,   -- Port A 10-bit Address Input
ADDRB => ADDRB,   -- Port B 11-bit Address Input
CLKA => CLKA,     -- Port A 1-bit Clock
CLKB => CLKB,     -- Port B 1-bit Clock
DIA => DIA,       -- Port A 16-bit Data Input
DIB => DIB,       -- Port B 8-bit Data Input
DIPA => DIPA,     -- Port A 2-bit parity Input
DIPB => DIPB,     -- Port B 1-bit parity Input
ENA => ENA,       -- Port A 1-bit RAM Enable Input
ENB => ENB,       -- Port B 1-bit RAM Enable Input
SSRA => SSRA,     -- Port A 1-bit Synchronous Set/Reset Input
SSRB => SSRB,     -- Port B 1-bit Synchronous Set/Reset Input
WEA => WEA,       -- Port A 2-bit Write Enable Input
WEB => WEB        -- Port B 1-bit Write Enable Input
);

-- End of RAMB16BWE_S18_S9_inst instantiation
```

Verilog Instantiation Template

```
// RAMB16BWE_S18_S9: 1k/2k x 16/8 + 2/1 Parity bits Dual-Port byte-wide write RAM
//                   Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S18_S9 #(
.INIT_A(18'h00000),       // Value of output RAM registers on Port A at startup
.INIT_B(9'h000),  // Value of output RAM registers on Port B at startup
.SIM_COLLISION_CHECK("ALL"),  // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
.SRVAL_A(18'h00000),      // Port A output value upon SSR assertion
.SRVAL_B(9'h000), // Port B output value upon SSR assertion
.WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
.WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

// The following INIT_xx declarations specify the initial contents of the RAM
// Port A Address 0 to 255, Port B Address 0 to 127
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 256 to 511, Port B Address 128 to 255
.INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

```
.INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 767, Port B Address 256 to 383
.INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 768 to 1023, Port B Address 384 to 511
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 255, Port B Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 256 to 511, Port B Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 767, Port B Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 768 to 1023, Port B Address 384 to 511
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16BWE_S18_S9_inst (
.DOA(DOA),       // Port A 16-bit Data Output
.DOB(DOB),       // Port B 8-bit Data Output
.DOPA(DOPA),     // Port A 2-bit Parity Output
.DOPB(DOPB),     // Port B 1-bit Parity Output
.ADDRA(ADDRA),   // Port A 10-bit Address Input
.ADDRB(ADDRB),   // Port B 11-bit Address Input
.CLKA(CLKA),     // Port A 1-bit Clock
.CLKB(CLKB),     // Port B 1-bit Clock
.DIA(DIA),       // Port A 16-bit Data Input
.DIB(DIB),       // Port B 8-bit Data Input
.DIPA(DIPA),     // Port A 2-bit parity Input
.DIPB(DIPB),     // Port-B 1-bit parity Input
.ENA(ENA),       // Port A 1-bit RAM Enable Input
```

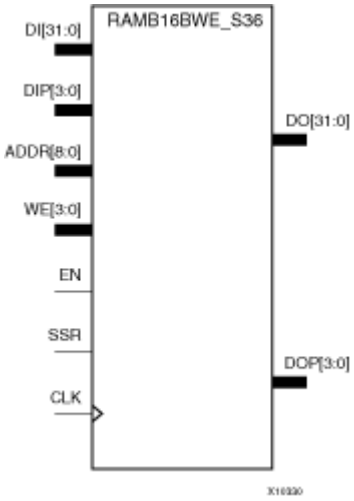
```
.ENB(ENB),       // Port B 1-bit RAM Enable Input
.SSRA(SSRA),     // Port A 1-bit Synchronous Set/Reset Input
.SSRB(SSRB),     // Port B 1-bit Synchronous Set/Reset Input
.WEA(WEA),       // Port A 2-bit Write Enable Input
.WEB(WEB)        // Port B 1-bit Write Enable Input
);

// End of RAMB16BWE_S18_S9_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

RAMB16BWE_S36

Primitive: 16K-bit Data and 2K-bit Parity Synchronous Single Port Block RAM with 36-Bit Port



Introduction

This design element can be configured and used as a 1-bit wide by 16K deep to a 36-bit wide by 512 deep single-port or dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component however Port A and Port B may operate fully independent and asynchronous to each other accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible. This Block RAM memory offers fast and flexible storage of large amounts of on-chip data.

Port Descriptions

Name	Direction	Width (Bits)	Function
DOA, DOB	Output	32-bits	Port A/B data output bus.
DOPA, DOPB	Output	4-bits	Port A/B parity output bus.
DIA, DIB	Input	32-bits	Port A/B data input bus.
DIPA, DIPB	Input	4-bits	Port A/B parity input bus.
ADDRA[13:5], ADDRB[13:5]	Input	14-bits	Port A/B address input bus. LSB always exists on ADDRA/B[0] while the MSB is determined by the settings for DATA_WIDTH_A/B.
WEA[0:0], WEB[0:0]	Input	4-bits	Port A/B byte-wide write enable.
ENA, ENB	Input	1-bit	Port A/B enable
SSRA, SSRB	Input	1-bit	Port A/B output registers synchronous reset.
CLKA, CLKB	Input	1-bit	Port A/B clock input.

Design Entry Method

Instantiation	Yes
Inference	Recommended

Coregen and wizards	Yes
Macro support	No

This element can be inferred by most synthesis tools by properly describing the RAM behavior in standard RTL code (consult synthesis tool documentation for details). Alternatively, CORE Generator can also create the desired macro for this RAM. If it is desired to have more control over the implementation or placement of this component, it may also be directly instantiated. In order to instantiate this component, use the ISE HDL Templates or instantiation template below and paste into your code. Connect all necessary inputs to the desired signals in the design. The CLKA/CLKB clock signals must be tied to an active clock for RAM operation and the SSRA/SSRB reset signals must be either tied to a logic zero or to the proper reset signal. ENA/ENB must either be tied to a logic one or a proper RAM port enable signal. Refer to the DATA_WIDTH table below for the necessary data input, data output, write enable and address connection information for each DATA_WIDTH setting as the necessary connections for these signals change based on this attribute. All other output signals can be left unconnected (open) and all unused input signals should be tied to a logic zero.

The following table shows the necessary input and output connections for the variable input ports for each DATA_WIDTH values for either Port A or Port B:

DATA_WIDTH Value	DI, DIP Connections	ADDR Connections	WE Connections	DO, DOP Connections
1	DI[0]	ADDR[13:0]	Connect WE[3:0] to single user WE signal	DO[0]
2	DI[1:0]	ADDR[13:1]	Connect WE[3:0] to single user WE signal	DO[1:0]
4	DI[3:0]	ADDR[13:2]	Connect WE[3:0] to single user WE signal	DO[3:0]
9	DI[7:0], DIP[0]	ADDR[13:3]	Connect WE[3:0] to single user WE signal	DO[7:0], DOP[0]
18	DI[15:0], DIP[1:0]	ADDR[13:4]	Connect WE[0] and WE[2] to user WE[0] and WE[1] and WE[3] to user WE[1]	DO[15:0], DOP[1:0]
36	DI[31:0], DIP[3:0]	ADDR[13:5]	Connect each WE[3:0] signal to the associated byte write enable.	DO[31:0], DOP[3:0]

Alternatively, the prior RAMB16_Sm_Sn design elements can be instantiated if a byte-enable operation is not necessary. Also new convenience macros called RAMB16BWE_Sm_Sn are provided to allow for easier instantiation of this RAM with byte-enable operation. If either of these components is used, the software automatically re-target them to a properly configured RAMB16BWE component.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH_A, DATA_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, or 36	0	Specifies the configurable data width for Ports A and B.
INIT_A, INIT_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the initial value on the Port B output after configuration.

Attribute	Type	Allowed Values	Default	Description
SIM_COLLISION_CHECK	String	ALL, "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"	"ALL"	Allows modification of the simulation behavior so that if a memory collission occurs: "ALL" = warning produced and affected outputs/memory location go unknown (X). WARNING_ONLY = warning produced and affected outputs/memory retain last value. GENERATE_X_ ONLY = no warning, however, affected outputs/memory go unknown (X). NONE = no warning and affected outputs/memory retain last value. *Note:* Setting this to a value other than "ALL" can allow problems in the design to go unnoticed during simulation. Care should be taken when changing the value of this attribute.
SRVAL_A, SRVAL_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the output value of Port B upon the assertion of the synchronous reset (SSRB) signal.
WRITE_MODE_A, WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"	"WRITE_ FIRST"	Specifies output behavior of the port being written to: WRITE_FIRST" = written value appears on output port of the RAM. READ_FIRST = previous RAM contents for that memory location appear on the output port. NO_CHANGE = previous value on the output port remains the same.
INIT_00 to INIT_3F	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 16Kb data memory array.
INITP_00 to INITP_07	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 2Kb parity data memory array.

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;
```

```
-- RAMB16BWE_S36: 512 x 32 + 4 Parity bits Single-Port byte-wide write RAM
--                Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S36_inst : RAMB16BWE_S36
generic map (
INIT => X"000000000",  --  Value of output RAM registers at startup
SRVAL => X"000000000", --  Ouput value upon SSR assertion
WRITE_MODE => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the initial contents of the RAM
-- Address 0 to 127
INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 128 to 255
INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 383
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
-- Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DO => DO,       -- 32-bit Data Output
DOP => DOP,     -- 4-bit parity Output
ADDR => ADDR,   -- 9-bit Address Input
CLK => CLK,     -- 1-bit Clock
DI => DI,       -- 32-bit Data Input
DIP => DIP,     -- 4-bit parity Input
EN => EN,       -- 1-bit RAM Enable Input
SSR => SSR,     -- 1-bit Synchronous Set/Reset Input
WE => WE        -- 4-bit Write Enable Input
);

-- End of RAMB16BWE_S36_inst instantiation
```

Verilog Instantiation Template

```
// RAMB16BWE_S36: 512 x 32 + 4 Parity bits Single-Port byte-wide write RAM
//                Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S36 #(
.INIT(36'h000000000),  // Value of output RAM registers at startup
.SRVAL(36'h000000000), // Output value upon SSR assertion
.WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

// The following INIT_xx declarations specify the initial contents of the RAM
// Address 0 to 127
.INIT_00(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_01(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_02(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_03(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_04(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_05(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_06(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_07(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_08(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_09(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
// Address 128 to 255
.INIT_10(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_11(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_12(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_13(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_14(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_15(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_16(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_17(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_18(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_19(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
// Address 256 to 383
.INIT_20(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_21(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_22(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_23(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_24(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
```

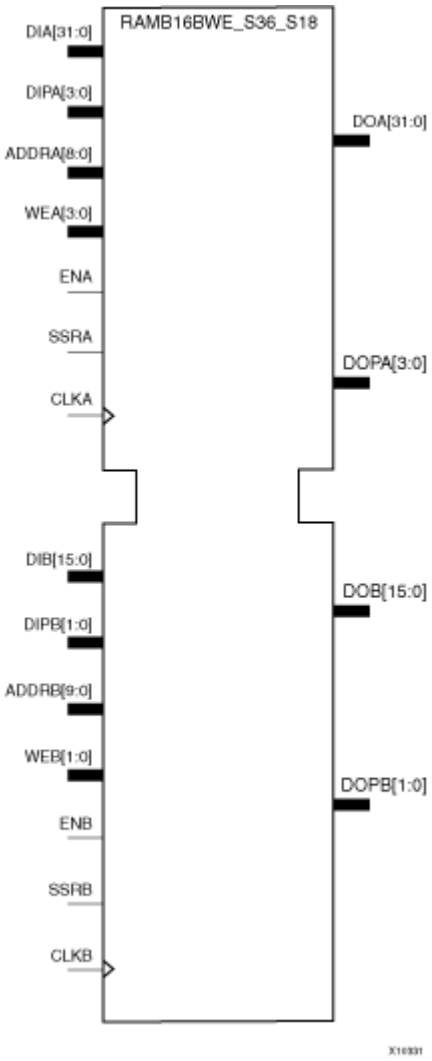
```
.INIT_25(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_26(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_27(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_28(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_29(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
// Address 384 to 511
.INIT_30(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_31(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_32(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_33(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_34(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_35(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_36(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_37(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_38(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_39(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),

// The next set of INITP_xx are for the parity bits
// Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 384 to 511
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16BWE_S36_inst (
.DO(DO),      // 32-bit Data Output
.DOP(DOP),    // 4-bit parity Output
.ADDR(ADDR),  // 9-bit Address Input
.CLK(CLK),    // 1-bit Clock
.DI(DI),      // 32-bit Data Input
.DIP(DIP),    // 4-bit parity Input
.EN(EN),      // 1-bit RAM Enable Input
.SSR(SSR),    // 1-bit Synchronous Set/Reset Input
.WE(WE)       // 4-bit Write Enable Input
);

// End of RAMB16BWE_S36_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

RAMB16BWE_S36_S18

Primitive: 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit and 18-bit Ports



Introduction

This design element can be configured and used as a 1-bit wide by 16K deep to a 36-bit wide by 512 deep single-port or dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component however Port A and Port B may operate fully independent and asynchronous to each other accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible. This Block RAM memory offers fast and flexible storage of large amounts of on-chip data.

Port Descriptions

Name	Direction	Width (Bits)	Function
DOA, DOB	Output	32-bits	Port A/B data output bus.
DOPA, DOPB	Output	4-bits	Port A/B parity output bus.
DIA, DIB	Input	32-bits	Port A/B data input bus.

Name	Direction	Width (Bits)	Function
DIPA, DIPB	Input	4-bits	Port A/B parity input bus.
ADDRA[13:5], ADDRB[13:5]	Input	14-bits	Port A/B address input bus. LSB always exists on ADDRA/B[0] while the MSB is determined by the settings for DATA_WIDTH_A/B.
WEA[0:0], WEB[0:0]	Input	4-bits	Port A/B byte-wide write enable.
ENA, ENB	Input	1-bit	Port A/B enable
SSRA, SSRB	Input	1-bit	Port A/B output registers synchronous reset.
CLKA, CLKB	Input	1-bit	Port A/B clock input.

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	Yes
Macro support	No

This element can be inferred by most synthesis tools by properly describing the RAM behavior in standard RTL code (consult synthesis tool documentation for details). Alternatively, CORE Generator can also create the desired macro for this RAM. If it is desired to have more control over the implementation or placement of this component, it may also be directly instantiated. In order to instantiate this component, use the ISE HDL Templates or instantiation template below and paste into your code. Connect all necessary inputs to the desired signals in the design. The CLKA/CLKB clock signals must be tied to an active clock for RAM operation and the SSRA/SSRB reset signals must be either tied to a logic zero or to the proper reset signal. ENA/ENB must either be tied to a logic one or a proper RAM port enable signal. Refer to the DATA_WIDTH table below for the necessary data input, data output, write enable and address connection information for each DATA_WIDTH setting as the necessary connections for these signals change based on this attribute. All other output signals can be left unconnected (open) and all unused input signals should be tied to a logic zero.

The following table shows the necessary input and output connections for the variable input ports for each DATA_WIDTH values for either Port A or Port B:

DATA_WIDTH Value	DI, DIP Connections	ADDR Connections	WE Connections	DO, DOP Connections
1	DI[0]	ADDR[13:0]	Connect WE[3:0] to single user WE signal	DO[0]
2	DI[1:0]	ADDR[13:1]	Connect WE[3:0] to single user WE signal	DO[1:0]
4	DI[3:0]	ADDR[13:2]	Connect WE[3:0] to single user WE signal	DO[3:0]
9	DI[7:0], DIP[0]	ADDR[13:3]	Connect WE[3:0] to single user WE signal	DO[7:0], DOP[0]
18	DI[15:0], DIP[1:0]	ADDR[13:4]	Connect WE[0] and WE[2] to user WE[0] and WE[1] and WE[3] to user WE[1]	DO[15:0], DOP[1:0]
36	DI[31:0], DIP[3:0]	ADDR[13:5]	Connect each WE[3:0] signal to the associated byte write enable.	DO[31:0], DOP[3:0]

Alternatively, the prior RAMB16_Sm_Sn design elements can be instantiated if a byte-enable operation is not necessary. Also new convenience macros called RAMB16BWE_Sm_Sn are provided to allow for easier instantiation of this RAM with byte-enable operation. If either of these components is used, the software automatically re-target them to a properly configured RAMB16BWE component.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH_A, DATA_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, or 36	0	Specifies the configurable data width for Ports A and B.
INIT_A, INIT_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the initial value on the Port B output after configuration.
SIM_COLLISION_CHECK	String	ALL, "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"	"ALL"	Allows modification of the simulation behavior so that if a memory collission occurs: "ALL" = warning produced and affected outputs/memory location go unknown (X). WARNING_ONLY = warning produced and affected outputs/memory retain last value. GENERATE_X_ ONLY = no warning, however, affected outputs/memory go unknown (X). NONE = no warning and affected outputs/memory retain last value. *Note:* Setting this to a value other than "ALL" can allow problems in the design to go unnoticed during simulation. Care should be taken when changing the value of this attribute.
SRVAL_A, SRVAL_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the output value of Port B upon the assertion of the synchronous reset (SSRB) signal.
WRITE_MODE_A, WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"	"WRITE_ FIRST"	Specifies output behavior of the port being written to: WRITE_FIRST" = written value appears on output port of the RAM. READ_FIRST = previous RAM contents for that memory location appear on the output port. NO_CHANGE = previous value on the output port remains the same.
INIT_00 to INIT_3F	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 16Kb data memory array.
INITP_00 to INITP_07	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 2Kb parity data memory array.

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;

-- RAMB16BWE_S36_S18: 1k/2k x 16/8 + 2/1 Parity bits Dual-Port byte-wide write RAM
--                    Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S36_S18_inst : RAMB16BWE_S36_S18
generic map (
INIT_A => X"000000000", --  Value of output RAM registers on Port A at startup
INIT_B => X"00000", --  Value of output RAM registers on Port B at startup
SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
SRVAL_A => X"000000000", --  Port A ouput value upon SSR assertion
SRVAL_B => X"00000", --  Port B ouput value upon SSR assertion
WRITE_MODE_A => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
WRITE_MODE_B => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the intiial contents of the RAM
-- Port A Address 0 to 255, Port B address 0 to 127
INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port A Address 0 to 255, Port B Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DOA => DOA,       -- Port A 32-bit Data Output
DOB => DOB,       -- Port B 16-bit Data Output
DOPA => DOPA,     -- Port A 4-bit Parity Output
DOPB => DOPB,     -- Port B 2-bit Parity Output
ADDRA => ADDRA,   -- Port A 9-bit Address Input
ADDRB => ADDRB,   -- Port B 10-bit Address Input
CLKA => CLKA,     -- Port A 1-bit Clock
CLKB => CLKB,     -- Port B 1-bit Clock
DIA => DIA,       -- Port A 32-bit Data Input
DIB => DIB,       -- Port B 16-bit Data Input
DIPA => DIPA,     -- Port A 4-bit parity Input
DIPB => DIPB,     -- Port B 2-bit parity Input
ENA => ENA,       -- Port A 1-bit RAM Enable Input
ENB => ENB,       -- Port B 1-bit RAM Enable Input
SSRA => SSRA,     -- Port A 1-bit Synchronous Set/Reset Input
SSRB => SSRB,     -- Port B 1-bit Synchronous Set/Reset Input
WEA => WEA,       -- Port A 4-bit Write Enable Input
WEB => WEB        -- Port B 2-bit Write Enable Input
);

-- End of RAMB16BWE_S36_S18_inst instantiation
```

Verilog Instantiation Template

```
// RAMB16BWE_S36_S18: 1k/512 x 16/32 + 2/4 Parity bits Dual-Port byte-wide write RAM
//                    Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S36_S18 #(
.INIT_A(36'h000000000),   // Value of output RAM registers on Port A at startup
.INIT_B(18'h00000),       // Value of output RAM registers on Port B at startup
.SIM_COLLISION_CHECK("ALL"),  // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
.SRVAL_A(36'h000000000),  // Port A output value upon SSR assertion
.SRVAL_B(18'h00000),      // Port B output value upon SSR assertion
.WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
.WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

// The following INIT_xx declarations specify the initial contents of the RAM
// Port A Address 0 to 255, Port B Address 0 to 127
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

```
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 256 to 511, Port B Address 128 to 255
.INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 767, Port B Address 256 to 383
.INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 768 to 1023, Port B Address 384 to 511
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 255, Port B Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 256 to 511, Port B Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

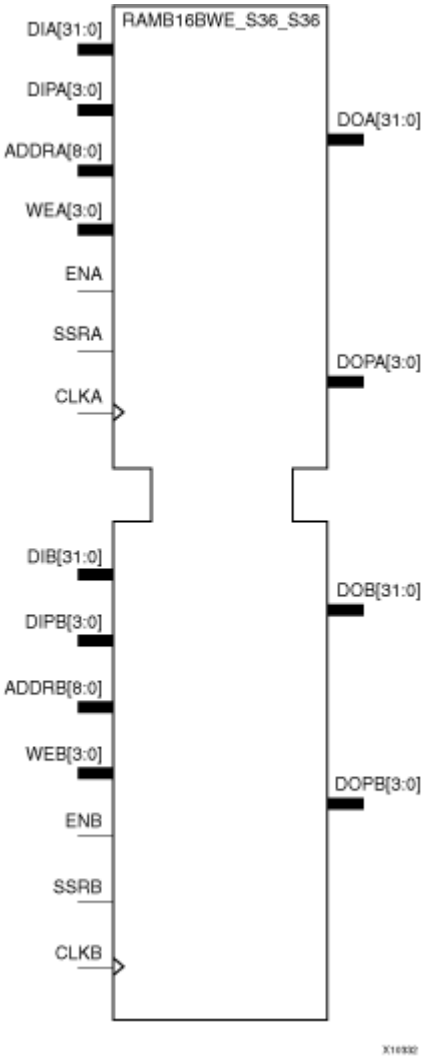
```
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 767, Port B Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 768 to 1023, Port B Address 384 to 511
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16BWE_S36_S18_inst (
.DOA(DOA),       // Port A 32-bit Data Output
.DOB(DOB),       // Port B 16-bit Data Output
.DOPA(DOPA),     // Port A 4-bit Parity Output
.DOPB(DOPB),     // Port B 2-bit Parity Output
.ADDRA(ADDRA),   // Port A 9-bit Address Input
.ADDRB(ADDRB),   // Port B 10-bit Address Input
.CLKA(CLKA),     // Port A 1-bit Clock
.CLKB(CLKB),     // Port B 1-bit Clock
.DIA(DIA),       // Port A 32-bit Data Input
.DIB(DIB),       // Port B 16-bit Data Input
.DIPA(DIPA),     // Port A 4-bit parity Input
.DIPB(DIPB),     // Port-B 2-bit parity Input
.ENA(ENA),       // Port A 1-bit RAM Enable Input
.ENB(ENB),       // Port B 1-bit RAM Enable Input
.SSRA(SSRA),     // Port A 1-bit Synchronous Set/Reset Input
.SSRB(SSRB),     // Port B 1-bit Synchronous Set/Reset Input
.WEA(WEA),       // Port A 4-bit Write Enable Input
.WEB(WEB)        // Port B 2-bit Write Enable Input
);

// End of RAMB16BWE_S36_S18_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

RAMB16BWE_S36_S36

Primitive: 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit Ports



Introduction

This design element can be configured and used as a 1-bit wide by 16K deep to a 36-bit wide by 512 deep single-port or dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component however Port A and Port B may operate fully independent and asynchronous to each other accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible. This Block RAM memory offers fast and flexible storage of large amounts of on-chip data.

Port Descriptions

Name	Direction	Width (Bits)	Function
DOA, DOB	Output	32-bits	Port A/B data output bus.
DOPA, DOPB	Output	4-bits	Port A/B parity output bus.
DIA, DIB	Input	32-bits	Port A/B data input bus.
DIPA, DIPB	Input	4-bits	Port A/B parity input bus.

Name	Direction	Width (Bits)	Function
ADDRA[13:5], ADDRB[13:5]	Input	14-bits	Port A/B address input bus. LSB always exists on ADDRA/B[0] while the MSB is determined by the settings for DATA_WIDTH_A/B.
WEA[0:0], WEB[0:0]	Input	4-bits	Port A/B byte-wide write enable.
ENA, ENB	Input	1-bit	Port A/B enable
SSRA, SSRB	Input	1-bit	Port A/B output registers synchronous reset.
CLKA, CLKB	Input	1-bit	Port A/B clock input.

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	Yes
Macro support	No

This element can be inferred by most synthesis tools by properly describing the RAM behavior in standard RTL code (consult synthesis tool documentation for details). Alternatively, CORE Generator can also create the desired macro for this RAM. If it is desired to have more control over the implementation or placement of this component, it may also be directly instantiated. In order to instantiate this component, use the ISE HDL Templates or instantiation template below and paste into your code. Connect all necessary inputs to the desired signals in the design. The CLKA/CLKB clock signals must be tied to an active clock for RAM operation and the SSRA/SSRB reset signals must be either tied to a logic zero or to the proper reset signal. ENA/ENB must either be tied to a logic one or a proper RAM port enable signal. Refer to the DATA_WIDTH table below for the necessary data input, data output, write enable and address connection information for each DATA_WIDTH setting as the necessary connections for these signals change based on this attribute. All other output signals can be left unconnected (open) and all unused input signals should be tied to a logic zero.

The following table shows the necessary input and output connections for the variable input ports for each DATA_WIDTH values for either Port A or Port B:

DATA_WIDTH Value	DI, DIP Connections	ADDR Connections	WE Connections	DO, DOP Connections
1	DI[0]	ADDR[13:0]	Connect WE[3:0] to single user WE signal	DO[0]
2	DI[1:0]	ADDR[13:1]	Connect WE[3:0] to single user WE signal	DO[1:0]
4	DI[3:0]	ADDR[13:2]	Connect WE[3:0] to single user WE signal	DO[3:0]
9	DI[7:0], DIP[0]	ADDR[13:3]	Connect WE[3:0] to single user WE signal	DO[7:0], DOP[0]
18	DI[15:0], DIP[1:0]	ADDR[13:4]	Connect WE[0] and WE[2] to user WE[0] and WE[1] and WE[3] to user WE[1]	DO[15:0], DOP[1:0]
36	DI[31:0], DIP[3:0]	ADDR[13:5]	Connect each WE[3:0] signal to the associated byte write enable.	DO[31:0], DOP[3:0]

Alternatively, the prior RAMB16_Sm_Sn design elements can be instantiated if a byte-enable operation is not necessary. Also new convenience macros called RAMB16BWE_Sm_Sn are provided to allow for easier instantiation of this RAM with byte-enable operation. If either of these components is used, the software automatically re-target them to a properly configured RAMB16BWE component.

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;

-- RAMB16BWE_S36_S36: 512 x 32 + 4 Parity bits Dual-Port byte-wide write RAM
--                    Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S36_S36_inst : RAMB16BWE_S36_S36
generic map (
INIT_A => X"000000000",  --  Value of output RAM registers on Port A at startup
INIT_B => X"000000000",  --  Value of output RAM registers on Port B at startup
SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
SRVAL_A => X"000000000", --  Port A ouput value upon SSR assertion
SRVAL_B => X"000000000", --  Port B ouput value upon SSR assertion
WRITE_MODE_A => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
WRITE_MODE_B => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the initial contents of the RAM
-- Address 0 to 127
INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 128 to 255
INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 383
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DOA => DOA,       -- Port A 32-bit Data Output
DOB => DOB,       -- Port B 32-bit Data Output
DOPA => DOPA,     -- Port A 4-bit Parity Output
DOPB => DOPB,     -- Port B 4-bit Parity Output
ADDRA => ADDRA,   -- Port A 9-bit Address Input
ADDRB => ADDRB,   -- Port B 9-bit Address Input
CLKA => CLKA,     -- Port A 1-bit Clock
CLKB => CLKB,     -- Port B 1-bit Clock
DIA => DIA,       -- Port A 32-bit Data Input
DIB => DIB,       -- Port B 32-bit Data Input
DIPA => DIPA,     -- Port A 4-bit parity Input
DIPB => DIPB,     -- Port-B 4-bit parity Input
ENA => ENA,       -- Port A 1-bit RAM Enable Input
ENB => ENB,       -- Port B 1-bit RAM Enable Input
SSRA => SSRA,     -- Port A 1-bit Synchronous Set/Reset Input
SSRB => SSRB,     -- Port B 1-bit Synchronous Set/Reset Input
WEA => WEA,       -- Port A 4-bit Write Enable Input
WEB => WEB        -- Port B 4-bit Write Enable Input
);

-- End of RAMB16BWE_S36_S36_inst instantiation
```

Verilog Instantiation Template

```
// RAMB16BWE_S36_S36: 512 x 32 + 4 Parity bits byte-wide write Dual-Port RAM
//                    Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S36_S36 #(
.INIT_A(36'h000000000),  // Value of output RAM registers on Port A at startup
.INIT_B(36'h000000000),  // Value of output RAM registers on Port B at startup
.SIM_COLLISION_CHECK("ALL"),  // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
```

```
.SRVAL_A(36'h000000000), // Port A output value upon SSR assertion
.SRVAL_B(36'h000000000), // Port B output value upon SSR assertion
.WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
.WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

// The following INIT_xx declarations specify the initial contents of the RAM
// Address 0 to 127
.INIT_00(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_01(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_02(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_03(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_04(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_05(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_06(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_07(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_08(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_09(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_0F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
// Address 128 to 255
.INIT_10(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_11(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_12(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_13(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_14(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_15(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_16(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_17(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_18(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_19(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_1F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
// Address 256 to 383
.INIT_20(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_21(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_22(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_23(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_24(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_25(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_26(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_27(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_28(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_29(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
// Address 384 to 511
.INIT_30(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_31(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_32(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_33(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_34(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_35(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_36(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_37(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_38(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_39(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
```

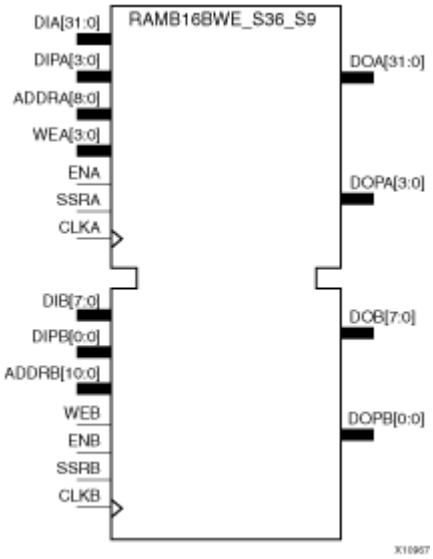
```
.INIT_3F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),

// The next set of INITP_xx are for the parity bits
// Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 384 to 511
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16BWE_S36_S36_inst (
.DOA(DOA),      // Port A 32-bit Data Output
.DOB(DOB),      // Port B 32-bit Data Output
.DOPA(DOPA),    // Port A 4-bit Parity Output
.DOPB(DOPB),    // Port B 4-bit Parity Output
.ADDRA(ADDRA),  // Port A 9-bit Address Input
.ADDRB(ADDRB),  // Port B 9-bit Address Input
.CLKA(CLKA),    // Port A 1-bit Clock
.CLKB(CLKB),    // Port B 1-bit Clock
.DIA(DIA),      // Port A 32-bit Data Input
.DIB(DIB),      // Port B 32-bit Data Input
.DIPA(DIPA),    // Port A 4-bit parity Input
.DIPB(DIPB),    // Port-B 4-bit parity Input
.ENA(ENA),      // Port A 1-bit RAM Enable Input
.ENB(ENB),      // Port B 1-bit RAM Enable Input
.SSRA(SSRA),    // Port A 1-bit Synchronous Set/Reset Input
.SSRB(SSRB),    // Port B 1-bit Synchronous Set/Reset Input
.WEA(WEA),      // Port A 4-bit Write Enable Input
.WEB(WEB)       // Port B 4-bit Write Enable Input
);

// End of RAMB16BWE_S36_S36_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

RAMB16BWE_S36_S9

Primitive: 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit and 9-bit Ports



Introduction

This design element can be configured and used as a 1-bit wide by 16K deep to a 36-bit wide by 512 deep single-port or dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component however Port A and Port B may operate fully independent and asynchronous to each other accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible. This Block RAM memory offers fast and flexible storage of large amounts of on-chip data.

Port Descriptions

Name	Direction	Width (Bits)	Function
DOA, DOB	Output	32-bits	Port A/B data output bus.
DOPA, DOPB	Output	4-bits	Port A/B parity output bus.
DIA, DIB	Input	32-bits	Port A/B data input bus.
DIPA, DIPB	Input	4-bits	Port A/B parity input bus.
ADDRA[13:5], ADDRB[13:5]	Input	14-bits	Port A/B address input bus. LSB always exists on ADDRA/B[0] while the MSB is determined by the settings for DATA_WIDTH_A/B.
WEA[0:0], WEB[0:0]	Input	4-bits	Port A/B byte-wide write enable.
ENA, ENB	Input	1-bit	Port A/B enable
SSRA, SSRB	Input	1-bit	Port A/B output registers synchronous reset.
CLKA, CLKB	Input	1-bit	Port A/B clock input.

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	Yes
Macro support	No

This element can be inferred by most synthesis tools by properly describing the RAM behavior in standard RTL code (consult synthesis tool documentation for details). Alternatively, CORE Generator can also create the desired macro for this RAM. If it is desired to have more control over the implementation or placement of this component, it may also be directly instantiated. In order to instantiate this component, use the ISE HDL Templates or instantiation template below and paste into your code. Connect all necessary inputs to the desired signals in the design. The CLKA/CLKB clock signals must be tied to an active clock for RAM operation and the SSRA/SSRB reset signals must be either tied to a logic zero or to the proper reset signal. ENA/ENB must either be tied to a logic one or a proper RAM port enable signal. Refer to the DATA_WIDTH table below for the necessary data input, data output, write enable and address connection information for each DATA_WIDTH setting as the necessary connections for these signals change based on this attribute. All other output signals can be left unconnected (open) and all unused input signals should be tied to a logic zero.

The following table shows the necessary input and output connections for the variable input ports for each DATA_WIDTH values for either Port A or Port B:

DATA_WIDTH Value	DI, DIP Connections	ADDR Connections	WE Connections	DO, DOP Connections
1	DI[0]	ADDR[13:0]	Connect WE[3:0] to single user WE signal	DO[0]
2	DI[1:0]	ADDR[13:1]	Connect WE[3:0] to single user WE signal	DO[1:0]
4	DI[3:0]	ADDR[13:2]	Connect WE[3:0] to single user WE signal	DO[3:0]
9	DI[7:0], DIP[0]	ADDR[13:3]	Connect WE[3:0] to single user WE signal	DO[7:0], DOP[0]
18	DI[15:0], DIP[1:0]	ADDR[13:4]	Connect WE[0] and WE[2] to user WE[0] and WE[1] and WE[3] to user WE[1]	DO[15:0], DOP[1:0]
36	DI[31:0], DIP[3:0]	ADDR[13:5]	Connect each WE[3:0] signal to the associated byte write enable.	DO[31:0], DOP[3:0]

Alternatively, the prior RAMB16_Sm_Sn design elements can be instantiated if a byte-enable operation is not necessary. Also new convenience macros called RAMB16BWE_Sm_Sn are provided to allow for easier instantiation of this RAM with byte-enable operation. If either of these components is used, the software automatically re-target them to a properly configured RAMB16BWE component.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH_A, DATA_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, or 36	0	Specifies the configurable data width for Ports A and B.
INIT_A, INIT_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the initial value on the Port B output after configuration.

Attribute	Type	Allowed Values	Default	Description
SIM_COLLISION_CHECK	String	ALL, "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"	"ALL"	Allows modification of the simulation behavior so that if a memory collission occurs: "ALL" = warning produced and affected outputs/memory location go unknown (X). WARNING_ONLY = warning produced and affected outputs/memory retain last value. GENERATE_X_ ONLY = no warning, however, affected outputs/memory go unknown (X). NONE = no warning and affected outputs/memory retain last value. *Note:* Setting this to a value other than "ALL" can allow problems in the design to go unnoticed during simulation. Care should be taken when changing the value of this attribute.
SRVAL_A, SRVAL_B	Hexadecimal	Any 36-Bit Value	All zeros	Specifies the output value of Port B upon the assertion of the synchronous reset (SSRB) signal.
WRITE_MODE_A, WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"	"WRITE_ FIRST"	Specifies output behavior of the port being written to: WRITE_FIRST" = written value appears on output port of the RAM. READ_FIRST = previous RAM contents for that memory location appear on the output port. NO_CHANGE = previous value on the output port remains the same.
INIT_00 to INIT_3F	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 16Kb data memory array.
INITP_00 to INITP_07	Hexadecimal	Any 256-Bit Value	All zeros	Allows specification of the initial contents of the 2Kb parity data memory array.

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;
```

```
-- RAMB16BWE_S36_S9: 2k/512 x 8/32 + 1/4 Parity bits Dual-Port byte-wide write RAM
--                   Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S36_S9_inst : RAMB16BWE_S36_S9
generic map (
INIT_A => X"000000000", --  Value of output RAM registers on Port A at startup
INIT_B => X"000", --  Value of output RAM registers on Port B at startup
SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
SRVAL_A => X"000000000", --  Port A ouput value upon SSR assertion
SRVAL_B => X"000", --  Port B ouput value upon SSR assertion
WRITE_MODE_A => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
WRITE_MODE_B => "WRITE_FIRST", --  WRITE_FIRST, READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the intiial contents of the RAM
-- Port A Address 0 to 255, Port B address 0 to 127
```

```
INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port A Address 0 to 255, Port B Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DOA => DOA,       -- Port A 32-bit Data Output
DOB => DOB,       -- Port B 8-bit Data Output
DOPA => DOPA,     -- Port A 4-bit Parity Output
DOPB => DOPB,     -- Port B 1-bit Parity Output
ADDRA => ADDRA,   -- Port A 9-bit Address Input
ADDRB => ADDRB,   -- Port B 11-bit Address Input
CLKA => CLKA,     -- Port A 1-bit Clock
CLKB => CLKB,     -- Port B 1-bit Clock
DIA => DIA,       -- Port A 32-bit Data Input
DIB => DIB,       -- Port B 8-bit Data Input
DIPA => DIPA,     -- Port A 4-bit parity Input
DIPB => DIPB,     -- Port B 1-bit parity Input
ENA => ENA,       -- Port A 1-bit RAM Enable Input
ENB => ENB,       -- Port B 1-bit RAM Enable Input
SSRA => SSRA,     -- Port A 1-bit Synchronous Set/Reset Input
SSRB => SSRB,     -- Port B 1-bit Synchronous Set/Reset Input
WEA => WEA,       -- Port A 4-bit Write Enable Input
WEB => WEB        -- Port B 1-bit Write Enable Input
);

-- End of RAMB16BWE_S36_S9_inst instantiation
```

Verilog Instantiation Template

```
// RAMB16BWE_S36_S9: 2k/512 x 8/32 + 1/4 Parity bits Dual-Port byte-wide write RAM
//                   Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAMB16BWE_S36_S9 #(
.INIT_A(36'h000000000),   // Value of output RAM registers on Port A at startup
.INIT_B(9'h000),          // Value of output RAM registers on Port B at startup
.SIM_COLLISION_CHECK("ALL"),  // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
.SRVAL_A(36'h000000000),  // Port A output value upon SSR assertion
.SRVAL_B(9'h000),         // Port B output value upon SSR assertion
.WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
.WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

// The following INIT_xx declarations specify the initial contents of the RAM
// Port A Address 0 to 255, Port B Address 0 to 127
.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 256 to 511, Port B Address 128 to 255
.INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

```
.INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 767, Port B Address 256 to 383
.INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 768 to 1023, Port B Address 384 to 511
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 255, Port B Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 256 to 511, Port B Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 767, Port B Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 768 to 1023, Port B Address 384 to 511
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16BWE_S36_S9_inst (
.DOA(DOA),       // Port A 32-bit Data Output
.DOB(DOB),       // Port B 8-bit Data Output
.DOPA(DOPA),     // Port A 4-bit Parity Output
.DOPB(DOPB),     // Port B 1-bit Parity Output
.ADDRA(ADDRA),   // Port A 9-bit Address Input
.ADDRB(ADDRB),   // Port B 11-bit Address Input
.CLKA(CLKA),     // Port A 1-bit Clock
.CLKB(CLKB),     // Port B 1-bit Clock
.DIA(DIA),       // Port A 32-bit Data Input
.DIB(DIB),       // Port B 8-bit Data Input
.DIPA(DIPA),     // Port A 4-bit parity Input
.DIPB(DIPB),     // Port-B 1-bit parity Input
.ENA(ENA),       // Port A 1-bit RAM Enable Input
```

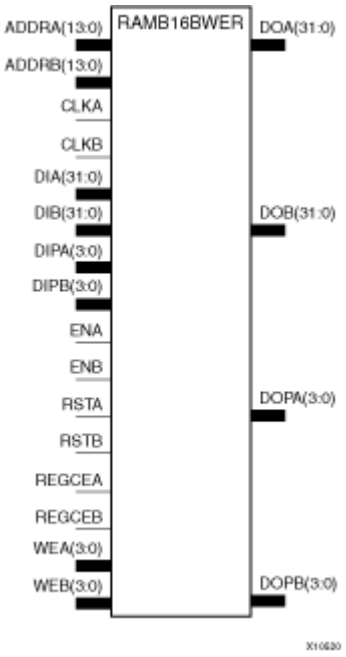
```
.ENB(ENB),       // Port B 1-bit RAM Enable Input
.SSRA(SSRA),     // Port A 1-bit Synchronous Set/Reset Input
.SSRB(SSRB),     // Port B 1-bit Synchronous Set/Reset Input
.WEA(WEA),       // Port A 4-bit Write Enable Input
.WEB(WEB)        // Port B 1-bit Write Enable Input
);

// End of RAMB16BWE_S36_S9_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

RAMB16BWER

Primitive: 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers



Introduction

Note This element is available only for Spartan-3A DSP parts.

This design element contains several block RAM memories that can be configured as general-purpose 16Kb data + 2Kb parity RAM/ROM memories. These block RAM memories offer fast and flexible storage of large amounts of on-chip data. This component can be configured and used as a 1-bit wide by 16K deep to a 36-bit wide by 512 deep, single-port or dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component. However, Port A and Port B can operate fully independent and asynchronous to each other, accessing the same memory array. When these ports are configured in the wider data width modes, byte-enable write operations are possible. This RAM also offers a configurable output register that can be enabled in order to improve clock-to-out times of the RAM while incurring an extra clock cycle of latency during the read operation.

Port Descriptions

The following table shows the necessary input and output connections for the variable input ports for each DATA_WIDTH values for either Port A or Port B.

DATA_WIDTH Value	DI, DIP Connections	ADDR Connections	WE Connections	DO, DOP Connections
1	DI[0]	ADDR[13:0]	Connect WE[3:0] to single user WE signal.	DO[0]
2	DI[1:0]	ADDR[13:1]	Connect WE[3:0] to single user WE signal.	DO[1:0]
4	DI[3:0]	ADDR[13:2]	Connect WE[3:0] to single user WE signal.	DO[3:0]
9	DI[7:0], DIP[0]	ADDR[13:3]	Connect WE[3:0] to single user WE signal.	DO[7:0], DOP[0]

DATA_WIDTH Value	DI, DIP Connections	ADDR Connections	WE Connections	DO, DOP Connections
18	DI[15:0], DIP[1:0]	ADDR[13:4]	Connect WE[0] and WE[2] to user WE[0] and WE[1] and WE[3] to user WE[1].	DO[15:0], DOP[1:0]
36	DI[31:0], DIP[3:0]	ADDR[13:5]	Connect each WE[3:0] signal to the associated byte write enable/.	DO[31:0], DOP[3:0]

Alternatively, the prior RAMB16_Sm_Sn and RAMB16BWER_Sm_Sn elements can be instantiated if the output registers are not necessary. If any of these components are used, the software will automatically re-target them the a properly configured RAMB16BWER element.

Name	Direction	Width (Bits)	Function
DOA, DOB	Output	32-bits	Port A/B data output bus.
DOPA, DOPB	Output	4-bits	Port A/B parity output bus.
DIA, DIB	Input	32-bits	Port A/B data input bus.
DIPA, DIPB	Input	4-bits	Port A/B parity input bus.
ADDRA, ADDRB	Input	14-bits	Port A/B address input bus. MSB always exists on ADDRA/B[13] while the LSB is determined by the settings for DATA_WIDTH_A/B.
WEA, WEB	Input	4-bits	Port A/B byte-wide write enable.
ENA, ENB	Input	1-bit	Port A/B enable.
REGCEA, RECEB	Input	1-bit	Output register clock enable.
RSTA, RSTB	Input	1-bit	Port A/B output registers set/reset. This reset is configurable to be synchronous or asynchronous, depending on the value of the RSTTYPE attribute.
CLKA, CLKB	Input	1-bit	Port A/B clock input.

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	Yes
Macro support	No

Connect all necessary inputs to the desired signals in the design. The CLKA/CLKB clock signals must be tied to an active clock for RAM operation, and the SRA/SRB reset signals must be either tied to a logic zero or to the proper reset signal. ENA/ENB must either be tied to a logic one or a proper RAM port enable signal. REGCEA and REGCEB must be tied to the proper output register clock enable, or a logic one if the respective DOA_REG or DOB_REG attribute is set to 1. If DOA_REG is set to 0, then tie REGCEA and REGCEB must be set to a logic 0.

Refer to the DATA_WIDTH column in the “Port Description” table (above) for the necessary data input, data output, write enable and address connection information for each DATA_WIDTH setting, since the necessary connections for these signals change, based on this attribute. All other output signals can be left unconnected (open) and all unused input signals should be tied to a logic zero.

Available Attributes

Attribute(s)	Type	Allowed Values	Default	Description
DATA_WIDTH_A, DATA_WIDTH_B	Integer	1, 2, 4, 9, 18, or 36	0	Specifies the configurable data width for Ports A and B.
DOA_REG, DOB_REG	Integer	0 or 1	0	Specifies to use or bypass the output registers for the RAM.
INIT_A, INIT_B	Hexadecimal	Any 36-bit Hexadecimal Value	All zeroes	Specifies the initial value on the Port B output after configuration.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"	"ALL"	Allows modification of the simulation behavior so that if a memory collision occurs: "ALL" = warning produced and affected outputs/memory location go unknown (X). "WARNING_ONLY" = warning produced and affected outputs/memory retain last value. "GENERATE_X_ONLY" = no warning, however affected outputs/memory go unknown (X) "NONE" = no warning and affected outputs/memory retain last value. *Note:* Setting this to a value other than "ALL" can allow problems in the design to go unnoticed during simulation. Care should be taken when changing the value of this attribute.
SRVAL_A	Hexadecimal	Any 36-Bit Value	All zeroes	Specifies the output value of Port A upon the assertion of the reset (RSTA) signal.
SRVAL_B	Hexadecimal	Any 36-Bit Value	All zeroes	Specifies the output value of Port B upon the assertion of the reset (RSTB) signal.

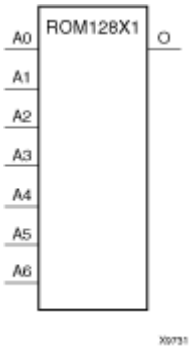
Attribute(s)	Type	Allowed Values	Default	Description
RSTTYPE	String	"ASYNC", "SYNC"	"SYNC"	Selects whether the RAM outputs should have a synchronous or asynchronous reset capability. Due to improved timing and circuit stability, it is recommended to always have this set to SYNC" unless an asynchronous reset is absolutely necessary.
WRITE_MODE_A, WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"	"WRITE_ FIRST"	Specifies output behavior of the port being written to: "WRITE_FIRST" = written value appears on output port of the RAM. "READ_FIRST" = previous RAM contents for that memory location appear on the output port. "NO_CHANGE" = previous value on the output port remains the same.
INIT_00 to INIT_3F	Hexadecimal	Any 256-bit hexadecimal value	All zeroes	Allows specification of the initial contents of the 16Kb data memory array.
INITP_00 to INITP_07	Hexadecimal	Any 256-bit hexadecimal value	All zeroes	Allows specification of the initial contents of the 2Kb parity data memory array.

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

ROM128X1

Primitive: 128-Deep by 1-Wide ROM



Introduction

This design element is a 128-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 7-bit address (A6 – A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of 32 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H. An error occurs if the INIT=value is not specified.

Logic Table

Input				Output
I0	**I1**	**I2**	**I3**	**O**
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

Design Entry Method

Instantiation	Yes
Inference	Recommended

Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 128-Bit Value	All zeros	Specifies the contents of the ROM.

VHDL Instantiation Template

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

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

```
-- ROM128X1: 128 x 1 Asynchronous Distributed  => LUT ROM
--           Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

ROM128X1_inst : ROM128X1
generic map (
INIT => X"00000000000000000000000000000000")
port map (
O => O,   -- ROM output
A0 => A0, -- ROM address[0]
A1 => A1, -- ROM address[1]
A2 => A2, -- ROM address[2]
A3 => A3, -- ROM address[3]
A4 => A4, -- ROM address[4]
A5 => A5, -- ROM address[5]
A6 => A6  -- ROM address[6]
);

-- End of ROM128X1_inst instantiation
```

Verilog Instantiation Template

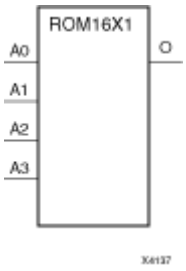
```
// ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
//           Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

ROM128X1 #(
.INIT(128'h00000000000000000000000000000000) // Contents of ROM
) ROM128X1_inst (
.O(O),   // ROM output
.A0(A0), // ROM address[0]
.A1(A1), // ROM address[1]
.A2(A2), // ROM address[2]
.A3(A3), // ROM address[3]
.A4(A4), // ROM address[4]
.A5(A5), // ROM address[5]
.A6(A6)  // ROM address[6]
);

// End of ROM128X1_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

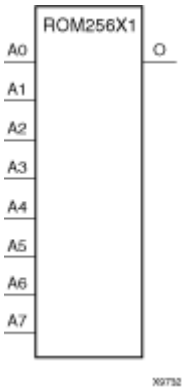
```
// ROM16X1: 16 x 1 Asynchronous Distributed (LUT) ROM
//          All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

ROM256X1

Primitive: 256-Deep by 1-Wide ROM



Introduction

This design element is a 256-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 8-bit address (A7– A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of 64 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H.

An error occurs if the INIT=value is not specified.

Logic Table

Input				Output
I0	**I1**	**I2**	**I3**	**O**
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 256-Bit Value	All zeros	Specifies the contents of the ROM.

VHDL Instantiation Template

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

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

```
-- ROM256X1: 256 x 1 Asynchronous Distributed  => LUT ROM
--           Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

ROM256X1_inst : ROM256X1
generic map (
INIT => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
O => O,   -- ROM output
A0 => A0, -- ROM address[0]
A1 => A1, -- ROM address[1]
A2 => A2, -- ROM address[2]
A3 => A3, -- ROM address[3]
A4 => A4, -- ROM address[4]
A5 => A5, -- ROM address[5]
A6 => A6,  -- ROM address[6]
A7 => A7  -- ROM address[7]
);

-- End of ROM256X1_inst instantiation
```

Verilog Instantiation Template

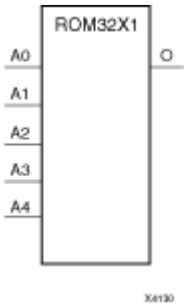
```
// ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
//           Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

ROM256X1 #(
.INIT(256'h0000000000000000000000000000000000000000000000000000000000000000) // Contents of ROM
) ROM256X1_inst (
.O(O),   // ROM output
.A0(A0), // ROM address[0]
.A1(A1), // ROM address[1]
.A2(A2), // ROM address[2]
.A3(A3), // ROM address[3]
.A4(A4), // ROM address[4]
.A5(A5), // ROM address[5]
.A6(A6), // ROM address[6]
.A7(A7)  // ROM address[7]
);

// End of ROM256X1_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

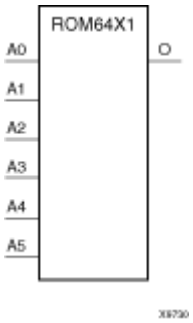
```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
//          All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

ROM64X1

Primitive: 64-Deep by 1-Wide ROM



Introduction

This design element is a 64-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 6-bit address (A5 – A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of 16 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H. An error occurs if the INIT=value is not specified.

Logic Table

Input				Output
I0	**I1**	**I2**	**I3**	**O**
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

Design Entry Method

Instantiation	Yes
Inference	Recommended

Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 64-Bit Value	All zeros	Specifies the contents of the ROM.

VHDL Instantiation Template

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

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

```
-- ROM64X1: 64 x 1 Asynchronous Distributed  => LUT ROM
--          Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

ROM64X1_inst : ROM64X1
generic map (
INIT => X"0000000000000000")
port map (
O => O,   -- ROM output
A0 => A0, -- ROM address[0]
A1 => A1, -- ROM address[1]
A2 => A2, -- ROM address[2]
A3 => A3, -- ROM address[3]
A4 => A4, -- ROM address[4]
A5 => A5  -- ROM address[5]
);

-- End of ROM64X1_inst instantiation
```

Verilog Instantiation Template

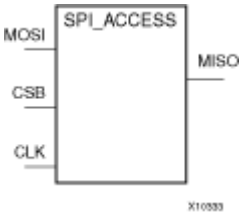
```
// ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
//          Virtex-II/II-Pro/4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

ROM64X1 #(
.INIT(64'h0000000000000000) // Contents of ROM
) ROM64X1_inst (
.O(O),   // ROM output
.A0(A0), // ROM address[0]
.A1(A1), // ROM address[1]
.A2(A2), // ROM address[2]
.A3(A3), // ROM address[3]
.A4(A4), // ROM address[4]
.A5(A5)  // ROM address[5]
);

// End of ROM64X1_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

SPI_ACCESS

Primitive: Internal Logic Access to the Serial Peripheral Interface (SPI) PROM Data



Introduction

This design element allows connection from the internal logic of the FPGA to an In-System Flash (ISF) Memory contained within the Spartan-3AN devices through an SPI serial protocol.

Port Descriptions

Name	Direction	Width	Function
MISO	Output	1-bit	Serial output data from the ISF Memory.
MOSI	Input	1-bit	Serial input instructions/data to the ISF Memory.
CSB	Input	1-bit	ISF Memory enable.
CLK	Input	1-bit	ISF Memory clock.

Design Entry Method

Instantiation	Yes
Inference	Recommended
Coregen and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
SIM_DEVICE	String	3S50AN, 3S200AN, 3S400AN, 3S700AN or 3S1400AN	"UNSPECIFIED"	Specifies the target device so that the proper size ISF Memory is used. This attribute *must* be set.
SIM_USER_ID	64-bit Hex Value	Any 64-bit Hex Value	All locations default to 0xFF	Specifies the programmed USER ID in the Security Register for the ISF Memory.
SIM_MEM_FILE	String	Specified file and directory name.	"NONE"	Optionally specifies a hex file containing the initialization memory content for the ISF Memory.

Attribute	Type	Allowed Values	Default	Description
SIM_FACTORY_ID	64-bit Hex Value	Any 64-bit Hex Value	All locations default to 0xFF	Specifies the Unique Identifier value in the Security Register for simulation purposes (the actual HW value will be specific to the particular device used).
SIM_DELAY_TYPE	String	"ACCURATE", "SCALED"	"SCALED"	Scales down some timing delays for faster simulation run. "ACCURATE" = timing and delays consistent with datasheet specs. "SCALED" = timing numbers scaled back to run faster simulation, behavior not affected.

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;

-- SPI_ACCESS: Internal Logic Access to the Serial Peripheral
--             Interface (SPI) PROM Data
--             Spartan-3AN
-- Xilinx HDL Libraries Guide, version 10.1.2

SPI_ACCESS_inst : SPI_ACCESS
generic map (
SIM_DELAY_TYPE => "SCALED", -- "ACCURATE" spec timing delays, "SCALED" shorten delays (faster sim)
SIM_DEVICE => "3S1400AN", -- "3S50AN", "3S200AN", "3S400AN", "3S700AN", "3S1400AN"
SIM_FACTORY_ID => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",  -- Specifies the Pre-programmed fa
SIM_MEM_FILE => "NONE",      -- Name/location of file containing memory contents
SIM_USER_ID => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000") -- Specifies the programmed User ID va
port map (
MISO => MISO,  -- Serial output data from SPI PROM
CLK => CLK,    -- SPI PROM clock input
CSB => CSB,    -- SPI PROM enable input
MOSI => MOSI   -- Serial input data to SPI PROM
);

-- End of SPI_ACCESS_inst instantiation
```

Verilog Instantiation Template

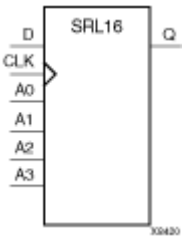
```
// SPI_ACCESS: Internal Logic Access to the Serial Peripheral
//             Interface (SPI) PROM Data
//             Spartan-3AN
// Xilinx HDL Libraries Guide, version 10.1.2

SPI_ACCESS #(
.SIM_DELAY_TYPE("SCALED"), // "ACCURATE" spec timing delays, "SCALED" shorten delays (faster sim)
.SIM_DEVICE("3S1400AN"),   // "3S50AN", "3S200AN", "3S400AN", "3S700AN", "3S1400AN"
.SIM_FACTORY_ID(64'h0),    // Specifies the Pre-programmed factory ID value
.SIM_MEM_FILE("NONE"),     // Name/location of file containing memory contents
.SIM_USER_ID(64'h0)        // Specifies the programmed User ID value
) SPI_ACCESS_inst (
.MISO(MISO),  // Serial output data from SPI PROM
.CLK(CLK),    // SPI PROM clock input
.CSB(CSB),    // SPI PROM enable input
.MOSI(MOSI)   // Serial input data to SPI PROM
);

// End of SPI_ACCESS_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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

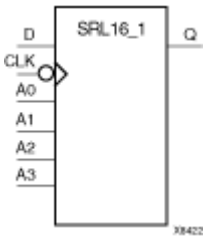
```
// SRL16: 16-bit shift register LUT operating on posedge of clock
//        All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

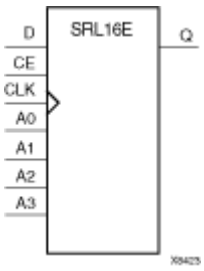
```
// SRL16_1: 16-bit shift register LUT operating on negedge of clock
//          All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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

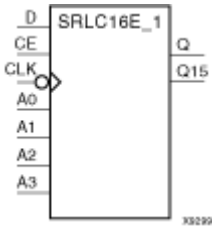
```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//         All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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	16-Bit Hexadecimal	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

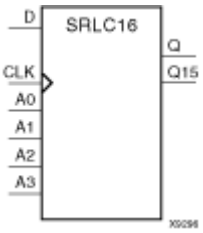
```
// SRL16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
//           All FPGAs
// Xilinx HDL Libraries Guide, version 10.1.2

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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

SRLC16

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry



Introduction

This design element is a shift register look-up table (LUT) with Carry. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register -** Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: 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 as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached. The Q15 output is available for you in cascading to multiple shift register LUTs to create larger shift registers.

Logic Table

Inputs			**Output**
Am	**CLK**	**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 Valuel	All zeros	Sets the initial value of content and output of shift register after configuration.

VHDL Instantiation Template

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

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

```
-- SRLC16: 16-bit cascadable shift register LUT operating on posedge of clock
--         Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

SRLC16_inst : SRLC16
generic map (
INIT => X"0000")
port map (
Q => Q,       -- SRL data output
Q15 => Q15,   -- Carry output (connect to next SRL)
A0 => A0,     -- Select[0] input
A1 => A1,     -- Select[1] input
A2 => A2,     -- Select[2] input
A3 => A3,     -- Select[3] input
CLK => CLK,   -- Clock input
D => D        -- SRL data input
);

-- End of SRLC16_inst instantiation
```

Verilog Instantiation Template

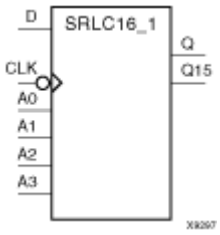
```
// SRLC16: 16-bit cascadable shift register LUT operating on posedge of clock
//         Virtex-II/II-Pro/4, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

SRLC16 #(
.INIT(16'h0000) // Initial Value of Shift Register
) SRLC16_inst (
.Q(Q),       // SRL data output
.Q15(Q15),   // Carry output (connect to next SRL)
.A0(A0),     // Select[0] input
.A1(A1),     // Select[1] input
.A2(A2),     // Select[2] input
.A3(A3),     // Select[3] input
.CLK(CLK),   // Clock input
.D(D)        // SRL data input
);

// End of SRLC16_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

SRLC16_1

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry and Negative-Edge Clock



Introduction

This design element is a shift register look-up table (LUT) with carry and a negative-edge clock. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register -** Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: 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 Q15 output is available for your use in cascading multiple shift register LUTs to create larger shift registers.

Logic Table

Inputs			Output	
Am	**CLK**	**D**	**Q**	**Q15**
Am	X	X	Q(Am)	No Change
Am	↓	D	Q(Am - 1)	Q14
m= 0, 1, 2, 3				

Design Entry Method

Instantiation	Yes
Inference	Recommended
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;
```

```
-- SRLC16_1: 16-bit cascadable shift register LUT operating on negedge of clock
--           Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

SRLC16_1_inst : SRLC16_1
generic map (
INIT => X"0000")
port map (
Q => Q,       -- SRL data output
Q15 => Q15,   -- Carry output (connect to next SRL)
A0 => A0,     -- Select[0] input
A1 => A1,     -- Select[1] input
A2 => A2,     -- Select[2] input
A3 => A3,     -- Select[3] input
CLK => CLK,   -- Clock input
D => D        -- SRL data input
);

-- End of SRLC16_1_inst instantiation
```

Verilog Instantiation Template

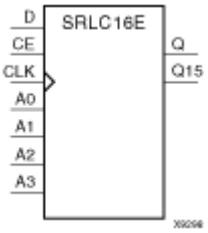
```
// SRLC16_1: 16-bit cascadable shift register LUT operating on negedge of clock
//           Virtex-II/II-Pro/4, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

SRLC16_1 #(
.INIT(16'h0000) // Initial Value of Shift Register
) SRLC16_1_inst (
.Q(Q),       // SRL data output
.Q15(Q15),   // Carry output (connect to next SRL)
.A0(A0),     // Select[0] input
.A1(A1),     // Select[1] input
.A2(A2),     // Select[2] input
.A3(A3),     // Select[3] input
.CLK(CLK),   // Clock input
.D(D)        // SRL data input
);

// End of SRLC16_1_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

SRLC16E

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry and Clock Enable



Introduction

This design element is a shift register look-up table (LUT) with carry and clock enable. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register -** Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: 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. When CE is High, during subsequent Low-to-High clock transitions, data shifts to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached.

The Q15 output is available for you in cascading to multiple shift register LUTs to create larger shift registers.

Logic Table

Inputs				**Output**	
Am	**CLK**	**CE**	**D**	**Q**	**Q15**
Am	X	0	X	Q(Am)	Q(15)
Am	X	1	X	Q(Am)	Q(15)
Am	↑	1	D	Q(Am - 1)	Q15
m= 0, 1, 2, 3					

Design Entry Method

Instantiation	Yes
Inference	Recommended
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;
```

```
-- SRLC16E: 16-bit cascable shift register LUT with clock enable operating on posedge of clock
--          Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

SRLC16E_inst : SRLC16E
generic map (
INIT => X"0000")
port map (
Q => Q,       -- SRL data output
Q15 => Q15,   -- Carry output (connect to next SRL)
A0 => A0,     -- Select[0] input
A1 => A1,     -- Select[1] input
A2 => A2,     -- Select[2] input
A3 => A3,     -- Select[3] input
CE => CE,     -- Clock enable input
CLK => CLK,   -- Clock input
D => D        -- SRL data input
);

-- End of SRLC16E_inst instantiation
```

Verilog Instantiation Template

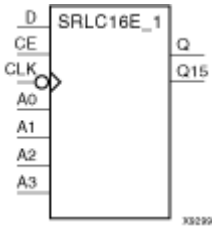
```
// SRLC16E: 16-bit cascadable shift register LUT with clock enable operating on posedge of clock
//          Virtex-II/II-Pro/4, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

SRLC16E #(
.INIT(16'h0000) // Initial Value of Shift Register
) SRLC16E_inst (
.Q(Q),       // SRL data output
.Q15(Q15),   // Carry output (connect to next SRL)
.A0(A0),     // Select[0] input
.A1(A1),     // Select[1] input
.A2(A2),     // Select[2] input
.A3(A3),     // Select[3] input
.CE(CE),     // Clock enable input
.CLK(CLK),   // Clock input
.D(D)        // SRL data input
);

// End of SRLC16E_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

SRLC16E_1

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry, Negative-Edge Clock, and Clock Enable



Introduction

This design element is a shift register look-up table (LUT) with carry, clock enable, and negative-edge clock. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register -** Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: 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 data shifts to the next highest bit position as new data is loaded when CE is High. The data appears on the Q output when the shift register length determined by the address inputs is reached.

The Q15 output is available for your use in cascading multiple shift register LUTs to create larger shift registers.

Logic Table

Inputs				Output	
Am	**CE**	**CLK**	**D**	**Q**	**Q15**
Am	0	X	X	Q(Am)	No Change
Am	1	X	X	Q(Am)	No Change
Am	1	↓	D	Q(Am -1)	Q14
m= 0, 1, 2, 3					

Design Entry Method

Instantiation	Yes
Inference	Recommended

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;
```

```
-- SRLC16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
--            Virtex-II/II-Pro, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 10.1.2

SRLC16E_1_inst : SRLC16E_1
generic map (
INIT => X"0000")
port map (
Q => Q,       -- SRL data output
Q15 => Q15,   -- Carry output (connect to next SRL)
A0 => A0,     -- Select[0] input
A1 => A1,     -- Select[1] input
A2 => A2,     -- Select[2] input
A3 => A3,     -- Select[3] input
CE => CE,     -- Clock enable input
CLK => CLK,   -- Clock input
D => D        -- SRL data input
);

-- End of SRLC16E_1_inst instantiation
```

Verilog Instantiation Template

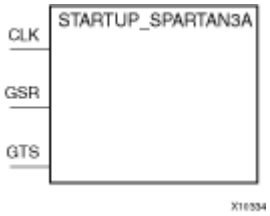
```
// SRLC16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
//            Virtex-II/II-Pro/4, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

SRLC16E_1 #(
.INIT(16'h0000) // Initial Value of Shift Register
) SRLC16E_1_inst (
.Q(Q),       // SRL data output
.Q15(Q15),   // Carry output (connect to next SRL)
.A0(A0),     // Select[0] input
.A1(A1),     // Select[1] input
.A2(A2),     // Select[2] input
.A3(A3),     // Select[3] input
.CE(CE),     // Clock enable input
.CLK(CLK),   // Clock input
.D(D)        // SRL data input
);

// End of SRLC16E_1_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.

STARTUP_SPARTAN3A

Primitive: Spartan-3A Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface



Introduction

This design element is used to either interface device pins and logic to the global asynchronous set/reset (GSR) signal, or for global, 3-state (GTS) dedicated routing. This primitive can also be used to specify a different clock for the device startup sequence at the end of configuring the device.

Port Descriptions

Name	Direction	Width	Function
GSR	Input	1-bit	Input connection to the global set / reset (GSR) routing.
GTS	Input	1-bit	Input connection to the global 3-state (GTS) routing.
CLK	Input	1-bit	Input connection to the configuration startup sequence clock (GSR) routing.

Design Entry Method

Instantiation	Recommended
Inference	No
Coregen and wizards	No
Macro support	No

To use the dedicated GSR circuitry, connect the sourcing pin or logic to the GSR pin. However, avoid using the GSR circuitry of this component unless certain precautions are taken first. Since the skew of the GSR net cannot be guaranteed, either use general routing for the set/reset signal in which routing delays and skew can be calculated as a part of the timing analysis of the design or to take preventative measures to ensure that possible skew on the release of the clock cycle won't interfere with circuit operation.

Similarly, if the dedicated global 3-state is used, connect the appropriate sourcing pin or logic to the GTS input pin of the primitive. In order to specify a clock for the startup sequence of configuration, connect a clock from the design to the CLK pin of this design 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;

-- STARTUP_SPARTAN3A: Startup primitive for GSR, GTS or startup sequence
--                    control.
--                    Spartan-3A
-- Xilinx HDL Libraries Guide, version 10.1.2

STARTUP_SPARTAN3A_inst : STARTUP_SPARTAN3A
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_SPARTAN3A_inst instantiation
```

Verilog Instantiation Template

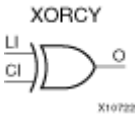
```
// STARTUP_SPARTAN3A: Startup primitive for GSR, GTS or startup sequence
//                    control.
//                    Spartan-3A
// Xilinx HDL Libraries Guide, version 10.1.2

STARTUP_SPARTAN3A STARTUP_SPARTAN3A_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_SPARTAN3A_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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
```

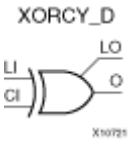
```
XORCY XORCY_inst (
.O(O),   // XOR output signal
.CI(CI), // Carry input signal
.LI(LI)  // LUT4 input signal
);

// End of XORCY_inst instantiation
```

For More Information

- See the *Spartan-3A User Guide*.
- See the *Spartan-3A 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 *Spartan-3A User Guide*.
- See the *Spartan-3A 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 *Spartan-3A User Guide*.
- See the *Spartan-3A Data Sheets*.